

Project	<b>IEEE 802.20 Working Group on Mobile Broadband Wireless Access</b> < <a href="http://ieee802.org/20/">http://ieee802.org/20/</a> >	
Title	<b>MBFDD and MBTDD: Proposed Draft Air Interface Specification</b>	
Date Submitted	<b>2006-01-06</b>	
Source(s)	Jim Tomcik Qualcomm, Incorporated 5775 Morehouse Drive San Diego, CA, 92121  Voice: 858-658-3231 Fax: 858-658-3231 Email: <a href="mailto:jtomcik@qualcomm.com">jtomcik@qualcomm.com</a>	Radhakrishna Canchi Kyocera Telecom Research Corp. 2480 N. First Street, Suite #280 San Jose, CA 95131  Voice: 408-952-4701 Fax: 408-954-8709 Email: <a href="mailto:cradhak@ktrc-na.com">cradhak@ktrc-na.com</a>
Re:	<b>MBWA Call for Proposals</b>	
Abstract	This contribution (part of the MBFDD and MBTDD proposal packages for 802.20), contains the proposed Draft Air Interface Specification. The Mobile Broadband Frequency Division Duplex (MBFDD) proposal is a revision of the previously submitted QFDD proposal; the Mobile Broadband Time Division Duplex (MBTDD) proposal is a revision of the previously submitted QTDD proposal, and also a merged proposal including the BEST-WINE proposal previously submitted to 802.20.	
Purpose	For consideration of 802.20 in its efforts to adopt FDD and TDD proposals for MBWA.	
Notice	This document has been prepared to assist the IEEE 802.20 Working Group. It is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributor(s) reserve(s) the right to add, amend or withdraw material contained herein.	
Release	The contributor grants a free, irrevocable license to the IEEE to incorporate material contained in this contribution, and any modifications thereof, in the creation of an IEEE Standards publication; to copyright in the IEEE's name any IEEE Standards publication even though it may include portions of this contribution; and at the IEEE's sole discretion to permit others to reproduce in whole or in part the resulting IEEE Standards publication. The contributor also acknowledges and accepts that this contribution may be made public by IEEE 802.20.	
Patent Policy	The contributor is familiar with IEEE patent policy, as outlined in <a href="http://standards.ieee.org/guides/opman/sect6.html#6.3">Section 6.3 of the IEEE-SA Standards Board Operations Manual</a> < <a href="http://standards.ieee.org/guides/opman/sect6.html#6.3">http://standards.ieee.org/guides/opman/sect6.html#6.3</a> > and in <i>Understanding Patent Issues During IEEE Standards Development</i> < <a href="http://standards.ieee.org/board/pat/guide.html">http://standards.ieee.org/board/pat/guide.html</a> >.	

**Abstract:**

These technical requirements form a compatibility standard for mobile broadband wireless access systems. The requirements ensure that a compliant access terminal can obtain service through any access network conforming to this standard, thus providing a framework for the rapid development of cost-effective, interoperable multivendor mobile broadband wireless access systems. This compatibility standard is targeted for use in a wide variety of licensed frequency bands.

This specification includes provisions for future service additions and expansion of system capabilities. The architecture defined by this specification permits such expansion without the loss of backward compatibility to older access terminals.

# Contents

<b>Foreword .....</b>	<b>1</b>
<b>References .....</b>	<b>2</b>
<b>1 Overview .....</b>	<b>4</b>
1.1 Scope of this document .....	4
1.2 Modes of the specification .....	4
1.3 Requirements language .....	4
1.4 Wideband Mode Overview .....	5
1.4.1 Architecture reference model .....	5
1.4.2 Protocol architecture .....	5
1.4.2.1 Layers .....	6
1.4.3 Physical layer channels .....	7
1.4.3.1 Forward physical channels .....	8
1.4.3.2 Reverse physical channels .....	9
1.4.4 Protocols .....	10
1.4.4.1 Interfaces .....	10
1.4.4.2 States .....	11
1.4.4.3 SessionConfigurationToken .....	11
1.4.4.4 InUse and Suspended protocol/transport instances .....	12
1.4.4.5 Procedures and messages .....	13
1.4.4.6 Common commands .....	13
1.4.4.7 Attribute negotiation .....	13
1.4.4.8 Protocol overview .....	14
1.4.5 Default transports .....	17
1.4.6 Sessions and connections .....	17
1.4.7 Security .....	17
1.4.8 Physical Layer modes .....	17
1.4.8.1 MIMO support .....	18
1.4.9 Management Information Base .....	18
1.4.10 Definitions .....	18
1.4.11 Abbreviations and acronyms .....	20
1.4.12 Notation .....	23
1.4.13 Malfunction detection .....	24
1.4.14 System time .....	24
1.4.15 Revision number .....	24
<b>2 Session Control Sublayer .....</b>	<b>25</b>
2.1 Introduction .....	25
2.1.1 General overview .....	25
2.2 Default Session Management Protocol .....	26
2.2.1 Overview .....	26

2.2.2 Primitives.....	27
2.2.2.1 Commands .....	27
2.2.2.2 Return indications .....	28
2.2.3 Public data .....	28
2.2.3.1 Static public data.....	28
2.2.3.2 Dynamic public data .....	28
2.2.4 Protocol initialization and swap procedures.....	28
2.2.4.1 Protocol initialization.....	28
2.2.4.2 Protocol swap.....	28
2.2.5 Procedures .....	28
2.2.5.1 Processing the Activate command .....	28
2.2.5.2 Processing the Deactivate command .....	29
2.2.5.3 Processing the SessionOpen message.....	29
2.2.5.4 Processing the SessionClose message .....	29
2.2.5.5 Processing failed indications.....	30
2.2.5.6 Processing the ForwardTrafficChannelMAC.SessionLost indication .....	30
2.2.5.7 Procedures for closing the session .....	30
2.2.5.8 Inactive state .....	30
2.2.5.9 AMP setup state .....	31
2.2.5.10 Open state .....	32
2.2.5.11 Close state .....	32
2.2.6 Message formats.....	33
2.2.6.1 SessionOpen.....	33
2.2.6.2 SessionClose .....	33
2.2.6.3 KeepAliveRequest .....	34
2.2.6.4 KeepAliveResponse.....	34
2.2.7 Interface to other protocols.....	35
2.2.7.1 Commands sent.....	35
2.2.7.2 Indications.....	35
2.2.8 Configuration attributes.....	36
2.2.9 Protocol numeric constants .....	36
2.2.10 Session state information.....	36
2.2.10.1 SessionSeed parameter.....	36
2.3 Default Address Management Protocol .....	37
2.3.1 Overview .....	37
2.3.2 Primitives.....	38
2.3.2.1 Commands .....	38
2.3.2.2 Return indications .....	38
2.3.3 Public data .....	38
2.3.3.1 Static public data.....	38
2.3.3.2 Dynamic public data .....	38
2.3.4 Connection endpoints.....	39
2.3.5 Protocol initialization and swap procedures.....	39
2.3.5.1 Protocol initialization.....	39
2.3.5.2 Protocol swap.....	39
2.3.6 Procedures .....	39

2.3.6.1	Processing the Activate command .....	39
2.3.6.2	Processing the Deactivate command .....	39
2.3.6.3	Processing the UpdateUATI command.....	39
2.3.6.4	Processing the RetrieveHWID command .....	40
2.3.6.5	UATIAssignment message validation .....	40
2.3.6.6	Processing HardwareIDRequest message .....	40
2.3.6.7	Inactive state .....	40
2.3.6.8	Setup state .....	41
2.3.6.9	Open state .....	42
2.3.7	Message formats.....	44
2.3.7.1	UATIUpdateRequest.....	44
2.3.7.2	UATIAssignment.....	44
2.3.7.3	UATIComplete .....	45
2.3.7.4	HardwareIDRequest.....	45
2.3.7.5	HardwareIDResponse .....	46
2.3.8	Interface to other protocols.....	47
2.3.8.1	Commands .....	47
2.3.8.2	Indications.....	47
2.3.9	Configuration attributes.....	47
2.3.10	Protocol numeric constants.....	47
2.3.11	Session state information.....	47
2.3.11.1	UATI parameter .....	48
2.4	Default Session Configuration Protocol .....	48
2.4.1	Overview .....	48
2.4.2	Primitives.....	50
2.4.2.1	Commands .....	50
2.4.2.2	Return indications .....	51
2.4.3	Public data .....	51
2.4.3.1	Static public data.....	51
2.4.3.2	Dynamic public data .....	51
2.4.4	Protocol initialization and swap procedures.....	51
2.4.4.1	Protocol initialization.....	51
2.4.4.2	Protocol swap.....	51
2.4.5	Procedures .....	51
2.4.5.1	Processing the Activate command .....	51
2.4.5.2	Processing the Deactivate command .....	52
2.4.5.3	Commit procedure .....	52
2.4.5.4	Save Procedure .....	52
2.4.5.5	TokenAssignment message validation.....	52
2.4.5.6	Inactive state .....	53
2.4.5.7	Active state .....	53
2.4.6	Message formats.....	55
2.4.6.1	TokensSupportedRequest .....	55
2.4.6.2	TokensSupportedResponse .....	55
2.4.6.3	TokenUpdateRequest.....	56
2.4.6.4	TokenAssignment .....	56

2.4.6.5 TokenComplete.....	57
2.4.6.6 LockConfiguration.....	57
2.4.6.7 LockConfigurationAck.....	58
2.4.6.8 UnLockConfiguration.....	58
2.4.6.9 UnLockConfigurationAck.....	59
2.4.6.10 ConfigurationRequest.....	59
2.4.6.11 ConfigurationAccept.....	60
2.4.6.12 ConfigurationReject.....	60
2.4.7 Interface to other protocols.....	60
2.4.7.1 Commands.....	60
2.4.7.2 Indications.....	61
2.4.8 Configuration attributes.....	61
2.4.9 Protocol numeric constants.....	61
2.4.10 Session State information.....	61
2.4.10.1 ConfigurationLock parameter.....	61
2.4.10.2 SessionConfigurationToken parameter.....	62
2.5 Default Capabilities Discovery Protocol.....	62
2.5.1 Overview.....	62
2.5.2 Primitives.....	63
2.5.2.1 Commands.....	63
2.5.2.2 Return indications.....	63
2.5.3 Public data.....	63
2.5.3.1 Static public data.....	63
2.5.3.2 Dynamic public data.....	63
2.5.4 Protocol initialization and swap procedures.....	64
2.5.4.1 Protocol initialization.....	64
2.5.4.2 Protocol swap.....	64
2.5.5 Procedures.....	64
2.5.5.1 Processing the Activate command.....	64
2.5.5.2 Processing the Deactivate command.....	64
2.5.5.3 Inactive state.....	64
2.5.5.4 Active state.....	64
2.5.6 Message formats.....	65
2.5.6.1 CapabilitiesRequest.....	65
2.5.6.2 CapabilitiesResponse.....	66
2.5.7 Interface to other protocols.....	66
2.5.7.1 Commands.....	66
2.5.7.2 Indications.....	66
2.5.8 Configuration attributes.....	67
2.5.8.1 Simple attributes.....	67
2.5.8.2 Complex attributes.....	70
2.5.9 Protocol numeric constants.....	71
2.5.10 Session state information.....	71
2.6 Default Inter Radio Access Technology (RAT) Protocol.....	72
2.6.1 Overview.....	72
2.6.2 Primitives.....	72

2.6.2.1	Commands .....	72
2.6.2.2	Return indications .....	72
2.6.3	Public data .....	73
2.6.3.1	Static public data .....	73
2.6.3.2	Dynamic public data .....	73
2.6.4	Protocol initialization and swap procedures .....	73
2.6.4.1	Protocol initialization .....	73
2.6.4.2	Protocol swap .....	73
2.6.5	Procedures .....	73
2.6.5.1	Processing the Activate command .....	73
2.6.5.2	Processing the Deactivate command .....	73
2.6.5.3	Inactive state .....	73
2.6.5.4	Active state .....	73
2.6.6	Message formats .....	74
2.6.6.1	InterRATBlob .....	74
2.6.6.2	InterRATIDRequest .....	74
2.6.6.3	InterRATIDResponse .....	75
2.6.7	Interface to other protocols .....	75
2.6.7.1	Commands .....	75
2.6.7.2	Indications .....	76
2.6.8	Configuration attributes .....	76
2.6.9	Protocol numeric constants .....	76
2.6.10	Session state information .....	76
<b>3</b>	<b>Convergence Sublayer .....</b>	<b>77</b>
3.1	Introduction .....	77
3.1.1	General Overview .....	77
3.2	Default Signaling Transport .....	78
3.2.1	Introduction .....	78
3.2.1.1	General overview .....	78
3.2.1.2	Public data .....	78
3.2.1.3	Data encapsulation .....	79
3.2.2	Transport initialization and swap procedures .....	80
3.2.2.1	Transport initialization .....	80
3.2.2.2	Transport swap .....	80
3.2.3	General signaling requirements .....	80
3.2.3.1	General requirements .....	80
3.2.3.2	Message information .....	81
3.2.4	Signaling Network Protocol .....	82
3.2.4.1	Overview .....	82
3.2.4.2	Primitives .....	82
3.2.4.3	Protocol data unit .....	82
3.2.4.4	Procedures .....	83
3.2.4.5	Type definitions .....	83
3.2.4.6	SNP packet header .....	84
3.2.4.7	Message formats .....	84

3.2.4.8 Interface to other protocols .....	84
3.2.5 Signaling Link Protocol.....	84
3.2.5.1 Overview.....	84
3.2.5.2 Primitives .....	85
3.2.5.3 Protocol data unit .....	85
3.2.5.4 Procedures.....	85
3.2.5.5 SLP packet header.....	93
3.2.5.6 Message formats .....	94
3.2.5.7 Interface to other protocols .....	97
3.2.5.8 Protocol numeric constants .....	97
3.2.6 Configuration attributes.....	97
3.2.7 Session state information.....	98
3.2.7.1 SignalingLinkState parameter.....	98
3.3 Default Data Transport .....	98
3.3.1 Introduction .....	98
3.3.1.1 General overview .....	98
3.3.1.2 Public data.....	101
3.3.1.3 Data encapsulation.....	101
3.3.2 Transport initialization and swap procedures.....	101
3.3.2.1 Transport initialization.....	101
3.3.2.2 Transport swap.....	102
3.3.3 Route Selection Protocol.....	102
3.3.3.1 Overview.....	102
3.3.3.2 Primitives .....	102
3.3.3.3 Protocol data unit .....	102
3.3.3.4 Procedures.....	102
3.3.3.5 Message formats .....	106
3.3.3.6 Interface to other protocols .....	108
3.3.3.7 Protocol numeric constants .....	108
3.3.4 Radio Link Protocol .....	108
3.3.4.1 Overview.....	108
3.3.4.2 Primitives .....	108
3.3.4.3 Protocol data unit .....	109
3.3.4.4 Procedures.....	109
3.3.4.5 Message formats .....	128
3.3.4.6 In-band message formats .....	137
3.3.4.7 Interface to other protocols .....	140
3.3.4.8 RLP packet priorities .....	140
3.3.4.9 Protocol numeric constants .....	141
3.3.5 Flow Control Protocol .....	141
3.3.5.1 Overview.....	141
3.3.5.2 Primitives .....	142
3.3.5.3 Protocol data unit .....	142
3.3.5.4 Procedures.....	142
3.3.5.5 Message formats .....	144
3.3.5.6 In-band message formats .....	145



3.3.5.7 Interface to other protocols .....	146
3.3.5.8 Protocol numeric constants .....	146
3.3.6 Configuration attributes for the default data transport .....	147
3.3.6.1 Simple attributes .....	147
3.3.6.2 Complex attributes .....	153
3.3.7 Session state information.....	178
3.3.7.1 FlowControlState parameter .....	179
3.3.7.2 ReservationState parameter .....	179
3.3.7.3 RouteState parameter .....	180
3.3.7.4 RadioLinkState parameter .....	181
3.4 Default Packet Consolidation Protocol .....	181
3.4.1 Overview .....	181
3.4.2 Data encapsulation .....	182
3.4.3 Primitives.....	182
3.4.3.1 Commands .....	182
3.4.3.2 Return indications .....	183
3.4.4 Public data .....	183
3.4.4.1 Static public data.....	183
3.4.4.2 Dynamic public data .....	183
3.4.5 Protocol data unit.....	183
3.4.6 Protocol initialization and swap procedures.....	183
3.4.6.1 Protocol initialization.....	183
3.4.6.2 Protocol swap.....	183
3.4.7 Procedures .....	183
3.4.7.1 Destination channels.....	184
3.4.7.2 Priority order.....	184
3.4.7.3 Forced single encapsulation.....	184
3.4.7.4 Transmit procedures.....	185
3.4.7.5 Access network procedures.....	185
3.4.8 Packet Consolidation Protocol header.....	185
3.4.9 Message formats.....	186
3.4.10 Interface to other protocols.....	186
3.4.10.1 Commands .....	186
3.4.10.2 Indications.....	186
3.4.11 Configuration attributes.....	186
3.4.11.1 TransportConfiguration attribute .....	187
3.4.12 Protocol numeric constants.....	188
3.4.13 Session state information.....	188
<b>4 Security Control Sublayer .....</b>	<b>189</b>
4.1 Introduction.....	189
4.1.1 General overview .....	189
4.2 Default Key Exchange Protocol.....	189
4.2.1 Overview .....	189
4.2.2 Primitives.....	189
4.2.2.1 Commands .....	189

4.2.2.2 Return indications .....	189
4.2.3 Public data .....	189
4.2.3.1 Static public data .....	189
4.2.3.2 Dynamic public data .....	190
4.2.4 Protocol data unit .....	190
4.2.5 Protocol initialization and swap .....	190
4.2.5.1 Protocol initialization .....	190
4.2.5.2 Protocol swap .....	190
4.2.6 Procedures .....	190
4.2.6.1 Access terminal requirements .....	191
4.2.6.2 Access network requirements .....	193
4.2.6.3 MIC Key, Authentication Key, and Encryption Key generation .....	196
4.2.6.4 Pseudorandom function, $PRF(K, A, B, Len)$ .....	196
4.2.6.5 HMAC-SHA256(K, Message) .....	196
4.2.7 Message format and flows .....	197
4.2.7.1 Message flows .....	197
4.2.7.2 Message formats .....	198
4.2.8 Interface to other protocols .....	203
4.2.8.1 Commands .....	203
4.2.8.2 Indications .....	203
4.2.9 Configuration attributes .....	204
4.2.10 Protocol numeric constants .....	204
4.2.11 Session state information .....	205
4.2.11.1 SKey parameter .....	205
4.2.11.2 Nonce parameter .....	207
4.2.11.3 LastValidTransactionID parameter .....	207
4.2.11.4 PMK parameter .....	208
<b>5 Security Sublayer .....</b>	<b>209</b>
5.1 Introduction .....	209
5.1.1 General overview .....	209
5.2 Packet encapsulation for the protocol instances .....	209
5.2.1 Packet encapsulation with IsSecure set .....	210
5.2.2 Packet encapsulation with IsSecure not set .....	211
5.2.3 Security Sublayer data transmit operation overview .....	211
5.2.4 Security Sublayer data receive operation overview .....	212
5.3 Default Encryption Protocol .....	212
5.3.1 Overview .....	212
5.3.2 Primitives .....	212
5.3.2.1 Commands .....	212
5.3.2.2 Return indications .....	212
5.3.3 Public data .....	212
5.3.3.1 Static public data .....	212
5.3.3.2 Dynamic public data .....	212
5.3.4 Protocol data unit .....	213
5.3.5 Protocol initialization and swap .....	213

5.3.5.1 Protocol initialization.....	213
5.3.5.2 Protocol swap.....	213
5.3.6 Procedures .....	213
5.3.7 Default Encryption Protocol header and trailer.....	213
5.3.8 Message formats.....	213
5.3.9 Interface to other protocols.....	213
5.3.9.1 Commands .....	213
5.3.9.2 Indications.....	213
5.3.10 Configuration attributes.....	213
5.3.11 Protocol numeric constants.....	213
5.3.12 Session state information.....	214
5.4 Default Security Protocol.....	214
5.4.1 Overview .....	214
5.4.2 Primitives.....	214
5.4.2.1 Commands .....	214
5.4.2.2 Return indications .....	214
5.4.3 Public data .....	214
5.4.3.1 Static public data.....	214
5.4.3.2 Dynamic public data .....	214
5.4.4 Protocol data unit.....	215
5.4.5 Protocol initialization and swap .....	215
5.4.5.1 Protocol Initialization.....	215
5.4.5.2 Protocol swap.....	215
5.4.6 Procedures .....	215
5.4.6.1 Secure State Procedures.....	215
5.4.6.2 Generation of the Cryptosync .....	215
5.4.6.3 Transmit procedures.....	216
5.4.6.4 Receive procedures .....	216
5.4.7 Header and trailer .....	217
5.4.8 Message formats.....	217
5.4.9 Interface to other protocols.....	217
5.4.9.1 Commands .....	217
5.4.9.2 Indications.....	217
5.4.10 Configuration attributes.....	217
5.4.11 Protocol numeric constants.....	217
5.4.12 Session state information.....	218
5.5 Default Authentication Protocol .....	218
5.5.1 Overview .....	218
5.5.2 Primitives.....	218
5.5.2.1 Commands .....	218
5.5.2.2 Return indications .....	218
5.5.3 Public data .....	218
5.5.3.1 Static public data.....	218
5.5.3.2 Dynamic public data .....	218
5.5.4 Protocol data unit.....	218
5.5.5 Protocol initialization and swap .....	218

5.5.5.1 Protocol initialization.....	218
5.5.5.2 Protocol swap.....	218
5.5.6 Procedures .....	219
5.5.6.1 Access terminal requirements .....	219
5.5.6.2 Access network requirements .....	220
5.5.7 Header and trailer .....	222
5.5.7.1 Header.....	222
5.5.7.2 Trailer.....	222
5.5.8 Message formats.....	222
5.5.9 Interface to other protocols.....	222
5.5.9.1 Commands .....	222
5.5.9.2 Indications.....	222
5.5.10 Configuration attributes.....	222
5.5.11 Protocol numeric constants.....	223
5.5.12 Session state information.....	223
5.6 Generic Encryption Protocol.....	223
5.6.1 Overview .....	223
5.6.2 Primitives.....	223
5.6.2.1 Commands .....	223
5.6.2.2 Return indications .....	223
5.6.3 Public data .....	224
5.6.3.1 Static public data.....	224
5.6.3.2 Dynamic public data .....	224
5.6.4 Protocol data unit.....	224
5.6.5 Protocol initialization and swap .....	224
5.6.5.1 Protocol initialization.....	224
5.6.5.2 Protocol swap.....	224
5.6.6 Procedures .....	224
5.6.6.1 Constructing the encryption key for the FL.....	224
5.6.6.2 Constructing the encryption key for the RL.....	225
5.6.6.3 Transmit procedures.....	225
5.6.6.4 Receive procedures .....	225
5.6.7 Generic Encryption Protocol header and trailer .....	226
5.6.8 Message formats.....	226
5.6.9 Interface to other protocols.....	226
5.6.9.1 Commands .....	226
5.6.9.2 Indications.....	226
5.6.10 Configuration attributes.....	226
5.6.11 Protocol numeric constants.....	226
5.6.12 Session state information.....	226
<b>6 Lower MAC Control Sublayer .....</b>	<b>227</b>
6.1 Introduction.....	227
6.1.1 General overview .....	227
6.1.2 Data encapsulation .....	228
6.2 Default Air Link Management Protocol .....	228

6.2.1 Overview .....	228
6.2.2 Primitives.....	231
6.2.2.1 Commands .....	231
6.2.2.2 Return indications .....	231
6.2.3 Public data .....	231
6.2.3.1 Static public data .....	231
6.2.3.2 Dynamic public data .....	231
6.2.4 Protocol initialization and swap procedures.....	231
6.2.4.1 Protocol initialization.....	231
6.2.4.2 Protocol swap.....	231
6.2.5 Procedures .....	232
6.2.5.1 Command processing.....	232
6.2.5.2 Initialization state.....	232
6.2.5.3 Idle state .....	233
6.2.5.4 Connected state .....	234
6.2.6 Message formats.....	236
6.2.6.1 Redirect .....	236
6.2.7 Interface to other protocols.....	237
6.2.7.1 Commands .....	237
6.2.7.2 Indications.....	238
6.2.8 Configuration attributes.....	238
6.2.9 Protocol numeric constants .....	238
6.2.10 Session state information.....	238
6.3 Default Idle State Protocol.....	239
6.3.1 Overview .....	239
6.3.2 Primitives.....	241
6.3.2.1 Commands .....	241
6.3.2.2 Return indications .....	241
6.3.3 Public data .....	241
6.3.3.1 Static public data.....	241
6.3.3.2 Dynamic public data .....	241
6.3.4 Protocol initialization and swap procedures.....	241
6.3.4.1 Protocol initialization.....	241
6.3.4.2 Protocol swap.....	241
6.3.5 Procedures .....	242
6.3.5.1 Command processing.....	242
6.3.5.2 General overview of paging.....	243
6.3.5.3 General overview of sleep cycle .....	243
6.3.5.4 General procedures .....	245
6.3.5.5 Inactive state .....	247
6.3.5.6 Sleep state .....	247
6.3.5.7 Monitor state .....	248
6.3.5.8 Access state.....	250
6.3.5.9 BindUATI state.....	250
6.3.6 Message formats.....	252
6.3.6.1 ConnectionOpenRequest.....	253

6.3.6.2	ConnectionOpenResponse .....	253
6.3.6.3	PageUATI .....	254
6.3.6.4	PreferredChannelRequest .....	255
6.3.7	Interface to other protocols .....	255
6.3.7.1	Commands .....	255
6.3.7.2	Indications .....	256
6.3.8	Configuration attributes .....	256
6.3.8.1	Preferred paging attribute .....	256
6.3.8.2	SlottedMode attribute .....	257
6.3.8.3	FastRepage attribute .....	258
6.3.8.4	MaxAccessAttempts attribute .....	258
6.3.9	Protocol numeric constants .....	259
6.3.10	Session state information .....	259
6.4	Default Connected State Protocol .....	259
6.4.1	Overview .....	259
6.4.2	Primitives .....	260
6.4.2.1	Commands .....	260
6.4.2.2	Return indications .....	260
6.4.3	Public data .....	260
6.4.3.1	Static public data .....	260
6.4.3.2	Dynamic public data .....	261
6.4.4	Protocol initialization and swap procedures .....	261
6.4.4.1	Protocol initialization .....	261
6.4.4.2	Protocol swap .....	261
6.4.5	Procedures .....	261
6.4.5.1	Command processing .....	261
6.4.5.2	Open state .....	262
6.4.5.3	Close state .....	267
6.4.6	Message formats .....	267
6.4.6.1	ConnectionClose .....	267
6.4.6.2	MIMOResponse .....	268
6.4.6.3	SelectedInterlaceRequest .....	269
6.4.6.4	SelectedInterlaceAssignment .....	270
6.4.6.5	SelectedInterlaceAck .....	271
6.4.6.6	TuneAwayRequest .....	271
6.4.6.7	TuneAwayResponse .....	272
6.4.6.8	ChannelMeasurementReportRequest .....	273
6.4.6.9	ChannelMeasurementReport .....	274
6.4.7	Interface to other protocols .....	275
6.4.7.1	Commands .....	275
6.4.7.2	Indications .....	275
6.4.8	Configuration attributes .....	276
6.4.8.1	Simple attributes .....	276
6.4.8.2	Complex attributes .....	276
6.4.8.3	TuneAwayScheduleN attribute .....	276
6.4.9	Protocol numeric constants .....	277

6.4.10 Session state information.....	278
6.4.10.1 ConnectedState parameter .....	278
6.5 Overhead Messages Protocol.....	279
6.5.1 Overview .....	279
6.5.2 Primitives.....	279
6.5.2.1 Commands .....	279
6.5.2.2 Return indications .....	279
6.5.3 Public data .....	280
6.5.3.1 Static public data.....	280
6.5.3.2 Dynamic public data .....	280
6.5.4 Protocol initialization and swap procedures.....	280
6.5.4.1 Protocol initialization.....	280
6.5.4.2 Protocol swap.....	281
6.5.5 Procedures .....	281
6.5.5.1 Extensibility requirements .....	281
6.5.5.2 Command processing.....	281
6.5.5.3 Inactive state .....	281
6.5.5.4 Active state .....	282
6.5.6 Message and block formats .....	287
6.5.6.1 SystemInfo block .....	288
6.5.6.2 QuickChannelInfo block.....	290
6.5.6.3 ExtendedChannelInfo .....	292
6.5.6.4 SectorParameters.....	300
6.5.6.5 EncapsulatedQuickChannelInfo .....	303
6.5.7 Interface to other protocols.....	304
6.5.7.1 Commands .....	304
6.5.7.2 Indications.....	304
6.5.8 Configuration attributes.....	304
6.5.9 Protocol numeric constants .....	305
6.5.10 Session state information.....	305
6.6 Default Active Set Management Protocol .....	305
6.6.1 Overview .....	305
6.6.2 Primitives.....	307
6.6.2.1 Commands .....	307
6.6.2.2 Return indications .....	307
6.6.3 Public data .....	307
6.6.3.1 Static public data.....	307
6.6.3.2 Dynamic public data .....	307
6.6.4 Protocol initialization and swap procedures.....	307
6.6.4.1 Protocol initialization.....	307
6.6.4.2 Protocol swap.....	308
6.6.5 Procedures .....	308
6.6.5.1 Command processing.....	308
6.6.5.2 Processing the RegistrationRadiusUpdated indications.....	308
6.6.5.3 Pilots and pilot sets .....	309
6.6.5.4 Message sequence numbers .....	313

6.6.5.5 Inactive state .....	314
6.6.5.6 Idle state .....	314
6.6.5.7 Connected state .....	318
6.6.6 Message formats .....	322
6.6.6.1 PilotReport .....	322
6.6.6.2 VCQIReportSISO .....	324
6.6.6.3 VCQIReportMIMO .....	325
6.6.6.4 ActiveSetAssignment .....	326
6.6.6.5 ActiveSetComplete .....	332
6.6.6.6 ResetReport .....	333
6.6.6.7 PilotReportRequest .....	333
6.6.7 Interface to other protocols .....	333
6.6.7.1 Commands .....	333
6.6.7.2 Indications .....	333
6.6.8 Configuration attributes .....	334
6.6.8.1 SearchParameters attribute .....	334
6.6.8.2 SetManagementSameChannelBandParameters attribute .....	335
6.6.8.3 SetManagementDifferentChannelBandParameters attribute .....	337
6.6.8.4 InitialSetupAttribute .....	339
6.6.8.5 IdleStateRegistrationTimeOut attribute .....	340
6.6.9 Protocol numeric constants .....	340
6.6.10 Session state information .....	340
6.6.10.1 ActiveSetManagement parameter .....	341
6.7 Default Initialization State Protocol .....	342
6.7.1 Overview .....	342
6.7.2 Primitives and public data .....	343
6.7.2.1 Commands .....	343
6.7.2.2 Return indications .....	343
6.7.3 Public data .....	343
6.7.3.1 Static public data .....	343
6.7.3.2 Dynamic public data .....	344
6.7.4 Protocol initialization and swap procedures .....	344
6.7.4.1 Protocol initialization .....	344
6.7.4.2 Protocol swap .....	344
6.7.5 Procedures .....	344
6.7.5.1 Command processing .....	344
6.7.5.2 Inactive state .....	344
6.7.5.3 Network determination state .....	345
6.7.5.4 Pilot acquisition state .....	345
6.7.5.5 Read SystemInfo state .....	345
6.7.6 Message formats .....	346
6.7.7 Interface to other protocols .....	346
6.7.7.1 Commands .....	346
6.7.7.2 Indications .....	346
6.7.8 Configuration attributes .....	346
6.7.9 Protocol numeric constants .....	346



6.7.10 Session state information.....	347
<b>7 Lower MAC Sublayer .....</b>	<b>348</b>
7.1 Introduction.....	348
7.1.1 General overview .....	348
7.1.2 Data encapsulation .....	348
7.1.3 Superframe timing.....	349
7.1.3.1 FDD .....	349
7.1.3.2 TDD .....	353
7.1.4 Common definitions and terms .....	358
7.1.4.1 Channel trees.....	358
7.1.4.2 Power density.....	359
7.2 Default Control Channel MAC Protocol .....	359
7.2.1 Overview .....	359
7.2.2 Primitives.....	360
7.2.2.1 Commands .....	360
7.2.2.2 Return indications .....	360
7.2.3 Public data .....	360
7.2.3.1 Static public data.....	360
7.2.3.2 Dynamic public data .....	360
7.2.4 Protocol data unit.....	360
7.2.5 Protocol initialization and swap .....	361
7.2.5.1 Protocol initialization.....	361
7.2.5.2 Protocol swap.....	361
7.2.6 Procedures .....	361
7.2.6.1 Procedures for transmission and reception of the F-pBCH1 Physical Layer channel.....	361
7.2.6.2 Procedures for transmission and reception of the F-pBCH0 Physical Layer channel .....	366
7.2.6.3 Procedures for transmission and reception of the F-OSICH Physical Layer channel.....	366
7.2.6.4 Command processing.....	367
7.2.6.5 Inactive state .....	367
7.2.6.6 Active state .....	367
7.2.7 Header and trailer formats.....	367
7.2.8 Interface to other protocols.....	368
7.2.8.1 Commands .....	368
7.2.8.2 Indications.....	368
7.2.9 Configuration attributes.....	368
7.2.10 Protocol numeric constants.....	368
7.2.11 Session state information.....	368
7.3 Default Access Channel MAC Protocol .....	368
7.3.1 Overview .....	368
7.3.2 Primitives.....	370
7.3.2.1 Commands .....	370
7.3.2.2 Return indications .....	370
7.3.3 Public data .....	370

7.3.3.1	Static public data.....	370
7.3.3.2	Dynamic public data.....	370
7.3.4	Protocol data unit.....	370
7.3.5	Protocol initialization and swap.....	370
7.3.5.1	Protocol initialization.....	370
7.3.5.2	Protocol swap.....	371
7.3.6	Procedures.....	371
7.3.6.1	Command processing.....	371
7.3.6.2	Access channel structure.....	372
7.3.6.3	Inactive state.....	372
7.3.6.4	Active state.....	373
7.3.7	Header formats.....	379
7.3.8	Message formats.....	379
7.3.9	Interface to other protocols.....	379
7.3.10	Configuration attributes.....	379
7.3.11	Protocol numeric constants.....	381
7.3.12	Session state information.....	381
7.4	Default Shared Signaling Channel MAC Protocol.....	381
7.4.1	Overview.....	381
7.4.2	Primitives.....	383
7.4.2.1	Commands.....	383
7.4.2.2	Return indications.....	383
7.4.3	Public data.....	383
7.4.3.1	Static public data.....	383
7.4.3.2	Dynamic public data.....	383
7.4.4	Protocol data unit.....	383
7.4.5	Protocol initialization and swap.....	383
7.4.5.1	Protocol initialization.....	383
7.4.5.2	Protocol swap.....	384
7.4.6	Procedures.....	384
7.4.6.1	Command processing.....	384
7.4.6.2	Inactive state.....	384
7.4.6.3	Active state.....	384
7.4.7	Header and trailer formats.....	400
7.4.8	Interface to other protocols.....	400
7.4.9	Configuration attributes.....	400
7.4.10	Protocol numeric constants.....	400
7.4.11	Session state information.....	401
7.5	Default Forward Traffic Channel MAC Protocol.....	401
7.5.1	Overview.....	401
7.5.2	Primitives.....	402
7.5.2.1	Commands.....	402
7.5.2.2	Return indications.....	402
7.5.3	Public data.....	402
7.5.3.1	Static public data.....	402
7.5.3.2	Dynamic public data.....	402

7.5.4	Protocol data unit.....	403
7.5.5	Protocol initialization and swap .....	403
7.5.5.1	Protocol initialization.....	403
7.5.5.2	Protocol Swap.....	403
7.5.6	Procedures .....	403
7.5.6.1	Command processing.....	403
7.5.6.2	Forward traffic channel addressing.....	404
7.5.6.3	Inactive state .....	406
7.5.6.4	Active state .....	406
7.5.6.5	Supervision procedures.....	420
7.5.6.6	Channel trees.....	420
7.5.6.7	Packet formats.....	422
7.5.7	Header and trailer formats.....	424
7.5.7.1	Header and trailer for unicast transmissions.....	424
7.5.7.2	Header and trailer for broadcast transmissions.....	428
7.5.8	Message formats.....	428
7.5.9	Interface to other protocols.....	428
7.5.9.1	Commands sent.....	428
7.5.9.2	Indications.....	429
7.5.10	Configuration attributes.....	429
7.5.11	Protocol numeric constants.....	429
7.5.12	Session state information.....	429
7.6	Default Reverse Control Channel MAC Protocol.....	430
7.6.1	Overview .....	430
7.6.2	Primitives.....	432
7.6.2.1	Commands .....	432
7.6.2.2	Return indications .....	432
7.6.3	Public data .....	432
7.6.3.1	Static public data.....	432
7.6.3.2	Dynamic public data.....	432
7.6.4	Protocol data unit.....	432
7.6.5	Protocol initialization and swap .....	432
7.6.5.1	Protocol initialization.....	432
7.6.5.2	Protocol swap.....	433
7.6.6	Procedures .....	433
7.6.6.1	Command processing.....	433
7.6.6.2	Inactive State.....	433
7.6.6.3	Active State.....	433
7.6.7	Message formats.....	457
7.6.8	Interface to other protocols.....	457
7.6.9	Configuration attributes.....	458
7.6.9.1	PowerControl attribute.....	458
7.6.10	Protocol numeric constants.....	459
7.6.11	Session state information.....	459
7.7	Default Reverse Traffic Channel MAC Protocol.....	459
7.7.1	Overview .....	459

7.7.2 Primitives.....	461
7.7.2.1 Commands .....	461
7.7.2.2 Return indications .....	461
7.7.3 Public data .....	461
7.7.3.1 Static public data.....	461
7.7.3.2 Dynamic public data .....	461
7.7.4 Protocol data unit.....	461
7.7.5 Protocol initialization and swap .....	462
7.7.5.1 Protocol initialization.....	462
7.7.5.2 Protocol Swap .....	462
7.7.6 Procedures .....	462
7.7.6.1 Command processing.....	462
7.7.6.2 Reverse traffic channel addressing .....	463
7.7.6.3 Inactive state .....	464
7.7.6.4 Active state .....	464
7.7.6.5 Reverse link silence interval .....	476
7.7.6.6 Supervision procedures .....	476
7.7.6.7 Channel trees.....	477
7.7.6.8 Packet formats.....	479
7.7.7 Header and trailer and formats .....	480
7.7.7.1 Header .....	480
7.7.7.2 Trailer.....	481
7.7.8 Message formats.....	484
7.7.8.1 OSIReportRequest .....	484
7.7.8.2 OSIReport .....	485
7.7.9 Interface to other protocols.....	486
7.7.10 Configuration attributes.....	487
7.7.10.1 PowerParameters attribute .....	487
7.7.11 Protocol numeric constants.....	488
7.7.12 Session state information.....	488
<b>8 Physical Layer.....</b>	<b>489</b>
8.1 Default Physical Layer Protocol .....	489
8.1.1 Overview .....	489
8.1.2 Primitives.....	489
8.1.2.1 Commands .....	489
8.1.2.2 Return indications .....	489
8.1.3 Public data .....	489
8.1.3.1 Static public data.....	489
8.1.3.2 Dynamic public data .....	489
8.1.4 Protocol data unit.....	489
8.1.5 Protocol initialization and swap procedures .....	489
8.1.5.1 Protocol initialization.....	489
8.1.6 Protocol swap .....	490
8.1.7 Procedures .....	490
8.1.8 Message formats.....	490

8.1.8.1 TimingCorrection.....	490
8.1.9 Interface to other protocols.....	490
8.1.9.1 Commands .....	490
8.1.9.2 Indications.....	490
8.1.10 Configuration attributes.....	490
8.1.11 Protocol numeric constants and parameters .....	491
8.1.12 Session state information.....	492
<b>9 Default Physical Layer.....</b>	<b>493</b>
9.1 Physical layer modes.....	493
9.2 Encoding and modulation .....	493
9.2.1 Packet-splitting and CRC insertion .....	494
9.2.2 Core encoders .....	494
9.2.2.1 Rate 1/3 convolutional encoding .....	494
9.2.2.2 Rate 1/5 turbo encoding .....	495
9.2.3 Channel interleaving.....	500
9.2.3.1 Bit demultiplexing .....	500
9.2.3.2 Bit permuting.....	501
9.2.4 Sequence repetition .....	501
9.2.5 Data scrambling.....	502
9.2.6 Modulation .....	503
9.2.6.1 QPSK modulation .....	504
9.2.6.2 8-PSK modulation.....	505
9.2.6.3 16-QAM modulation.....	506
9.2.6.4 64-QAM modulation.....	507
9.3 Access network requirements .....	510
9.3.1 Transmitter .....	510
9.3.2 Modulation characteristics .....	511
9.3.2.1 Superframe timing .....	511
9.3.2.2 OFDM symbol characteristics .....	514
9.3.2.3 Multiple transmit antennas.....	516
9.3.2.4 Superframe preamble modulation.....	517
9.3.2.5 FL PHY Frame modulation .....	520
9.3.2.6 Sector-specific scrambling.....	551
9.3.2.7 Cell-specific scrambling for F-DPICH .....	553
9.3.2.8 Time-domain processing.....	554
9.3.3 Synchronization and timing.....	556
9.3.3.1 Timing reference source .....	556
9.3.3.2 Sector transmission time .....	557
9.4 Access terminal requirements .....	557
9.4.1 Modulation characteristics .....	557
9.4.1.1 Superframe timing .....	557
9.4.1.2 OFDM symbol characteristics .....	559
9.4.1.3 Hopping sequence generation .....	562
9.4.1.4 R-ACKCH .....	570
9.4.1.5 Control segment modulation.....	573

9.4.1.6 Data segment modulation .....	582
9.4.1.7 Time-domain processing .....	591
9.4.2 Synchronization and timing .....	593
<b>10 Common Algorithms and Data Structures .....</b>	<b>595</b>
10.1 Channel band record .....	595
10.1.1 Definition of ChannelBandRecord record for Band Class .....	596
10.1.2 Definition of ChannelBandRecord record for Frequency Specified .....	596
10.2 Access terminal identifier record .....	597
10.3 Attribute record .....	597
10.4 Hash function .....	598
10.5 Computation of the CRC bits .....	599
10.6 Pseudorandom number generator .....	600
10.6.1 General procedures .....	600
10.6.2 Initialization .....	600
10.7 Sequence number .....	601
10.7.1 Sequence number initialization .....	601
10.7.2 Sequence number validation .....	601
10.8 TransactionID number .....	601
10.8.1 TransactionID initiating procedures .....	601
10.9 Generic Attribute Update Protocol .....	601
10.9.1 Introduction .....	601
10.9.2 Procedures .....	602
10.9.2.1 Initiator requirements .....	602
10.9.2.2 Responder requirements .....	603
10.9.3 Message formats .....	603
10.9.3.1 AttributeUpdateRequest .....	603
10.9.3.2 AttributeUpdateAccept .....	604
10.9.3.3 AttributeUpdateReject .....	604
10.9.3.4 AttributeValueRequest .....	605
10.9.3.5 AttributeValueResponse .....	605
10.9.4 Protocol numeric constants .....	606
10.10 Session state information record .....	606
10.11 SectorID provisioning .....	607
10.11.1 SectorID construction .....	607
<b>11 Assigned Names and Numbers .....</b>	<b>608</b>
11.1 Protocols .....	608
11.2 Transport subtype assignments .....	609
11.3 Messages .....	609
11.4 Other RAT Types .....	612
11.5 Session Configuration Tokens .....	612
11.5.1 SessionConfigurationToken 0x0000 .....	612
11.5.2 SessionConfigurationToken 0x0001 .....	613
11.5.3 SessionConfigurationToken 0x0002 .....	615
11.6 Flow profile identifier assignments .....	616

11.6.1 Flow Profile Identifier Assignments .....	617
11.6.1.1 Generic data service flow profile identifier assignments .....	617
11.6.1.2 Speech service flow profile identifier assignments .....	618
11.6.1.3 Audio service flow profile identifier assignments .....	619
11.6.1.4 Video service flow profile identifier assignments .....	620
11.6.1.5 Text service flow profile identifier assignments .....	621
11.6.1.6 Signaling service flow profile identifier assignments .....	621
11.6.1.7 Gaming Service Flow Profile Identifier Assignments .....	622
<b>12 Precoding and SDMA Codebooks .....</b>	<b>623</b>
12.1 BFCHBeamCodeBookIndex = 0000 .....	623
12.1.1 Cluster 1 : .....	623
12.1.2 Cluster 2: .....	623
12.2 BFCHBeamCodeBookIndex = 0001 .....	624
12.2.1 Precoding Matrices .....	624
12.2.2 Cluster 1: .....	625
12.2.3 Cluster 2: .....	625
<b>13 MAC and PHY MIB .....</b>	<b>626</b>
13.1 Overview .....	626
13.2 MIB Structure .....	626
13.3 Definition .....	626
<b>14 System Overview of 625k-MC (625kiloHertz-spaced MultiCarrier) Mode ...</b>	<b>664</b>
14.1 Scope .....	664
14.2 Architecture Reference Model .....	664
14.3 Acronyms .....	665
14.4 Conventions .....	665
14.5 625k-MC Application Overview .....	665
14.6 625k-MC Protocol Overview .....	666
14.6.1 625k-MC Protocol Features .....	666
14.6.2 625k-MC Protocol Reference Model and Interfaces .....	666
<b>15 625k-MC Spectral Layout Terminology and Requirements .....</b>	<b>667</b>
<b>16 625k-MC Slot and Frame Structure .....</b>	<b>668</b>
16.1 Overview .....	668
16.2 RF Channel and Frame Structure .....	668
16.3 Burst Formats .....	668
16.3.1 Frequency Synchronization .....	668
16.3.2 Timing Synchronization .....	668
16.3.3 Broadcast Burst .....	668
16.3.4 Page Burst .....	668
16.3.5 Configuration Request Burst .....	668
16.3.6 Standard Uplink Burst (FACCH, RACH & TCH) .....	668
16.3.7 Standard Downlink Burst (CM, <u>SMB</u> , AA & TCH) .....	668

<b>17 625k-MC Modulation and Channel Coding .....</b>	<b>669</b>
17.1 625k-MC Modulation and Channel Coding Overview .....	669
17.2 Standard Modulation and Coding .....	670
17.2.1 Encryption .....	672
17.2.2 Cyclic redundancy check.....	672
17.2.3 Multiplexing .....	672
17.2.4 Tail append .....	672
17.2.5 Convolutional Encoding.....	672
17.2.6 Puncturing and repeating.....	673
17.2.7 Block Coding.....	673
17.2.7.1 Extended Hamming Code .....	673
17.2.7.2 Parity check Code .....	673
17.2.8 Block Shaper .....	673
17.2.9 Symbol Mapping .....	674
17.2.10 Interleaving.....	675
17.2.11 Scrambling.....	676
17.2.12 $\pi/2$ Rotation and Scaling .....	676
17.3 Broadcast Channel Modulation and Coding .....	676
<b>18 625k-MC User Terminal Radio Transmission and Reception.....</b>	<b>677</b>
<b>19 625k-MC Base Station Radio Transmission and Reception.....</b>	<b>678</b>
<b>20 625k-MC L2 MAC Protocol Sublayer Specification .....</b>	<b>679</b>
20.1 Logical Channels .....	679
20.1.1 <u>Short Message Broadcast (SMB)</u> .....	680
20.2 <u>625k-MC Minimized RMU header</u> .....	680
20.2.1 Message Format for AA-cts and AA-short .....	680
20.2.2 625k-MC RMU Field Definition.....	681
20.2.3 625k-MC Header Field Insertion.....	682
20.2.4 UM Message Insertion .....	683
<b>21 625k-MC L2 RLC Protocol Sublayer Specification.....</b>	<b>683</b>
21.1 625k-MC AM RMU .....	684
21.1.1 <u>RMU Header</u> .....	684
21.2 625k-MC Transmit Procedure .....	684
21.2.1 625k-MC Transmit State Execution.....	685
21.2.2 mapTx Update Procedure.....	686
21.3 Receive Procedure .....	686
21.3.1 Receive Task Execution .....	686
21.3.2 MapTx Retransmit Update .....	688
21.3.3 Reset Procedure.....	688



<b>22 625k-MC L3 Protocol Specification .....</b>	<b>689</b>
<b>23 625k-MC Protocol Layer Primitives (Informative) .....</b>	<b>690</b>
23.1 Interface list .....	690
23.2 Individual Interfaces .....	690
23.2.1 L2 MAC to L3 RM Interface Primitives .....	690
23.2.2 L3 RM to L2 MAC Interface Primitives .....	691
23.2.3 L3 RM to L3 CM Interface Primitives .....	691
23.2.4 L3 CM to L3 RM Interface Primitives .....	693
23.2.5 L3 CM to L4 Interface Primitives .....	694
23.2.6 L4 to L3 CM Interface Primitives .....	695
<b>24 625k-MC QoS Enhancements .....</b>	<b>697</b>
24.1 Classes of Services .....	697
24.2 Session QoS Information Exchange Procedures .....	698
24.3 QoS Priority .....	699
24.3.1 QoS Priority 1 .....	699
24.3.2 QoS Priority 2 .....	699
24.3.3 QoS Priority 3 .....	699
<b>25 625k-MC Broadcast and Multicast Service (BCMCS) Support Enhancement .....</b>	<b>700</b>
25.1 Overview .....	700
25.2 Broadcast Service .....	700
25.3 Multicast Service .....	701
25.3.1 Overview .....	701
25.3.2 Multicast handshake .....	701
25.3.3 Message Format .....	703
25.3.3.1 MCStream.req Message .....	703
25.3.3.2 MCStream.resp Message .....	705
<b>26 625k-MC Privacy and Authentication Enhancement.....</b>	<b>708</b>
26.1 Overview .....	708
26.2 625k-MC Handshake and BS Authentication Protocol, i-HAP .....	708
26.3 625k-MC Terminal Authentication Protocol, i-TAP .....	709
26.4 625k-MC Secure Communications Protocol, i-SEC .....	710
26.4.1 TCH Streams .....	710
26.4.2 625k-MC Symmetric Key Stream Cipher Algorithm.....	710
26.4.2.1 Initialization Vector Selection .....	710
26.4.2.2 Stream Counter Test .....	711
26.4.2.3 Determination of Encryption Key from Shared Secret .....	711
26.4.2.4 Stream Cipher State Initialization .....	711
26.4.2.5 Keystream Generation .....	711
26.4.2.6 Burst Counter Test .....	711
26.4.2.7 Encryption and Decryption Using i-SEC Keystream .....	711
26.4.3 625k-MC AES Cipher Algorithm .....	712

26.4.3.1 AES Initialization Vector Selection.....	712
26.4.3.2 AES Stream Counter Test.....	712
26.4.3.3 Determination of AES Encryption Key from Shared Secret .....	712
26.4.3.4 AES Cipher State Initialization.....	713
26.4.3.5 AES Keystream Generation.....	713
26.4.3.6 AES Burst Counter Test.....	713
26.4.3.7 Encryption and Decryption Using i-SEC AES Keystream.....	714
<b>27 625k-MC Sleep Mode Control Protocol .....</b>	<b>715</b>
<b>28 625k-MC OA &amp; M Radio Network Quality Monitor and Control Enhancement .....</b>	<b>716</b>
28.1 625k-MC Mode MIB .....	716
28.1.1 Overview .....	716
28.1.2 Definition.....	716
<b>625K-MC Appendix – A (Informative) .....</b>	<b>772</b>

## Figures

Figure 1	Architecture reference model .....	5
Figure 2	Air interface layering architecture .....	6
Figure 3	Forward channel structure .....	7
Figure 4	Reverse channel structure .....	8
Figure 5	SessionConfigurationToken state diagram .....	12
Figure 6	Protocol and transport types .....	14
Figure 7	Session Control Sublayer protocols .....	26
Figure 8	Session Management Protocol state diagram (access terminal) .....	27
Figure 9	Session Management Protocol state diagram (access network) .....	27
Figure 10	Address Management Protocol state diagram (access terminal) .....	37
Figure 11	Address Management Protocol state diagram (access network) .....	38
Figure 12	SessionConfigurationToken state diagram .....	49
Figure 13	Session Configuration Protocol state diagram .....	50
Figure 14	Session Capabilities Discovery Protocol state diagram .....	63
Figure 15	Inter RAT Protocol state diagram .....	72
Figure 16	Convergence Sublayer protocols .....	77
Figure 17	Default signaling transport protocols .....	78
Figure 18	Message encapsulation (non-fragmented) .....	79
Figure 19	Message encapsulation (fragmented) .....	79
Figure 20	Sample message information .....	81
Figure 21	SNP packet structure .....	82
Figure 22	SLP reset procedure initiated by SLP transmitter .....	88
Figure 23	SLP reset procedure initiated by SLP receiver .....	88
Figure 24	SLP transmit sequence number variable .....	90
Figure 25	SLP receive sequence number variables .....	91
Figure 26	Reference architecture for a forward link flow .....	99
Figure 27	Reference architecture for a reverse link flow .....	99
Figure 28	Default data transport protocols .....	100
Figure 29	Relationship between default data transport and higher layer protocols .....	100
Figure 30	Default data transport encapsulation .....	101
Figure 31	Route selection protocol state diagram (access terminal) .....	103
Figure 32	RLP reset procedure initiated by RLP transmitter .....	115
Figure 33	RLP reset procedure initiated by RLP receiver .....	115
Figure 34	RLP transmit sequence number variable .....	118
Figure 35	Reservation state diagram (access terminal) .....	119
Figure 36	Reservation state diagram (access network) .....	120
Figure 37	RLP receive sequence number variables .....	124
Figure 38	Flow control protocol state diagram (access terminal) .....	141
Figure 39	Flow control protocol state diagram (access network) .....	142
Figure 40	Packet Consolidation Protocol encapsulation .....	182
Figure 41	Default Key Exchange Protocol message flow .....	197
Figure 42	Security Key Change Protocol [??Figure updated above] .....	197
Figure 43	Security Sublayer protocols .....	209

Figure 44 Security Sublayer data encapsulation for IsSecure=1 .....	210
Figure 45 Security Sublayer data encapsulation for IsSecure=0 .....	211
Figure 46 Air Link Management Protocol state diagram (access terminal) .....	229
Figure 47 Air Link Management Protocol state diagram (access network).....	230
Figure 48 Default Idle State Protocol (access terminal) .....	239
Figure 49 Default Idle State Protocol state (access network) .....	240
Figure 50 Slotted mode operation when access terminal and access network are synchronized.....	244
Figure 51 Slotted mode operation when access terminal and access network are not synchronized.....	245
Figure 52 Default Connected State Protocol state diagram (access terminal) .....	259
Figure 53 Default Connected State Protocol state diagram (access network) .....	260
Figure 54 Overhead Messages Protocol state diagram .....	279
Figure 55 Default Active Set Management Protocol state diagram.....	306
Figure 56 Default Initialization State Protocol state diagram (access terminal).....	343
Figure 57 Lower MAC Sublayer packet encapsulation .....	349
Figure 58 FDD superframe timing.....	350
Figure 59 FDD FL H-ARQ interlace structure .....	351
Figure 60 FDD RL H-ARQ interlace structure.....	351
Figure 61 FDD FL H-ARQ interlace structure for Extended Transmission Duration Assignments .....	352
Figure 62 FDD RL H-ARQ interlace structure for Extended Transmission Duration Assignments .....	352
Figure 63 TDD superframe timing for 1:1 partitioning .....	354
Figure 64 TDD FL H-ARQ interlace structure for 1:1 partitioning.....	354
Figure 65 TDD RL H-ARQ interlace structure for 1:1 partitioning .....	355
Figure 66 TDD superframe timing for 2:1 partitioning .....	355
Figure 67 TDD FL H-ARQ interlace structure for 2:1 partitioning.....	356
Figure 68 TDD RL H-ARQ interlace structure for 2:1 partitioning .....	356
Figure 69 TDD Superframe Structure for other partitionings.....	357
Figure 70 Example channel tree .....	358
Figure 71 Default Control Channel MAC Protocol state diagram.....	360
Figure 72 Default Access Channel MAC Protocol state diagram.....	369
Figure 73 Access probe sequences. Ns sequences with Np probes per sequence.....	372
Figure 74 Default Shared Signaling Channel MAC Protocol state diagram .....	382
Figure 75 Default Forward Traffic Channel MAC Protocol state diagram .....	401
Figure 76 F-DCH addressing example.....	405
Figure 77 FL channel trees with index 0.....	421
Figure 78 Default Reverse Control Channel MAC Protocol state diagram.....	430
Figure 79 Default Reverse Traffic Channel MAC Protocol state diagram .....	460
Figure 80 R-DCH addressing example .....	464
Figure 81 RL channel trees with index 0 .....	478
Figure 82 Encoding and modulation structure.....	493
Figure 83 Rate 1/3 convolutional encoder .....	495
Figure 84 Turbo encoder.....	497
Figure 85 Turbo interleaver output address calculation procedure.....	499
Figure 86 Data scrambler.....	503

Figure 87	Signal constellation for QPSK modulation .....	504
Figure 88	Signal constellation for 8-PSK modulation .....	505
Figure 89	Signal constellation for 16-QAM modulation .....	507
Figure 90	Signal constellation for 64-QAM modulation .....	510
Figure 91	Forward link superframe structure: FDD .....	511
Figure 92	Forward link superframe structure: TDD ( $N_{FL\_BURST}=1, N_{RL\_BURST}=1$ ) .....	512
Figure 93	Forward link superframe structure: TDD ( $N_{FL\_BURST}=2, N_{RL\_BURST}=1$ ) .....	512
Figure 94	Forward link superframe structure: TDD ( $N_{FL\_BURST}=3, N_{RL\_BURST}=2$ ) .....	512
Figure 95	Forward channel structure: SymbolRateHopping mode.....	513
Figure 96	Forward channel structure: BlockHopping mode.....	514
Figure 97	Offset <sub>p</sub> Determination.....	518
Figure 98	PN Register for generating pseudorandom bits .....	521
Figure 99	F-DPICH Format 0 .....	536
Figure 100	F-DPICH Format 1 .....	537
Figure 101	F-DPICH Format 2 .....	538
Figure 102	Cyclic spatial multiplexing.....	547
Figure 103	Sector-specific scrambler – I sequence .....	552
Figure 104	Sector-specific scrambler – Q sequence.....	553
Figure 105	Time-domain processing .....	554
Figure 106	Overlap and add operation.....	556
Figure 107	Reverse link superframe structure: FDD .....	558
Figure 108	Reverse link superframe structure: TDD ( $N_{FL\_BURST}=1, N_{RL\_BURST}=1$ ).....	558
Figure 109	Reverse link superframe structure: TDD ( $N_{FL\_BURST}=2, N_{RL\_BURST}=1$ ).....	558
Figure 110	Reverse link superframe structure ( $N_{FL\_BURST}=3, N_{RL\_BURST}=2$ ).....	559
Figure 111	Reverse channel structure.....	559
Figure 112	PN Register for generating pseudorandom bits .....	563
Figure 113	R-DPICH Format 0.....	584
Figure 114	R-DPICH Format 1 .....	585
Figure 115	Sector-specific scrambler for the data segments – I sequence.....	589
Figure 116	Sector-specific scrambler for the data segments – Q sequence.....	589
Figure 117	Time-domain processing .....	591
Figure 118	Overlap and add operation.....	593
Figure 119	Relationship between forward link and reverse link timings .....	593
Figure 120	CRC calculation.....	600
Figure 121	Architecture reference model.....	664
Figure 122	625k-MC PHY/MAC/LLC Protocol Reference .....	666
Figure 123	Block diagram for error control coding scheme. The notation $\{i, j\}$ indicates that a block is active only for modulation classes $i$ and $j$ . .....	670
Figure 124	QoS Message sequence .....	698
Figure 125	BCH superframe with SMB after $B_3$ .....	700
Figure 126	Message Flow for short message channel .....	701
Figure 127	Message Flow for Multicast channel.....	702
Figure 128	<u>MCStream.req</u> format.....	704
Figure 129	<u>MCStream.resp</u> format .....	707
Figure 130	HC-SDMA BS certificate message fields. Mandatory field values are identified in square brackets. Not to scale. ....	709

## Tables

Table 1	Encoding of CloseReason field.....	34
Table 2	Configurable attributes .....	36
Table 3	Format of the parameter record for the SessionSeed parameter .....	36
Table 4	HardwareIDType encoding.....	46
Table 5	Format of the parameter record for the UATI parameter.....	48
Table 6	Format of the parameter record for the ConfigurationLock parameter.....	61
Table 7	Format of the parameter record for the SessionConfigurationToken parameter .....	62
Table 8	Configurable attributes .....	67
Table 9	Default protocol stack type values.....	83
Table 10	Format of the parameter record for the SignalingLinkState parameter .....	98
Table 11	Configurable values .....	147
Table 12	Traffic Class.....	162
Table 13	FilterSpecType for Packet Filter.....	166
Table 14	ProtocolID for Flow Protocol .....	173
Table 15	ProtocolID for Route Protocol.....	177
Table 16	Format of the parameter record for the FlowControlState parameter.....	179
Table 17	Format of the parameter record for the ReservationState parameter .....	179
Table 18	Format of the parameter record for the RouteState parameter.....	180
Table 19	RouteSelectionProtocolState encoding.....	180
Table 20	The format of the parameter record for the RadioLinkNNState parameter .....	181
Table 21	Transport subtypes for transports defined in this specification .....	182
Table 22	Definition of result field .....	200
Table 23	Definition of result field .....	201
Table 24	Configurable values .....	204
Table 25	Format of the parameter record for the SKey parameter .....	205
Table 26	Format of the parameter record for the Nonce parameter.....	207
Table 27	Format of the parameter record for the LastValidTransactionID parameter .....	207
Table 28	Format of the parameter record for the PMK parameter .....	208
Table 29	Subfield of the Cryptosync .....	215
Table 30	Encoding of the CryptoAttribute Field .....	216
Table 31	Configurable values .....	217
Table 32	Message bits for AT PAC computation.....	219
Table 33	Message bits for AN PAC computation.....	221
Table 34	Configurable values .....	223
Table 35	Active protocols per Air Link Management Protocol state .....	230
Table 36	Encoding of the RedirectReason field .....	237
Table 37	Computation of $Period_i$ from $SlotCycle_i$ .....	246
Table 38	Encoding of the ConnectRequestReason field.....	253
Table 39	Encoding of the ConnectionStatus field .....	254
Table 40	Encoding of the CloseReason field.....	268
Table 41	Encoding of the SupportedMIMOMode field.....	269
Table 42	Configurable values .....	276
Table 43	The Format of the parameter record for the ActiveSetManagement parameter .....	278

Table 44	Interpretation of FLReservedInterlaces .....	289
Table 45	Interpretation of SSCHNumHopports.....	292
Table 46	Interpretation of SSCHModulationSymbolsPerBlock .....	292
Table 47	SectorInformation group.....	294
Table 48	Interpretation of MACIDRange .....	297
Table 49	Interpretation of AccessCycleDuration.....	299
Table 50	Active Set Management Protocol parameters that are public data of the Overhead Messages Protocol.....	306
Table 51	Interpretation of CQIReportingMode .....	329
Table 52	Interpretation of CQIReportInterval .....	330
Table 53	Interpretation of CQIPilotInterval.....	330
Table 54	Pilot drop timer values .....	336
Table 55	The Format of the parameter record for the ActiveSetManagement parameter .....	341
Table 56	QuickPage format with NumPages=0.....	363
Table 57	QuickPage format with NumPages=1 .....	364
Table 58	QuickPage format with NumPages=2.....	364
Table 59	QuickPage format with NumPages=3.....	364
Table 60	QuickPage format with NumPages=4.....	364
Table 61	QuickPage format with NumPages=5.....	365
Table 62	QuickPage format with NumPages=6.....	365
Table 63	QuickPage format with NumPages=7.....	365
Table 64	Access sequence partitions .....	376
Table 65	Constants for interpreting the access sequence partition table.....	376
Table 66	Mapping the BufferLevel and PilotLevel to access sequence segment .....	377
Table 67	Configurable values .....	380
Table 68	F-SSCH blocks .....	387
Table 69	F-SSCH Block Structure.....	395
Table 70	Base node NodeID to Hop-port Mapping Example ( $N_{\text{CARRIER\_SIZE}}=512$ , $\text{MinHopPortsPerNode}=16$ , and $Q_{\text{SDMA}}=4$ ) .....	421
Table 71	FL packet formats – SISO mode.....	422
Table 72	Encoding of the UATISStatus field .....	425
Table 73	Encoding of the FailureCode field.....	425
Table 74	Configurable attributes .....	429
Table 75	CQI Reports for each CQIReportingMode .....	442
Table 76	Format for CQICHPIlot report.....	442
Table 77	Format for CQICHCTRL report .....	442
Table 78	Format for CQICHSCW report.....	443
Table 79	Format of first part of CQICHMCW report.....	443
Table 80	Format of second part of CQICHMCW report .....	444
Table 81	CQI Mapping to the FL Packet Format and Number of FL-PHY Frames.....	445
Table 82	SFCH Report for each CQIReportingMode .....	447
Table 83	Format for SFCHSISO report .....	447
Table 84	Format for SFCHSCW report.....	448
Table 85	Format of first part of SFCHMCW report .....	448
Table 86	Format of second part of SFCHMCW report .....	449
Table 87	Format of third part of SFCHMCW report .....	449
Table 88	Format for BFCHBeamIndexType1 .....	451

Table 89	Format for BFCHBeamIndexType2 .....	452
Table 90	R-REQCH message format.....	454
Table 91	MaxNumSubCarriers lookup table .....	454
Table 92	Base node NodeID to Hop-port Mapping Example ( $N_{\text{CARRIER\_SIZE}}=512$ , $\text{MinHopPortsPerNode}=16$ , and $Q_{\text{SDMA}}=4$ ) .....	478
Table 93	RL packet formats.....	479
Table 94	Chip duration as a function of $N_{\text{FFT}}$ .....	492
Table 95	Turbo interleaver lookup table definition .....	499
Table 96	QPSK modulation table .....	504
Table 97	8-PSK modulation table.....	505
Table 98	16-QAM modulation table.....	506
Table 99	64-QAM modulation table.....	507
Table 100	Values of the parameters $a_k$ and $b_k$ .....	532
Table 101	ChannelBandRecordType for channel band record .....	595
Table 102	ATIType field encoding.....	597
Table 103	Format of the session state information record .....	606
Table 104	Encoding of the DataType field.....	607
Table 105	Protocol types and subtypes.....	608
Table 106	Transport subtypes assignments .....	609
Table 107	RAT Types and IDs .....	612
Table 108	Protocol types and subtypes.....	612
Table 109	Protocol types and subtypes.....	613
Table 110	Configuration attributes that shall be set to non-default values .....	614
Table 111	Protocol types and subtypes.....	615
Table 112	Configuration attributes that shall be set to non-default values .....	616
Table 113	FlowProfileID format .....	616
Table 114	FlowProfileIDType values.....	616
Table 115	Generic data service profile identifier assignments .....	617
Table 116	Speech service profile identifier assignments .....	618
Table 117	Audio service profile identifier assignments .....	619
Table 118	Video service profile identifier assignments.....	620
Table 119	Text service profile identifier assignments .....	621
Table 120	Signaling service profile identifier assignments .....	621
Table 121	Gaming service profile identifier assignments.....	622
Table 122	User data throughput at various ModClasses.....	669
Table 123	Modulation and Coding Rates .....	670
Table 124	ModClass versus Burst Type .....	671
Table 125	Block Lengths in Downlink Traffic Burst .....	671
Table 126	Block Lengths in Uplink Traffic Burst .....	672
Table 127	Block Lengths in Short Message Broadcast .....	672
Table 128	Mapping for Rate 5/5 Block shaper.....	673
Table 129	Mapping for Rate 6/6 Block shaper .....	674
Table 130	Symbol mapper for Modulation classes 9.....	674
Table 131	Symbol mapper for Modulation classes 10.....	675
Table 132	Modulation Scaling.....	676
Table 133	Logical Cannels, Messages, and Burst Types.....	679
Table 134	SMB fields .....	680



Table 135	Message format for subtypes AA-cts and AA-short.....	681
Table 136	<u>Basic</u> RMU Header Fields (see Table 7.29) .....	682
Table 137	Minimized RMU Header Fields.....	682
Table 138	Classes of QoS Service .....	697
Table 139	QoS Message .....	698
Table 140	Multicast connection rejection values.....	705

1 **Foreword**

2 This foreword is not part of this Standard.

3

## 1   References

- 2           [1] FIPS PUB 180-2, Federal Information Processing Standards Publication 180-2<sup>1</sup>
- 3           [2] IETF RFC 2409, The Internet Key Exchange (IKE)<sup>2</sup>
- 4           [3] Internet Assigned Numbers Authority at <http://www.iana.org/>
- 5           [4] IETF RFC 791, Internet Protocol
- 6           [5] IETF RFC 2460, Internet Protocol, Version 6 (IPv6) Specification
- 7           [6] ITU-T Recommendation E.212, Identification Plan for Land Mobile Stations, 1988<sup>3</sup>
- 8           [7] IETF RFC 3056, Connection of IPv6 Domains via IPv4 Clouds, February 2001
- 9           [8] TR45.AHAG, Enhanced Cryptographic Algorithms, Revision B, March 5, 2002<sup>4</sup>
- 10          [9] IEEE Std 802.11i/D7.0<sup>TM</sup>, “Part 11: Wireless Medium Access Control (MAC) and  
11           Physical layer (PHY) specifications: Specifications for Enhanced Security”, October  
12           2003<sup>5,6</sup>
- 13          [10] IETF RFC 2104, HMAC: Keyed Hashing for Message Authentication, February 1997
- 14          [11] IETF RFC 3095, Robust Header Compression (ROHC): Framework and four profiles:  
15           RTP, UDP, ESP, and uncompressed.
- 16          [12] IETF RFC 2210, The Use of RSVP with IETF Integrated Services
- 17          [13] IETF RFC 2212, Specification of Guaranteed Quality of Service
- 18          [14] IETF RFC 2215, General Characterization Parameters for Integrated Service Network  
19           Elements
- 20          [15] Internet Assigned Numbers Authority ROHC Profile  
21           Identifiers at <http://www.iana.org/assignments/rohc-pro-ids>

---

<sup>1</sup> FIPS publications are available from the National Technical Information Service (NTIS), U. S. Dept. of Commerce, 5285 Port Royal Road, Springfield, VA 22161 (<http://www.ntis.gov/>).

<sup>2</sup> IETF publications are available from the Internet Engineering Task Force at <http://www.ietf.org/>.

<sup>3</sup> ITU-T publications are available from the International Telecommunications Union, Place des Nations, CH-1211, Geneva 20, Switzerland/Suisse (<http://www.itu.int/>).

<sup>4</sup> TR-45.AHAG publications available from <http://ftp.tiaonline.org/TR-45/TR45AHAG/Public/>

<sup>5</sup> IEEE and 802 are registered trademarks in the U.S. Patent & Trademark Office, owned by the Institute of Electrical and Electronics Engineers, Incorporated.

<sup>6</sup> IEEE publications are available from the Institute of Electrical and Electronics Engineers, Inc., 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA (<http://standards.ieee.org/>).

- 1 [16] 3GPP2, S.S0055Av2.0, Enhanced Cryptographic Algorithms, Jan 2005. Available at  
2 [http://www.3gpp2.org/Public\\_html/specs/S.S0055-A\\_v2.0\\_050120.pdf](http://www.3gpp2.org/Public_html/specs/S.S0055-A_v2.0_050120.pdf), also published as  
3 TIA 946-1
- 4 [17] IEEE Std 802.3<sup>TM</sup>-2002, IEEE Standard for Information technology—  
5 Telecommunications and information exchange between systems—Local and  
6 metropolitan area networks—Specific requirements—Part 3: Carrier sense multiple  
7 access with collision detection (CSMA/CD) access method and physical layer  
8 specifications.
- 9 [18] IETF RFC 2578, Structure of Management Information Version 2 (SMIv2), April 1999
- 10 [19] IETF RFC 2579, Textual Conventions for SMIv2, April 1999
- 11 [20] IETF RFC 2580, Conformance Statements for SMIv2, April 1999
- 12 [21] IETF RFC 3513, IP Version 6 Addressing Architecture, April 2003
- 13 [22] IETF RFC 3748, Extensible Authentication Protocol, June 2004
- 14 [23] IETF RFC 4193 Unique Local IPv6 Unicast Addresses, October 2005
- 15 [24] 3GPP2, C.R1001-E v1.0, Administration of Parameter Value Assignments for  
16 cdma2000<sup>TM</sup> Spread Spectrum Standards. <sup>7</sup>
- 17 [25] ATIS-PP-0700004-2005, High Capacity-Spatial Division Multiple Access (HC-SDMA),  
18 September 2005

---

<sup>7</sup> cdma2000<sup>®</sup> publications are available from the Third Generation Partnership Project 2 (3GPP2) (<http://www.3gpp2.org/>).

# 1 Overview

## 1.1 Scope of this document

These technical requirements form a compatibility standard for mobile broadband wireless access systems. The requirements ensure that a compliant access terminal can obtain service through any access network conforming to this standard, thus providing a framework for the rapid development of cost-effective, interoperable multivendor mobile broadband wireless access systems. This compatibility standard is targeted for use in a wide variety of licensed frequency bands.

This specification includes provisions for future service additions and expansion of system capabilities. The architecture defined by this specification permits such expansion without the loss of backward compatibility to older access terminals.

## 1.2 Modes of the specification

This specification has two modes of operation, a Wideband mode and a 625k-MC mode. The Wideband mode is designed to operate for all FDD and TDD bandwidths and is described in Chapters 1 through 12. The 625k-MC mode is designed with 625 KHz carrier bandwidth supporting aggregation of multiple carriers for TDD operation and is described in Chapters 14 through 28.1 and Appendix A.

## 1.3 Requirements language

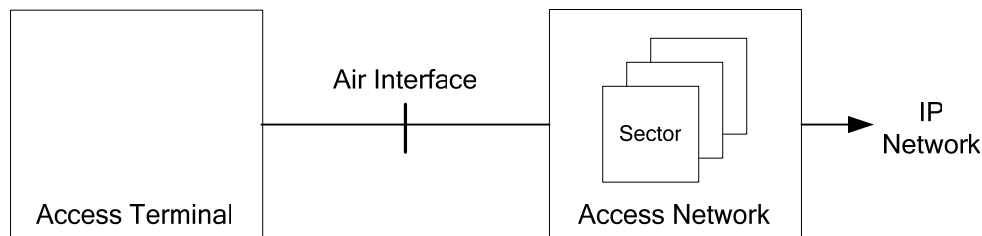
Compatibility, as used in connection with this standard, is understood to mean: Any access terminal can obtain service through any access network conforming to this standard. Conversely, all access networks conforming to this standard can service access terminals.

“Shall” and “shall not” identify requirements to be followed strictly to conform to the standard and from which no deviation is permitted. “Should” and “should not” indicate that one of several possibilities is recommended as particularly suitable, without mentioning or excluding others, that a certain course of action is preferred but not necessarily required, or that (in the negative form) a certain possibility or course of action is discouraged but not prohibited. “May” and “need not” indicate a course of action permissible within the limits of the standard. “Can” and “cannot” are used for statements of possibility and capability, whether material, physical, or causal.

## 1.4 Wideband Mode Overview

### 1.4.1 Architecture reference model

The architecture reference model is presented in Figure 1. The reference model includes the air interface between the access terminal and the access network. The protocols used over the air interface are defined in this document.



**Figure 1 Architecture reference model**

The functional units of the reference architecture in Figure 1 are:

**Access Network (AN)** The network equipment providing Layer 3 connectivity between an IP network (typically the Internet) and the access terminals.

**Access Terminal (AT)** A device providing data connectivity to a user. An access terminal may be connected to a computing device such as a laptop personal computer or it may be a self-contained data device such as a personal digital assistant.

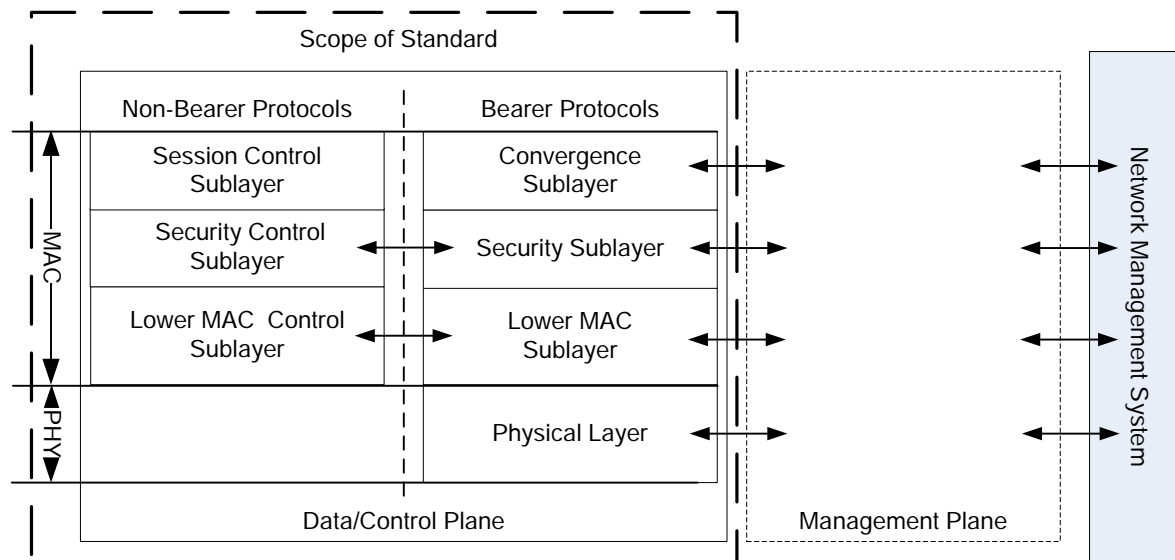
**Sector** One set of physical layer channels transmitted between the access network and the access terminals within a given frequency assignment. A sector consists of a reverse link ChannelBand and a forward link ChannelBand.

### 1.4.2 Protocol architecture

The air interface is layered, with interfaces defined for each layer (and for each protocol within each layer). This architecture allows future modifications to a layer or to a protocol to be isolated.

### 1.4.2.1 Layers

Figure 2 describes the layering architecture for the air interface. Each layer consists of one or more protocols that perform the layer's functionality.



**Figure 2 Air interface layering architecture**

The protocols and layers specified in Figure 2 are:

#### Session Control Sublayer

The Session Control Sublayer provides address management, protocol negotiation, protocol configuration, and state maintenance services. The Session Control Sublayer is a non-bearer layer and, therefore, it does not carry payload on behalf of other layers. The Session Control Sublayer is defined in Chapter 2.

#### Convergence Sublayer

The Convergence Sublayer provides protocols and transports used to transport messages and data, and provides multiplexing of distinct transports. For example, it provides the Signaling Transport for transporting air interface protocol messages and the Data Transport for transporting user data. The Convergence Sublayer is defined in Chapter 3.

#### Security Control Sublayer

The Security Control Sublayer provides key exchange for use by the Security Sublayer. The Security Control Sublayer is defined in Chapter 4.

#### Security Sublayer

The Security Sublayer provides authentication and encryption services. The Security Sublayer is defined in Chapter 5.

#### Lower MAC Control Sublayer

The Lower Medium Access Control (MAC) Control Sublayer provides air-link connection establishment and maintenance services. The Lower MAC Control Sublayer is defined in Chapter 6.

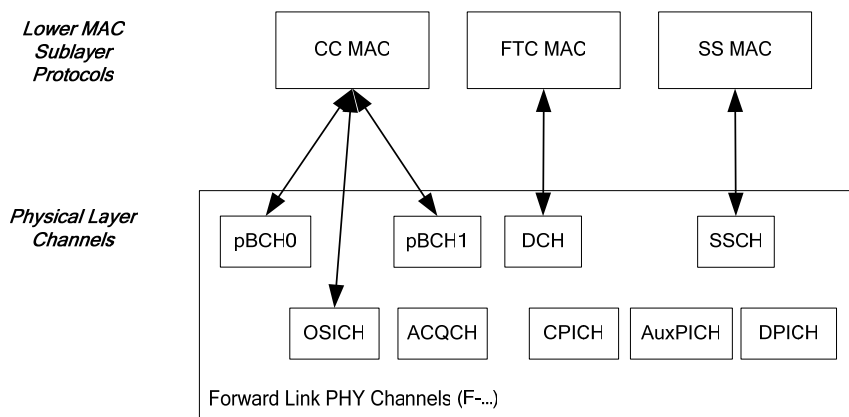
1 Lower MAC Sublayer The Lower MAC Sublayer defines the procedures used to receive and to  
 2 transmit over the Physical Layer. The Lower MAC Sublayer is defined in  
 3 Chapter 7.

4 Physical Layer The Physical Layer provides the channel structure, frequency, power output,  
 5 modulation, and encoding specifications for the Forward and Reverse  
 6 Channels. The Physical Layer is defined in Chapter 8.

7 Each layer may contain one or more protocols or transports. Protocols use signaling messages, in-  
 8 band messages, blocks, or headers to convey information to their peer protocols at the other side of  
 9 the air-link. When protocols send messages, they use the Signaling Network Protocol (SNP) to  
 10 transmit these messages. Transports send signaling messages using the Signaling Network Protocol.  
 11 Blocks are information conveyed to a peer protocol using an encapsulation that is specific to a  
 12 Physical Layer Channel. For example, the Lower MAC Control Sublayer Overhead Messages  
 13 Protocol uses the SystemInfo block to carry information to its peer protocol at the access terminal on  
 14 the forward primary broadcast channel 0 (pBCH0).

### 15 1.4.3 Physical layer channels

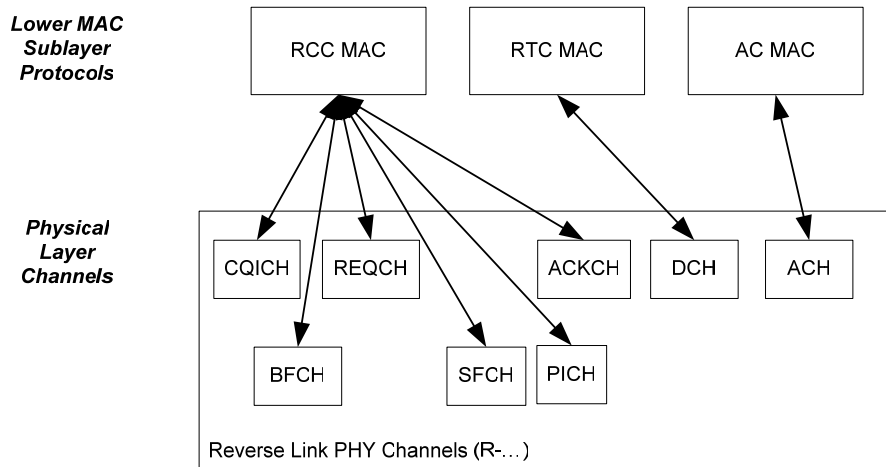
16 The hierarchies between the Lower MAC Sublayer Protocols and the Physical Layer Channels for the  
 17 forward and reverse links are shown in Figure 3 and Figure 4. The following is a brief description of  
 18 each Physical Layer Channel. A more complete description is provided in Chapter 9. When the  
 19 context is clear, the complete qualified name is usually omitted (e.g., Quick Paging Channel as  
 20 opposed to Forward Quick Paging Channel or Data Channel as opposed to Reverse Data Channel).



21

22 **Figure 3 Forward channel structure**





1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26

**Figure 4 Reverse channel structure**

### 1.4.3.1 Forward physical channels

#### Forward Acquisition Channel (F-ACQCH)

Carries an acquisition pilot for an access terminal to use to acquire the system.

#### Forward Auxiliary Pilot Channel (F-AuxPICH)

Carries auxiliary pilots for channel estimation from multiple transmit antennas. The Forward Primary Broadcast Channel 1 (F-pBCH1) indicates whether the F-AuxPICH is present.

#### Forward Common Pilot Channel (F-CPICH)

Carries the common pilot.

#### Forward Data Channel (F-DCH)

Carries information for a specific access terminal. A Forward Data Channel assignment is assigned to an access terminal by a Forward Shared Signaling Channel (F-SSCH) assignment. Also carries broadcast information including pages and sector specific messages.

#### Forward Dedicated Pilot Channel (F-DPICH)

Carries the dedicated pilot. This channel is present in BlockHopping mode, which is indicated over the Forward Primary Broadcast Channel 0 (F-pBCH0).

#### Forward Other Sector Interference Pilot Channel (F-OSICH)

Carries information about the interference from other sectors to be received by all access terminals.

#### Forward Primary Broadcast Channel 0 (F-pBCH0)

Carries information about the system to be received by all access terminals.

1 Forward Primary Broadcast Channel 1 (F-pBCH1)  
 2 Carries information about the sector to be received by all access terminals.  
 3 Also carries quick pages.

4 Forward Shared Signaling Channel (F-SSCH)  
 5 Carries forward and reverse link data channel assignments, access grants,  
 6 power control commands, and acknowledgement information for Reverse  
 7 Data Channel (R-DCH) receptions.

### 8 **1.4.3.2 Reverse physical channels**

9 Reverse Access Channel (R-ACH)  
 10 Used by access terminals to initiate communication with the access network.  
 11 The Reverse Access Channel is also used by access terminals to obtain  
 12 timing corrections.

13 Reverse Acknowledgement Channel (R-ACKCH)  
 14 Carries acknowledgement information of a Forward Data Channel (F-DCH)  
 15 reception.

16 Reverse Beam Feedback Channel (R-BFCH)  
 17 Carries information about the beam index and the quality of the forward link  
 18 channel.

19 Reverse Channel Quality Indicator Channel (R-CQICH)  
 20 Carries information about the quality of the forward link channel of a sector  
 21 as received by an access terminal. The Reverse Channel Quality Indicator  
 22 Channel also carries information about the desired forward link serving  
 23 sector.

24 Reverse Data Channel (R-DCH)  
 25 Carries information from an access terminal. The Reverse Data Channel is  
 26 assigned to an access terminal by a Forward Shared Signaling Channel (F-  
 27 SSCH) assignment.

28 Reverse Pilot Channel (R-PICH)  
 29 Carries the pilot.

30 Reverse Request Channel (R-REQCH)  
 31 Carries information about the buffer level at different quality of service  
 32 classes for an access terminal. The Reverse Request Channel also carries  
 33 information about the desired reverse link serving sector.

34 Reverse Subband Feedback Channel (R-SFCH)  
 35 Carries information about the quality of a subband of the forward link  
 36 channel.

## 1.4.4 Protocols

### 1.4.4.1 Interfaces

This standard defines a set of interfaces for communications between protocols in the same entity and between a protocol executing in one entity and the same protocol executing in a peer entity.

In the following the generic term “entity” is used to refer to the access terminal and the access network.

Protocols in this specification have five types of interfaces:

#### Headers, messages, and blocks

Used for communications between a protocol executing in one entity and the same protocol executing in a peer entity.

#### Commands

Used by a protocol to obtain a service from another protocol within the same entity. For example, *AccessChannelMAC.Deactivate* causes the Access Channel MAC Protocol to abort any access attempt currently in progress.

#### Indications

Used by a protocol to convey information regarding the occurrence of an event to another protocol within the same entity. Any protocol can register to receive these indications. For example, the access terminal Control Channel MAC Protocol returns a “Supervision Failed” indication when it is unable to receive the sector parameters message for a certain amount of time. This notification is then used by the Lower MAC Control Sublayer Air Link Management Protocol to close the connection.

#### Static Public Data

Used to share information in a controlled way between protocols/ transports. Static public data is shared between protocols/ transports in the same layer, as well as between protocols/ transports in different layers. Static public data is independent of the InUse SessionConfigurationToken and is supported by all subtypes of a protocol. For example the UATI is static public data for a protocol in the Session Control Sublayer. New protocol subtypes may define additional static public data. The additional static public data shall be initialized by the new protocol subtype when it is created.

#### Dynamic Public Data

Used to share information in a controlled way between protocols/ transports. Dynamic public data is shared between protocols/ transports in the same layer, as well as between protocols/ transports in different layers. Dynamic public data is a function of the InUse SessionConfigurationToken and is defined separately for each subtype of a protocol. For example the protocol subtype is always dynamic public data for a protocol.

Commands and indications are written in the form of *Protocol.Command* and *Protocol.Indication*. For example, *AccessChannelMAC.Activate* is a command activating the Access Channel MAC, and *IdleState.ConnectionOpened* is an indication provided by the Lower MAC Control Sublayer Idle State Protocol indicating that the connection is now open. When the context is clear, the *Protocol* part is dropped (e.g., within the Idle State Protocol, *Activate* refers to *IdleState.Activate*).

1 Commands are always written in the imperative form, since they direct an action. Indications are  
2 always written in the past tense since they notify of events that have happened (e.g., *OpenConnection*  
3 for a command and *ConnectionOpened* for an indication).

4 Headers, messages, and blocks are binding on all implementations. Commands, indications, and  
5 public data are used as devices to help ensure a clear and precise specification. Access terminals and  
6 access networks can be compliant with this specification while choosing a different implementation  
7 that exhibits identical behavior.

#### 8 **1.4.4.2 States**

9 When protocols exhibit different behavior as a function of the environment (e.g., if a connection is  
10 opened or not, if a session is opened or not, etc.), this behavior is captured in a set of states and events  
11 leading to a transition between states.

12 Unless otherwise specifically mentioned, the state of the access network refers to the state of a  
13 protocol engine in the access network as it applies to a particular access terminal. Since the access  
14 network communicates with multiple access terminals, multiple independent instantiations of a  
15 protocol will exist in the access network, each with its own independent state machine.

16 Unless otherwise specifically shown, the state transitions due to failure are not shown in the figures  
17 for the state transition diagrams.

18 Typical events leading to a transition from one state to another are the receipt of a message, a  
19 command an indication, or the expiration of a timer.

20 When a protocol is not functional at a particular time (e.g., the Access Channel MAC protocol at the  
21 access terminal when the access terminal has an open connection), the protocol is placed in a state  
22 called the Inactive state. This state is common for most protocols.

23 Other common state names are Open, indicating that the session or connection (as applicable to the  
24 protocol) is open, and Close, indicating that the session or connection is closed.

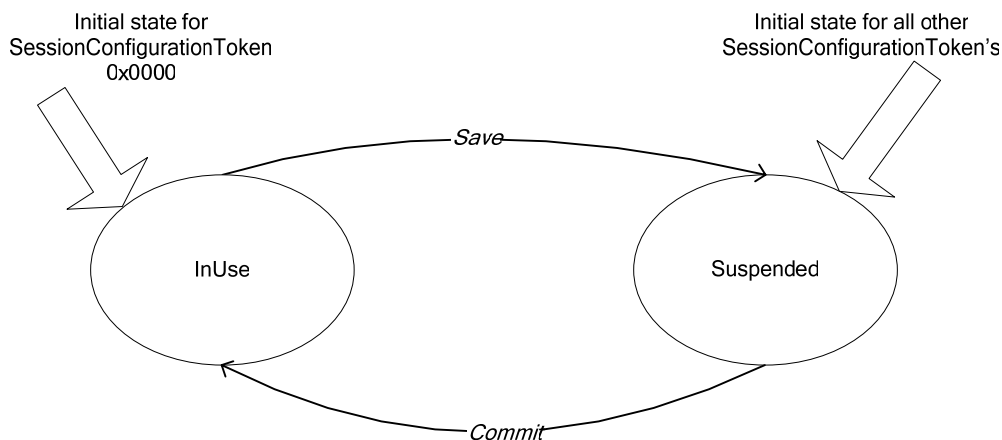
25 If a protocol has a single state other than the Inactive state, that state is usually called the Active state.  
26 If a protocol has more than one state other than the Inactive state, all of these states are considered  
27 active, and are given individual names (e.g., the Address Management Protocol at the access network  
28 has three states: Inactive, Setup, and Open).

#### 29 **1.4.4.3 SessionConfigurationToken**

30 The SessionConfigurationToken is a 16 bit value that defines a complete set of protocol and transport  
31 instances that can be used to communicate between the access terminal and the access network.

32 A SessionConfigurationToken is InUse if the set of protocol and transport instances specified by the  
33 SessionConfigurationToken are currently being used to communicate between the access terminal and  
34 the access network. Otherwise, a SessionConfigurationToken is Suspended. Only one  
35 SessionConfigurationToken shall be InUse at a time.

1 The Session Configuration Protocol executes its save and commit procedures to swap the InUse  
 2 SessionConfigurationToken with a Suspended SessionConfigurationToken as shown in Figure 5.



3

4 **Figure 5 SessionConfigurationToken state diagram**

#### 5 **1.4.4.4 InUse and Suspended protocol/transport instances**

6 A protocol or transport instance is InUse if it is currently being used to communicate between the  
 7 access terminal and the access network. Otherwise, a protocol or transport instance is Suspended.  
 8 Only one protocol instance of a protocol type shall be InUse at a time. Each transport maps to a  
 9 Transport defined in the Packet Consolidation Protocol. Only one transport instance corresponding to  
 10 a Transport shall be InUse at a time. A protocol or transport instance shall correspond to exactly one  
 11 SessionConfigurationToken.

12 The Session Configuration Protocol executes its save and commit procedures to swap the InUse  
 13 protocol and transport instances associated with the current InUse SessionConfigurationToken with  
 14 the Suspended protocol and transport instances associated with a Suspended  
 15 SessionConfigurationToken.

16 Once the access terminal and access network agree upon using a new SessionConfigurationToken, the  
 17 InUse protocol or transport instances associated with the current InUse SessionConfigurationToken  
 18 are saved and the Suspended protocol or transport instances associated with the new  
 19 SessionConfigurationToken are swapped in.

#### 20 **1.4.4.4.1 Protocol initialization and swap**

21 The initialization procedures for a protocol/transport instance are invoked upon creation of the  
 22 protocol/transport instance. A protocol/transport instance shall be created before it can become an  
 23 InUse or Suspended protocol/transport instance.

24 The swap procedures for a Suspended protocol/transport instance are invoked when the Session  
 25 Configuration Protocol performs its commit procedure to change the InUse  
 26 SessionConfigurationToken.

1 If the swap procedure for a Suspended protocol/transport instance sets the state of the InUse protocol  
2 instance to a particular initial state, the procedures associated with entering the initial state are  
3 executed upon entering the initial state.

#### 4 **1.4.4.5 Procedures and messages**

5 Each protocol/transport specifies procedures, blocks, and messages corresponding to the InUse  
6 protocol/transport instances.

#### 7 **1.4.4.6 Common commands**

8 Most protocols support the following two commands:

- 9 ■ *Activate*, which commands the protocol to transition from the Inactive state to some other  
10 state.
- 11 ■ *Deactivate*, which commands the protocol to transition to the Inactive state. Some  
12 protocols do not transition immediately to the Inactive state, due to requirements on  
13 orderly cleanup procedures.

14 Other common commands are *Open* and *Close*, which command protocols to perform session  
15 open/close or connection open/close related functions.

#### 16 **1.4.4.7 Attribute negotiation**

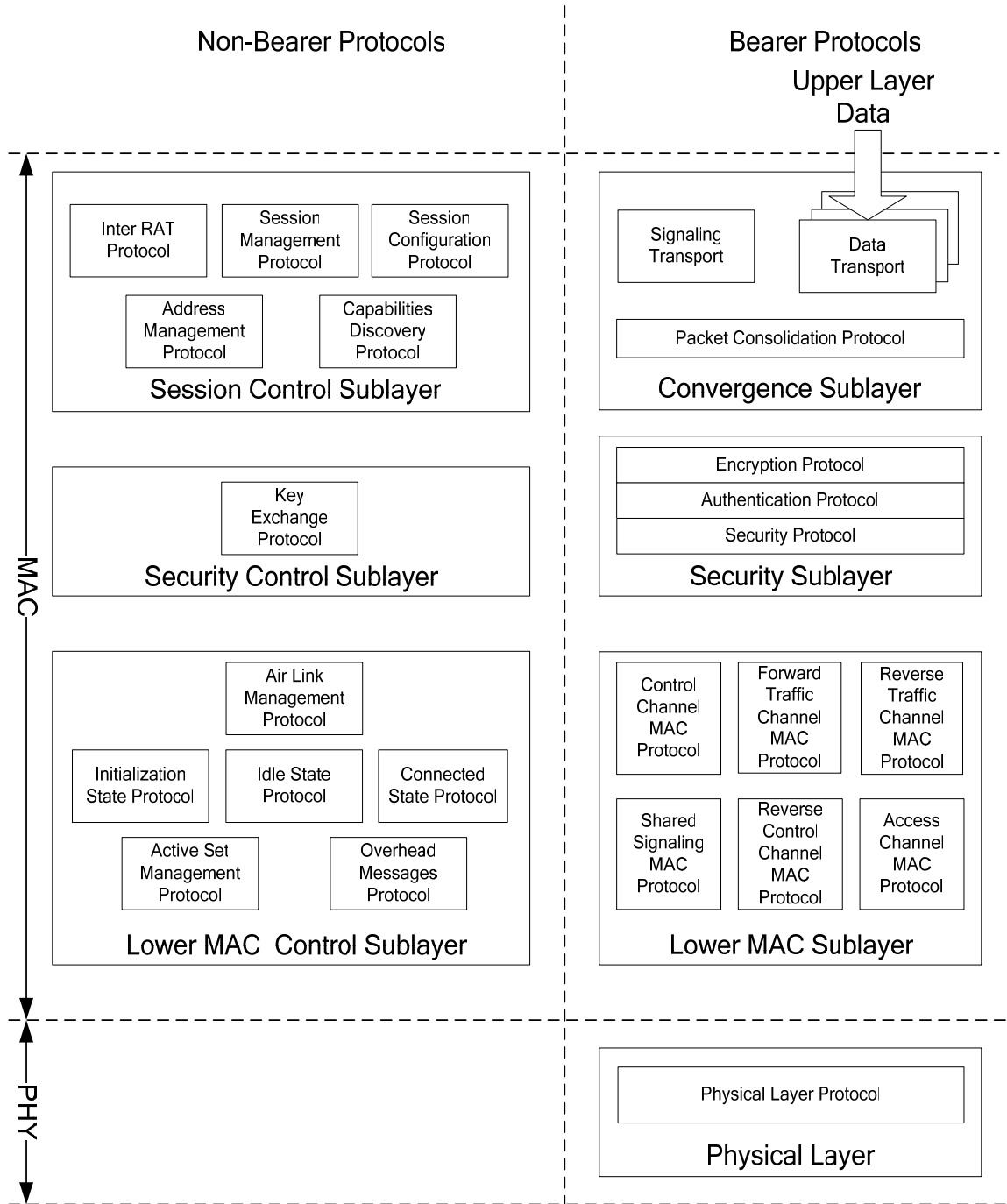
17 The Generic Attribute Update Protocol provides a means to update protocol and transport attributes.  
18 The protocol uses an AttributeUpdateRequest message, an AttributeUpdateAccept message, and an  
19 AttributeUpdateReject message to negotiate a mutually acceptable configuration. Only the protocol  
20 and transport attributes of the InUse protocol and transport instances may be configured using the  
21 Generic Attribute Update Protocol.

22 The Default Session Configuration Protocol defines a ConfigurationRequest message, a  
23 ConfigurationAccept message, and a ConfigurationReject message to update protocol and transport  
24 attributes for Suspended protocol and transport instances.

25 Protocols are associated with a Type that denotes the type of the protocol (e.g., Access Channel MAC  
26 Protocol) and with a Subtype that denotes a specific instance of a protocol (e.g., the Default Access  
27 Channel MAC Protocol).

1 **1.4.4.8 Protocol overview**

2 Figure 6 presents the protocol and transport types defined for each of the layers shown in Figure 2.  
 3 The following is a brief description of each protocol and transport. A more complete description is  
 4 provided in the introduction section of each layer.



5  
 6 **Figure 6 Protocol and transport types**

- 1
- 2 ■ Session Control Sublayer:
    - 3 □ Session Management Protocol: Provides means to control the activation and the
    - 4 deactivation of the Address Management Protocol, Capabilities Discovery Protocol
    - 5 and the Session Configuration Protocol. It also provides a session keep-alive
    - 6 mechanism.
    - 7 □ Address Management Protocol: Provides unicast access terminal identifier (UATI)
    - 8 management.
    - 9 □ Capabilities Discovery Protocol: Provides means for the access network to discover
    - 10 the capabilities of the access terminal.
    - 11 □ Session Configuration Protocol: Provides means for negotiation of the
    - 12 SessionConfigurationToken used in the session.
    - 13 □ Inter RAT Protocol: Provides the means to send messages for other radio access
    - 14 technologies.
  - 15 ■ Convergence Sublayer:
    - 16 □ Signaling Transport: Provides message transmission services, including
    - 17 fragmentation mechanisms, along with reliable and best-effort delivery mechanisms
    - 18 for signaling messages.
    - 19 □ Data Transport: Provides two route instances for packets for a higher layer packet
    - 20 flow, as well as retransmission and duplicate detection for the packet flow of each
    - 21 route instance. It also provides procedures to enable and disable the Data Transport
    - 22 data flow.
    - 23 □ Packet Consolidation Protocol: Adds the Packet Consolidation Protocol header to
    - 24 transport packets prior to transmission; and, after reception, removes the Packet
    - 25 Consolidation Protocol header and forwards the transport packets to the correct
    - 26 transport. Provides transmit prioritization and packet encapsulation for the
    - 27 Convergence Sublayer.
  - 28 ■ Security Control Sublayer:
    - 29 □ Key Exchange Protocol: Provides the procedures followed by the access network and
    - 30 the access terminal to exchange security keys for authentication and encryption.
  - 31 ■ Security Sublayer:
    - 32 □ Authentication Protocol: Provides the procedures followed by the access network and
    - 33 the access terminal for authenticating traffic.
    - 34 □ Encryption Protocol: Provides the procedures followed by the access network and the
    - 35 access terminal for encrypting traffic.
    - 36 □ Security Protocol: Provides procedures for generating a cryptosync based on the
    - 37 information fetched from the Lower MAC Sublayer that can be used by the
    - Authentication Protocol and the Encryption Protocol.



- 1       ■ Lower MAC Control Sublayer:
    - 2           □ Air Link Management Protocol: Provides the overall state machine management that
3           □           an access terminal and an access network follow for the Lower MAC Control
4           □           Sublayer.  - 5           □ Initialization State Protocol: Provides the procedures that an access terminal follows
6           □           to acquire a network and that an access network follows to support network7           □           acquisition.  - 8           □ Idle State Protocol: Provides the procedures that an access terminal and an access
9           □           network follow when a connection is not open.  - 10          □ Connected State Protocol: Provides the procedures that an access terminal and an
11          □           access network follow when a connection is open.  - 12          □ Active Set Management Protocol: Provides the means to maintain the active set
13          □           between the access terminal and the access network.  - 14          □ Overhead Messages Protocol: Provides broadcast messages and blocks containing
15          □           information that is mostly used by Lower MAC Control Sublayer protocols.
- 16       ■ Lower MAC Sublayer:
  - 17          □ Control Channel MAC Protocol: Provides the procedures followed by the access
18          □           network to transmit, and by the access terminal to receive, the Control Channels.
- 19          □ Access Channel MAC Protocol: Provides the procedures followed by the access
- 20          □           terminal to transmit, and by the access network to receive, the Access Channel.
- 21          □ Shared Signaling MAC Protocol: Provides the procedures followed by the access
- 22          □           network to transmit, and by the access terminal to receive, the physical layer channels23          □           controlled by this protocol.
- 24          □ Forward Traffic Channel MAC Protocol: Provides the procedures followed by the
- 25          □           access network to transmit, and by the access terminal to receive, the Forward Traffic26          □           Channel.
- 27          □ Reverse Control Channel MAC Protocol: Provides the procedures for the access
- 28          □           terminal to transmit, and the access network to receive, the Reverse Control29          □           Channels.
- 30          □ Reverse Traffic Channel MAC Protocol: Provides the procedures followed by the
- 31          □           access terminal to transmit, and by the access network to receive, the Reverse Traffic32          □           Channel.
- 33       ■ Physical Layer:
  - 34          □ Physical Layer Protocol: Provides channel structure, frequency, power output, and
35          □           modulation specifications for the forward and reverse links.

## 1.4.5 Default transports

This document defines two default transports that all compliant access terminals and access networks support:

- Default Signaling Transport: Provides the means to carry messages between a protocol/transport in one entity and the same protocol/transport in a peer entity. The Default Signaling Transport consists of a messaging protocol (Signaling Network Protocol) and a link layer protocol that provides message fragmentation, retransmission, and duplicate detection (Signaling Link Protocol).
- Default Data Transport: Consists of a link layer protocol that provides fragmentation, retransmission, and duplicate detection (Radio Link Protocol); a Route Selection Protocol that provides two route instances for a higher layer packet flow; and a Flow Control Protocol that provides flow control of data traffic.

The transports used are negotiated as part of session negotiation.

The air interface can support up to 8 parallel transports. The first transport (Transport0) always carries Signaling. Other transport can be used to carry, for example, the Default Data Transport to support different Quality of Service (QoS) requirements for data or other transports.

## 1.4.6 Sessions and connections

A session refers to a shared state between the access terminal and the access network. This shared state stores the protocols and protocol configurations that were negotiated and are used for communications between the access terminal and the access network.

Other than to open a session, an access terminal cannot communicate with an access network without having an open session.

A connection is a particular state of the air-link in which the access terminal is assigned a MACID. It also may be assigned a Forward Traffic Channel and/or a Reverse Traffic Channel.

During a single session, the access terminal and the access network can open and close a connection multiple times.

## 1.4.7 Security

The air interface supports a Security Sublayer, which can be used for authentication and encryption of access terminal traffic transported by the Control Channel, the Access Channel, the Forward Traffic Channel, and the Reverse Traffic Channel.

## 1.4.8 Physical Layer modes

The Physical Layer consists of two different duplexing modes, two different forward link hopping modes, two different synchronization modes and two different multi-carrier modes. The possible duplexing modes are Time Division Duplexing (TDD) and Frequency Division Duplexing (FDD). The different forward link hopping modes are SymbolRateHopping and BlockHopping. The possible synchronization modes are SemiSynchronous and Asynchronous. The possible multi-carrier modes are MultiCarrierOn and MultiCarrierOff.

### 1.4.8.1 MIMO support

The air interface supports two modes of operation for the Lower MAC Sublayer and Physical Layer

- Single-input-single-output (SISO) mode, and
- Multiple-input-multiple-output (MIMO) mode.

The MIMO mode is divided into two sub-modes: multiple codeword (MCW) and single codeword (SCW).

#### 1.4.8.1.1 Access terminal requirements

All terminals shall support SISO mode. A MIMO-capable terminal shall support either the MIMO SCW or MCW sub-mode. The MIMO mode requires at least two antennas at the access network and at least two antennas at the access terminal.

### 1.4.9 Management Information Base

The standard includes the definition of a Management Information Base (MIB) module for managing the MAC and PHY. The objects in this MIB are defined using the mechanisms specified in the Structure of Management Information (SMI). The MIB module specified is compliant to SMIV2 which is described in RFC 2578 [18], RFC 2579 [19] and RFC 2580 [20]. The MIB is defined in Chapter 13.

### 1.4.10 Definitions

**ChannelBand** The set of channels transmitted between the access network and the access terminals within a given frequency assignment. A ChannelBand consists of a Forward Link and a Reverse Link.

**Cell** A group of one or more sectors that transmit from a common geographical location.

**Dedicated Resource** An access network resource required to provide data service to the access terminal that is granted to the access terminal only when an access terminal has an open connection. Power control and rate control are not considered dedicated resources.

**Effective Isotropically Radiated Power (EIRP)** The product of the power supplied to the antenna and the antenna gain in a direction relative to an isotropic antenna.

**Effective Radiated Power (ERP)** The product of the power supplied to the antenna and its gain relative to a half-wave dipole in a given direction.

**Empty Procedure** A procedure that performs no operations.

**Forward Link PHY Frame** The forward link PHY frame consists of  $N_{\text{FRAME}, F}$  OFDM symbols, see 8.1.11.

1	Global Positioning System (GPS)	
2		A US government satellite system that provides location and time
3		information to users. See Navstar GPS Space Segment/Navigation User
4		Interfaces ICD-GPS-200 for specifications.
5	NULL	A value which is not in the specified range of the field.
6	OFDM Symbol	An OFDM symbol is comprised of individually modulated subcarriers which
7		carry complex-valued data.
8	PHY Frame Index	An integer value $f$ such that $f = (n \times s) + m$ , where $n$ is the number of PHY
9		frames in a superframe, $s$ is the superframe index, and $m$ is the offset of the
10		frame in the current superframe, where $0 \leq m < n$ . The current frame index is
11		specific to a sector and link.
12	Reservation.	Air interface resources set up by the access network to carry a higher layer
13		flow. A Reservation is identified by its ReservationLabel. ReservationLabels
14		are bound to Link Flows that carry higher layer flows. A Reservation can be
15		either in the Open or Close state.
16	Reverse Link PHY Frame	
17		The reverse link PHY frame consists of $N_{\text{FRAME, R}}$ OFDM symbols,
18		see 8.1.11.
19	Subnet Mask (of length $n$ )	
20		A 128-bit value whose binary representation consists of $n$ consecutive ‘1’s
21		followed by $128-n$ consecutive ‘0’s.
22	Superframe	One of the fundamental units of transmission on the forward and reverse
23		links. On the Forward Link, a superframe consists of $N_{\text{PREAMBLE}}$ OFDM
24		symbols followed by $N_{\text{FDD, FL PHY Frames}}$ Forward Link PHY Frames in FDD
25		mode and $N_{\text{TDD, FL PHY Frames}}$ Forward Link PHY Frames in TDD mode. On the
26		Reverse Link, a superframe consists of $N_{\text{FDD, RL PHY Frames}}$ RL PHY Frames in
27		FDD mode and $N_{\text{TDD, RL PHY Frames}}$ RL PHY Frames in TDD mode. $N_{\text{PREAMBLE}}$
28		is defined in Chapter 8; $N_{\text{FDD, FL PHY Frames}}$ , $N_{\text{TDD, FL PHY Frames}}$ , $N_{\text{FDD, RL PHY Frames}}$
29		and $N_{\text{TDD, RL PHY Frames}}$ are defined in Chapter 7.
30	Superframe Index	An integer value $s$ such that: $s = \lfloor t / x \rfloor$ , where $t$ represents System Time in
31		seconds and $x$ represents the time for a superframe in seconds as defined in
32		Chapter 9. Whenever the document refers to the System Time in
33		superframes, it is referring to the value $s$ .
34	System Time	The time reference used by the system measured in seconds. System time is
35		defined in 1.4.14.
36	Universal Coordinated Time (UTC)	
37		An internationally agreed-upon time scale maintained by the Bureau
38		International de l’Heure (BIH) used as the time reference by nearly all
39		commonly available time and frequency distribution systems.
40	VCQI	Long term averaged CQI value per interlace and subband.

### 1.4.11 Abbreviations and acronyms

ACK	Acknowledgment
ALMP	Air Link Management Protocol
AMP	Address Management Protocol
AN	Access network
ASMP	Active Set Management Protocol
AT	Access terminal
ATA	Access terminal assignment
ATI	Access terminal identifier
BATI	Broadcast access terminal identifier
BCD	Binary-coded decimal
BE	Best effort
BPSK	Binary phase shift keying
CC	Control channel
CDMA	Code division multiple access
CQI	Channel quality indicator
CRC	Cyclic redundancy check
CSP	Connected State Protocol
EIRP	Effective isotropically radiated power
F-ACQCH	Forward Acquisition Channel
F-AuxPICH	Forward Auxiliary Pilot Channel
F-CPICH	Forward Common Pilot Channel
F-DCH	Forward Data Channel
F-DPICH	Forward Dedicated Pilot Channel
F-pBCH	Forward Primary Broadcast Channel
F-OSICH	Forward Other Sector Interference Channel
F-SSCH	Forward Shared Signaling Channel
FCP	Flow control protocol
FCS	Frame check sequence
FDD	Frequency Division Duplex
FER	Frame error ratio
FFT	Fast Fourier Transform
FIFO	First in first out
FL	Forward link
FLAB	Forward link assignment block
FTC	Forward traffic channel
FWD	Forward
GPS	Global positioning system

H-ARQ	Hybrid automatic retransmission request
HW	Hardware
IFFT	Inverse Fast Fourier Transform
IFT	Inverse Fourier Transform
IP	Internet protocol
LAN	Local area network
LSB	Least significant bit
MAC	Medium access control
MATI	Multicast access terminal identifier
MCC	Mobile Country Code
MCW	Multiple codeword
MIB	Management Information Base
MIC	Message Integrity Check
MIMO	Multiple input multiple output
MNC	Mobile Network Code
MSB	Most significant bit
N <sub>fft</sub>	FFT size
N/A	Not applicable
NAK	Negative acknowledgement
OFDM	Orthogonal frequency division multiplexing
OMP	Overhead Messages Protocol
OSICH	Other sector interference channel
PBRI	Pruned Bit-Reversal Interleaver
PCP	Packet Consolidation Protocol
PDU	Protocol data unit
PER	Packet error rate
PF	Packet format
PHY	Physical layer
PMK	Pairwise Master Key
PN	Pseudo noise (code sequence)
QAM	Quadrature amplitude modulation
QoS	Quality of service
QPSK	Quadrature phase shift keying
R-ACH	Reverse Access Channel
R-ACKCH	Reverse Acknowledgement Channel
R-BFCH	Reverse Beam Feedback Channel
R-CQICH	Reverse Channel Quality Indicator Channel
R-DCH	Reverse Data Channel

R-PICH	Reverse Pilot Channel
R-REQCH	Reverse Request Channel
R-SFCH	Reverse Subband Feedback Channel
RD	Reliable delivery
REV	Reverse
RF	Radio frequency
RLAB	Reverse link assignment block
RLP	Radio link protocol
ROHC	Robust header compression
RSP	Route selection protocol
RTC	Reverse traffic channel
RX	Receive or receiver
SCP	Session Configuration Protocol
SCW	Single codeword
SDMA	Space Division Multiple Access
SID	System Identifier
SISO	Single input single output
SLP	Signaling link protocol
SMP	Session Management Protocol
SNP	Signaling network protocol
SS	Shared signaling
SYNC	Synchronization
T2P	Traffic to pilot transmit power ratio
TCP	Transmission control protocol
TDD	Time Division Duplex
TL	Time limited
TX	Transmit or transmitter
UATI	Unicast access terminal identifier
UTC	Universal Temps Coordinate (See Universal Coordinated Time)
WLAN	Wireless LAN

## 1.4.12 Notation

1		
2	$A[i]$	The $i^{\text{th}}$ element of array A. The first element of the array is $A[0]$ .
3	$\langle e_1, e_2, \dots, e_n \rangle$	A <i>structure</i> with elements ‘ $e_1$ ’, ‘ $e_2$ ’, ..., ‘ $e_n$ ’.
4		Two structures $E = \langle e_1, e_2, \dots, e_n \rangle$ and $F = \langle f_1, f_2, \dots, f_m \rangle$ are equal if ‘ $m$ ’ is
5		equal to ‘ $n$ ’ and $e_i$ is equal to $f_i$ for $i=1, \dots, n$ .
6		Given $E = \langle e_1, e_2, \dots, e_n \rangle$ and $F = \langle f_1, f_2, \dots, f_n \rangle$ , the assignment “ $E = F$ ”
7		denotes the following set of assignments: $e_i = f_i$ , for $i=1, \dots, n$ .
8	$S.e$	The member of the structure ‘S’ that is identified by ‘ $e$ ’.
9	$M[i:j]$	Bits $i^{\text{th}}$ through $j^{\text{th}}$ inclusive ( $i \geq j$ ) of the binary representation of variable M.
10		$M[0:0]$ denotes the least significant bit of M.
11		Concatenation operator. ( $A   B$ ) denotes variable A concatenated with
12		variable B.
13	$\times$	Indicates multiplication.
14	$\lfloor x \rfloor$	Indicates the largest integer less than or equal to $x$ : $\lfloor 1.1 \rfloor = 1, \lfloor 1.0 \rfloor = 1$ .
15	$\lceil x \rceil$	Indicates the smallest integer greater than or equal to $x$ : $\lceil 1.1 \rceil = 2, \lceil 2.0 \rceil = 2$ .
16	$ x $	Indicates the absolute value of $x$ : $ -17 =17,  17 =17$ .
17	$\oplus$	Indicates exclusive OR (modulo-2 addition).
18	$\min(x, y)$	Indicates the minimum of $x$ and $y$ .
19	$\max(x, y)$	Indicates the maximum of $x$ and $y$ .
20	$x \bmod y$	Indicates the remainder after dividing $x$ by $y$ : $x \bmod y = x - (y \times \lfloor x/y \rfloor)$ .
21		Unless otherwise specified, the format of field values is unsigned binary.
22		Unless indicated otherwise, this standard presents numbers in decimal form. Binary numbers are
23		distinguished in the text by the use of single quotation marks. Hexadecimal numbers are distinguished
24		by the prefix ‘0x’.
25		Unless specified otherwise, each field of a packet shall be transmitted in sequence such that the most
26		significant bit (MSB) is transmitted first and the least significant bit (LSB) is transmitted last. The
27		MSB is the left-most bit in the figures in this document. If there are multiple rows in a table, the top-
28		most row is transmitted first. If a table is used to show the sub-fields of a particular field or variable,
29		the top-most row consists of the MSBs of the field. Within a row in a table, the left-most bit is
30		transmitted first. Notations of the form “repetition factor of N” or “repeated N times” mean that a
31		total of N versions of the item are used.



### 1.4.13 Malfunction detection

The access terminal shall have a malfunction timer that is separate from and independent of all other functions and that runs continuously whenever power is applied to the transmitter of the access terminal. The timer shall be reset and restarted periodically if the access terminal is functioning properly. If the timer expires, the access terminal shall cease transmission. The maximum time allowed for expiration of the timer is two seconds.

### 1.4.14 System time

System Time counts the number of seconds that have elapsed since the start of System Time on a per sector basis.

In a synchronous access network, all sector air interface transmissions are referenced to a common system-wide timing reference that uses the Global Positioning System (GPS) time, which is traceable to and synchronous with Universal Coordinated Time (UTC). GPS and UTC differ by an integer number of seconds, specifically the number of leap second corrections added to UTC since January 6, 1980. The start of System Time is January 6, 1980 00:00:00 UTC, which coincides with the start of GPS time.

In a synchronous access network, System Time keeps track of leap second corrections to UTC but does not use these corrections for physical adjustments to the System Time clocks.

In an asynchronous access network, System Time need not be traceable and synchronous to a common timing reference.

### 1.4.15 Revision number

Access terminals and access networks complying with the requirements of this specification shall set their revision number to 0x01.

## 2 Session Control Sublayer

### 2.1 Introduction

#### 2.1.1 General overview

The Session Control Sublayer contains protocols used to negotiate a session between the access terminal and the access network. The Session Control Sublayer consists of non-bearer protocols and does not modify transmitted or received packets.

A session is a shared state maintained between the access terminal and the access network, including information such as:

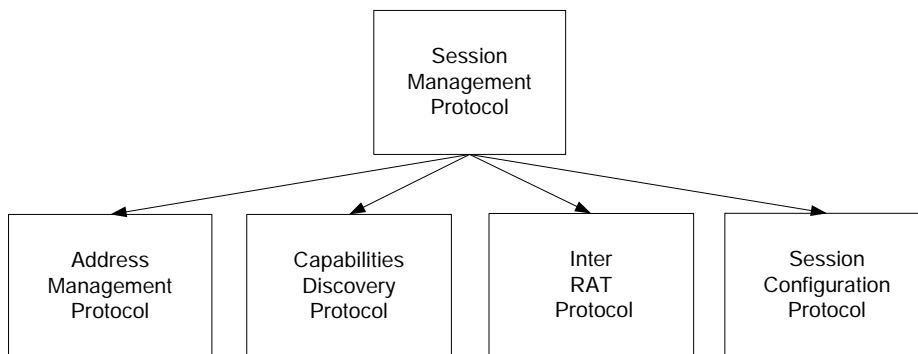
- A Unicast address assigned to the access terminal (UATI).
- The set of protocols and applications used by the access terminal and the access network to communicate over the air-link.
- Configuration settings for these protocols and applications (e.g., authentication keys, parameters for Lower MAC Sublayer Protocols, etc.).

During a single session, the access terminal and the access network can open and close a connection multiple times. Therefore, sessions will be closed rarely and only on occasions, such as when the access terminal leaves the coverage area or during prolonged periods in which the access terminal is unavailable.

The Session Control Sublayer contains the following protocols:

- Session Management Protocol: Provides the means to control the activation of the other Session Control Sublayer protocols. In addition, this protocol ensures that the session is still valid and manages closing of the session.
- Address Management Protocol: Specifies procedures for the initial UATI assignment and maintains the access terminal addresses.
- Session Configuration Protocol: Provides the means to negotiate the *SessionConfigurationToken*'s used during the session.
- Capabilities Discovery Protocol: Provides the means for the access network to discover the capabilities of the access terminal.
- Inter RAT Protocol: Provides the means to send messages for other radio access technologies.

1 The relationship between the Session Control Sublayer protocols is illustrated in Figure 7.



2

3

**Figure 7 Session Control Sublayer protocols**

4

## 2.2 Default Session Management Protocol

5

### 2.2.1 Overview

6

The Default Session Management protocol provides the means to control the activation of the Address Management Protocol, the Capabilities Discovery Protocol, Inter RAT Protocol and the Session Configuration Protocol when a session is established. This protocol also periodically ensures that the session is still valid and manages closing the session.

10

This protocol uses the Signaling Transport to transmit and receive messages.

11

The actual behavior and message exchange in each state of this protocol are mainly governed by protocols that are activated by the Default Session Management Protocol. These protocols return indications, which trigger the state transitions of this protocol.

13

14

This protocol can be in one of four states:

15

- Inactive State: This state applies only to the access terminal. In this state, there are no communications between the access terminal and the access network.

16

17

- AMP Setup State: In this state, the access terminal and access network perform exchanges governed by the Address Management Protocol, and the access network assigns a UATI to the access terminal.

18

19

- Open State: In this state, a session is open.

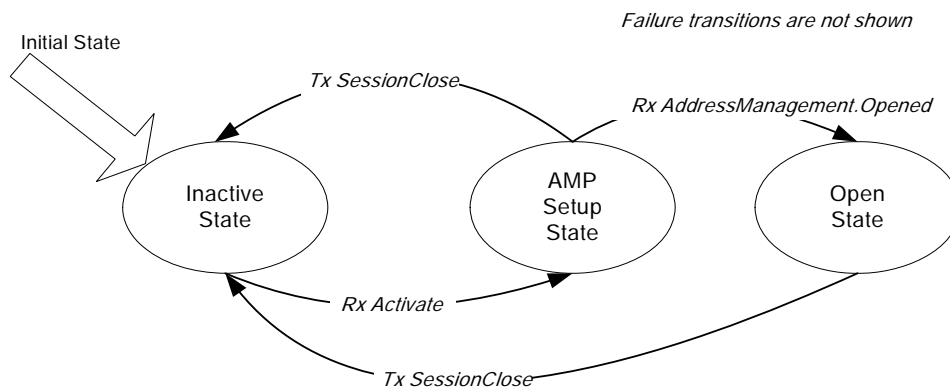
20

- Close State: This state applies only to the access network. In this state, the access network waits for the close procedure to complete.

21

22

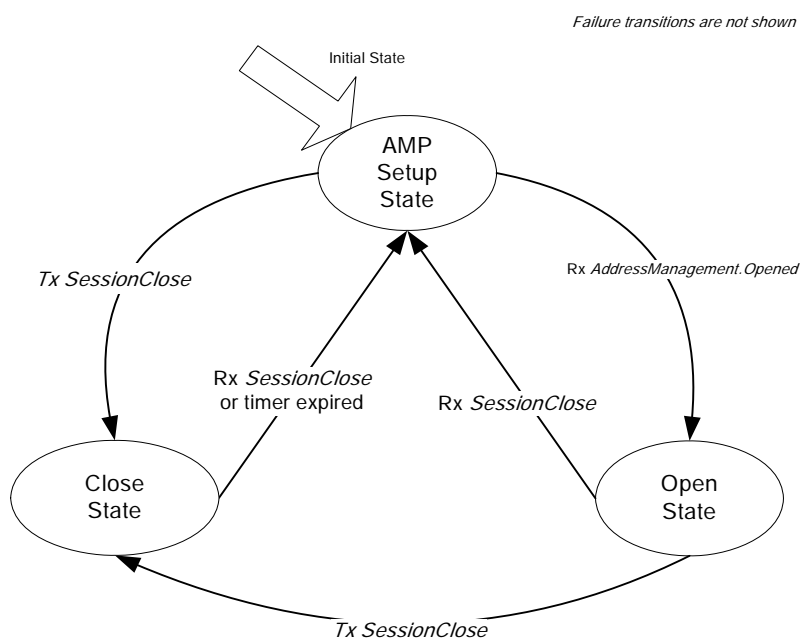
1 Figure 8 provides an overview of the access terminal states and state transitions.



2

3 **Figure 8 Session Management Protocol state diagram (access terminal)**

4 Figure 9 provides an overview of the access network states and state transitions.



5

6 **Figure 9 Session Management Protocol state diagram (access network)**

## 7 2.2.2 Primitives

### 8 2.2.2.1 Commands

9 This protocol defines the following commands:

- 10 ■ *Activate*
- 11 ■ *Deactivate*

## 2.2.2.2 Return indications

This protocol returns the following indications:

- *SessionOpened*
- *SessionClosed*

## 2.2.3 Public data

### 2.2.3.1 Static public data

- *SessionSeed*

### 2.2.3.2 Dynamic public data

- Subtype for this protocol

## 2.2.4 Protocol initialization and swap procedures

### 2.2.4.1 Protocol initialization

Upon creation, the instance of this protocol in the access terminal and access network shall perform the following:

- The value of the attributes for this protocol instance shall be set to the default values specified for each attribute.
- The protocol at the access network shall enter the AMP Setup State.
- The protocol at the access terminal shall enter the Inactive State.

### 2.2.4.2 Protocol swap

Upon swap, the protocol in the access terminal and the access network shall enter the Open State

## 2.2.5 Procedures

### 2.2.5.1 Processing the Activate command

If the access terminal receives the *Activate* command in the Inactive State, it shall transition to the AMP Setup State.

If the access terminal receives the *Activate* command in any state other than the Inactive State, the command shall be ignored.

The access network shall ignore the command.

The list of events that causes an *Activate* command to be sent to this protocol is outside the scope of this specification.

### 2.2.5.2 Processing the Deactivate command

If the access terminal receives a *Deactivate* command in the Inactive State, the command shall be ignored.

If the access terminal receives a *Deactivate* command in any state other than the Inactive State, the access terminal shall perform the following:

- Send a SessionClose message to the access network.
- Perform the procedures for closing the session in 2.2.5.7.
- Transition to the Inactive State.

If the access network receives a *Deactivate* command in the Close State, the command shall be ignored.

If the access network receives a *Deactivate* command in any state other than the Close State, the access network shall send a SessionClose message and transition to the Close State.

The list of events that causes a *Deactivate* command to be sent to this protocol is outside the scope of this specification.

### 2.2.5.3 Processing the SessionOpen message

If the access network receives the SessionOpen message in any state other than AMP Setup state, it shall ignore the message.

If the access network receives the SessionOpen message in the AMP Setup state, it shall issue an *AddressManagement.Activate* command.

### 2.2.5.4 Processing the SessionClose message

If the access terminal receives a SessionClose message in the Inactive State, it shall ignore the message.

If the access terminal receives a SessionClose message in any state other than the Inactive State, the access terminal shall perform the following:

- Send a SessionClose message to the access network.
- Perform the procedures for closing the session in 2.2.5.7.
- Transition to the Inactive State.

If the access network receives a SessionClose message in the Close State, the access network shall process it as specified in 2.2.5.11.

If the access network receives a SessionClose message in any state other than the Close State, the access network shall perform the following:

- Perform the procedures for closing the session in 2.2.5.7.
- Transition to the AMP Setup State.

### 2.2.5.5 Processing failed indications

The access terminal shall ignore an *AddressManagement.Failed* or a *SessionConfiguration.Failed* indication, if the access terminal receives the indication in the Inactive State.

If the access terminal receives an *AddressManagement.Failed* or a *SessionConfiguration.Failed* indication while in any state other than the Inactive State, then the access terminal shall perform the following:

- Send a *SessionClose* message to the access network.
- Perform the procedures for closing the session in 2.2.5.7.
- Transition to the Inactive State.

If the access network receives an *AddressManagement.Failed* or a *SessionConfiguration.Failed* indication, the access network shall perform the following:

- Send a *SessionClose* message to the access terminal.
- Transition to the Close State.

### 2.2.5.6 Processing the *ForwardTrafficChannelMAC.SessionLost* indication

If the access terminal receives a *ForwardTrafficChannelMAC.SessionLost* indication in the Inactive State, it shall ignore the indication.

If the access terminal receives a *ForwardTrafficChannelMAC.SessionLost* in any state other than the Inactive State, the access terminal shall perform the following:

- Perform the procedures for closing the session in 2.2.5.7.
- Transition to the Inactive State.

### 2.2.5.7 Procedures for closing the session

The access terminal or access network shall perform the following to close the session:

- Issue an *AirLinkManagement.CloseConnection* command.
- Issue an *AddressManagement.Deactivate* command.
- Issue a *CapabilitiesDiscovery.Deactivate* command.
- Issue a *InterRAT.Deactivate* command.
- Issue a *SessionConfiguration.Deactivate* command.
- Return a *SessionClosed* indication.

### 2.2.5.8 Inactive state

This state only applies to the access terminal. In this state there are no communications between the access terminal and the access network. The access terminal does not maintain any session-related state and the access network may be unaware of the access terminal's existence within its coverage area when the access terminal's Session Management Protocol is in this state.

1 Upon entering this state the access terminal shall perform the following:

- 2 ■ Set public data *SessionSeed* to the 32-bit pseudorandom number generated using output  
3 of the pseudorandom number generator specified in 10.6.

#### 4 **2.2.5.9 AMP setup state**

5 In this state the Session Management Protocol in the access terminal sends a request to the access  
6 network asking for the session to be opened and waits for the Address Management Protocol to  
7 respond.

##### 8 **2.2.5.9.1 Access terminal requirements**

9 Upon entering the AMP Setup State, the access terminal shall perform the following:

- 10 ■ Send *SessionOpen* message to the access network
- 11 ■ Send an *AddressManagement.Activate* command to the Address Management Protocol

12 If the access terminal receives an *AddressManagement.Opened* indication, it shall perform the  
13 following:

- 14 ■ Issue a *CapabilitiesDiscovery.Activate* command.
- 15 ■ Issue a *InterRAT.Activate* command.
- 16 ■ Issue a *SessionConfiguration.Activate* command.
- 17 ■ Return a *SessionOpened* indication.
- 18 ■ Transition to the Open State.

##### 19 **2.2.5.9.2 Access network requirements**

20 Upon entering the AMP Setup State, the access network is waiting for a *SessionOpen* message.

21 When the access network receives a *SessionOpen* message, it shall issue an  
22 *AddressManagement.Activate* command to the Address Management Protocol.

23 If the access network receives an *AddressManagement.Opened* indication, it shall perform the  
24 following:

- 25 ■ Issue a *CapabilitiesDiscovery.Activate* command.
- 26 ■ Issue a *InterRAT.Activate* command.
- 27 ■ Issue a *SessionConfiguration.Activate* command.
- 28 ■ Return a *SessionOpened* indication.
- 29 ■ Transition to the Open State.



### 2.2.5.10 Open state

In the Open State, the access terminal has an assigned UATI and the access terminal and the access network have a session.

The access terminal and the access network shall support the keep-alive mechanism defined in 2.2.5.10.1.

#### 2.2.5.10.1 Keep-alive functions

The access terminal and the access network shall monitor the traffic on the transports, directed to or from the access terminal. If either the access terminal or the access network detects a period of inactivity of at least  $\text{SessionCloseTimer}/N_{\text{SMPKeepAlive}}$  minutes, it may send a KeepAliveRequest message. The recipient of the message shall respond by sending the KeepAliveResponse message. When a KeepAliveResponse message is received, the access terminal shall not send another KeepAliveRequest message for at least  $\text{SessionCloseTimer}/N_{\text{SMPKeepAlive}}$  minutes.

If the access terminal does not detect any traffic from the access network directed to it for a period of at least SessionCloseTimer minutes, it shall perform the following:

- Perform the procedures for closing the session in 2.2.5.7.
- Transition to the Inactive State.

If the access network does not detect any traffic from the access terminal directed to it for a period of at least SessionCloseTimer minutes, it should perform the following:

- Perform the procedures for closing the session in 2.2.5.7.
- Transition to the AMP Setup State.

If the value of SessionCloseTimer is set to zero, the access terminal and the access network shall not send or expect keep-alive messages, and shall disable the transitions occurring as a consequence of not receiving these messages.

#### 2.2.5.11 Close state

The Close State is associated only with the protocol in the access network. In this state, the protocol in the access network waits for a SessionClose message from the access terminal or an expiration of a timer.

The access network shall set the Close State timer upon entering this state. The value of this timer shall be set to  $T_{\text{SMPMinClose}}$ .

When the access network receives a SessionClose message or when the Close State timer expires, the protocol shall:

- Perform the procedures for closing the session in 2.2.5.7.
- Transition to the AMP Setup State.

1 While in this state, if the access network receives any packet from the access terminal which is  
 2 addressed by the UATI assigned during this session and contains anything but a SessionClose  
 3 message, it shall stay in the Close State and perform the following:

- 4 ■ Discard the packet.
- 5 ■ Respond with a SessionClose message.

## 6 2.2.6 Message formats

7 The protocol uses the AttributeUpdateRequest, AttributeUpdateAccept, and AttributeUpdateReject  
 8 messages of the Generic Attribute Update Protocol in 10.9 to update configurable attributes.

### 9 2.2.6.1 SessionOpen

10 The access terminal sends the SessionOpen message to initiate a session with the access network.

Field	Length (bits)
MessageID	8
SessionSeed	32

12 MessageID The access terminal shall set this field to 0x00.

13 SessionSeed This field shall be set to the value of the public data *SessionSeed* associated  
 14 with the access terminal's session.

<b>Channels</b>	RTC	<b>SLP</b>	Reliable
<b>Addressing</b>	Unicast	<b>Security</b>	Required

### 17 2.2.6.2 SessionClose

18 The sender sends the SessionClose message to terminate the session.

Field	Length (bits)
MessageID	8
CloseReason	8
MoreInfoLen	8
MoreInfo	8 × MoreInfoLen

20 MessageID The sender shall set this field to 0x01.

21 CloseReason The sender shall set this field to the close reason as shown in Table 1.

**Table 1 Encoding of CloseReason field**

Field Value	Meaning	MoreInfoLen	MoreInfo
0x00	Normal Close	0	N/A
0x01	Close Reply	0	N/A
0x02	Protocol Error	0	N/A
0x03	Protocol Negotiation Error	variable	Zero or more Type followed by Subtype followed by offending attribute records (see 10.3 for attribute record definition).
0x04	Session Configuration Failure	2	<i>SessionConfigurationToken</i>
0x05	Session Lost	0	N/A
0x06	Session Unreachable	0	N/A
0x07	All session resources busy	0	N/A
All other values are reserved			

MoreInfoLen Length in octets of the MoreInfo field.

MoreInfo Additional information pertaining to the closure. The format of this field is determined by the particular close reason.

<b>Channels</b>	FTC    RTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Unicast	<b>Security</b>	Required

### 2.2.6.3 KeepAliveRequest

The sender sends the KeepAliveRequest to verify that the peer is still alive.

Field	Length (bits)
MessageID	8

MessageID The sender shall set this field to 0x02.

<b>Channels</b>	FTC    RTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Unicast	<b>Security</b>	Required

### 2.2.6.4 KeepAliveResponse

The sender sends the KeepAliveResponse message as an answer to the KeepAliveRequest message.

Field	Length (bits)
MessageID	8

1 MessageID The sender shall set this field to 0x03.

<b>Channels</b>	FTC    RTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Unicast	<b>Security</b>	Required

## 2.2.7 Interface to other protocols

### 2.2.7.1 Commands sent

6 This protocol issues the following commands:

- 7     ■ *AddressManagement.Activate*
- 8     ■ *CapabilitiesDiscovery.Activate*
- 9     ■ *InterRAT.Activate*
- 10    ■ *SessionConfiguration.Activate*
- 11    ■ *AddressManagement.Deactivate*
- 12    ■ *CapabilitiesDiscovery.Deactivate*
- 13    ■ *InterRAT.Deactivate*
- 14    ■ *SessionConfiguration.Deactivate*
- 15    ■ *AirLinkManagement.CloseConnection*

### 2.2.7.2 Indications

17 This protocol registers to receive the following indications:

- 18    ■ *AddressManagement.Failed*
- 19    ■ *SessionConfiguration.Failed*
- 20    ■ *AddressManagement.Opened*
- 21    ■ *ForwardTrafficChannelMAC.SessionLost*

## 2.2.8 Configuration attributes

The negotiable attributes for this protocol are listed in Table 2. The access terminal shall use as defaults the values in Table 2 that are listed in *bold italics*.

Unless specified otherwise, the access terminal and the access network shall use the Generic Attribute Update Protocol in 10.9 to update configurable attributes belonging to the Default Session Management Protocol.

**Table 2 Configurable attributes**

Attribute ID	Attribute	Values	Meaning
0xff	SessionCloseTimer	<i>0x0CA8</i>	Default is 54 hours.
		0x0000 to 0xFFFF	0x0000 means disable keep-alive messages; all other values are in minutes.

## 2.2.9 Protocol numeric constants

Constant	Meaning	Value
N <sub>SMPType</sub>	Type field for this protocol	Table 9
N <sub>SMPDefault</sub>	Subtype field for this protocol	0x0000
N <sub>SMPKeepAlive</sub>	Maximum number of keep-alive transactions within SessionCloseTimer.	3
T <sub>SMPMinClose</sub>	Minimum recommended timer setting for Close State	300 seconds

### 2.2.10 Session state information

The Session State Information record (see 10.10) consists of parameter records.

The parameter records for this protocol consist of the configuration attributes of this protocol and the following parameters.

#### 2.2.10.1 SessionSeed parameter

**Table 3 Format of the parameter record for the SessionSeed parameter**

Field	Length (bits)
ParameterType	8
Length	8
SessionSeed	32

ParameterType This field shall be set to 0x01 for this parameter record.

1	Length	This field shall be set to the length of this parameter record in units of octets, excluding the Length field.
2		
3	SessionSeed	This field shall be set to the value of the SessionSeed associated with the access terminal's session.
4		

## 2.3 Default Address Management Protocol

### 2.3.1 Overview

The Default Address Management Protocol provides the following functions:

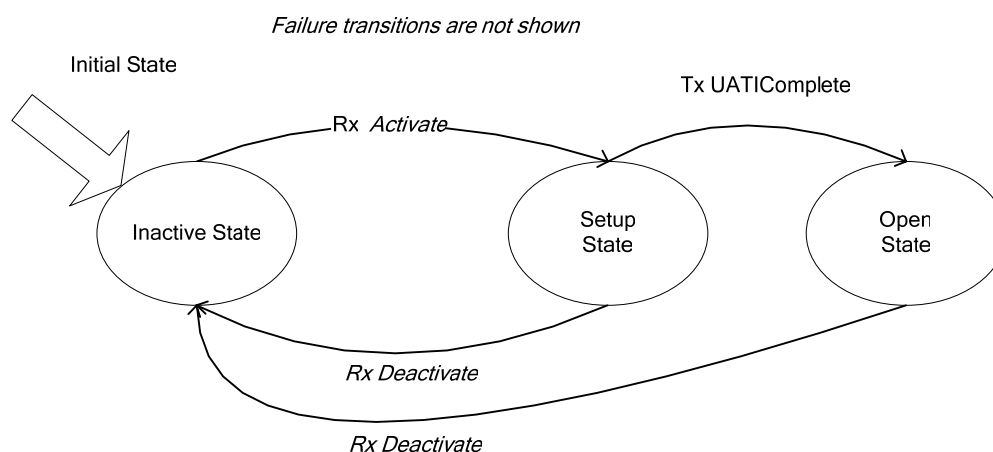
- Initial UATI assignment
- Maintaining the access terminal UATI as the access terminal moves between subnets.

This protocol uses the Signaling Transport to transmit and receive messages.

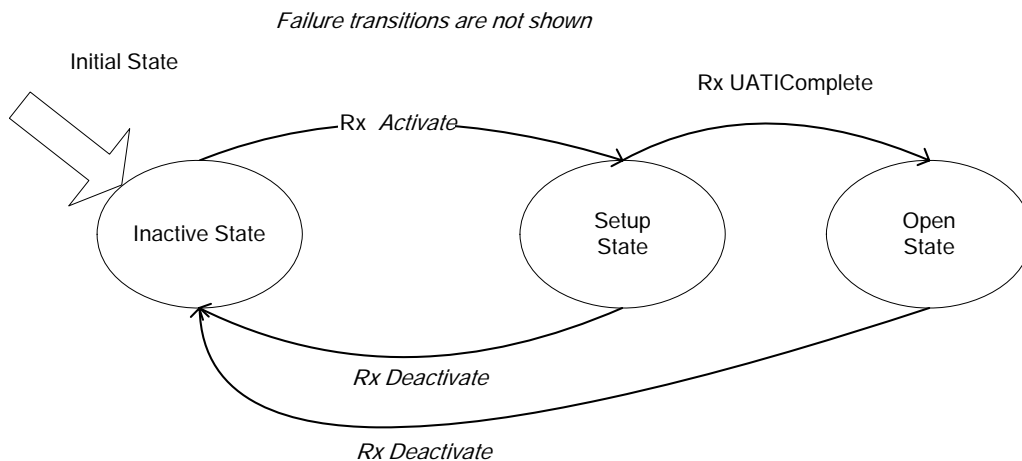
This protocol operates in one of three states:

- Inactive State: In this state there are no communications between the access terminal and the access network.
- Setup State: In this state the access terminal and the access network perform a UATIAssignment/UATIComplete exchange to assign the access terminal a UATI.
- Open State: In this state the access terminal has been assigned a UATI. The access terminal and access network may also perform a UATIUpdateRequest/UATIAssignment/UATIComplete or a UATIAssignment/UATIComplete exchange respectively, so that the access terminal obtains a new UATI.

The protocol states and the messages and events causing the transition between the states are shown in Figure 10 and Figure 11.



**Figure 10 Address Management Protocol state diagram (access terminal)**



**Figure 11 Address Management Protocol state diagram (access network)**

## 2.3.2 Primitives

### 2.3.2.1 Commands

This protocol defines the following commands:

- *Activate*
- *Deactivate*
- *UpdateUATI*
- *RetrieveHWID*

### 2.3.2.2 Return indications

This protocol returns the following indications:

- *Opened*
- *UATIReleased*
- *UATIAssigned*
- *Failed*
- *SubnetChanged*

## 2.3.3 Public data

### 2.3.3.1 Static public data

- *ReceiveATIList*
- *TransmitUATI*

### 2.3.3.2 Dynamic public data

- *Subtype for this protocol*

## 2.3.4 Connection endpoints

The following Connection Endpoints are defined (to be used by the SLP protocol):

- The addresses specified by entries in the ReceiveATIList list whose ATIType is equal to '10' (i.e., UATI) all define the same connection endpoint.
- Each unique <ATI, SectorID> defines a separate connection endpoint. The ATI is an entry in the ReceiveATIList with ATIType equal to '00' (i.e., BATI), and the SectorID is defined in the SectorParameters message in the Lower MAC Control Sublayer.

## 2.3.5 Protocol initialization and swap procedures

### 2.3.5.1 Protocol initialization

Upon creation, the protocol in the access terminal and access network shall perform the following:

- The value of the attributes for this protocol instance shall be set to the default values specified for each attribute.
- The protocol at the access terminal and the access network shall enter the Inactive State.

### 2.3.5.2 Protocol swap

Upon swap, the protocol in the access terminal and the access network shall enter the Open State.

## 2.3.6 Procedures

### 2.3.6.1 Processing the Activate command

If the protocol receives the *Activate* command in the Inactive State, the protocol shall transition to the Setup State.

If the protocol receives the *Activate* command in any state other than the Inactive State, the command shall be ignored.

### 2.3.6.2 Processing the Deactivate command

If the protocol receives the *Deactivate* command in the Inactive State, the command shall be ignored.

If the protocol receives the *Deactivate* command in any state other than the Inactive State, the protocol shall transition to the Inactive State and return a *UATIReleased* indication.

### 2.3.6.3 Processing the UpdateUATI command

If the protocol receives the *UpdateUATI* command in any state other than the Open State, the command shall be ignored.

If the access terminal receives an *UpdateUATI* command in the Open State, it shall send a *UATIUpdateRequest* message.

If the access network receives an *UpdateUATI* command in the Open State, it may send a *UATIAssignment* message.



1 A comprehensive list of events causing the *UpdateUATI* command is beyond the scope of this  
2 specification.

### 3 **2.3.6.4 Processing the RetrieveHWID command**

4 The access terminal shall ignore the *RetrieveHWID* command in all states. The access network shall  
5 ignore the *RetrieveHWID* command when it is received in any state other than the Open State.

6 If the access network receives a *RetrieveHWID* command in the Open State, it may send a  
7 HardwareIDRequest message.

### 8 **2.3.6.5 UATIAssignment message validation**

9 When the access network first sends a UATIAssignment message, it shall set the MessageSequence  
10 field of the message to zero. Subsequently, the access network shall increment this field modulo 256  
11 each time it sends a UATIAssignment message.

12 The access terminal shall initialize a receive pointer for the UATIAssignment message validation,  
13  $V(R)$ , to 255. When the access terminal receives a UATIAssignment message, it shall validate the  
14 message, using the procedure defined in 10.7 (S is equal to 8). The access terminal shall discard the  
15 message if it is invalid.

### 16 **2.3.6.6 Processing HardwareIDRequest message**

17 Upon reception of a HardwareIDRequest message, the access terminal shall respond with a  
18 HardwareIDResponse message. The access terminal shall set the HardwareID record of the  
19 HardwareIDResponse message to the unique ID that has been assigned to the terminal by the  
20 manufacturer.

### 21 **2.3.6.7 Inactive state**

22 In this state, there are no communications between the access terminal and the access network. The  
23 access terminal does not have an assigned UATI, and the access network does not maintain a UATI  
24 for the access terminal and may be unaware of the access terminal's existence within its coverage  
25 area.

#### 26 **2.3.6.7.1 Access terminal requirements**

27 Upon entering the Inactive State, the access terminal shall perform the following:

- 28 ■ Clear the ReceiveATIList
- 29 ■ Add the following entry to the ReceiveATIList:  
30 <ATIType = '00', ATI = NULL>.
- 31 ■ Set the TransmitUATI to NULL.
- 32 ■ Set the UATI to NULL.
- 33 ■ Set the UATISubnetMask to NULL.
- 34 ■ Disable the Address timers.

35 If the access terminal receives an *Activate* command, it shall transition to the Setup State.

### 2.3.6.7.2 Access network requirements

Upon entering the Inactive State, the access network shall perform the following:

- Set the value of the access terminal's UATI to NULL.
- Set the value of the access terminal's UATISubnetMask to NULL.

The access network shall transition to the Setup State if it receives an *Activate* command.

### 2.3.6.8 Setup state

In this state, the access network assigns a UATI to the access terminal.

#### 2.3.6.8.1 Access terminal requirements

Upon entering the Setup State, the access terminal expects a UATIAssignment message.

If the access terminal does not receive a UATIAssignment message within  $T_{ADMPATResponse}$  seconds after entering the Setup state, it shall return a *Failed* indication and transition to the Inactive State.

If the access terminal receives a UATIAssignment message, the access terminal shall validate the message sequence number as defined in 10.7. If the message is valid, it shall perform the following:

- Set its UATI and UATISubnetMask to the UATI and UATISubnetMask fields specified in the message.
- Add the following entry to the ReceiveATIList:  
<ATIType = '10', ATI = UATI[31:0]>.
- Set the TransmitUATI to UATI.
- Return an *Opened* indication.
- Return a *UATIAssigned* indication.
- Send a UATIComplete message.
- Transition to the Open State.

#### 2.3.6.8.2 Access network requirements

In this state the access network shall perform the following:

- Assign a Unicast Access Terminal Identifier (UATI) to the access terminal for the session.
- Send a UATIAssignment message.

When the access network receives the corresponding UATIComplete message with the MessageSequence field that is equal to the MessageSequence field of the UATIAssignment message sent, it shall perform the following:

- Return an *Opened* indication.
- Return a *UATIAssigned* indication.
- Transition to the Open State.

1 If the access network does not receive the corresponding UATIComplete message in response to the  
 2 UATIAssignment message, it may re-transmit the UATIAssignment message. If the access network  
 3 does not receive the UATIComplete message after an implementation-specific number of re-  
 4 transmissions of the UATIAssignment message, it shall return a *Failed* indication and transition to  
 5 the Inactive State.

### 6 **2.3.6.9 Open state**

7 In this state the access terminal has been assigned a UATI.

#### 8 **2.3.6.9.1 Access terminal requirements**

9 If the access terminal receives a *ActiveSetManagement.IdleHO* indication or a  
 10 *ConnectedState.ConnectionClosed*, and then receives an *OverheadMessages.Updated* indication, and  
 11 if either of the following two conditions is true, it shall send a UATIUpdateRequest message:

- 12 ■ The UATISubnetMask is not equal to the SubnetMask of the sector in the active set, or
- 13 ■ The result of bitwise logical AND of the UATI and its subnet mask specified by  
 14 UATISubnetMask is different from the result of bitwise logical AND of SectorID and its  
 15 subnet mask specified by SubnetMask (where SectorID and SubnetMask correspond to  
 16 the sector in the active set).

17 If the access terminal receives an *UpdateUATI* command, it shall process the command as specified  
 18 in 2.3.6.3.

19 Upon sending a UATIUpdateRequest message, the access terminal shall start a UATIResponse timer  
 20 with a timeout value of  $T_{\text{ADM PAT Response}}$  seconds. The access terminal shall disable this timer if either of  
 21 the following conditions is true:

- 22 ■ The UATISubnetMask is equal to the SubnetMask of the sector in the active set, and the  
 23 result of bitwise logical AND of the UATI and its subnet mask specified by  
 24 UATISubnetMask is the same as the result of bitwise logical AND of SectorID and its  
 25 subnet mask specified by SubnetMask (where SectorID and SubnetMask correspond to  
 26 the sector in the active set), or
- 27 ■ The access terminal receives a valid UATIAssignment message.

28 If the UATIResponse timer expires, the access terminal shall return a *Failed* indication and transition  
 29 to the Inactive State.

30 If the access terminal receives a UATIAssignment message, the access terminal shall validate the  
 31 message sequence number as defined in 10.7. If the message is valid, it shall perform the following:

- 32 ■ Set its UATI and UATISubnetMask to the UATI and UATISubnetMask fields in the  
 33 message.
- 34 ■ Add the following entry to the ReceiveATIList:  
 35 <ATIType = '10', ATI = UATI[31:0]>.
- 36 ■ Set the TransmitUATI to UATI.
- 37 ■ Return a *UATIAssigned* indication.
- 38 ■ Send a UATIComplete message.

- 1           ■ Reset and start an Address timer with a timeout value of  $T_{ADMPAddress}$  for the added entry  
2 to the ReceiveATIList.

3 The access terminal shall perform the following when an Address timer corresponding to an entry in  
4 the ReceiveATIList expires:

- 5           ■ Disable the Address timer for that entry.  
6           ■ Delete all of the entries in the ReceiveATIList that are older than the entry whose  
7 Address timer has expired. An entry X in the list is considered older than another entry Y,  
8 if the entry X has been added to the list prior to the entry Y.

### 9 **2.3.6.9.2 Access network requirements**

10 The access network may send a UATIAssignment message at any time in this state. The following are  
11 some of the possible triggers for sending a UATIAssignment message:

- 12           ■ Receiving an *ActiveSetManagement.ActiveSetUpdated* indication.  
13           ■ Receiving an *UpdateUATI* command.  
14           ■ Receiving a valid UATIUpdateRequest message.

15 The access network may return a *SubnetChanged* indication and send a UATIAssignment message  
16 after reception of a *ActiveSetManagement.ActiveSetUpdated* indication. The triggers for returning a  
17 *SubnetChanged* indication after reception of a *ActiveSetManagement.ActiveSetUpdated* indication are  
18 outside the scope of this specification.

19 When the access network sends a UATIAssignment message, it shall perform the following:

- 20           ■ Assign a Unicast Access Terminal Identifier (UATI) to the access terminal for the session  
21 and include it in a UATIAssignment message.

22 When the access network receives a UATIComplete message with the MessageSequence field that is  
23 equal to the MessageSequence field of the last UATIAssignment message that it has sent, it shall  
24 return a *UATIAssigned* indication.

25 If the access network does not receive the UATIComplete message in response to the corresponding  
26 UATIAssignment message within a certain time interval that is specified by the access network<sup>8</sup>, it  
27 should re-transmit the UATIAssignment message. If the access network does not receive the  
28 UATIComplete message after an implementation-specific number of re-transmissions of the  
29 UATIAssignment message, it shall return a *Failed* indication and transition to the Inactive State.

---

<sup>8</sup> The value of this timeout is determined by the access network, and the specification of the timeout value is outside the scope of this document.

## 2.3.7 Message formats

### 2.3.7.1 UATIUpdateRequest

The access terminal sends the UATIUpdateRequest message to request that a UATI is reassigned to it by the access network.

Field	Length (bits)
MessageID	8
MessageSequence	8
TransmitUATI	128

**MessageID** The access terminal shall set this field to 0x00.

**MessageSequence** The access terminal shall increment this field modulo 256 for each new UATIUpdateRequest message sent. If this is the first UATIUpdateRequest message sent by the access terminal, the access terminal shall set this field to zero.

**TransmitUATI** The current value of the TransmitUATI.

<b>Channels</b>	RTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Unicast	<b>Security</b>	Required

### 2.3.7.2 UATIAssignment

The access network sends the UATIAssignment message to assign or re-assign a UATI to the access terminal.

Field	Length (bits)
MessageID	8
MessageSequence	8
UATISubnetMask	8
SessionSeed	32
UATI	128

**MessageID** The access network shall set this field to 0x01.

**MessageSequence** The access network shall increment this field modulo 256 for each new UATIAssignment message sent to this access terminal. If this is the first UATIAssignment message sent to this access terminal, the access network shall set this field to zero.

**UATISubnetMask** The access network shall set this field to the number of consecutive 1's in the subnet mask of the subnet to which the assigned UATI belongs.

1 SessionSeed This field shall be set to the value of the public data  
 2 *SessionManagement.SessionSeed* associated with the access terminal's  
 3 session.

4 UATI The access network shall set this field to the UATI that it is assigning to the  
 5 access terminal.

<b>Channels</b>	FTC	<b>SLP</b>	Reliable
<b>Addressing</b>	Unicast	<b>Security</b>	Required

### 8 2.3.7.3 UATIComplete

9 The access terminal sends this message to notify the access network that it has received the  
 10 UATIAssignment message.

Field	Length (bits)
MessageID	8
MessageSequence	8

12 MessageID The access terminal shall set this field to 0x02.

13 MessageSequence The access terminal shall set this field to the MessageSequence field of the  
 14 UATIAssignment message whose receipt this message is acknowledging.

<b>Channels</b>	RTC	<b>SLP</b>	Reliable
<b>Addressing</b>	Unicast	<b>Security</b>	Required

### 17 2.3.7.4 HardwareIDRequest

18 The access network uses this message to query the access terminal of its Hardware ID information.

Field	Length (bits)
MessageID	8
TransactionID	8

20 MessageID The access network shall set this field to 0x03.

21 TransactionID The access network shall set this field according to 10.8.

<b>Channels</b>	FTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Unicast	<b>Security</b>	Required

### 2.3.7.5 HardwareIDResponse

The access terminal sends this message in response to the HardwareIDRequest message.

Field	Length (bits)
MessageID	8
TransactionID	8
HardwareIDType	24
HardwareIDLength	8
HardwareIDValue	8 × HardwareIDLength

**MessageID** The access terminal shall set this field to 0x04.

**TransactionID** The access terminal shall set this field the TransactionID field of the corresponding HardwareIDRequest message.

**HardwareIDType** The access terminal shall set this field according to Table 4.

**Table 4 HardwareIDType encoding**

HardwareIDType field value	Meaning
0x300000	48-bit extended unique identifier (EUI-48)
0x400000	64-bit extended unique identifier (EUI-64)
0x100000	32-bit Electronic Serial Number (ESN)
0x00NNNN	Hardware ID “NNNN” from [3]
0xFFFFFFFF	Null
All other values	Invalid

**HardwareIDLength** If HardwareIDType is not set to 0xFFFFFFFF, the access terminal shall set this field to the length in octets of the HardwareIDValue field; otherwise the access terminal shall set this field to 0x00.

**HardwareIDValue** The access terminal shall set this field to the unique ID (specified by HardwareIDType) that has been assigned to the terminal by the manufacturer.

<b>Channels</b>	RTC	<b>SLP</b>	Reliable
<b>Addressing</b>	Unicast	<b>Security</b>	Required

## 2.3.8 Interface to other protocols

### 2.3.8.1 Commands

This protocol does not issue any commands.

### 2.3.8.2 Indications

This protocol registers to receive the following indications:

- *ActiveSetManagement.IdleHO*
- *ActiveSetManagement.ActiveSetUpdated*
- *InitializationState.NetworkAcquired*
- *OverheadMessages.Updated*
- *ConnectedState.ConnectionClosed*

### 2.3.9 Configuration attributes

No configuration attributes are defined for this protocol.

### 2.3.10 Protocol numeric constants

Constant	Meaning	Value
N <sub>ADMPT</sub> Type	Type field for this protocol	Table 9
N <sub>ADMP</sub> Default	Subtype field for this protocol	0x0000
T <sub>ADMP</sub> ATResponse	Time to receive UATIAssignment after sending UATIUpdateRequest	120 seconds
T <sub>ADMP</sub> Address	The duration of time that the access terminal declares an address match if it receives a message that is addressed using either the old or the new UATI	180 seconds

### 2.3.11 Session state information

The Session State Information record (see 10.10) consists of parameter records.

The parameter records for this protocol consist of the configuration attributes of this protocol and the following parameters.



### 2.3.11.1 UATI parameter

**Table 5 Format of the parameter record for the UATI parameter**

Field	Length (bits)
ParameterType	8
Length	8
MessageSequence	8
UATISubnetLength	8
UATI	128

ParameterType This field shall be set to 0x01 for this parameter record.

Length This field shall be set to the length of this parameter record in units of octets excluding the Length field.

MessageSequence This field shall be set to the MessageSequence field of the last UATIAssignment message that was sent by the source access network.

UATISubnetLength This field shall be set to the number of consecutive 1's in the subnet mask of the subnet to which the assigned UATI belongs.

UATI This field shall be set to the UATI that it is assigned to the access terminal.

## 2.4 Default Session Configuration Protocol

### 2.4.1 Overview

The Default Session Configuration Protocol provides for the negotiation and configuration of the set of *SessionConfigurationToken*'s used during a session.

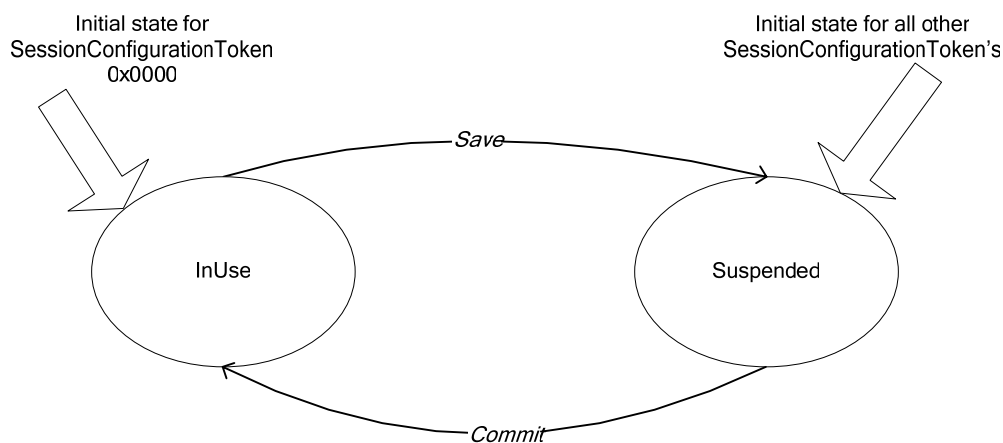
This protocol uses the Signaling Transport to transmit and receive messages.

The *SessionConfigurationToken* is a 16-bit value that defines a complete set of protocol and transport instances that can be used to communicate between the access terminal and the access network. A protocol instance consists of a protocol subtype, dynamic public data and attribute values. A transport instance consists of a transport subtype, dynamic public data and attribute values. A transport instance is bound to a Transport in the Packet Consolidation Protocol. A listing of *SessionConfigurationToken*'s, including the subtype, dynamic public data and attribute values for each protocol and transport instance defined by the *SessionConfigurationToken* can be found in 11.5.

A *SessionConfigurationToken* is InUse if the set of protocol and transport instances specified by the *SessionConfigurationToken* are currently being used to communicate between the access terminal and the access network. Otherwise, a *SessionConfigurationToken* is Suspended. Only one *SessionConfigurationToken* shall be InUse at a time.

1 A protocol or transport instance is InUse if it is currently being used to communicate between the  
 2 access terminal and the access network. Otherwise, a protocol or transport instance is Suspended.  
 3 Only one protocol instance of a protocol type shall be InUse at a time<sup>9</sup>. Only one transport instance  
 4 corresponding to a Transport in the Packet Consolidation Protocol shall be InUse at a time. A  
 5 protocol or transport instance shall correspond to exactly one *SessionConfigurationToken*.

6 The Session Configuration Protocol executes its Save and Commit procedures to swap the InUse  
 7 protocol and transport instances associated with the current InUse *SessionConfigurationToken* with  
 8 the Suspended protocol and transport instances associated with a Suspended  
 9 *SessionConfigurationToken*. A state diagram for each *SessionConfigurationToken* is shown in  
 10 Figure 12.



11  
 12 **Figure 12 SessionConfigurationToken state diagram**

13 The access network and the access terminal shall use the Generic Attribute Update Protocol in 10.9 to  
 14 negotiate the configurable attributes of the protocol and transport instances of the InUse  
 15 *SessionConfigurationToken*. The access network and the access terminal shall not use the Generic  
 16 Attribute Update Protocol in 10.9 to negotiate the configurable attributes of the protocol and transport  
 17 instances of a Suspended *SessionConfigurationToken*.

18 The access network and the access terminal shall use the ConfigurationRequest, ConfigurationAccept  
 19 and ConfigurationReject messages to negotiate the configurable attributes of the protocol and  
 20 transport instances of a Suspended *SessionConfigurationToken*. The access network and the access  
 21 terminal shall not use the ConfigurationRequest, ConfigurationAccept and ConfigurationReject  
 22 messages to negotiate the configurable attributes of the protocol and transport instances of the InUse  
 23 *SessionConfigurationToken*.

---

<sup>9</sup>The Session Configuration Protocol shall have two protocol instances that are temporarily InUse at the same time when a Suspended *SessionConfigurationToken* is swapped with the InUse *SessionConfigurationToken* while the connection is in the Closed state.

1 A ConfigurationRequest message to update an attribute for a Suspended *SessionConfigurationToken*  
 2 is defined as legal if the attribute may be updated by an AttributeUpdateRequest message if the  
 3 Suspended *SessionConfigurationToken* were to become the InUse *SessionConfigurationToken*.  
 4 Otherwise a ConfigurationRequest message is illegal.

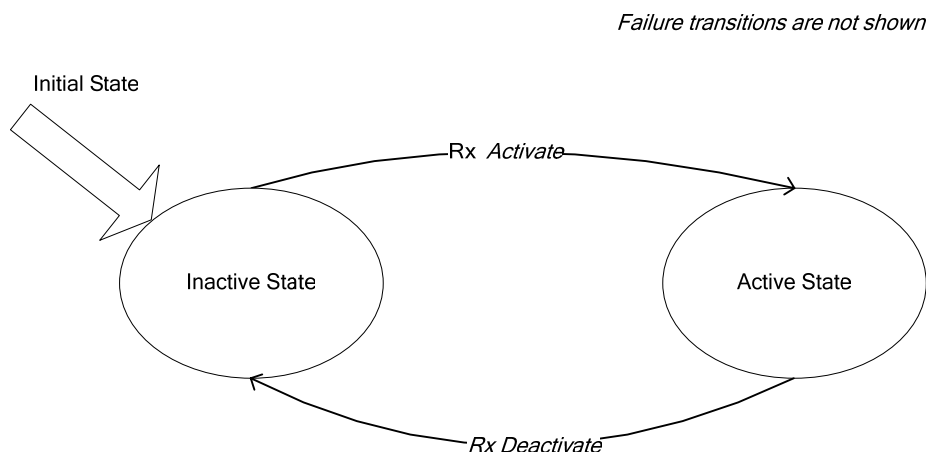
5 The access network and access terminal shall not send an illegal ConfigurationRequest message.

6 If the access terminal receives an illegal ConfigurationRequest message, it shall ignore the message.  
 7 If the access network receives an illegal ConfigurationRequest message, it shall either ignore the  
 8 message or respond with a ConfigurationReject message.

9 This protocol operates in one of two states:

- 10 ■ Inactive State: In this state, the protocol waits for an *Activate* command. There are no  
 11 communications between the access terminal and the access network.
- 12 ■ Active State: In this state the access network may query the access terminal as to which  
 13 *SessionConfigurationToken*'s are supported and may change the InUse  
 14 *SessionConfigurationToken*.

15 The protocol states and the messages and events causing the transition between the states are shown  
 16 in Figure 13.



17  
 18 **Figure 13 Session Configuration Protocol state diagram**

## 19 2.4.2 Primitives

### 20 2.4.2.1 Commands

21 This protocol defines the following commands:

- 22 ■ *Activate*
- 23 ■ *Deactivate*

## 2.4.2.2 Return indications

This protocol returns the following indications:

- *Reconfigured*
- *Failed*

## 2.4.3 Public data

### 2.4.3.1 Static public data

- *SessionConfigurationToken*
- *ConfigurationLock*

### 2.4.3.2 Dynamic public data

- Subtype for this protocol

## 2.4.4 Protocol initialization and swap procedures

### 2.4.4.1 Protocol initialization

Upon creation, the protocol in the access terminal and access network shall perform the following:

- The protocol at the access terminal and the access network shall enter the Inactive State.
- The access network and the access terminal shall set the *ConfigurationLock* to *UnLocked*.
- The access terminal and the access network shall set the *SessionConfigurationToken* to *0x0000*.

### 2.4.4.2 Protocol swap

Upon swap, the protocol in the access terminal and the access network shall perform the following:

- The protocol at the access network and the access terminal shall execute its *Commit* procedure for the *SessionConfigurationToken* in the static public data.
- The protocol at the access terminal and the access network shall enter the Active State.

## 2.4.5 Procedures

The access terminal and the access network shall maintain a parameter called *ConfigurationLock*.

### 2.4.5.1 Processing the *Activate* command

If the protocol receives the *Activate* command in the Inactive State, the protocol shall transition to the Active State.

If the protocol receives the *Activate* command in the Active State, the command shall be ignored.

### 2.4.5.2 Processing the Deactivate command

If the protocol receives the *Deactivate* command in the Inactive State, the command shall be ignored.

If the protocol receives this command in the Active State, it shall transition to the Inactive State.

### 2.4.5.3 Commit procedure

The access terminal and the access network shall perform the procedures specified in this section when directed to execute its Commit procedure.

The Session Configuration Protocol shall direct all of the protocol instances specified by the *SessionConfigurationToken*, except for the Session Configuration Protocol instance, to perform the following in the order specified:

- Restore the dynamic public data and attributes of the protocol instance.
- The Suspended protocol instance shall become the InUse instance for this protocol type.
- The protocol instance shall perform its Swap Procedure.

The Session Configuration Protocol shall direct all of the transport instances specified by the *SessionConfigurationToken* to perform the following in the order specified:

- Restore the dynamic public data and attributes of the transport instance.
- The Suspended transport instance shall become the InUse transport instance.
- The transport instance shall perform its Swap procedure.

### 2.4.5.4 Save Procedure

The access terminal and the access network shall perform the procedures specified in this section when directed to execute its Save procedure.

The Session Configuration Protocol shall direct all of the protocol instances specified by the *SessionConfigurationToken*, except for the Session Configuration Protocol instance, to perform the following in the order specified:

- Store the dynamic public data and attributes of the protocol instance.
- The InUse protocol instance shall become Suspended.

The Session Configuration Protocol shall direct all of the transport instances specified by the *SessionConfigurationToken* to perform the following in the order specified:

- Store the dynamic public data and attributes of the transport instance.
- The InUse transport instance shall become Suspended.

### 2.4.5.5 TokenAssignment message validation

When the access network first sends a TokenAssignment message, it shall set the MessageSequence field of the message to zero. Subsequently, the access network shall increment this field modulo 256 each time it sends a TokenAssignment message.

1 The access terminal shall initialize a receive pointer for the TokenAssignment message validation,  
 2  $V(R)$ , to 255. When the access terminal receives a TokenAssignment message, it shall validate the  
 3 message, using the procedure defined in 10.7 (S is equal to 8). The access terminal shall discard the  
 4 message if it is invalid.

#### 5 **2.4.5.6 Inactive state**

6 Upon entering this state, the protocol shall set the *SessionConfigurationToken* to 0x0000.

7 If the protocol receives the *Activate* command in the Inactive State, the protocol shall transition to the  
 8 Active State

#### 9 **2.4.5.7 Active state**

##### 10 **2.4.5.7.1 Access terminal requirements**

11 While in this state, the access terminal may send a TokenUpdateRequest message to request the  
 12 access network to change the value of the *SessionConfigurationToken* if the value of the  
 13 ConfigurationLock parameter is UnLocked. The access terminal shall not send a  
 14 TokenUpdateRequest or ConfigurationRequest message if the value of the ConfigurationLock  
 15 parameter is Locked.

16 If the access terminal receives a TokenAssignment message requesting to update the value of the  
 17 *SessionConfigurationToken* in this state, the access terminal shall validate the message sequence  
 18 number as defined in 10.7. If the message is valid, the access terminal shall perform the following in  
 19 the order specified:

- 20 ■ If the *SessionConfigurationToken* specified by the TokenAssignment message is the same  
 21 as the InUse *SessionConfigurationToken*, the Session Configuration Protocol shall send a  
 22 TokenComplete message.
- 23 ■ Otherwise if the *SessionConfigurationToken* does not specify a Suspended  
 24 *SessionConfigurationToken*, the access terminal shall return a *Failed* indication and  
 25 transition to the Inactive state.
- 26 ■ Otherwise, if the *SessionConfigurationToken* specified by the TokenAssignment message  
 27 is different from the InUse *SessionConfigurationToken* the access terminal shall perform  
 28 the following:
  - 29 □ Send a TokenComplete message.
  - 30 □ If the Air Link Management Protocol is in the Connected State
    - 31 – Issue an *AirLinkManagement.CloseConnection* command.
    - 32 – Wait to receive a *ConnectedState.ConnectionClosed* indication.
  - 33 □ Execute the Save procedure for the InUse *SessionConfigurationToken*.
  - 34 □ Store the dynamic public data and attributes of the InUse Session Configuration  
 35 Protocol instance.
  - 36 □ Set the *SessionConfigurationToken* static public data to the value specified in the  
 37 TokenAssignment message.
  - 38 □ Restore the dynamic public data and attributes of the Suspended Session  
 39 Configuration Protocol instance specified by the new *SessionConfigurationToken*.

- 1           □ Return a *Reconfigured* indication.
- 2           □ The Session Configuration Protocol instance specified by the new
- 3           *SessionConfigurationToken* shall execute its Swap procedure and shall become the
- 4           InUse instance for this protocol type.
- 5           □ This Session Configuration Protocol instance shall become Suspended.

6 If the access terminal receives a LockConfiguration message, then the access terminal shall respond  
7 with a ConfigurationLockAck message and shall set ConfigurationLock to Locked. If the access  
8 terminal receives an UnLockConfiguration message, then the access terminal shall respond with an  
9 UnLockConfigurationAck message and shall set ConfigurationLock to UnLocked.

#### 10 **2.4.5.7.2 Access network requirements**

11 While in this state, the access network may send a TokenAssignment message to change the value of  
12 the *SessionConfigurationToken* if the value of the ConfigurationLock parameter is UnLocked. The  
13 access network shall not send a TokenAssignment or ConfigurationRequest message if the value of  
14 the ConfigurationLock parameter is Locked.

15 Upon receiving a TokenComplete message in response to the TokenAssignment message, the access  
16 network shall perform the following:

- 17       ■ If the *SessionConfigurationToken* specified by the TokenAssignment message is different  
18       from the InUse *SessionConfigurationToken*, the access network shall perform the  
19       following:
  - 20       □ If the Air Link Management Protocol is in the Connected State
    - 21           – Issue an *AirLinkManagement.CloseConnection* command.
    - 22           – Wait to receive a *ConnectedState.ConnectionClosed* indication.
  - 23       □ Execute the Save procedure for the InUse *SessionConfigurationToken*.
  - 24       □ Store the dynamic public data and attributes of the InUse Session Configuration  
25       Protocol instance.
  - 26       □ Set the *SessionConfigurationToken* static public data to the value specified in the  
27       TokenAssignment message.
  - 28       □ Restore the dynamic public data and attributes of the Suspended Session  
29       Configuration Protocol instance specified by the new *SessionConfigurationToken*.
  - 30       □ Return a *Reconfigured* indication.
  - 31       □ The Session Configuration Protocol instance specified by the new  
32       *SessionConfigurationToken* shall execute its Swap procedure and shall become the  
33       InUse instance for this protocol type.
  - 34       □ This Session Configuration Protocol instance shall become Suspended.

## 2.4.6 Message formats

### 2.4.6.1 TokensSupportedRequest

The access network sends the TokensSupportedRequest message to discover the set of *SessionConfigurationToken*'s supported by the access terminal.

Field	Length (bits)
MessageID	8

MessageID                      The access network shall set this field to 0x00.

<b>Channels</b>	FTC	<b>SLP</b>	Reliable
<b>Addressing</b>	Unicast	<b>Security</b>	Required

### 2.4.6.2 TokensSupportedResponse

The access terminal sends the TokensSupportedResponse message in response to the TokensSupportedRequest message.

Field	Length (bits)
MessageID	8
TokenCount	8

TokenCount occurrences of the following field:

{	
SessionConfigurationToken	16
}	

MessageID                      The access terminal shall set this field to 0x01.

TokenCount                      The access terminal shall set this field to the number of *SessionConfigurationToken* fields included in this message. The access terminal shall include TokenCount occurrences of the following field with the message.

SessionConfigurationToken

The access terminal shall set this field to a *SessionConfigurationToken* supported by the access terminal.

<b>Channels</b>	RTC	<b>SLP</b>	Reliable
<b>Addressing</b>	Unicast	<b>Security</b>	Required



### 2.4.6.3 TokenUpdateRequest

The access terminal sends the TokenUpdateRequest message to request a new InUse *SessionConfigurationToken* assignment from the access network.

Field	Length (bits)
MessageID	8
MessageSequence	8
TokenCount	8

TokenCount occurrences of the following field:

{	
SessionConfigurationToken	16
}	

**MessageID** The access terminal shall set this field to 0x02.

**MessageSequence** The access terminal shall increment this field modulo 256 for each new TokenUpdateRequest message sent. If this is the first TokenUpdateRequest message sent by the access terminal, the access terminal shall set this field to zero.

**TokenCount** The access terminal shall set this field to the number of SessionConfigurationToken fields included in this message, where the *SessionConfigurationToken* values are in descending order of preference. The access terminal shall include TokenCount occurrences of the following field with the message.

#### SessionConfigurationToken

The access terminal shall set this field to a *SessionConfigurationToken* supported by the access terminal.

<b>Channels</b>	RTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Unicast	<b>Security</b>	Required

### 2.4.6.4 TokenAssignment

The access network sends the TokenAssignment message to change the InUse *SessionConfigurationToken*.

Field	Length (bits)
MessageID	8
MessageSequence	8
SessionConfigurationToken	16

- 1 MessageID The access network shall set this field to 0x03.
- 2 MessageSequence The access network shall increment this field modulo 256 for each new  
3 TokenAssignment message sent to this access terminal. If this is the first  
4 TokenAssignment message sent to this access terminal, the access network  
5 shall set this field to zero.
- 6 SessionConfigurationToken  
7 The access network shall set this field to the *SessionConfigurationToken* that  
8 it is assigning to the access terminal.

<b>Channels</b>	FTC	<b>SLP</b>	Reliable
<b>Addressing</b>	Unicast	<b>Security</b>	Required

10

### 11 2.4.6.5 TokenComplete

- 12 The access terminal sends the TokenComplete message to notify the access network that it has  
13 received the TokenAssignment message.

14

Field	Length (bits)
MessageID	8
MessageSequence	8

- 15 MessageID The access terminal shall set this field to 0x04.
- 16 MessageSequence The access terminal shall set this field to the MessageSequence field of the  
17 TokenAssignment message whose receipt this message is acknowledging.

18

<b>Channels</b>	RTC	<b>SLP</b>	Reliable
<b>Addressing</b>	Unicast	<b>Security</b>	Required

19

### 20 2.4.6.6 LockConfiguration

- 21 The access network sends the LockConfiguration message to set the ConfigurationLock parameter in  
22 the access terminal to Locked.

23

Field	Length (bits)
MessageID	8
TransactionID	8

- 24 MessageID The access network shall set this field to 0x05.
- 25 TransactionID The access network shall set this field according to 10.8.

26

<b>Channels</b>	FTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Unicast	<b>Security</b>	Required

### 2.4.6.7 LockConfigurationAck

The access terminal sends the LockConfigurationAck message to acknowledge the receipt of a LockConfiguration message.

Field	Length (bits)
MessageID	8
TransactionID	8

**MessageID** The access terminal shall set this field to 0x06.

**TransactionID** The access terminal shall set this field to the TransactionID field of the LockConfiguration message that is being acknowledged.

<b>Channels</b>	RTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Unicast	<b>Security</b>	Required

### 2.4.6.8 UnLockConfiguration

The access network sends the UnLockConfiguration message to set the ConfigurationLock parameter in the access terminal to UnLocked.

Field	Length (bits)
MessageID	8
TransactionID	8

**MessageID** The access network shall set this field to 0x07.

**TransactionID** The access network shall set this field according to 10.8.

<b>Channels</b>	FTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Unicast	<b>Security</b>	Required

### 2.4.6.9 UnLockConfigurationAck

The access terminal sends the UnLockConfigurationAck message to acknowledge the receipt of an UnLockConfiguration message.

Field	Length (bits)
MessageID	8
TransactionID	8

**MessageID** The access terminal shall set this field to 0x08.

**TransactionID** The access terminal shall set this field to the TransactionID field of the UnLockConfiguration message that is being acknowledged.

<b>Channels</b>	RTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Unicast	<b>Security</b>	Required

### 2.4.6.10 ConfigurationRequest

The sender sends an ConfigurationRequest message to offer a set of attribute-values for a given attribute of a protocol or transport instance for a Suspended *SessionConfigurationToken*.

Field	Length (bits)
MessageID	8
TransactionID	8
SessionConfigurationToken	16
ProtocolType	8

One or more instances of the following field:

{	
AttributeRecord	Attribute dependent
}	

**MessageID** The sender shall set this field to 0x09.

**TransactionID** The sender shall set this field according to 10.8.

**ProtocolType** The sender shall set this field to the Type value in Table 9 for the protocol or transport associated with the attributes being negotiated.

**AttributeRecord** The format of this record is specified in 10.3.

<b>Channels</b>	FTC RTC	<b>SLP</b>	Reliable
<b>Addressing</b>	Unicast	<b>Security</b>	Required

### 2.4.6.11 ConfigurationAccept

The sender sends a ConfigurationAccept message in response to a ConfigurationRequest message to accept the offered attribute values.

Field	Length (bits)
MessageID	8
TransactionID	8

**MessageID** The sender shall set this field to 0x0a.

**TransactionID** The sender shall set this field to the TransactionID field of the ConfigurationRequest message that is being accepted.

<b>Channels</b>	FTC    RTC	<b>SLP</b>	Reliable
<b>Addressing</b>	Unicast	<b>Security</b>	Required

### 2.4.6.12 ConfigurationReject

The access network sends a ConfigurationReject message in response to a ConfigurationRequest message to reject the offered attribute values.

Field	Length (bits)
MessageID	8
TransactionID	8

**MessageID** The sender shall set this field to 0x0b.

**TransactionID** The sender shall set this field to the TransactionID field of the ConfigurationRequest message that is being rejected.

<b>Channels</b>	FTC	<b>SLP</b>	Reliable
<b>Addressing</b>	Unicast	<b>Security</b>	Required

## 2.4.7 Interface to other protocols

### 2.4.7.1 Commands

This protocol issues the following command:

- *AirLinkManagement.CloseConnection*

### 2.4.7.2 Indications

This protocol registers to receive the following indication:

- *ConnectedState.ConnectionClosed*

### 2.4.8 Configuration attributes

No configuration attributes are defined for this protocol.

### 2.4.9 Protocol numeric constants

Constant	Meaning	Value
N <sub>SCPType</sub>	Type field for this protocol	Table 9
N <sub>SCPDefault</sub>	Subtype field for this protocol	0x0000

### 2.4.10 Session State information

The Session State Information record (see 10.10) consists of parameter records.

The parameter records for this protocol consist of the configuration attributes of this protocol and the following parameters.

#### 2.4.10.1 ConfigurationLock parameter

**Table 6 Format of the parameter record for the ConfigurationLock parameter**

Field	Length (bits)
ParameterType	8
Length	8
ConfigurationLock	8

**ParameterType** This field shall be set to 0x01 for this parameter record.

**Length** This field shall be set to the length of this parameter record in units of octets excluding the Length field.

**ConfigurationLock** This field shall be set to 0x00 if the value of the ConfigurationLock is UnLocked and it shall be set to 0x01 if the value of the ConfigurationLock is set to Locked.

## 2.4.10.2 SessionConfigurationToken parameter

**Table 7 Format of the parameter record for the SessionConfigurationToken parameter**

Field	Length (bits)
ParameterType	8
Length	8
SessionConfigurationToken	16
MessageSequence	8

ParameterType This field shall be set to 0x02 for this parameter record.

Length This field shall be set to the length of this parameter record in units of octets excluding the Length field.

SessionConfigurationToken

This field shall be set to the value of the InUse *SessionConfigurationToken* assigned to the access terminal.

MessageSequence This field shall be set to the MessageSequence field of the last TokenAssignment message that was sent by the access network.

## 2.5 Default Capabilities Discovery Protocol

### 2.5.1 Overview

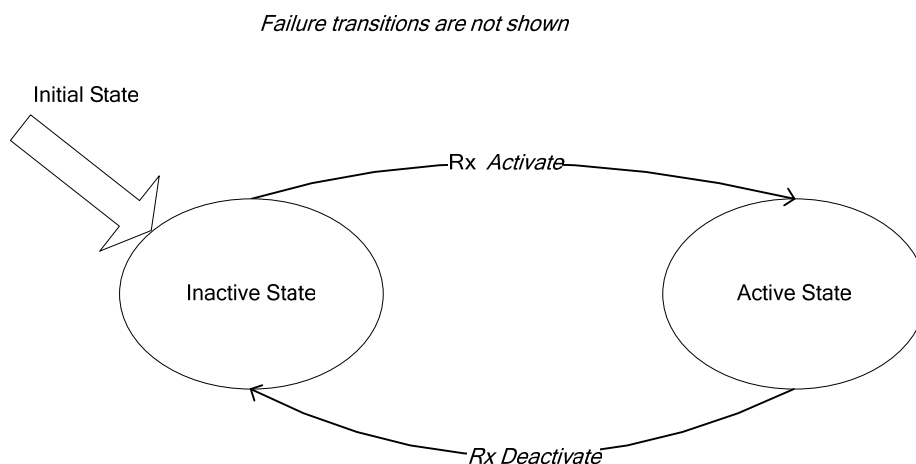
The Default Capabilities Discovery Protocol allows the access network to discover the capabilities of the access terminal.

This protocol uses the Signaling Transport to transmit and receive messages.

This protocol operates in one of two states:

- Inactive State: In this state, the protocol waits for an *Activate* command. There are no communications between the access terminal and the access network.
- Active State: In this state the access terminal and the access network perform a CapabilitiesRequest/CapabilitesResponse exchange.

1 The protocol states and the messages and events causing the transition between the states are shown  
 2 in Figure 14.



3

4 **Figure 14 Session Capabilities Discovery Protocol state diagram**

## 5 **2.5.2 Primitives**

### 6 **2.5.2.1 Commands**

7 This protocol defines the following commands:

- 8 ■ *Activate*
- 9 ■ *Deactivate*

### 10 **2.5.2.2 Return indications**

11 This protocol does not return any indications.

## 12 **2.5.3 Public data**

### 13 **2.5.3.1 Static public data**

14 This protocol does not define any static public data.

### 15 **2.5.3.2 Dynamic public data**

- 16 ■ Subtype for this protocol
- 17 ■ All the attributes listed in 2.5.8.



## 1 **2.5.4 Protocol initialization and swap procedures**

### 2 **2.5.4.1 Protocol initialization**

3 Upon creation, the protocol in the access terminal and access network shall perform the following:

- 4     ■ The value of the attributes for this protocol instance shall be set to the default values  
5         specified for each attribute.
- 6     ■ The protocol at the access terminal and the access network shall enter the Inactive State

### 7 **2.5.4.2 Protocol swap**

8 Upon swap, the protocol in the access terminal and the access network shall enter the Active State.

## 9 **2.5.5 Procedures**

### 10 **2.5.5.1 Processing the Activate command**

11 If the protocol receives the *Activate* command in the Inactive State, the protocol shall transition to the  
12 Active State.

13 If the protocol receives the *Activate* command in the Active State, the command shall be ignored.

### 14 **2.5.5.2 Processing the Deactivate command**

15 If the protocol receives the *Deactivate* command in the Inactive State, the command shall be ignored.

16 If the protocol receives the *Deactivate* command in the Active State, the protocol shall transition to  
17 the Inactive State

### 18 **2.5.5.3 Inactive state**

19 In this state, there are no communications between the access terminal and the access network.

20 In this state the protocol waits for the *Activate* command. See 2.5.5.1 for processing of the *Activate*  
21 command.

### 22 **2.5.5.4 Active state**

23 In this state the access terminal and the access network perform a  
24 CapabilitiesRequest/CapabilitiesResponse exchange.

## 2.5.6 Message formats

### 2.5.6.1 CapabilitiesRequest

The access network sends the CapabilitiesRequest message to discover the capabilities of the access terminal.

Field	Length (bits)
MessageID	8
TransactionID	8
AttributeCount	8

AttributeCount occurrences of the following field:

{	
AttributeID	16
}	

**MessageID** The access network shall set this field to 0x00.

**TransactionID** The access network shall set this field according to 10.8.

**AttributeCount** The access network shall set this field to the number of AttributeID fields included in this message. The sender shall set this field to 0x00 to request the value of all attributes defined in 2.5.8. The access network shall include AttributeCount occurrences of the following field with the message.

**AttributeID** The access network shall set this field to the AttributeID for which this request is generated.

<b>Channels</b>	FTC	<b>SLP</b>	Reliable
<b>Addressing</b>	Unicast	<b>Security</b>	Required

## 2.5.6.2 CapabilitiesResponse

The access terminal sends the CapabilitiesResponse message in response to the CapabilitiesRequest message.

Field	Length (bits)
MessageID	8
TransactionID	8
AttributeCount	8

AttributeCount occurrences of the following field:

{	
AttributeRecord	Attribute dependent
}	

**MessageID** The access terminal shall set this field to 0x01.

**TransactionID** The access terminal shall set this value to the TransactionID field of the corresponding CapabilitiesResponse message.

**AttributeCount** The access terminal shall set this field to the number of AttributeRecord fields included in this message. The access terminal shall include AttributeCount occurrences of the following field with the message.

**AttributeRecord** An attribute record containing a single attribute value. The format of the AttributeRecord is given in 10.3. The access terminal shall not include more than one attribute record with the same attribute identifier.

<b>Channels</b>	RTC	<b>SLP</b>	Reliable
<b>Addressing</b>	Unicast	<b>Security</b>	Required

## 2.5.7 Interface to other protocols

### 2.5.7.1 Commands

This protocol does not issue any commands.

### 2.5.7.2 Indications

This protocol does not register to receive any indications.

## 2.5.8 Configuration attributes

The access terminal and the access network shall not use the Generic Attribute Update Protocol in 10.9 to update configurable attributes belonging to the Default Capabilities Discovery Protocol.

### 2.5.8.1 Simple attributes

The negotiable simple attributes for this protocol are listed in Table 8. The access terminal shall use as defaults the values in Table 8 that are listed in *bold italics*.

**Table 8 Configurable attributes**

Attribute ID	Attribute	Values	Meaning
0x0000	NumRxAntennas	<b>0x0001</b>	1 receive antennas supported at the access terminal
		0x0001 to 0x0004	Number of receive antennas supported at the access terminal
		All other values	Reserved
0x0001	MaxPacketFormatFwd	<b>0x0001</b>	1 is the maximum Packet Format that can be supported by the access terminal on the forward link.
		0x0000 to 0x000f	Number of the maximum Packet Format that can be supported by the access terminal on the forward link.
		All other values	Reserved
0x0002	MaxPacketFormatRev	<b>0x0001</b>	1 is the maximum Packet Format that can be supported by the access terminal on the reverse link.
		0x0000 to 0x0009	Number of the maximum Packet Format that can be supported by the access terminal on the reverse link.
		All other values	Reserved
0x0003	MaxMIMOAssignmentFwd	<b>0x0000</b>	The access terminal does not support MIMO mode.
		0x0008 to 0x0800	The maximum number of sub-carriers that can be assigned to the access terminal on the forward link in units of carriers, when in MIMO mode.
		All other values	Reserved

<b>Attribute ID</b>	<b>Attribute</b>	<b>Values</b>	<b>Meaning</b>
0x0004	NumCarriers	<b>0x0001</b>	1 is the maximum number of carriers supported by the Access Terminal in multi-carrier mode.
		0x0001-0x0004	Maximum number of carriers supported by the access terminal in multi-carrier mode.
		All other values	Reserved
0x0005	MaxInterlaceAssignmentFwd	<b>0x0001</b>	1 is the maximum number of interlaces on which the access terminal can simultaneously receive MAC packets.
		0x0001 to 0x0006 (FDD)	The maximum number of interlaces on which the access terminal can simultaneously receive MAC packets.
		0x0001 to 0x000c (TDD)	
		All other values	Reserved
0x0006	MaxInterlaceAssignmentRev	<b>0x0001</b>	The maximum number of interlaces on which the access terminal can simultaneously transmit MAC packets.
		0x0001 to 0x0006 (FDD)	The maximum number of interlaces on which the access terminal can simultaneously transmit MAC packets.
		0x0001 to 0x000c (TDD)	
		All other values	Reserved
0x0007	MaxPacketSizeFwd	<b>0x0001</b>	Maximum MAC packet size of 1 kbits can be received by the access terminal per interlace on the forward link.
		0x0001 to 0x0190	The maximum packet size that can be received by the access terminal per interlace on the forward link in units of kbits.
		All other values	Reserved

Attribute ID	Attribute	Values	Meaning
0x0008	MaxPacketSizeRev	<b>0x0001</b>	A maximum MAC packet size of 1 kbits that can be transmitted by the access terminal per interlace on the reverse link.
		0x0001 to 0x0064	The maximum packet size that can be transmitted by the access terminal per interlace on the reverse link in units of kbits.
		All other values	Reserved
0x0009	SCWLayersSupported	<b>0x0000</b>	The access terminal does not support SCW transmission.
		0x0001 to NumRx Antennas	The maximum number of layers that the access terminal can support in MIMO SCW transmission.
		All other values	Reserved
0x000a	MCWLayersSupported	<b>0x0000</b>	The access terminal does not support MCW transmission.
		0x0001 to NumRx Antennas	The maximum number of layers that the access terminal can support in MIMO MCW transmission
		All other values	Reserved
0x000b	STTDSupport	<b>0x0000</b>	The access terminal does not support STTD transmission.
		0x0001	The access terminal supports STTD transmission.
		All other values	Reserved
0x000c	HalfDuplexSupportRequired	<b>0x0000</b>	The access terminal does not require half duplex support from the access network.
		0x0001	The access terminal requires half duplex support from the access network.
		All other values	Reserved
0x000d	MaxPHYSubPacketSize	<b>0x0000</b>	The access terminal supports a maximum packet size of 4096 at the Physical Layer.
		0x0001	The access terminal supports a maximum packet size of 8192 at the Physical Layer.
		All other values	Reserved

## 2.5.8.2 Complex attributes

The following complex attributes and default values are defined (see 10.3 for attribute record definition).

### 2.5.8.2.1 SupportPPRAT attribute

*PP* is the two-digit hexadecimal RAT type except for 0x00 according to 11.4, where hexadecimal digits A through F are specified in upper case letters.

Field	Length (bits)	Default
Length	8	N/A
AttributeID	16	N/A
RATSupported	8	0x00
SupportedRATParametersLength	8	0x00
SupportedRATParameters	SupportedRATParametersLength × 8	N/A

**Length** Length of the complex attribute in octets. The sender shall set this field to the length of the complex attribute excluding the Length field.

**AttributeID** The sender shall set this field to 0x01*PP*.

**RATSupported** The sender shall set this field to 0x00 if the RAT *PP* is not supported. Otherwise, the sender shall set this field to 0x01 if the RAT *PP* is supported. All other values are reserved.

**SupportedRATParametersLength** The sender shall set this field to the length of the SupportedRATParameters record in units of octets. If the RATSupported field is set to 0x00, the sender shall set this field to 0x00. If the RATSupported field is set to 0x01, the sender shall set this field to 0x00 for RAT type 0x01 to 0x06.

**SupportedRATParameters** If SupportedRATParametersLength is 0x00, the sender shall omit this record.

### 2.5.8.2.2 SupportedChannelBands attribute

Field	Length (bits)	Default
Length	8	N/A
AttributeID	16	N/A
ChannelBandCount	8	0x00

ChannelBandCount occurrences of the following field:

{		
LowerChannelBandRecord	ChannelBandRecordType Dependent	N/A
UpperChannelBandRecord	ChannelBandRecordType Dependent	N/A

**Length** Length of the complex attribute in octets. The sender shall set this field to the length of the complex attribute excluding the Length field.

**AttributeID** The sender shall set this field to 0x0200.

**ChannelBandCount** The sender shall set this field to the number of ChannelBandRecords associated with this attribute. The sender shall include ChannelBandCount occurrences of the following two fields with the attribute.

**LowerChannelBandRecord**  
The sender shall set this field to the lower channel band record definition according to 10.1 for the range of channel bands supported.

**UpperChannelBandRecord**  
The sender shall set this field to the upper channel band record definition according to 10.1 for the range of channel bands supported.

### 2.5.9 Protocol numeric constants

Constant	Meaning	Value
N <sub>CDPType</sub>	Type field for this protocol	Table 9
N <sub>CDPDefault</sub>	Subtype field for this protocol	0x0000

### 2.5.10 Session state information

The Session State Information record (see 10.10) consists of parameter records.

The parameter records for this protocol consist of the configuration attributes of this protocol.



## 2.6 Default Inter Radio Access Technology (RAT) Protocol

### 2.6.1 Overview

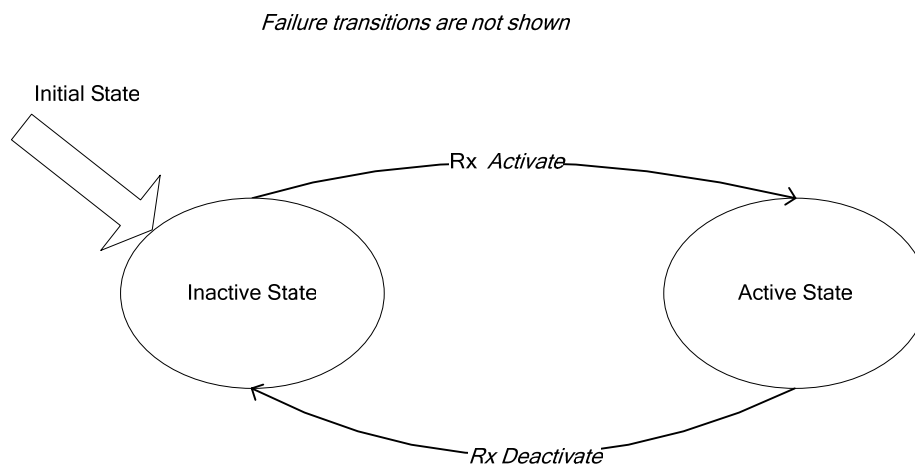
The Default Inter RAT Protocol allows the access network and access terminal to send messages for other radio access technologies.

This protocol uses the Signaling Transport to transmit and receive messages.

This protocol operates in one of two states:

- Inactive State: In this state, the protocol waits for an *Activate* command. There are no communications between the access terminal and the access network.
- Active State: In this state the access terminal or the access network may send a InterRATBlob message.

The protocol states and the messages and events causing the transition between the states are shown in Figure 15.



**Figure 15 Inter RAT Protocol state diagram**

### 2.6.2 Primitives

#### 2.6.2.1 Commands

This protocol defines the following commands:

- *Activate*
- *Deactivate*

#### 2.6.2.2 Return indications

This protocol does not return any indications.

## 2.6.3 Public data

### 2.6.3.1 Static public data

This protocol does not define any static public data.

### 2.6.3.2 Dynamic public data

- Subtype for this protocol

## 2.6.4 Protocol initialization and swap procedures

### 2.6.4.1 Protocol initialization

Upon creation, the protocol in the access terminal and access network shall perform the following:

- The protocol at the access terminal and the access network shall enter the Inactive State

### 2.6.4.2 Protocol swap

Upon swap, the protocol in the access terminal and the access network shall enter the Active State.

## 2.6.5 Procedures

### 2.6.5.1 Processing the Activate command

If the protocol receives the *Activate* command in the Inactive State, the protocol shall transition to the Active State.

If the protocol receives the *Activate* command in the Active State, the command shall be ignored.

### 2.6.5.2 Processing the Deactivate command

If the protocol receives the *Deactivate* command in the Inactive State, the command shall be ignored.

If the protocol receives the *Deactivate* command in the Active State, the protocol shall transition to the Inactive State

### 2.6.5.3 Inactive state

In this state, there are no communications between the access terminal and the access network.

In this state the protocol waits for the *Activate* command. See 2.6.5.1 for processing of the *Activate* command.

### 2.6.5.4 Active state

In this state the access terminal or the access network may send an InterRATBlob message.

## 2.6.6 Message formats

### 2.6.6.1 InterRATBlob

The access network or access terminal sends this message when it has another radio access technology's message to send.

Field	Length (bits)
MessageID	8
TechnologyType	8
TechnologyBlobLength	8
TechnologyBlob	8 × TechnologyBlobLength

**MessageID** The sender shall set this field to 0x00.

**TechnologyType** The sender shall include this field to indicate the type of technology as specified in 11.4.

**TechnologyBlobLength** The sender shall set this field to the length, in octets, of the TechnologyBlob.

**TechnologyBlob** The sender shall set this field to the message for the other technology. The interpretation of this field is beyond the scope of this specification.

<b>Channels</b>	RTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Unicast	<b>Security</b>	Required

### 2.6.6.2 InterRATIDRequest

The access network uses this message to query the access terminal of its RAT ID information.

Field	Length (bits)
MessageID	8
TransactionID	8

**MessageID** The access network shall set this field to 0x01.

**TransactionID** The access network shall set this field according to 10.8.

<b>Channels</b>	FTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Unicast	<b>Security</b>	Required

### 2.6.6.3 InterRATIDResponse

The access terminal sends this message in response to the InterRATIDRequest message.

Field	Length (bits)
MessageID	8
TransactionID	8
TechnologyTypeCount	8

TechnologyTypeCount occurrences of the following record:

{	
TechnologyIDType	8
TechnologyIDLength	8
TechnologyIDValue	$8 \times \text{TechnologyIDLength}$
}	

- MessageID** The access terminal shall set this field to 0x02.
- TransactionID** The access terminal shall set this field the TransactionID field of the corresponding InterRATIDRequest message.
- TechnologyTypeCount** The access terminal shall set this field to the number of TechnologyIDType fields in this message. The access terminal shall include TechnologyTypeCount occurrences of the following three fields with the message.
- TechnologyIDType** The access terminal shall include this field to indicate the type of technology as specified in 11.4. The access terminal shall not set this field to 0x00 or 0x01.
- TechnologyIDLength** The access terminal shall set this field to the length, in octets, of the TechnologyIDValue.
- TechnologyIDValue** The access terminal shall set this field to the unique ID (specified by TechnologyType) that has been assigned to the terminal by the radio access technology as specified in 11.4.

<b>Channels</b>	RTC	<b>SLP</b>	Reliable
<b>Addressing</b>	Unicast	<b>Security</b>	Required

## 2.6.7 Interface to other protocols

### 2.6.7.1 Commands

This protocol does not issue any commands.

### 2.6.7.2 Indications

This protocol does not register to receive any indications.

### 2.6.8 Configuration attributes

This protocol does not have any configurable attributes.

### 2.6.9 Protocol numeric constants

Constant	Meaning	Value
N <sub>IRATPType</sub>	Type field for this protocol	Table 9
N <sub>IRATPDefault</sub>	Subtype field for this protocol	0x0000

### 2.6.10 Session state information

The Session State Information record (see 10.10) consists of parameter records.

The parameter records for this protocol consist of the configuration attributes of this protocol.

## 3 Convergence Sublayer

### 3.1 Introduction

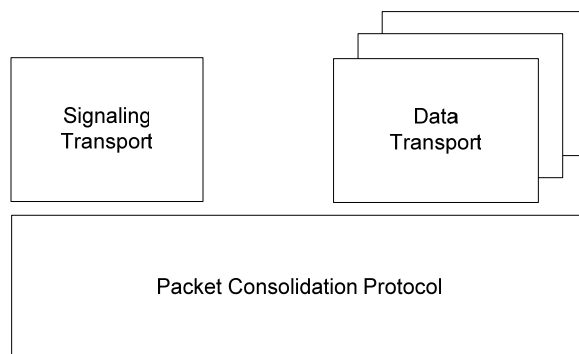
#### 3.1.1 General Overview

The Convergence Sublayer contains protocols and transports used to transport messages and data between the access terminal and the access network.

The Convergence Sublayer contains the following protocols and transports:

- **Signaling Transport:** Provides the means to carry messages between a protocol/transport in one entity and the same protocol/transport in the other entity. The Default Signaling Transport consists of a messaging protocol (Signaling Network Protocol) and a link layer protocol that provides message fragmentation, retransmission, and duplicate detection (Signaling Link Protocol).
- **Data Transport:** Provides the means to carry upper layer data. The Default Data Transport consists of a link layer protocol that provides fragmentation, retransmission, and duplicate detection (Radio Link Protocol); a Route Selection Protocol that provides two route instances for a higher layer packet flow; and a Flow Control Protocol that provides flow control of data traffic.
- **Packet Consolidation Protocol:** Provides multiplexing of distinct transports, transmit prioritization and packet encapsulation. The Default Packet Consolidation Protocol provides 8 Transports. Each Transport defined by the Default Packet Consolidation Protocol maps to a data-bearing transport such as the Signaling or Data Transport. The first Transport (Transport 0) always carries Signaling, and the other Transports can be used to carry, for example, the Default Packet Transport to support different Quality of Service (QoS) requirements for data or other transports.

The relationship between the Convergence Sublayer protocols is illustrated in Figure 16.



**Figure 16 Convergence Sublayer protocols**

## 3.2 Default Signaling Transport

### 3.2.1 Introduction

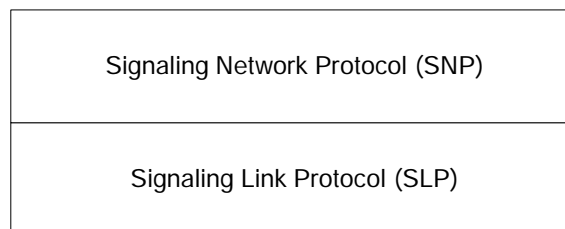
#### 3.2.1.1 General overview

The Default Signaling Transport is used to transport messages that manage air interface protocol objects in the access network and access terminal. The Default Signaling Transport encompasses the Signaling Network Protocol (SNP) and the Signaling Link Protocol (SLP). Protocols and transports use SNP to exchange messages.

SNP provides a one octet header that defines the Type of the protocol or transport with which the message is associated. The SNP uses the header to route the message to the appropriate protocol or transport instance.

SLP provides message fragmentation, reliable and best-effort message delivery and duplicate detection for messages that are delivered reliably.

The relationship between SNP and SLP is illustrated in Figure 17.



**Figure 17 Default signaling transport protocols**

#### 3.2.1.2 Public data

##### 3.2.1.2.1 Static public data

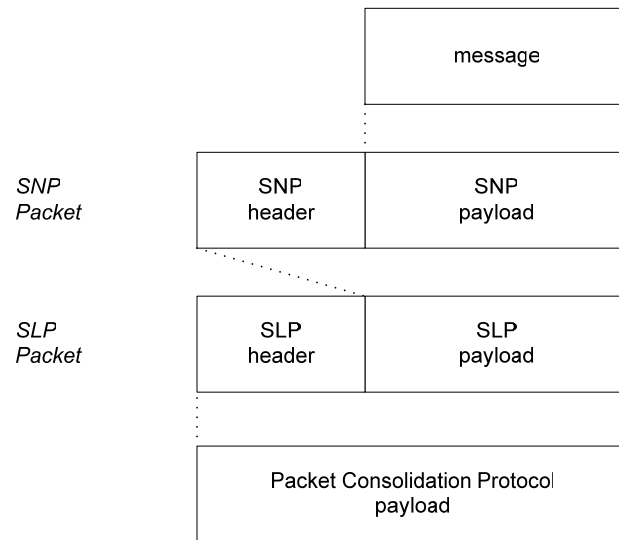
This transport does not define any static public data.

##### 3.2.1.2.2 Dynamic public data

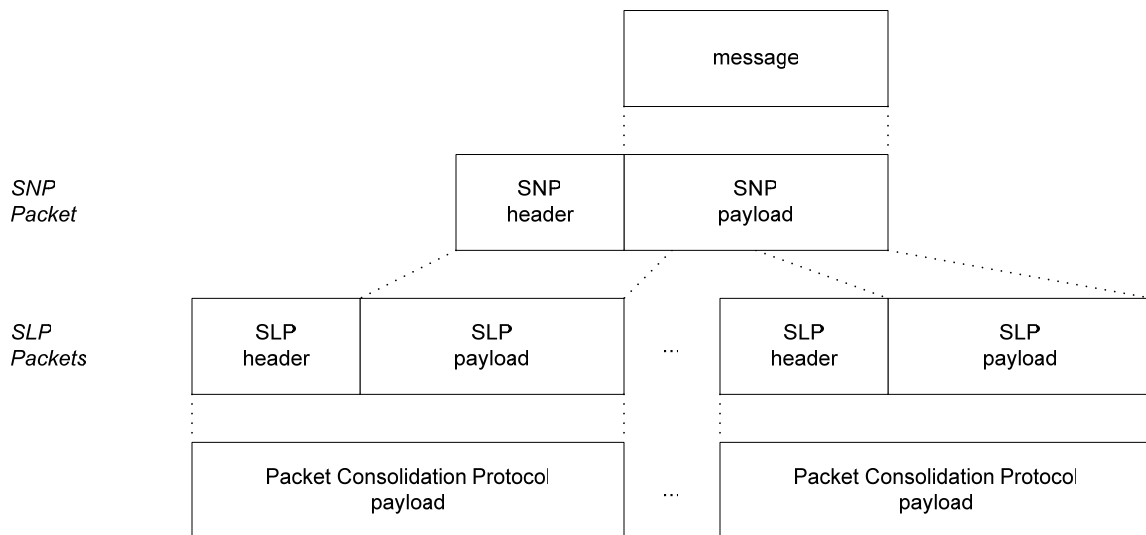
- Subtype for this transport

### 3.2.1.3 Data encapsulation

Figure 18 and Figure 19 illustrate the relationship between a message, SNP packets, SLP packets, and Packet Consolidation Protocol payloads. Figure 18 shows a case where SLP does not fragment the SNP packet. Figure 19 shows a case where the SNP packet is fragmented into more than one SLP payload.



**Figure 18 Message encapsulation (non-fragmented)**



**Figure 19 Message encapsulation (fragmented)**



## 1 **3.2.2 Transport initialization and swap procedures**

### 2 **3.2.2.1 Transport initialization**

3 Upon creation, the instance of the Signaling Transport in the access terminal and access network shall  
4 set the value of the attributes for this transport to the default values specified for each attribute.

### 5 **3.2.2.2 Transport swap**

6 This transport defines an empty swap procedure.

## 7 **3.2.3 General signaling requirements**

### 8 **3.2.3.1 General requirements**

9 The following requirements are common to all protocols and transports that carry messages using  
10 SNP and that provide for message extensibility. The access terminal and the access network shall  
11 abide by the following rules when generating and processing any signaling message carried by SNP:

- 12 ■ Messages shall be an integer number of octets in length.
- 13 ■ The fields of the message shall be generated in the order specified by the message format  
14 definition. Within each field, the most significant bit of the field shall be generated and  
15 processed first.
- 16 ■ Message identifiers shall be unambiguous for each protocol Type and for each protocol  
17 and transport Subtype for all protocols and transports compatible with the Air Interface,  
18 defined by MinimumRevision and above.
- 19 ■ For future revisions, the transmitter shall add new fields only at the end of a message.  
20 The transmitter shall not add fields if their addition makes the parsing of previous fields  
21 ambiguous for receivers whose protocol revision is equal to or greater than  
22 MinimumRevision.
- 23 ■ The receiver shall discard all unrecognized messages.
- 24 ■ The receiver shall discard all unrecognized fields.
- 25 ■ The receiver shall discard a message if any of the fields in the message is set to a value  
26 outside of the defined field range, unless the receiver is specifically directed to ignore this  
27 field. A field value is outside of the allowed range if a range was specified with the field  
28 and the value is not in this range, or the field is set to a value that is defined as invalid.  
29 The receiver shall discard a field in a message if the field is set to a reserved value.

### 3.2.3.2 Message information

Each message definition contains information regarding channels on which the message can be transmitted, whether the message requires SLP reliable or best-effort delivery, and the addressing modes applicable to the message. This information is provided in the form of a table, an example of which is given in Figure 20.

<b>Channels</b>	FTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Unicast	<b>Security</b>	Required

**Figure 20 Sample message information**

The following values are defined:

- Channels: This information field indicates the MAC protocols in the data path on which this message can be transmitted. The sender of the message shall send the message only on the MAC protocol(s) indicated by this information field. Values are:
  - FTC for Forward Traffic Channel MAC,
  - RTC for Reverse Traffic Channel MAC.
- SLP: Signaling Link Protocol requirements. The sender of the message shall send the message only using the SLP in the mode(s) indicated by this information field. Values are:
  - Best Effort: the message is sent once and is subject to erasure, and
  - Reliable: erasures are detected and the message is retransmitted one or more times, if necessary.
- Addressing: Addressing modes for the message. The sender of the message shall send the message only with an address type(s) indicated by this information field. Values are:
  - Broadcast if a broadcast address can be used with this message, and
  - Unicast if a unicast address can be used with this message.
- Security: Security modes for the message. The sender of the message shall send the message only with a security type(s) indicated by this information field. Values are:
  - Required: if SecurityEnabled public data of the Security Protocol is set to '1', then the message shall be sent with IsSecure field of the Lower MAC header set to '1'. Any message received when SecurityEnabled public data of the Security Protocol is set to '1' and the IsSecure field of the Lower MAC header is set to '0' shall be discarded, and
  - Optional: the message is always processed.

## 3.2.4 Signaling Network Protocol

### 3.2.4.1 Overview

The Signaling Network Protocol (SNP) routes messages to protocols and transports specified by the Type field provided in the SNP header.

The actual protocol indicated by the Type is defined by the InUse SessionConfigurationToken. For example, Type 0x11 is associated with the Session Management Protocol. The specific Session Management Protocol used (and, therefore, the Session Management protocol generating and processing the messages delivered by SNP) is defined by the InUse SessionConfigurationToken.

The Type field forms a single octet header.

The remainder of the message following the SNP header is processed by the protocol specified by the Type.

SNP is a protocol associated with the Default Signaling Transport.

### 3.2.4.2 Primitives

#### 3.2.4.2.1 Commands

This protocol does not define any commands.

#### 3.2.4.2.2 Return indications

This protocol does not return any indications.

### 3.2.4.3 Protocol data unit

The protocol data unit for this protocol is an SNP packet. Each SNP packet consists of one message sent by a protocol using SNP.

The protocol constructs an SNP packet by adding the SNP header (see 3.2.4.6) in front of the payload. The structure of the SNP packet is shown in Figure 21.

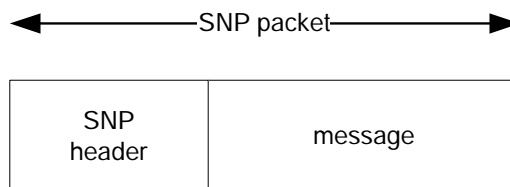


Figure 21 SNP packet structure

### 3.2.4.4 Procedures

SNP receives messages for transmission from multiple protocols and transports. SNP shall add the SNP header to each message and forward it for transmission to SLP.

SNP receives messages from SLP. SNP shall route these messages to their associated protocols and transports according to the value of the Type field in the SNP header.

If an SNP message is to be transmitted on the Forward Traffic Channel or on the Reverse Traffic Channel, and if a connection is not open, SNP shall issue an *AirLinkManagement.OpenConnection* command. SNP should queue all messages requiring transmission in the Forward Traffic Channel or in the Reverse Traffic Channel until the protocol receives an *IdleState.ConnectionOpened* indication.

### 3.2.4.5 Type definitions

Type definitions associated with the default protocol stack are presented in Table 9. The constant name and protocol layer are provided for informational purposes.

**Table 9 Default protocol stack type values**

Type	Protocol	Constant Name
0x00	Physical Layer Protocol	N <sub>PHYType</sub>
0x01	Control Channel MAC Protocol	N <sub>CCMPType</sub>
0x02	Access Channel MAC Protocol	N <sub>ACMPType</sub>
0x03	Forward Traffic Channel MAC Protocol	N <sub>FTCMPType</sub>
0x04	Reverse Traffic Channel MAC Protocol	N <sub>RTCMPType</sub>
0x05	Reverse Control Channel MAC Protocol	N <sub>RCCMPType</sub>
0x06	Shared Signaling MAC Protocol	N <sub>SSMPType</sub>
0x07	Air Link Management Protocol	N <sub>ALMPType</sub>
0x08	Initialization State Protocol	N <sub>ISPType</sub>
0x09	Idle State Protocol	N <sub>IDPType</sub>
0x0a	Connected State Protocol	N <sub>CSPType</sub>
0x0b	Active Set Management Protocol	N <sub>ASMPType</sub>
0x0c	Overhead Messages Protocol	N <sub>OMPType</sub>
0x0d	Authentication Protocol	N <sub>APType</sub>
0x0e	Encryption Protocol	N <sub>EPType</sub>
0x0f	Security Protocol	N <sub>SPType</sub>
0x10	Key Exchange Protocol	N <sub>KEPType</sub>
0x11	Session Management Protocol	N <sub>SMPType</sub>
0x12	Address Management Protocol	N <sub>ADMPType</sub>
0x13	Session Configuration Protocol	N <sub>SCPType</sub>
0x14	Capabilities Discovery Protocol	N <sub>CDPTyp</sub>
0x15	InterRAT Protocol	N <sub>IRATPType</sub>
0x16	Packet Consolidation Protocol	N <sub>PCPType</sub>
0x17	Transport 0	N <sub>TPT0Type</sub>

Type	Protocol	Constant Name
0x18	Transport 1	$N_{TPT1Type}$
0x19	Transport 2	$N_{TPT2Type}$
0x1a	Transport 3	$N_{TPT3Type}$
0x1b	Transport 4	$N_{TPT4Type}$
0x1c	Transport 5	$N_{TPT5Type}$
0x1d	Transport 6	$N_{TPT6Type}$
0x1e	Transport 7	$N_{TPT7Type}$

### 3.2.4.6 SNP packet header

The SNP shall place the following header in front of every message that it sends:

Field	Length (bits)
Type	8

Type This field shall be set the Type value in Table 9 for the protocol or Transport associated with the encapsulated message.

### 3.2.4.7 Message formats

No messages are defined for this protocol.

### 3.2.4.8 Interface to other protocols

#### 3.2.4.8.1 Commands

This protocol issues the following command:

- *AirLinkManagement.OpenConnection*

#### 3.2.4.8.2 Indications

This protocol registers to receive the following indications:

- *IdleState.ConnectionOpened*

## 3.2.5 Signaling Link Protocol

### 3.2.5.1 Overview

The purpose of the Signaling Link Protocol (SLP) is to provide best effort and reliable delivery for SNP packets. SLP provides retransmission and duplicate detection for messages using reliable delivery. SLP provides fragmentation and re-assembly for SNP packets. SLP does not ensure in-order delivery of SNP packets.

The delivery flow variable  $P$  takes value “BE” or ‘0’ for best effort delivery, and value “RD” or ‘1’ for reliable delivery. The reliable delivery flow provides two sequence spaces variables  $Q_{Tx}$  and  $Q_{Rx}$  for transmission and reception of SNP packets respectively. The transmitter toggles the sequence

1 space variable  $Q_{Tx}$  between '0' and '1' to indicate a reset. The receiver sequence space variable  $Q_{Rx}$   
 2 tracks the value of  $Q_{Tx}$  and detects when the transmitter has performed a reset. For best effort delivery  
 3 flow, the SequenceSpace field in the SLP header takes value '0' to indicate no sequence number or  
 4 packet framing fields are present in the SLP header, and value '1' to indicate the presence of the  
 5 sequence number and packet framing fields in the SLP header.

6 SLP is a protocol associated with the Default Signaling Transport.

### 7 **3.2.5.2 Primitives**

#### 8 **3.2.5.2.1 Commands**

9 This protocol does not define any commands.

#### 10 **3.2.5.2.2 Return indications**

11 This protocol does not return any indications.

#### 12 **3.2.5.3 Protocol data unit**

13 The transmission unit of this protocol is an SLP packet.

### 14 **3.2.5.4 Procedures**

15 Unless explicitly specified, SLP requirements for the access terminal and the access network are  
 16 identical; and are, therefore, presented in terms of transmitter and receiver.

17 SLP receives SNP packets for transmission and forms an SLP packet by prepending the SLP packet  
 18 header defined in 3.2.5.5 to a contiguous subset of the received octets. The policy SLP follows in  
 19 determining the number of octets to send in an SLP packet is beyond the scope of this specification. It  
 20 is subject to the following requirements:

- 21 ■ The size of an SLP packet shall not exceed the maximum payload length that can be  
 22 carried by the Packet Consolidation Protocol given the target channel and current  
 23 transmission rate on that channel.
- 24 ■ The SLP payload shall contain octets from no more than one SNP packet.

25 SLP shall construct the SLP payload(s) from an SNP packet. If the SNP packet exceeds the current  
 26 maximum SLP payload size, then the sender shall fragment the SNP packet. If the sender does not  
 27 fragment the SNP packet, then the SNP packet is the SLP payload. If the sender does fragment the  
 28 SNP packet, then each SNP packet fragment is an SLP payload.

29 SLP makes use of the ResetRxRequest, ResetRxAck, ResetTxIndication, ResetTxAck, and  
 30 ReceiverStatus messages to perform control related operations.

31 SLP is an Ack and Nak-based protocol with a sequence space size of  $2^{\text{SequenceLength}}$  bytes.

32 All operations and comparisons performed on SLP packet sequence numbers shall be carried out in  
 33 unsigned modulo  $2^J$  arithmetic, where J represents the value of SequenceLength. For any SLP octet  
 34 sequence number  $X$ , the sequence numbers in the range  $[X+1, X+2^{J-1}-1]$  shall be considered greater  
 35 than  $X$  and the sequence numbers in the range  $[X-2^{J-1}, X-1]$  shall be considered smaller than  $X$ .

### 3.2.5.4.1 Initialization and reset

The SLP initialization procedure initializes the SLP variables and data structures in one end of the link. The SLP reset procedure guarantees that SLP state variables on both sides are synchronized. The reset procedure includes initialization.

If the protocol receives an *IdleState.ConnectionOpened* indication then the access terminal and the access network shall perform the initialization procedures defined in 3.2.5.4.1.1.1 and 3.2.5.4.1.1.2 for both the reliable and best effort flows.

The SLP shall set the sequence space variables  $Q_{Tx}$  and  $Q_{Rx}$  to '0' after reception of an *IdleState.ConnectionOpened* indication. The SLP shall toggle the value of the sequence space variables  $Q_{Tx}$  and  $Q_{Rx}$  between '0' and '1' for every subsequent reset.

#### 3.2.5.4.1.1 Initialization procedure

##### 3.2.5.4.1.1.1 Initialization procedure for the SLP transmitter

When SLP transmitter performs the initialization procedure it shall:

- Reset the send state variable  $V(S)_P$  to zero.
- Clear the retransmission queue.

##### 3.2.5.4.1.1.2 Initialization procedure for the SLP receiver

When SLP receiver performs the initialization procedure it shall:

- Reset the receive state variables  $V(R)_P$  and  $V(N)_P$  to zero.
- Clear the re-assembly buffer.

#### 3.2.5.4.1.2 Reset procedure

The reset procedure shall only be used to reset the reliable delivery flows.

##### 3.2.5.4.1.2.1 Reset procedure for the initiating side when it is an SLP transmitter

If the side initiating a reset procedure for the reliable delivery flow is an SLP transmitter, then it shall:

- Perform the SLP transmitter initialization procedure defined in 3.2.5.4.1.1.1 for the reliable delivery flow.
- Toggle the value of the sequence space variable  $Q_{Tx}$ .
- Send a ResetTxIndication message.

The SLP transmitter shall not reset again until it receives a ReceiverStatus message with a SequenceSpace field equal to the new value of the sequence space variable  $Q_{Tx}$  from the SLP receiver, or a ResetTxAck message with a TransactionID field equal to the TransactionID sent in the ResetTxIndication message.

The SLP transmitter shall ignore any received ResetRxRequest messages until it receives a ReceiverStatus message with a SequenceSpace field with the new value of the sequence space

1 variable  $Q_{Tx}$  from the SLP receiver, or a ResetTxAck message with a TransactionID field equal to the  
2 TransactionID sent in the ResetTxIndication message.

3 The SLP transmitter may determine that the ResetTxIndication was lost if it does not receive a  
4 ReceiverStatus message with a SequenceSpace field equal to the new value of the sequence space  
5 variable  $Q_{Tx}$  from the SLP receiver, or a ResetTxAck message, within an implementation-dependent  
6 time interval based on  $T_{SLPResponse}$  and an estimate of the round-trip delay. If the SLP transmitter  
7 determines that the ResetTxIndication was lost, then the SLP transmitter shall send a new  
8 ResetTxIndication message.

#### 9 **3.2.5.4.1.2.2 Reset procedure for initiating side when it is an SLP receiver**

10 If the side initiating a reset procedure for the reliable delivery flow is an SLP receiver, then it shall  
11 enter the SLP Reset state. Upon entering the SLP Reset state, SLP shall:

- 12 ■ Perform the SLP receiver initialization procedure defined in 3.2.5.4.1.1.2 for the reliable  
13 delivery flow.
- 14 ■ Toggle the value of the sequence space variable  $Q_{Rx}$ .
- 15 ■ Send a ResetRxRequest message
- 16 ■ Ignore all SLP data octets received while in the SLP Reset state for the reliable delivery  
17 flow with SequenceSpace field not equal to the sequence space variable  $Q_{Rx}$ .
- 18 ■ If a ResetRxAck message is received with a TransactionID field equal to the  
19 TransactionID sent in the ResetRxRequest message, SLP shall leave the Reset state.
- 20 ■ If an SLP data octet is received with a SequenceSpace field equal to the sequence space  
21 variable  $Q_{Rx}$ , SLP shall leave the SLP Reset state.

22 If a ResetRxAck is received while not in the SLP Reset state, the message shall be ignored.

23 The SLP receiver may determine that the ResetRxRequest was lost if it does not leave the SLP Reset  
24 state within an implementation-dependent time interval based on  $T_{SLPResponse}$  and an estimate of the  
25 round-trip delay. If the SLP receiver determines that the ResetRxRequest was lost, then the SLP  
26 receiver shall send a new ResetRxRequest message.

#### 27 **3.2.5.4.1.2.3 Reset procedure for the responding side when it is an SLP receiver**

28 If the side responding to a reset procedure for the reliable delivery flow is an SLP receiver, then upon  
29 receiving a ResetTxIndication message, SLP shall perform the following procedures:

- 30 ■ If the sequence space variable  $Q_{Rx}$  is not equal to the value of the SequenceSpace field in  
31 the ResetTxIndication message, then SLP shall:
  - 32 □ Perform the SLP receiver initialization procedure defined in 3.2.5.4.1.1.2 for the  
33 reliable delivery flow.
  - 34 □ Toggle the value of the sequence space variable  $Q_{Rx}$ .
- 35 ■ Respond with a ResetTxAck message.



1 Upon receiving an SLP data octet for the reliable delivery flow with a SequenceSpace field not equal  
2 to the sequence space variable  $Q_{Rx}$ , SLP shall:

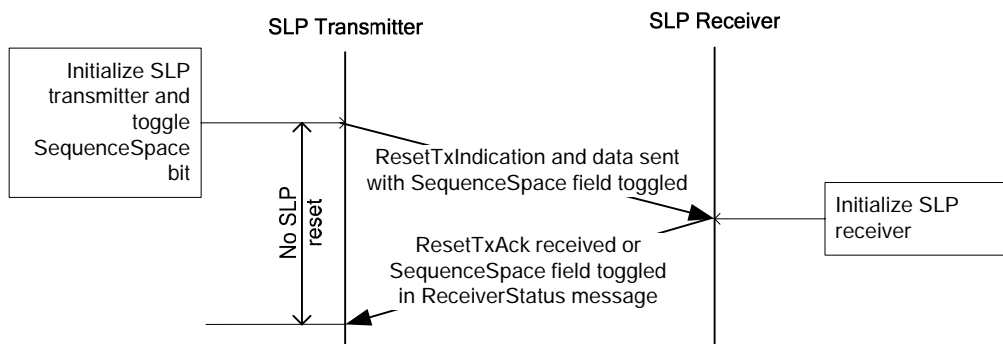
- 3 ■ Perform the SLP receiver initialization procedure defined in 3.2.5.4.1.1.2 for the reliable  
4 delivery flow.
- 5 ■ Toggle the value of the sequence space variable  $Q_{Rx}$ .

#### 6 3.2.5.4.1.2.4 Reset procedure for the responding side when it is a SLP transmitter

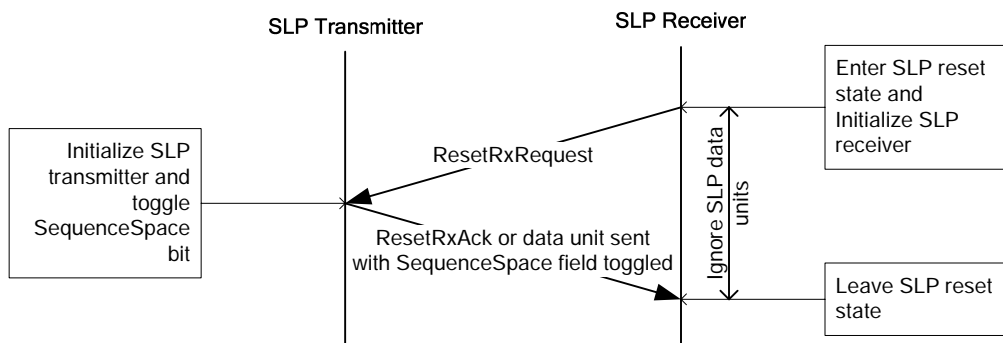
7 If the side responding to a reset procedure for the reliable delivery flow is an SLP transmitter, then  
8 upon receiving a ResetRxRequest message, SLP shall perform the following procedures:

- 9 ■ If the sequence space variable  $Q_{Tx}$  is not equal to the value of the SequenceSpace field in  
10 the ResetRxRequest message, then SLP shall:
  - 11 □ Perform the SLP transmitter initialization procedure defined in 3.2.5.4.1.1.1 for the  
12 reliable delivery flow.
  - 13 □ Toggle the value of the sequence space variable  $Q_{Tx}$ .
- 14 ■ Respond with a ResetRxAck message.

#### 15 3.2.5.4.1.2.5 SLP Reset message flows



16  
17 **Figure 22 SLP reset procedure initiated by SLP transmitter**



18  
19 **Figure 23 SLP reset procedure initiated by SLP receiver**

### 1 3.2.5.4.2 SLP transmit procedures

2 The SLP transmitter shall maintain a SequenceLength-bit variable  $V(S)_P$  for all transmitted SLP octets  
 3 (see Figure 24), where the delivery flow  $P$  takes on the value “BE” or ‘0’ for best effort delivery and  
 4 “RD” or ‘1’ for reliable delivery.  $V(S)_P$  is the sequence number of the next SLP octet to be sent on  
 5 delivery flow  $P$ . The sequence number field (SEQ) in each new SLP packet transmitted shall be set to  
 6  $V(S)_P$ , corresponding to the sequence number of the first octet in the payload. The sequence number  
 7 of the  $i^{\text{th}}$  octet in the payload (with the first octet being octet 0) is implicitly given by  $\text{SEQ}+i$ . If the  
 8 SEQ field is included in the SLP header, then  $V(S)_P$  shall be incremented for each octet contained in  
 9 the SLP payload. If the SEQ field is not included in the SLP header, then  $V(S)_P$  shall not be  
 10 incremented.

11 If the SLP payload contains the beginning of an SNP packet, then the sender shall set the SLP header  
 12 First field to ‘1’; otherwise, the sender shall set the SLP header First field to ‘0’.

13 If the SLP payload contains the end of an SNP packet, then the sender shall set the SLP header Last  
 14 field to ‘1’; otherwise, the sender shall set the SLP header Last field to ‘0’.

#### 15 3.2.5.4.2.1 Best effort delivery transmit procedures

16 If the SLP payload contains the beginning and end of an SNP packet, the sender shall set the SLP  
 17 header SequenceSpace field to ‘0’; otherwise, the sender shall set the SLP header SequenceSpace  
 18 field to ‘1’.

#### 19 3.2.5.4.2.2 Reliable delivery transmit procedures

20 If a ReceiverStatus message is received with the SequenceSpace field not equal to the value of  $Q_{Tx}$ ,  
 21 the message shall be ignored.

22 If the SLP transmitter is an access terminal, and if a  
 23 *ReverseTrafficChannelMAC.ReverseTrafficPacketsMissed* indication is received for octets sent with  
 24 the sequence space not equal to the value of  $Q_{Tx}$ , then the indication shall be ignored.

25 Upon receiving a *ReverseTrafficChannelMAC.ReverseTrafficPacketsMissed* indication, the SLP  
 26 transmitter in the access terminal shall transmit the requested octets(s) if the requested octets have not  
 27 been retransmitted  $N_{\text{SLPAttempt}}-1$  times before.

28 If the SLP transmitter is an access network, and if a  
 29 *ForwardTrafficChannelMAC.ForwardTrafficPacketsMissed* indication is received for octets sent  
 30 with the sequence space not equal to the value of  $Q_{Tx}$ , then the indication shall be ignored.

31 Upon receiving a *ForwardTrafficChannelMAC.ForwardTrafficPacketsMissed* indication, the SLP  
 32 transmitter in the access network shall transmit the requested octets(s) if the requested octets have not  
 33 been retransmitted  $N_{\text{SLPAttempt}}-1$  times before.

34 Upon receiving a ReceiverStatus message, SLP shall transmit the missing octet(s) (if any) conveyed  
 35 by the ReceiverStatus message if those octets have not been retransmitted  $N_{\text{SLPAttempt}}-1$  times before. If  
 36 the  $V(R)_{RD}$  conveyed in the ReceiverStatus message is smaller than  $V(S)_{RD} - 1$ , then the SLP  
 37 transmitter may re-transmit one or more of the octets with sequence numbers from  $V(R)_{RD}$  to  $V(S)_{RD} -$   
 38 1, inclusive, if those octets have not been retransmitted  $N_{\text{SLPAttempt}}-1$  times before.

1 The SLP transmitter shall meet the following requirements for each octet transmitted:

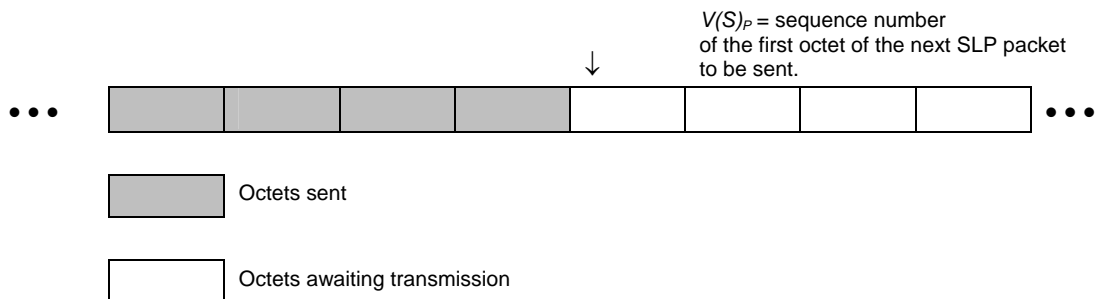
- 2 1. After transmitting an octet, the SLP transmitter shall start a wait Ack timer for time  
3  $T_{\text{SLPWaitAck}}$ .
- 4 2. If the SLP transmitter receives a ReceiverStatus message acknowledging the octet  
5 before the wait Ack timer expires, the SLP transmitter shall disable the timer.
- 6 3. If the timer expires and the octet has not been retransmitted  $N_{\text{SLPAttempt}}-1$  times before,  
7 the SLP transmitter shall retransmit the octets and repeat steps 1 and 2.

8 If the SLP transmitter is the access network, and the ReceiverStatus record includes any sequence  
9 number greater than or equal to  $V(S)_{RD}$ , SLP shall perform the reset procedures specified in  
10 3.2.5.4.1.2.1 for forward link reliable delivery flow.

11 If the SLP transmitter is the access terminal, and the ReceiverStatus record includes any sequence  
12 number greater than or equal to  $V(S)_{RD}$ , SLP shall perform the reset procedures specified in  
13 3.2.5.4.1.2.1 for reverse link reliable delivery flow.

14 If SLP has already transmitted  $2^{\text{SequenceLength}-1}$  SLP octets, SLP shall transmit an SLP octet with  
15 sequence number  $n$ , only after receiving acknowledgments for the SLP octets transmitted with  
16 sequence number  $n - 2^{\text{SequenceLength}-1}$  and below, or after determining that these SLP octets could not be  
17 delivered.

18 Reliable delivery SLP packets shall be stored in the buffer when they are first transmitted and may be  
19 deleted from the buffer, when they are acknowledged or when SLP determines that they could not be  
20 delivered.

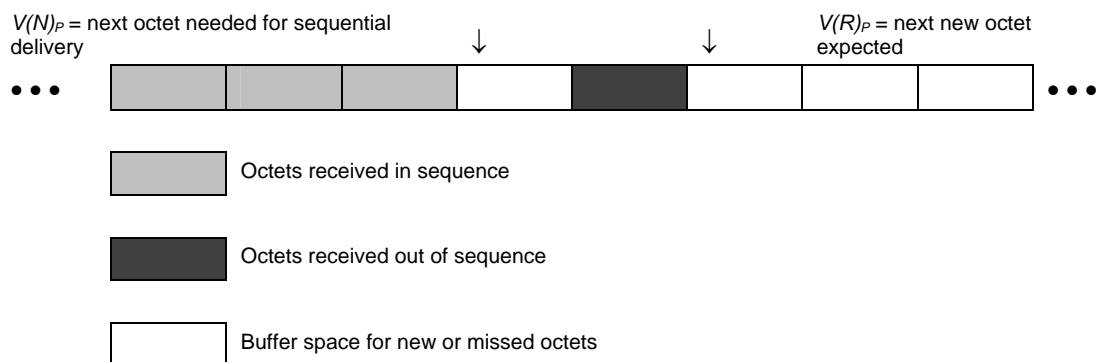


22 **Figure 24 SLP transmit sequence number variable**

### 23 3.2.5.4.3 SLP receive procedures

24 The SLP receiver shall maintain an independent re-assembly buffer for each Connection Endpoint as  
25 defined by the Address Management Protocol.

26 The SLP receiver shall maintain two SequenceLength-bit variables for receiving,  $V(R)_P$  and  $V(N)_P$   
27 (see Figure 25), where  $P$  is “BE” or ‘0’ for best effort delivery and “RD” or ‘1’ for reliable delivery.  
28  $V(R)_P$  contains the sequence number of the next octet expected to arrive.  $V(N)_P$  contains the sequence  
29 number of the first missing octet, as described below.



2 **Figure 25 SLP receive sequence number variables**

3 In addition, the SLP receiver shall keep track of the status of each octet in its re-assembly buffer  
 4 indicating whether the octet was received or not. Use of this status is implied in the following  
 5 procedures.

6 **3.2.5.4.3.1 Best effort delivery receive procedures**

7 If the SequenceSpace field in the SLP header is '0', SLP shall pass the complete SNP packet to the  
 8 SNP. Otherwise, in the following,  $X$  denotes the sequence number of a received octet. For each  
 9 received octet, SLP shall perform the following procedures:

- 10
- 11 ■ If  $X < V(R)_{BE}$ :
    - 12 □ SLP shall perform the initialization procedure defined in 3.2.5.4.1.1.2 for the best effort flow.
  - 13 ■ SLP shall store the received octet in the re-assembly buffer.
  - 14 ■ SLP shall set  $V(R)_{BE}$  to  $X+1$ .
  - 15 ■ SLP shall pass all complete SNP packets in the re-assembly buffer, that have not been  
 16 passed to the SNP, from the beginning of the re-assembly buffer upward, to the SNP.

17 The SLP receiver shall meet the following requirements for each SLP packet received on a best effort  
 18 flow:

- 19
- 20 ■ If the SLP packet is not carrying the last segment of a fragmented SNP packet, the SLP  
 21 receiver shall start a wait next segment timer for time  $T_{SLPWaitNextSegment}$ . If the wait next  
 22 segment timer is currently enabled, the SLP receiver shall reset and restart the timer.
  - 23 ■ If the SLP packet is carrying the last segment of a fragmented SNP packet, the SLP  
 24 receiver shall disable the timer.
  - 25 ■ If the timer expires and the last segment of a fragmented SNP packet has not been  
 26 received, the SLP receiver shall perform the initialization procedures defined in  
 3.2.5.4.1.1.2 for the best effort flow.

### 3.2.5.4.3.2 Reliable delivery receive procedures

The SLP receiver informs the SLP transmitter of the status of octets in its receive buffer by sending a ReceiverStatus message. The ReceiverStatus message shall convey all missing data from  $V(N)_{RD}$  onwards that has not been conveyed in a ReceiverStatus message  $N_{SLPAttempt}-1$  times before, and  $V(R)_{RD}$ . The ReceiverStatus message may convey missing data that has been conveyed in  $N_{SLPAttempt}-1$  previous ReceiverStatus messages. The ReceiverStatus message shall not convey status of octets with sequence number less than  $V(N)_{RD}$ .

In the following,  $X$  denotes the sequence number of a received octet. For each received octet, SLP shall perform the following procedures:

- The SLP receiver shall send a ReceiverStatus message for the octet such that the message arrives at the SLP transmitter before the  $T_{SLPWaitAck}$  timer expires.
- If  $X < V(N)_{RD}$ , the octet shall be discarded as a duplicate.
- If  $V(N)_{RD} \leq X < V(R)_{RD}$ , and the octet is not already stored in the re-assembly buffer, then:
  - SLP shall store the received octet in the re-assembly buffer.
  - SLP shall pass all complete SNP packets in the re-assembly buffer, that have not been passed to the SNP, from the beginning of the re-assembly buffer upward, to the SNP.
  - If  $X = V(N)_{RD}$ , then SLP shall set  $V(N)_{RD}$  to (LAST+1) where LAST is the sequence number of the last contiguous octet in the re-assembly buffer.
- If  $V(N)_{RD} < X < V(R)_{RD}$ , and the octet has already been stored in the re-assembly buffer, then the octet shall be discarded as a duplicate.
- If  $X = V(R)_{RD}$ , then:
  - SLP shall store the received octet in the re-assembly buffer.
  - SLP shall pass all complete SNP packets in the re-assembly buffer, that have not been passed to the SNP, from the beginning of the re-assembly buffer upward, to the SNP.
  - If  $V(R)_{RD} = V(N)_{RD}$ , then SLP shall increment  $V(N)_{RD}$  and  $V(R)_{RD}$ .
  - If  $V(R)_{RD} \neq V(N)_{RD}$ , then SLP shall increment  $V(R)_{RD}$ .
- If  $X > V(R)_{RD}$ , then:
  - SLP shall store the octet in the re-assembly buffer.
  - SLP shall pass all complete SNP packets in the re-assembly buffer, that have not been passed to the SNP, from the beginning of the re-assembly buffer upward, to the SNP.
  - SLP shall include a Nak for the missing SLP octets from  $V(R)_{RD}$  to  $X-1$ , inclusive in the ReceiverStatus message.
  - SLP shall set  $V(R)_{RD}$  to  $X+1$ .

The SLP receiver shall include all missing octets in each ReceiverStatus message sent. If  $N_{SLPAttempt}-1$  Naks have been sent for a missing octet, the SLP shall set  $V(N)_{RD}$  to the sequence number of the next missing octet, or to  $V(R)_{RD}$  if there are no remaining missing octets. If the SLP receiver determines that a missing octet shall not be retransmitted, the SLP shall set  $V(N)_{RD}$  to the sequence number of the

1 next missing octet, or to  $V(R)_{RD}$  if there are no remaining missing octets. The SLP may determine that  
 2 a missing octet shall not be retransmitted based on the arrival time of the first octet received after the  
 3 missing octet, the number of attempts for each octet  $N_{SLPAttempt}$  and the retransmission time  $T_{SLPWaitAck}$ .  
 4 Further recovery is the responsibility of the protocol sending the missing SNP packet(s).

### 5 3.2.5.5 SLP packet header

6 The SLP packet header, which precedes the SLP payload, has the following format:  
 7

Field	Length (bits)
ReliableDelivery	1
SequenceSpace	1
First	0 or 1
Last	0 or 1
SEQ	0 or SequenceLength
Reserved	0 or 6

8 **ReliableDelivery** Reliable or best effort delivery flag. The sender shall set this flag to '1' for  
 9 the reliable delivery flow. Otherwise the sender shall set this flag to '0'.

10 **SequenceSpace** Sequence space flag for reliable delivery, and sequence space and framing  
 11 present flag for best effort delivery. If the ReliableDelivery field is set to '1',  
 12 the sender shall set this flag to the value of the sequence space variable  $Q_{Tx}$ .  
 13 If the ReliableDelivery field is set to '0', the sender shall set this flag to '1' if  
 14 the First, Last and SEQ fields are included in the SLP packet header.  
 15 Otherwise, the sender shall set this flag to '0'.

16 **First** The sender shall include this field if the ReliableDelivery field is set to '1', or  
 17 the ReliableDelivery field is set to '0' and the SequenceSpace field is set to 1.  
 18 Otherwise the sender shall omit this field. If the payload of this SLP packet is  
 19 the first segment of a SNP packet, then the sender shall set this field to '1'.  
 20 Otherwise, the sender shall set this field to '0'.

21 **Last** The sender shall include this field if the ReliableDelivery field is set to '1', or  
 22 the ReliableDelivery field is set to '0' and the SequenceSpace field is set to 1.  
 23 Otherwise the sender shall omit this field. If the payload of this SLP packet is  
 24 the last segment of a SNP packet, then the sender shall set this field to '1'.  
 25 Otherwise, the sender shall set this field to '0'.

26 **SEQ** The sender shall include this field if the ReliableDelivery field is set to '1', or  
 27 the ReliableDelivery field is set to '0' and the SequenceSpace field is set to 1.  
 28 Otherwise the sender shall omit this field. The sender shall set this field to  
 29 the SLP sequence number of the first octet in the SLP payload.

30 **Reserved** The sender shall include this field and set it to '000000' if the  
 31 ReliableDelivery field is set to '0' and SequenceSpace field is set to '0'.  
 32 Otherwise, the sender shall omit this field.

### 3.2.5.6 Message formats

#### 3.2.5.6.1 ResetRxRequest

The SLP receiver in the access terminal or the access network sends the ResetRxRequest message to reset its peer SLP transmitter.

Field	Length (bits)
MessageID	8
TransactionID	8
Reserved	7
SequenceSpace	1

MessageID The sender shall set this field to 0x00.

TransactionID The sender shall set this field according to 10.8.

Reserved The sender shall set this field to '0000000'. The receiver shall ignore this field.

SequenceSpace The sender shall set this flag to the value of the sequence space variable  $Q_{Rx}$ .

<b>Channels</b>	FTC    RTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Unicast	<b>Security</b>	Required

#### 3.2.5.6.2 ResetRxAck

The SLP transmitter in the access terminal or the access network sends the ResetRxAck message to complete the SLP reset procedure.

Field	Length (bits)
MessageID	8
TransactionID	8

MessageID The sender shall set this field to 0x01.

TransactionID The sender shall set this field to the TransactionID of the associated ResetRxRequest message.

<b>Channels</b>	FTC    RTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Unicast	<b>Security</b>	Required

### 3.2.5.6.3 ResetTxIndication

The SLP transmitter in the access terminal or the access network sends the ResetTxIndication message to reset its peer SLP receiver.

Field	Length (bits)
MessageID	8
TransactionID	8
Reserved	7
SequenceSpace	1

MessageID The sender shall set this field to 0x02.

TransactionID The sender shall set this field according to 10.8.

Reserved The sender shall set this field to '0000000'. The receiver shall ignore this field.

SequenceSpace The sender shall set this flag to the value of the sequence space variable  $Q_{Tx}$ .

<b>Channels</b>	FTC RTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Unicast	<b>Security</b>	Required

### 3.2.5.6.4 ResetTxAck

The SLP receiver in the access terminal or the access network sends the ResetTxAck message in response to the ResetTxIndication message.

Field	Length (bits)
MessageID	8
TransactionID	8

MessageID The sender shall set this field to 0x03.

TransactionID The sender shall set this field to the TransactionID of the associated ResetTxIndication message.

<b>Channels</b>	FTC RTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Unicast	<b>Security</b>	Required



### 3.2.5.6.5 ReceiverStatus

The access terminal and the access network send the ReceiverStatus message to acknowledge the receipt of one or more SLP octets or to request the retransmission of one or more SLP octets for the reliable delivery flow.

Field	Length (bits)
MessageID	8
Reserved0	7
SequenceSpace	1
ReportCount	8

ReportCount occurrences of the following 4 fields:

{	
Reserved1	4
FirstErasedOctet	SequenceLength
Reserved2	4
WindowLen	SequenceLength
}	
Reserved3	4
VR	SequenceLength

6	MessageID	The sender shall set this field to 0x04.
7	Reserved0	The sender shall set this field to '0000000'. The receiver shall ignore this field.
8		
9	SequenceSpace	The sender shall set this flag to the value of the sequence space variable $Q_{Rx}$ .
10	ReportCount	The sender shall set this field to the number of Report records included in this message. The sender shall include ReportCount occurrences of the following four fields with the message.
11		
12		
13	Reserved1	The sender shall set this field to '0000'. The receiver shall ignore this field.
14	FirstErasedOctet	The sender shall set this field to the sequence number of the first SLP octet erased in a sequence of erased octets.
15		
16	Reserved2	The sender shall set this field to '0000'. The receiver shall ignore this field.
17	WindowLen	The sender shall set this field to the length of the erased window in octets.
18	Reserved3	The sender shall set this field to '0000'. The receiver shall ignore this field.
19	VR	The sender shall set this field to $V(R)_{RD}$ .
20		

<b>Channels</b>	FTC    RTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Unicast	<b>Security</b>	Required

1

2 **3.2.5.7 Interface to other protocols**3 **3.2.5.7.1 Commands**

4 This protocol does not issue any commands.

5 **3.2.5.7.2 Indications**

6 This protocol registers to receive the following indications:

- 7     ■ *IdleState.ConnectionOpened*
- 8     ■ *ReverseTrafficChannelMAC.ReverseTrafficPacketsMissed* along with parameters
- 9         indicating the missing octets.
- 10    ■ *ForwardTrafficChannelMAC.ForwardTrafficPacketsMissed* along with parameters
- 11         indicating the missing octets.

12 **3.2.5.8 Protocol numeric constants**

13

Constant	Meaning	Value
SequenceLength	Length of the sequence number in the SLP header	20
N <sub>SLPRequestLevelRev</sub>	QoSFlow field is set to '00' for signaling requests in the R-REQCH for a reverse Link SLP packet	'00'
N <sub>SLPAttempt</sub>	Maximum Number of attempts for sending a reliable-delivery SLP packet	3
T <sub>SLPWaitAck</sub>	Retransmission timer for a reliable delivery SLP packet	200 ms
T <sub>SLPWaitNextSegment</sub>	Wait timer for the next segment of a best effort delivery SLP packet	10 seconds
T <sub>SLPResponse</sub>	Time period an SLP receiver has to respond to ResetRxRequest and ResetTxIndication messages.	1 second

14

15 **3.2.6 Configuration attributes**

16 No configuration attributes are defined for this protocol.

## 3.2.7 Session state information

The Session State Information record (see 10.10) consists of parameter records.

This transport defines the following parameter records in addition to the configuration attributes for this transport.

### 3.2.7.1 SignalingLinkState parameter

**Table 10 Format of the parameter record for the SignalingLinkState parameter**

Field	Length (bits)
ParameterType	8
Length	8
QTxState	1
QRxState	1
Reserved	6

ParameterType This field shall be set to 0x01 for this parameter record.

Length This field shall be set to the length of this parameter record in units of octets excluding the Length field.

QTxState This field shall be set to the value of the sequence state variable  $Q_{Tx}$ .

QRxState This field shall be set to the value of the sequence state variable  $Q_{Rx}$ .

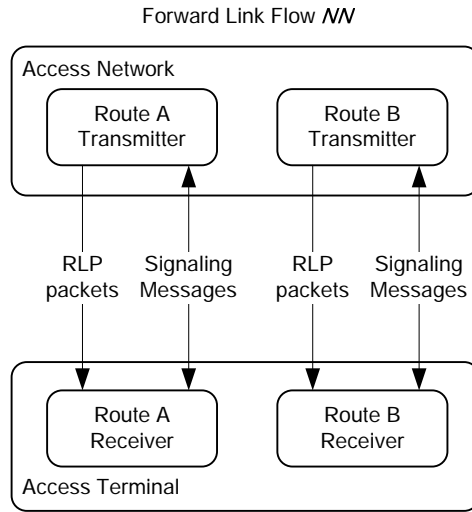
Reserved This field shall be set to '000000'. The receiver shall ignore this field.

## 3.3 Default Data Transport

### 3.3.1 Introduction

#### 3.3.1.1 General overview

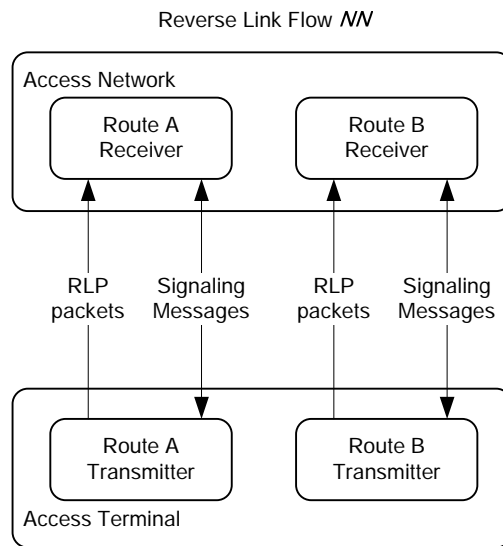
The Default Data Transport provides multiple packet streams that can be used to carry packets between the access terminal and the access network. Each packet stream is called a Link Flow. Each Link Flow provides two routes for transmission and reception of higher layer payloads. These routes are named Route A and Route B and can be carried using a single receiver-transmitter pair. Each route is associated with a transmitter-receiver pair. Figure 26 shows the association between a forward Link Flow and the transmitters and receivers for its two routes. Figure 27 shows the reference architecture for a reverse Link Flow.



1

2

**Figure 26 Reference architecture for a forward link flow**

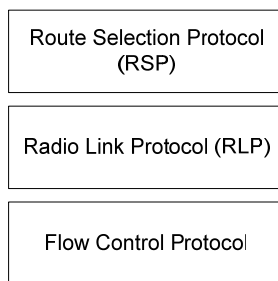


3

4

**Figure 27 Reference architecture for a reverse link flow**

1 The relationship between the Default Data Transport protocols is illustrated in Figure 28.

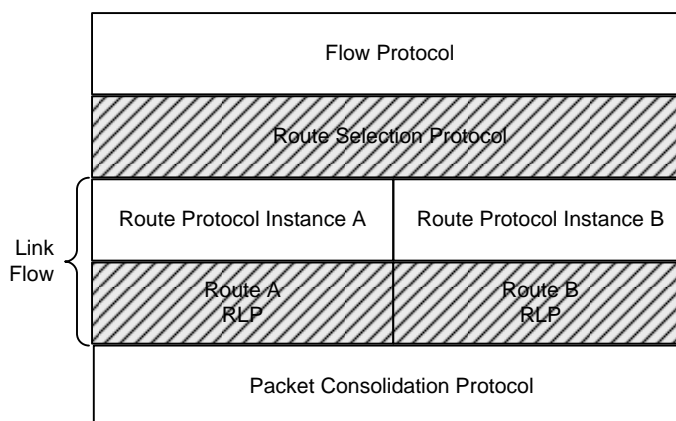


2

3

**Figure 28 Default data transport protocols**

4 Figure 29 illustrates the relationship for each Link Flow between the Default Data Transport and the  
 5 higher layer protocols supported by the Default Data Transport. The Flow Protocol and the Route  
 6 Protocol are referred to as higher layer protocols. The protocols defined in the Default Data Transport  
 7 are shown shaded. The Route Selection Protocol routes Flow Protocol PDUs to either instance A or  
 8 instance B of the Route Protocol. Instance A of the Route Protocol is bound to Route A of the Link  
 9 Flow. Instance B of the Route Protocol is bound to Route B of the Link Flow.



10

11 **Figure 29 Relationship between default data transport and higher layer protocols**

12 The Default Data Transport provides:

- 13 ■ The Route Selection Protocol, which routes Flow Protocol PDUs over either Route A or  
 14 Route B of a Link Flow.
- 15 ■ The Radio Link Protocol (RLP), which provides retransmission (if needed) and duplicate  
 16 detection of higher layer packets transmitted on each route.
- 17 ■ The Flow Control Protocol, which provides flow control for the Default Data Transport.
- 18 ■ The ability to negotiate protocol parameters for all protocols in the Default Data  
 19 Transport.

### 3.3.1.2 Public data

#### 3.3.1.2.1 Static public data

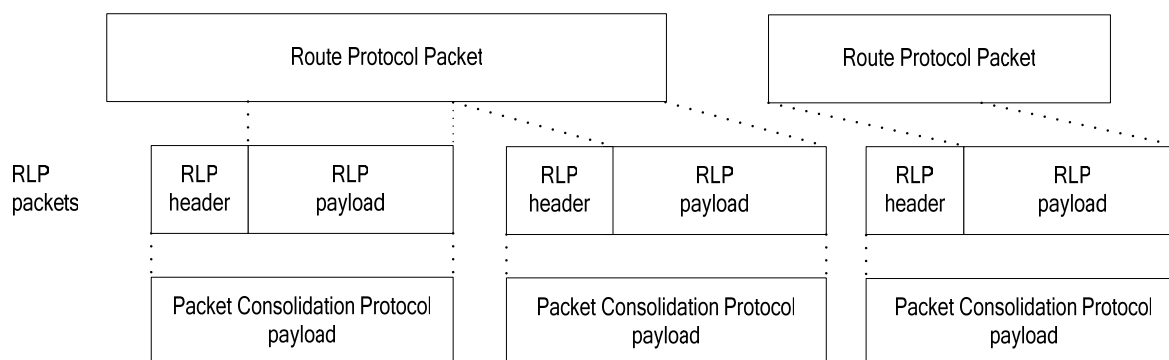
This transport does not define any static public data.

#### 3.3.1.2.2 Dynamic public data

- Subtype for this transport
- Flow $NN$ RequestLevelRev, where  $NN$  is the two-digit hexadecimal Link Flow number in the range 0x00 to  $N_{\text{LinkFlowMax}}-1$  inclusive, where hexadecimal digits A through F are specified in upper case letters.

### 3.3.1.3 Data encapsulation

Figure 30 illustrates the relationship between packets from the Route Protocol, RLP packets, and Packet Consolidation Protocol payload.



**Figure 30 Default data transport encapsulation**

The Default Data Transport uses the Signaling Transport to transmit and receive messages.

## 3.3.2 Transport initialization and swap procedures

### 3.3.2.1 Transport initialization

Upon creation, the instance of the Data Transport (i.e., corresponding to the Transport defined in the Packet Consolidation Protocol to which this transport is bound) in the access terminal and access network shall perform the following:

- The value of the attributes for this transport instance shall be set to the default values specified for each attribute.
- The Flow Control Protocol associated with the instance of the Data Transport at the access terminal and access network shall enter the Open State<sup>10</sup>.

<sup>10</sup> Forward and reverse link Reservations 0xff initialized in the Open state so that data can be sent without having to perform a state transition.

- 1           ■ Forward and reverse link Reservations with ReservationLabel 0xff shall enter the Open
- 2           state. All other Reservations shall enter the Close state.
- 3           ■ The Route Selection Protocol shall enter the A Open B Draining state.

### 4   **3.3.2.2 Transport swap**

5   Upon swap, the instance of the Data Transport (i.e., corresponding to the Transport defined in the  
6   Packet Consolidation Protocol to which this transport is bound) in the access terminal and access  
7   network shall perform the following:

- 8           ■ The Route Selection Protocol shall enter the A Open B Draining state.

## 9   **3.3.3 Route Selection Protocol**

### 10   **3.3.3.1 Overview**

11   The Route Selection Protocol provides means to select either instance A or instance B of the Route  
12   Protocol. The Route Selection Protocol routes Flow Protocol PDUs to the selected instance of the  
13   Route Protocol. Instance A of the Route Protocol is bound to Route A of the Link Flow. Instance B of  
14   the Route Protocol is bound to Route B of the Link Flow. The Route Selection Protocol is a protocol  
15   associated with the Default Data Transport.

### 16   **3.3.3.2 Primitives**

#### 17   **3.3.3.2.1 Commands**

18   This protocol does not define any commands.

#### 19   **3.3.3.2.2 Return indications**

20   This protocol does not return any indications.

### 21   **3.3.3.3 Protocol data unit**

22   The Route Selection Protocol routes Flow Protocol PDUs to the Route Protocol without modifying  
23   them. Hence, the transmission unit of this protocol is the same as a Flow Protocol PDU. The Flow  
24   Protocol for a forward Link Flow *NN* is identified by the ProtocolID field of the  
25   Flow*NN*FlowProtocolParametersFwd attribute. The Flow Protocol for a reverse Link Flow *NN* is  
26   identified by the ProtocolID field of the Flow*NN*FlowProtocolParametersRev attribute.

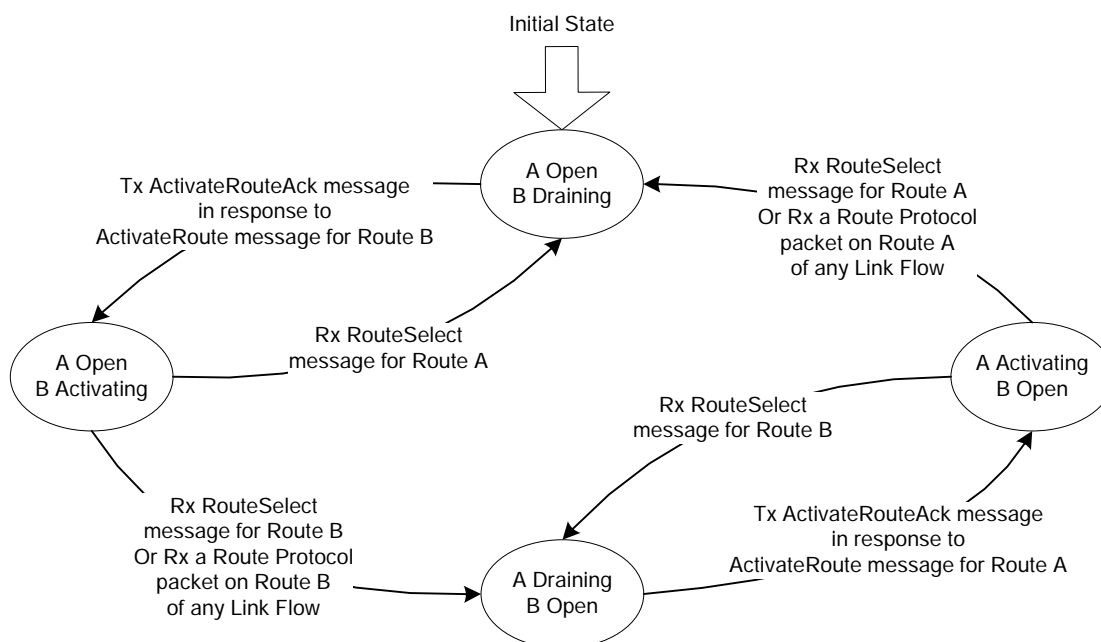
### 27   **3.3.3.4 Procedures**

#### 28   **3.3.3.4.1 General requirements**

29   If the Flow*NN*SimultaneousDeliveryOnBothRoutesFwd attribute of forward Link Flow *NN* is  
30   0x0000, then forward Link Flow *NN* delivers Flow Protocol PDUs in order. If the  
31   Flow*NN*SimultaneousDeliveryOnBothRoutesFwd attribute of forward Link Flow *NN* is 0x0001, then  
32   forward Link Flow *NN* may deliver Flow Protocol PDUs out of order.

### 3.3.3.4.2 Access terminal requirements

The Route Selection Protocol associated with an activated Link Flow can be in one of four states: A Open B Draining, A Open B Activating, A Draining B Open, or A Activating B Open. The Route Selection Protocol instance associated with all activated Link Flows shall be in the same state at any time. When a Link Flow is activated, the Route Selection Protocol shall enter the state that the Route Selection Protocols of other activated Link Flows are in. If no other Link Flows are activated when a Link Flow is activated, then the Route Selection Protocol shall enter the A Open B Draining state. Figure 31 shows the state diagram for the Route Selection Protocol at the access terminal.



**Figure 31 Route selection protocol state diagram (access terminal)**

#### 3.3.3.4.2.1 A Open B Draining state

##### 3.3.3.4.2.1.1 State transitions

Upon receiving an ActivateRoute message requesting to activate Route B, the access terminal shall perform the following:

- The Route Selection Protocol shall issue a *RadioLinkProtocol.InitializeRoute* command with Route B as the argument.
- The access terminal shall initialize the Route Protocol bound to Route B.
- After the Radio Link Protocol and the Route Protocol are initialized, the access terminal shall send an ActivateRouteAck message, and shall transition to the A Open B Activating state.

Upon receiving a RouteSelect message for Route A, the access terminal shall respond with a RouteSelectAck message.



### 3.3.3.4.2.1.2 Transmitter requirements

The access terminal shall route Flow Protocol PDUs to Route A. The access terminal shall not route Flow Protocol PDUs to Route B.

### 3.3.3.4.2.1.3 Receiver requirements

The access terminal shall pass Flow Protocol PDUs received on Route A to the Flow Protocol.

If the FlowNNSimultaneousDeliveryOnBothRoutesFwd attribute for Link Flow *NN* is 0x0001, then the access terminal shall pass Flow Protocol PDUs received on Route B of the Link Flow to the Flow Protocol if the access terminal has not received an ActivateRoute message requesting to activate Route B since the last time it entered this state.

If the FlowNNSimultaneousDeliveryOnBothRoutesFwd attribute for Link Flow *NN* is 0x0000, then the access terminal shall pass Flow Protocol PDUs received on Route B of the Link Flow to the Flow Protocol if the access terminal has not passed Flow Protocol PDUs received on Route A of the Link Flow to the Flow Protocol since the last time the access terminal entered this state and if the access terminal has not received an ActivateRoute message requesting to activate Route B since the last time it entered this state. If the FlowNNSimultaneousDeliveryOnBothRoutesFwd attribute for Link Flow *NN* is 0x0000, then the access terminal shall discard Flow Protocol PDUs received on Route B of the Link Flow if the access terminal has passed Flow Protocol PDUs received on Route A of the Link Flow to the Flow Protocol since the access terminal entered this state.

### 3.3.3.4.2.2 A Open B Activating state

#### 3.3.3.4.2.2.1 State transitions

Upon receiving a RouteSelect message requesting to select Route B, the access terminal shall respond with a RouteSelectAck message, and shall transition to the A Draining B Open state.

Upon receiving Flow Protocol PDU on Route B of any Link Flow, the access terminal shall store the Flow Protocol PDU received from Route B for processing in the A Draining B Open state and shall transition to the A Draining B Open state.

Upon receiving a RouteSelect message requesting to select Route A, the access terminal shall respond with a RouteSelectAck message, and shall transition to the A Open B Draining state.

#### 3.3.3.4.2.2.2 Transmitter requirements

The access terminal shall route Flow Protocol PDUs to Route A. The access terminal shall not route Flow Protocol PDUs to Route B.

#### 3.3.3.4.2.2.3 Receiver requirements

The access terminal shall pass Flow Protocol PDUs received on Route A to the Flow Protocol.

### 1 3.3.3.4.2.3 A Draining B Open state

#### 2 3.3.3.4.2.3.1 State transitions

3 Upon receiving an ActivateRoute message requesting to activate Route A, the access terminal shall  
4 perform the following:

- 5 ■ The Route Selection Protocol shall issue a *RadioLinkProtocol.InitializeRoute* command  
6 with Route A as the argument.
- 7 ■ The access terminal shall initialize the Route Protocol bound to Route A.
- 8 ■ After the Radio Link Protocol and the Route Protocol are initialized, the access terminal  
9 shall send respond with an ActivateRouteAck message, and shall transition to the A  
10 Activating B Open state.

11 Upon receiving a RouteSelect message for Route B, the access terminal shall respond with a  
12 RouteSelectAck message.

#### 13 3.3.3.4.2.3.2 Transmitter requirements

14 The access terminal shall route Flow Protocol PDUs to Route B. The access terminal shall not route  
15 Flow Protocol PDUs to Route A.

#### 16 3.3.3.4.2.3.3 Receiver requirements

17 The access terminal shall pass Flow Protocol PDUs received on Route B to the Flow Protocol.

18 If the *FlowNNSimultaneousDeliveryOnBothRoutesFwd* attribute for Link Flow *NN* is 0x0001, then  
19 the access terminal shall pass Flow Protocol PDUs received on Route A of the Link Flow to the Flow  
20 Protocol if the access terminal has not received an ActivateRoute message requesting to activate  
21 Route A since the last time it entered this state.

22 If the *FlowNNSimultaneousDeliveryOnBothRoutesFwd* attribute for Link Flow *NN* is 0x0000, then  
23 the access terminal shall pass Flow Protocol PDUs received on Route A of the Link Flow to the Flow  
24 Protocol if the access terminal has not passed Flow Protocol PDUs received on Route B of the Link  
25 Flow to the Flow Protocol since the access terminal entered this state and if the access terminal has  
26 not received an ActivateRoute message requesting to activate Route A since the last time it entered  
27 this state. If the *FlowNNSimultaneousDeliveryOnBothRoutesFwd* attribute for Link Flow *NN* is  
28 0x0000, then the access terminal shall discard Flow Protocol PDUs received on Route A of the Link  
29 Flow if the access terminal has passed Flow Protocol PDUs received on Route B of the Link Flow to  
30 the Flow Protocol since the last time the access terminal entered this state.

### 31 3.3.3.4.2.4 A Activating B Open state

#### 32 3.3.3.4.2.4.1 State transitions

33 Upon receiving a RouteSelect message requesting to select Route A, the access terminal shall respond  
34 with a RouteSelectAck message, and shall transition to the A Open B Draining state.

35 Upon receiving Flow Protocol PDU on Route A of any Link Flow, the access terminal shall store the  
36 Flow Protocol PDU received on Route A for processing in the A Open B Draining state and shall  
37 transition to the A Open B Draining state.

1 Upon receiving a RouteSelect message requesting to select Route B, the access terminal shall respond  
2 with a RouteSelectAck message, and shall transition to the A Draining B Open state.

### 3 3.3.3.4.2.4.2 Transmitter requirements

4 The access terminal shall route Flow Protocol PDUs to Route B. The access terminal shall not route  
5 Flow Protocol PDUs to Route A.

### 6 3.3.3.4.2.4.3 Receiver requirements

7 The access terminal shall pass Flow Protocol PDUs received on Route B to the Flow Protocol.

### 8 3.3.3.4.3 Access network requirements

9 Upon sending an ActivateRoute message requesting to activate Route A, the access network shall  
10 issue a *RadioLinkProtocol.InitializeRoute* command with Route A as the argument and initialize the  
11 Route Protocol bound to Route A.

12 Upon sending an ActivateRoute message requesting to activate Route B, the access network shall  
13 issue a *RadioLinkProtocol.InitializeRoute* command with Route B as the argument and initialize the  
14 Route Protocol bound to Route B.

### 15 3.3.3.5 Message formats

#### 16 3.3.3.5.1 RouteSelect

17 The access network sends this message to transition the access terminal to the A Open B Draining or  
18 the A Draining B Open state.

Field	Length (bits)
MessageID	8
TransactionID	8
Route	1
Reserved	7

20 MessageID The access network shall set this field to 0x00.

21 TransactionID The access network shall set this field according to 10.8.

22 Route The access network shall set this field to '0' to transition the access terminal  
23 to the A Open B Draining state. The access network shall set this field to '1'  
24 to transition the access terminal to the A Draining B Open state.

25 Reserved The access network shall set this field to '0000000'. The access terminal  
26 shall ignore this field.

<b>Channels</b>	FTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Unicast	<b>Security</b>	Required

### 3.3.3.5.2 RouteSelectAck

The access terminal sends this message to acknowledge the receipt of a RouteSelect message.

Field	Length (bits)
MessageID	8
TransactionID	8

**MessageID** The access terminal shall set this field to 0x01.

**TransactionID** The access terminal shall set this field to the TransactionID field to the RouteSelect message whose receipt is being acknowledged by this message.

<b>Channels</b>	RTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Unicast	<b>Security</b>	Required

### 3.3.3.5.3 ActivateRoute

The access network sends this message to transition the access terminal to the A Activating B Open state or the A Open B Activating state.

Field	Length (bits)
MessageID	8
TransactionID	8
Route	1
Reserved	7

**MessageID** The access network shall set this field to 0x02.

**TransactionID** The access network shall set this field according to 10.8.

**Route** The access network shall set this field to '0' to transition the access terminal to the A Activating B Open state. The access network shall set this field to '1' to transition the access terminal to the B Activating A Open state.

**Reserved** The access network shall set this field to '0000000'. The access terminal shall ignore this field.

<b>Channels</b>	FTC	<b>SLP</b>	Reliable
<b>Addressing</b>	Unicast	<b>Security</b>	Required

### 3.3.3.5.4 ActivateRouteAck

The access terminal sends this message to acknowledge the receipt of an ActivateRoute message.

Field	Length (bits)
MessageID	8
TransactionID	8

MessageID The access terminal shall set this field to 0x03.

TransactionID The access terminal shall set this field to the TransactionID field to the ActivateRoute message whose receipt is being acknowledged by this message.

<b>Channels</b>	RTC	<b>SLP</b>	Reliable
<b>Addressing</b>	Unicast	<b>Security</b>	Required

### 3.3.3.6 Interface to other protocols

#### 3.3.3.6.1 Commands

This protocol issues the following commands:

- *RadioLink.InitializeRoute* with the argument indicating which Route is to be initialized.

#### 3.3.3.6.2 Indications

This protocol does not register to receive any indications.

#### 3.3.3.7 Protocol numeric constants

This protocol does not define any protocol numeric constants.

### 3.3.4 Radio Link Protocol

#### 3.3.4.1 Overview

The Radio Link Protocol (RLP) provides one or more packet streams with an acceptably low erasure rate for efficient operation of higher layer protocols (e.g., TCP). When used as part of the Default Data Transport, the protocol carries one or more packet streams from the higher layer. RLP is a protocol associated with the Default Data Transport.

#### 3.3.4.2 Primitives

##### 3.3.4.2.1 Commands

This protocol defines the following commands:

- *InitializeRoute* with argument indicating which Route is to be initialized.

### 3.3.4.2 Return indications

This protocol does not return any indications.

### 3.3.4.3 Protocol data unit

The transmission unit of this protocol is an RLP packet.

### 3.3.4.4 Procedures

A forward Link Flow is defined to be activated if the Flow $NN$ ActivatedFwd attribute is set to 0x0001, where  $NN$  is the hexadecimal Link Flow number in the range 0x00 to  $N_{\text{LinkFlowMax}}-1$  inclusive.

A reverse Link Flow is defined to be activated if the Flow $NN$ ActivatedRev attribute is set to 0x0001.

A Link Flow is defined to be deactivated if it is not activated.

Each Route of the Link Flow receives packets for transmission from the corresponding instance of the Route Protocol and forms an RLP packet by prepending the RLP packet header defined in 3.3.4.4.3 with a number of received contiguous octets.

The Route Protocol for a forward Link Flow  $NN$  is identified by the ProtocolID field of the Flow $NN$ RouteProtocolParametersFwd attribute. The Route Protocol for a reverse Link Flow  $NN$  is identified by the ProtocolID field of the Flow $NN$ RouteProtocolParametersRev attribute.

If the Route Protocol is NULL<sup>11</sup>, then the transmitter shall set Route Protocol packets to Flow Protocol packets routed along the Route. If the Route Protocol is NULL, then the receiver shall set Flow Protocol packets to Route Protocol packets received on the Route.

If the Flow $NN$ OutOfOrderDeliveryToRouteProtocolFwd attribute of forward Link Flow  $NN$  is 0x0000, then each Route of forward Link Flow  $NN$  delivers packets of the corresponding instance of the Route Protocol in order. If the Flow $NN$ OutOfOrderDeliveryToRouteProtocolFwd attribute of forward Link Flow  $NN$  is 0x0001, then each Route of forward Link Flow  $NN$  may deliver packets of the corresponding instance of the Route Protocol out of order.

If the Flow $NN$ OutOfOrderDeliveryToRouteProtocolRev attribute of reverse Link Flow  $NN$  is 0x0000, then each Route of reverse Link Flow  $NN$  delivers packets of the corresponding instance of the Route Protocol in order. If the Flow $NN$ OutOfOrderDeliveryToRouteProtocolRev attribute of reverse Link Flow  $NN$  is 0x0001, then each Route of reverse Link Flow  $NN$  may deliver packets of the corresponding instance of the Route Protocol out of order.

The policy RLP follows in determining the number of octets to send in an RLP packet is beyond the scope of this specification. It is subject to the following requirements:

- The size of an RLP packet shall not exceed the maximum payload length that can be carried by a Packet Consolidation Protocol packet given the target channel and current transmission rate on that channel.
- An RLP packet shall contain octets from no more than one Route Protocol packet.

---

<sup>11</sup> Route Protocol being NULL means that a Route Protocol has not been negotiated.

1 RLP makes use of the ResetRxRequest, ResetRxAck, ResetTxIndication, ResetTxAck, and  
2 ReceiverStatus in-band messages to perform control related operations.

3 The access terminal shall not initiate negotiation of the ANSupportedQoSProfiles attribute. The  
4 access network shall not initiate negotiation of the ATSupportedQoSProfiles attribute.

5 The access network shall not initiate modification of the ReservationKKQoSListFwd or the  
6 ReservationKKQoSListRev attributes. If the access network receives an AttributeUpdateRequest  
7 message requesting to set the ReservationKKQoSListFwd or the ReservationKKQoSListRev attribute  
8 to its default value, then the access network shall respond with an AttributeUpdateAccept message.

9 The access terminal shall not initiate modification of the ReservationKKQoSUsedFwd or the  
10 ReservationKKQoSUsedRev attributes.

11 The access terminal uses the AttributeUpdateRequest message with the ReservationKKQoSListFwd  
12 attributes to add, modify, or remove the QoS for forward Reservation *KK*. The access terminal  
13 requests one or more QoSAttributeSets in order of preferences for forward Reservation *KK*. The  
14 access terminal uses the AttributeUpdateRequest message with the ReservationKKQoSListRev  
15 attributes to add, modify, or remove the QoS for reverse Reservation *KK*. The access terminal  
16 requests one or more QoSAttributeSets in order of preferences for reverse Reservation *KK*. Each  
17 QoSAttributeSet contains a group of detailed QoS parameters.

18 The access network stores the requested QoSAttributeSets in the AttributeUpdateRequest message for  
19 the ReservationKKQoSListFwd attribute and grants one. The access network informs the access  
20 terminal which QoSAttributeSet it granted by the QoSAttributeSet\_ID of the  
21 ReservationKKQoSUsedFwd attribute in an AttributeUpdateRequest message.

22 The access network stores the requested QoSAttributeSets in the AttributeUpdateRequest message for  
23 the ReservationKKQoSListRev attribute and grants one. The access network informs the access  
24 terminal which QoSAttributeSet it granted by the QoSAttributeSet\_ID of the  
25 ReservationKKQoSUsedRev attribute in an AttributeUpdateRequest message.

26 If the access terminal sends a new AttributeUpdateRequest for the ReservationKKQoSListFwd  
27 attribute, the new requested ReservationKKQoSListFwd attribute shall replace the previous requested  
28 ReservationKKQoSListFwd attribute for forward Reservation *KK*. If the access terminal sends a new  
29 AttributeUpdateRequest for the ReservationKKQoSListRev attribute, the new requested  
30 ReservationKKQoSListRev attribute shall replace the previous requested ReservationKKQoSListRev  
31 attribute for reverse Reservation *KK*. In any new requested ReservationKKQoSListFwd or  
32 ReservationKKQoSListRev attribute for Reservation *KK*, the access terminal shall not re-use values  
33 for QoSAttributeSet\_ID from the previous two ReservationKKQoSListFwd or  
34 ReservationKKQoSListRev attributes respectively, negotiated for Reservation *KK*.

35 The access network may change the granted QoSAttributeSet\_ID to another QoSAttributeSet\_ID in  
36 the group of QoSAttributeSet\_IDs most recently requested by the access terminal. The access  
37 network shall inform the access terminal of the new granted QoSAttributeSet\_ID.

38 The access network shall not initiate modification of the ReservationKKPacketFilterFwd or the  
39 ReservationKKPacketFilterRev attributes. If the access network receives an AttributeUpdateRequest  
40 message requesting to set the ReservationKKPacketFilterFwd or the ReservationKKPacketFilterRev  
41 attributes to their default values, then the access network shall respond with an  
42 AttributeUpdateAccept message. If the FilterSpecType is set to 0x02 according to Table 13, then the

1 packet filter for Reservation*KK*PacketFilterFwd or Reservation*KK*PacketFilterRev shall match all  
2 packets that do not match a packet filter with a lower value FilterPrecedence field.

3 The access network may send a FlowQoSdetect message to inform the access terminal that it should  
4 add a new Reservation or modify the Reservation*KK*QoSListFwd or Reservation*KK*QoSListRev  
5 attribute for an existing Reservation *KK*. If the access terminal determines that the FlowQoSdetect  
6 message corresponds to a Reservation that it has not already added or modified, the access terminal  
7 should send an AttributeUpdateRequest message for the Reservation*KK*QoSListFwd or  
8 Reservation*KK*QoSListRev attribute in response to the FlowQoSdetect message.

9 When forward Link Flow *NN* is activated, the access network and the access terminal shall not update  
10 the following attributes:

- 11 ■ Flow*NN*FlowProtocolParametersFwd
- 12 ■ Flow*NN*RouteProtocolParametersFwd
- 13 ■ Flow*NN*SequenceLengthFwd
- 14 ■ Flow*NN*DataUnitFwd
- 15 ■ Flow*NN*SimultaneousDeliveryOnBothRoutesFwd
- 16 ■ Flow*NN*OutOfOrderDeliveryToRouteProtocolFwd

17 When reverse Link Flow *NN* is activated, the access network and the access terminal shall not update  
18 the following attributes:

- 19 ■ Flow*NN*FlowProtocolParametersRev
- 20 ■ Flow*NN*RouteProtocolParametersRev
- 21 ■ Flow*NN*SequenceLengthRev
- 22 ■ Flow*NN*DataUnitRev
- 23 ■ Flow*NN*OutOfOrderDeliveryToRouteProtocolRev

24 The ProtocolID field of the Flow*NN*FlowProtocolParametersFwd attribute shall be set to a value that  
25 is supported by the access terminal as indicated in the ATSupportedFlowProtocolParameters*PP*  
26 attribute. The ProtocolID field of the Flow*NN*FlowProtocolParametersRev attribute shall be set to a  
27 value that is supported by the access terminal as indicated in the  
28 ATSupportedFlowProtocolParameters*PP* attribute. The ProtocolID field of the  
29 Flow*NN*RouteProtocolParametersFwd attribute shall be set to a value that is supported by the access  
30 terminal as indicated in the ATSupportedRouteProtocolParameters*PP* attribute. The ProtocolID field  
31 of the Flow*NN*RouteProtocolParametersRev attribute shall be set to a value that is supported by the  
32 access terminal as indicated in the ATSupportedRouteProtocolParameters*PP* attribute.

33 The fields of the ProtocolParameters record of the Flow*NN*FlowProtocolParametersFwd attribute  
34 shall be set to values that are in accordance with those supported by the AT as indicated in the  
35 SupportedProtocolsParametersValues record of the ATSupportedFlowProtocolParameters*PP*  
36 attribute. The fields of the ProtocolParameters record of the Flow*NN*FlowProtocolParametersRev  
37 attribute shall be set to values that are in accordance with those supported by the AT as indicated in  
38 the SupportedProtocolsParametersValues record of the ATSupportedFlowProtocolParameters*PP*  
39 attribute.



1 The fields of the ProtocolParameters record of the FlowNNRouteProtocolParametersFwd attribute  
 2 shall be set to values that are in accordance with those supported by the AT as indicated in the  
 3 SupportedProtocolsParametersValues record of the ATSupportedRouteProtocolParametersPP  
 4 attribute. The fields of the ProtocolParameters record of the FlowNNRouteProtocolParametersRev  
 5 attribute shall be set to values that are in accordance with those supported by the AT as indicated in  
 6 the SupportedProtocolsParametersValues record of the ATSupportedRouteProtocolParametersPP  
 7 attribute.

8 If the FlowNNDataUnitFwd attribute of forward Link Flow NN is 0x0000, then the data unit for the  
 9 Link Flow shall be octets. Otherwise the data unit for the Link Flow shall be RLP packet payloads. If  
 10 the FlowNNDataUnitRev attribute of reverse Link Flow NN is 0x0000, then the data unit for the Link  
 11 Flow shall be octets. Otherwise the data unit for the Link Flow shall be RLP packet payloads.

#### 12 **3.3.4.4.1 Initialization and reset**

13 The RLP initialization procedure initializes the RLP variables and data structures in one end of the  
 14 link. The RLP reset procedure guarantees that RLP state variables on both sides are synchronized.  
 15 The reset procedure includes initialization.

16 If the protocol receives an *IdleState.ConnectionOpened* indication, then the access terminal and the  
 17 access network shall perform the initialization procedures defined in 3.3.4.4.1.1.1 and 3.3.4.4.1.1.2  
 18 for both routes of all activated Link Flows.

19 The access network shall perform the initialization procedure defined in 3.3.4.4.1.1.1 for both routes  
 20 of forward Link Flow NN when forward Link Flow NN is activated. The access terminal shall  
 21 perform the initialization procedure defined in 3.3.4.4.1.1.2 for both routes of forward Link Flow NN  
 22 when forward Link Flow NN is activated.

23 The access terminal shall perform the initialization procedure defined in 3.3.4.4.1.1.1 for both routes  
 24 of reverse Link Flow NN when reverse Link Flow NN is activated. The access network shall perform  
 25 the initialization procedure defined in 3.3.4.4.1.1.2 for both routes of reverse Link Flow NN when  
 26 reverse Link Flow NN is activated.

27 Upon receiving an *InitializeRoute* command, the access terminal shall perform the initialization  
 28 procedures defined in 3.3.4.4.1.1 for the specified Route for all activated Link Flows. Upon receiving  
 29 an *InitializeRoute* command, the access network shall perform the initialization procedures defined in  
 30 3.3.4.4.1.1 for the specified Route of all activated Link Flows.

31 Each Link Flow provides sequence spaces variables  $Q_{NN,Tx}$  and  $Q_{NN,Rx}$  at the transmitter and receiver  
 32 respectively. The transmitter toggles the sequence space variable  $Q_{NN,Tx}$  between '0' and '1' to  
 33 indicate a reset. The receiver sequence space variable  $Q_{NN,Rx}$  tracks the value of  $Q_{NN,Tx}$  and detects  
 34 when the transmitter has performed a reset.

35 The RLP shall set the sequence space variables  $Q_{NN,Tx}$  and  $Q_{NN,Rx}$  to '0' after reception of an  
 36 *IdleState.ConnectionOpened* indication. The RLP shall toggle the value of the sequence space  
 37 variables  $Q_{NN,Tx}$  and  $Q_{NN,Rx}$  between '0' and '1' for every subsequent reset.

### 3.3.4.4.1.1 Initialization procedure

#### 3.3.4.4.1.1.1 Initialization procedure for the RLP transmitter

When RLP transmitter performs the initialization procedure it shall:

- Reset the send state variable  $V(S)_{NN,P}$  to zero, where  $NN$  indicates the Link Flow, and  $P$  indicates the Route which is being initialized.
- Clear the retransmission queue.

#### 3.3.4.4.1.1.2 Initialization procedure for the RLP receiver

When RLP receiver performs the initialization procedure it shall:

- Reset the receive state variables  $V(R)_{NN,P}$  and  $V(N)_{NN,P}$  to zero.
- Clear the resequencing buffer.

### 3.3.4.4.1.2 Reset procedure

#### 3.3.4.4.1.2.1 Reset procedure for the initiating side when it is an RLP transmitter

If the side initiating a reset procedure is an RLP transmitter for the Route of the Link Flow (or all Link Flows) being reset, then it shall:

- Perform the RLP transmitter initialization procedure defined in 3.3.4.4.1.1.1 for the Route of the Link Flow being reset.
- Toggle the value of the sequence space variable  $Q_{NN,Tx}$  for the Route of the Link Flow being reset.
- Send a ResetTxIndication message.

The RLP transmitter shall not reset again until it receives a ReceiverStatus message for the Route of the Link Flow being reset with a SequenceSpace field equal to the new value of the sequence space variable  $Q_{NN,Tx}$  from the RLP receiver, or a ResetTxAck message with a TransactionID field equal to the TransactionID sent in the ResetTxIndication message.

The RLP transmitter shall ignore any received ResetRxRequest messages for the Route of the Link Flow being reset until it receives a ReceiverStatus message with a SequenceSpace field with the new value of the sequence space variable  $Q_{NN,Tx}$  from the RLP receiver, or a ResetTxAck message with a TransactionID field equal to the TransactionID sent in the ResetTxIndication message.

The RLP transmitter may determine that the ResetTxIndication was lost if it does not receive a ReceiverStatus message for the Route of the Link Flow being reset with a SequenceSpace field equal to the new value of the sequence space variable  $Q_{NN,Tx}$  from the RLP receiver, or a ResetTxAck message, within an implementation-dependent time interval based on  $T_{RLPResponse}$  and an estimate of the round-trip delay. If the RLP transmitter determines that the ResetTxIndication was lost, then the RLP transmitter shall send a new ResetTxIndication message.

### 3.3.4.4.1.2.2 Reset procedure for initiating side when it is an RLP receiver

If the side initiating a reset procedure is an RLP receiver for the Route of the Link Flow being reset, then it shall enter the RLP Reset State. Upon entering the RLP Reset state, RLP shall:

- Perform the RLP receiver initialization procedure defined in 3.3.4.4.1.1.2 for the Route of the Link Flow being reset.
- Toggle the value of the sequence space variable  $Q_{NN,Rx}$  for the Route of the Link Flow being reset.
- Send a ResetRxRequest message.
- Ignore all RLP data units received for the Route of the Link Flow being reset while in the RLP Reset state with SequenceSpace field not equal to the sequence space variable  $Q_{NN,Rx}$ .
- If a ResetRxAck message is received for the Route of the Link Flow being reset with a TransactionID field equal to the TransactionID sent in the ResetRxRequest message, RLP shall leave the RLP reset state.
- If an RLP data unit is received with a SequenceSpace field equal to the sequence space variable  $Q_{NN,Rx}$  for the Route of the Link Flow being reset, RLP shall leave the RLP Reset state.

If a ResetRxAck is received for a Route while the Route is not in the RLP Reset state, the message shall be ignored.

The RLP receiver may determine that the ResetRxRequest was lost if it does not leave the RLP Reset state within an implementation-dependent time interval based on  $T_{RLPResponse}$  and an estimate of the round-trip delay. If the RLP receiver determines that the ResetRxRequest was lost, then the RLP receiver shall send a new ResetRxRequest message.

### 3.3.4.4.1.2.3 Reset procedure for the responding side when it is an RLP receiver

If the side responding to a reset procedure is an RLP receiver for the Route of the Link Flow being reset, then upon receiving a ResetTxIndication message, RLP shall perform the following procedures:

- If the sequence space variable  $Q_{NN,Rx}$  for the Route of the Link Flow being reset is not equal to the value of the SequenceSpace field in the ResetTxIndication message, then RLP shall:
  - Perform the RLP receiver initialization procedure defined in 3.3.4.4.1.1.2 for the Route of the Link Flow being reset.
  - Toggle the value of the sequence space variable  $Q_{NN,Rx}$  for the Route of the Link Flow being reset.
- Respond with a ResetTxAck message.

1 Upon receiving an RLP data unit for the Route of the Link Flow with a SequenceSpace field not  
2 equal to the sequence space variable  $Q_{NN,Rx}$ , RLP shall:

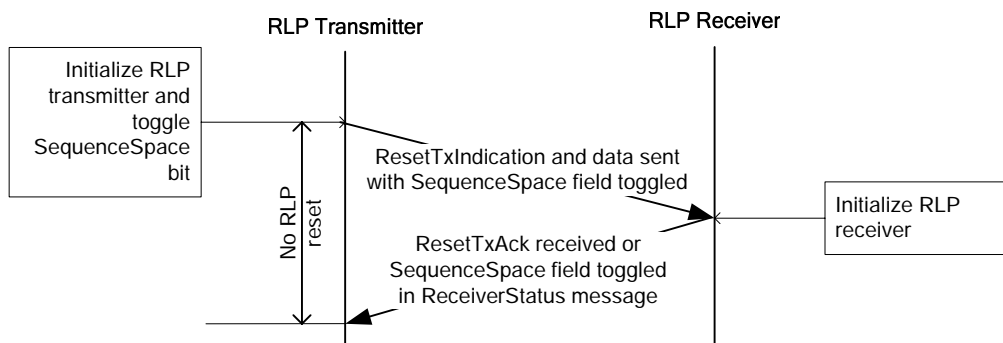
- 3 ■ Perform the RLP receiver initialization procedure defined in 3.3.4.4.1.1.2 for the Route of  
4 the Link Flow being reset.
- 5 ■ Toggle the value of the sequence space variable  $Q_{NN,Rx}$  for the Route of the Link Flow  
6 being reset.

#### 7 3.3.4.4.1.2.4 Reset procedure for the responding side when it is an RLP transmitter

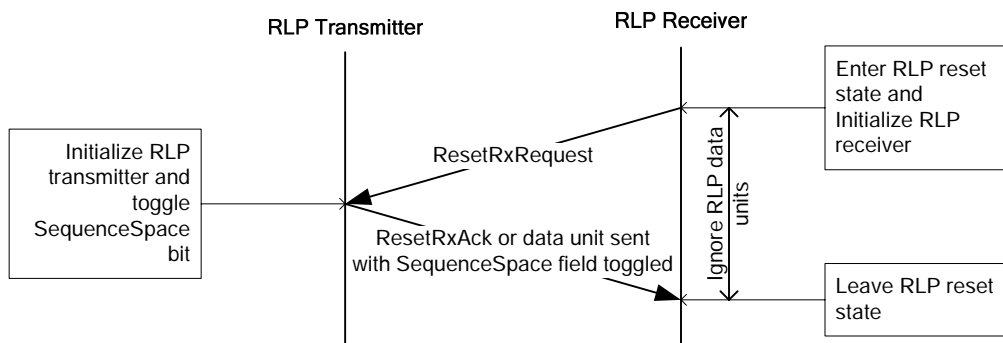
8 If the side responding to a reset procedure is an RLP transmitter for the Route of the Link Flow being  
9 reset, then upon receiving a ResetRxRequest message, RLP shall perform the following procedures:

- 10 ■ If the sequence space variable  $Q_{NN,Tx}$  is not equal to the value of the SequenceSpace field  
11 in the ResetRxRequest message, then RLP shall:
  - 12 □ Perform the RLP transmitter initialization procedure defined in 3.3.4.4.1.1.1 for the  
13 Route of the Link Flow being reset.
  - 14 □ Toggle the value of the sequence space variable  $Q_{NN,Tx}$  for the Route of the Link  
15 Flow being reset.
- 16 ■ Respond with a ResetRxAck message.

#### 17 3.3.4.4.1.2.5 RLP reset message flows



18  
19 **Figure 32 RLP reset procedure initiated by RLP transmitter**



20  
21 **Figure 33 RLP reset procedure initiated by RLP receiver**

### 3.3.4.4.2 Data transfer

RLP is an Ack and/or Nak-based protocol with a sequence space of SequenceLength bits, where SequenceLength is indicated by the FlowNNSequenceLengthFwd and FlowNNSequenceLengthRev attribute for forward and reverse Link Flow NN, respectively.

All operations and comparisons performed on RLP packet sequence numbers shall be carried out in unsigned modulo  $2^S$  arithmetic, where S represents the value of SequenceLength. For any RLP octet sequence number  $N$ , the sequence numbers in the range  $[N+1, N+2^{S-1}-1]$  shall be considered greater than  $N$  and the sequence numbers in the range  $[N-2^{S-1}, N-1]$  shall be considered smaller than  $N$ .

#### 3.3.4.4.2.1 RLP transmit procedures

The RLP transmitter shall maintain a SequenceLength-bit variable  $V(S)_{NN,P}$  for all transmitted RLP data units (see Figure 34), where NN is the two-digit hexadecimal Link Flow number in the range 0x00 to  $N_{\text{LinkFlowMax}}-1$  inclusive, and P is the Route indicator that takes values of either A or B.  $V(S)_{NN,P}$  is the sequence number of the next RLP data unit to be sent on Route P of Link Flow NN. The sequence number field (SEQ) in each new RLP packet transmitted shall be set to  $V(S)_{NN,P}$ , corresponding to the sequence number of the first data unit in the packet. If the data unit is octets, then the sequence number of the  $i^{\text{th}}$  octet in the packet (with the first octet being octet 0) is implicitly given by  $\text{SEQ}+i$ .  $V(S)_{NN,P}$  shall be incremented for each data unit contained in the packet.

If an RLP data unit is to be transmitted on the Forward Traffic Channel or on the Reverse Traffic Channel, and if a connection is not open, RLP shall issue an *AirLinkManagement.OpenConnection* command. RLP should queue all data units requiring transmission in the Forward Traffic Channel or in the Reverse Traffic Channel until the protocol receives an *IdleState.ConnectionOpened* indication.

If FlowNNSequenceLengthFwd is 0x0000, then the access network will follow the procedures in 3.3.4.4.2.1.1 when transmitting an RLP packet. If FlowNNSequenceLengthFwd is not 0x0000, then the access network will follow the procedures in 3.3.4.4.2.1.2 when transmitting an RLP packet.

If FlowNNSequenceLengthRev is 0x0000, then the access terminal will follow the procedures in 3.3.4.4.2.1.1 when transmitting an RLP packet. If FlowNNSequenceLengthRev is not 0x0000, then the access terminal will follow the procedures in 3.3.4.4.2.1.2 when transmitting an RLP packet.

#### 3.3.4.4.2.1.1 Transmit procedures for flows with SequenceLength of zero

If the FlowNNSequenceLengthFwd or the FlowNNSequenceLengthRev is 0x0000, the RLP transmitter shall set the First and Last fields of the RLP header to '1'.

If the FlowNNSequenceLengthFwd is 0x0000, then the FlowNNAckNakEnableFwd shall be set to 0x0000. If the FlowNNSequenceLengthFwd is 0x0000, then the FlowNNFTCMACNakEnableFwd attribute should be set to 0x0000. If the FlowNNSequenceLengthFwd is 0x0000, then the FlowNNOutOfOrderDeliveryToRouteProtocolFwd shall be set to 0x0001.

If the FlowNNSequenceLengthRev is 0x0000, then the FlowNNAckNakEnableRev shall be set to 0x0000. If the FlowNNSequenceLengthRev is 0x0000, then the FlowNNRTCMACNakEnableRev attribute should be set to 0x0000. If the FlowNNSequenceLengthRev is 0x0000, then the FlowNNOutOfOrderDeliveryToRouteProtocolRev shall be set to 0x0001.

### 3.3.4.4.2.1.2 Transmit procedures for flows with non-zero SequenceLength

The RLP transmitter should allow sufficient time before deleting an RLP packet payload transmitted for the first time.

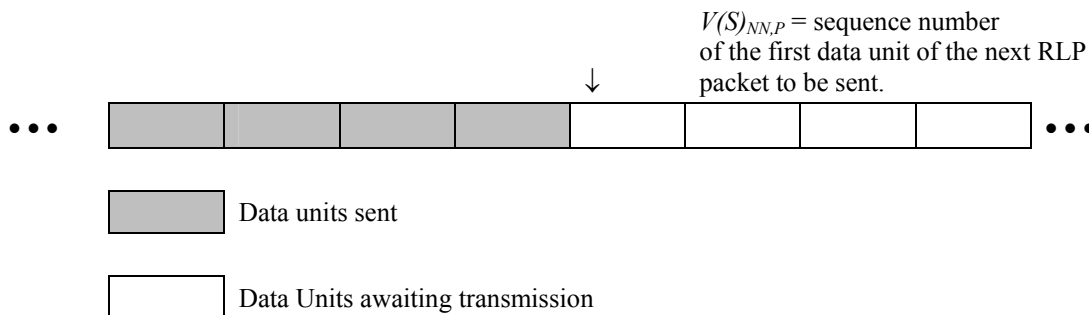
If a ReceiverStatus message is received with the SequenceSpace field not equal to the value of  $Q_{NN,Tx}$ , the message shall be ignored.

Upon receiving a ReceiverStatus message, RLP shall transmit the missing data unit(s) (if any) conveyed by the ReceiverStatus message if those data units are available and if those data units have not been retransmitted before in response to a ReceiverStatus message. Upon receiving a ReceiverStatus message, RLP may transmit the missing data unit(s) (if any) conveyed by the ReceiverStatus message if those data units are available and if those data units have been retransmitted before in response to a ReceiverStatus message.

If the RLP transmitter is the access network and if FlowNNAckNakEnableFwd is not 0x0000 or 0x0001, then the access network may determine that transmitted data units have been lost if it does not receive a ReceiverStatus message acknowledging the receipt of the data units within an implementation-dependent time interval based on the AckTimer and an estimate of the round-trip delay. If the RLP transmitter is the access network and if FlowNNAckNakEnableFwd is not 0x0000 or 0x0001 and the access network determines that transmitted data units were lost, then the access network shall re-transmit the data units if they have not been re-transmitted in response to a ReceiverStatus message. If the RLP transmitter is the access network and if FlowNNAckNakEnableFwd is not 0x0000 or 0x0001 and the access network determines that transmitted data units were lost, then the access network may re-transmit the data units if they have been re-transmitted in response to a ReceiverStatus message.

If the RLP transmitter is the access terminal and if FlowNNAckNakEnableRev is not 0x0000 or 0x0001, then the access terminal may determine that transmitted data units have been lost if it does not receive a ReceiverStatus message acknowledging the receipt of the data units within an implementation-dependent time interval based on the AckTimer and an estimate of the round-trip delay. If the RLP transmitter is the access terminal and if FlowNNAckNakEnableRev is not 0x0000 or 0x0001 and the access terminal determines that transmitted data units were lost, then the access terminal shall re-transmit the data units if they have not been re-transmitted in response to a ReceiverStatus message. If the RLP transmitter is the access terminal and if FlowNNAckNakEnableRev is not 0x0000 or 0x0001 and the access terminal determines that transmitted data units were lost, then the access terminal may re-transmit the data units if they have been re-transmitted in response to a ReceiverStatus message.

If the RLP transmitter is the access network, and the ReceiverStatus record includes any sequence number greater than or equal to  $V(S)_{NN,P}$ , RLP shall perform the reset procedures specified in 3.3.4.4.1.2.1 for Route  $P$  of forward Link Flow  $NN$ . If the RLP transmitter is the access terminal, and the ReceiverStatus record includes any sequence number greater than or equal to  $V(S)_{NN,P}$ , RLP shall perform the reset procedures specified in 3.3.4.4.1.2.1 for Route  $P$  of reverse Link Flow  $NN$ . If the ReceiverStatus record does not include any sequence number greater than or equal to  $V(S)_{NN,P}$  but the requested data units are not available for retransmission, RLP shall ignore the ReceiverStatus record for data units that are not available.



**Figure 34 RLP transmit sequence number variable**

Upon receiving a *ForwardTrafficChannelMAC.ForwardTrafficPacketsMissed* indication for forward Link Flow  $NN$ , the RLP transmitter in the access network shall transmit the requested data units(s) if and only if all of the following conditions are satisfied:

- Flow $NN$ FTCMACNakEnableFwd attribute is set to 0x0001.
- The requested data units have not been retransmitted before.
- The requested data units are available.
- The sequence space for the data units sent is equal to the value of  $Q_{NN,Tx}$ .

Upon receiving a *ReverseTrafficChannelMAC.ReverseTrafficPacketsMissed* indication for reverse Link Flow  $NN$ , the RLP transmitter in the access terminal shall transmit the requested data units(s) if and only if all of the following conditions are satisfied:

- Flow $NN$ RTCMACNakEnableRev attribute is set to 0x0001.
- The requested data units have not been retransmitted before.
- The requested data units are available.
- The sequence space for the data units sent is equal to the value of  $Q_{NN,Tx}$ .

If Flow $NN$ AckNakEnableFwd is 0x0001, then the transmitter at the access network for each Route of Link Flow  $NN$  shall meet the following requirements:

- After transmitting a packet, the RLP transmitter shall start a flush timer for time FlushTimer, where FlushTimer is a parameter of the Flow $NN$ TimersFwd attribute.
- If the RLP transmitter sends another packet before the flush timer expires, the RLP transmitter shall reset and restart the timer.
- If the timer expires, the RLP transmitter shall disable the flush timer and the RLP transmitter should send an RLP packet that contains at least the data unit with sequence number  $V(S)_{NN,P}-1$ .

If Flow $NN$ AckNakEnableRev is 0x0001, then the transmitter at the access terminal for each Route of Link Flow  $NN$  shall meet the following requirements:

- After transmitting a packet, the RLP transmitter shall start a flush timer for time FlushTimer, where FlushTimer is a parameter of the Flow $NN$ TimersRev attribute.
- If the RLP transmitter sends another packet before the flush timer expires, the RLP transmitter shall reset and restart the timer.

- 1           ■ If the timer expires, the RLP transmitter shall disable the flush timer and the RLP  
 2 transmitter should send an RLP packet that contains at least the data unit with sequence  
 3 number  $V(S)_{NN,P}-1$ .

4 The RLP transmitter should not transmit more than  $2^{\text{SequenceLength}-1}$  first-time data units in any  
 5 AbortTimer interval, where SequenceLength is the length of the SEQ field in the RLP header for the  
 6 corresponding Link Flow.

### 7 3.3.4.4.2.1.3 Reservation State Maintenance

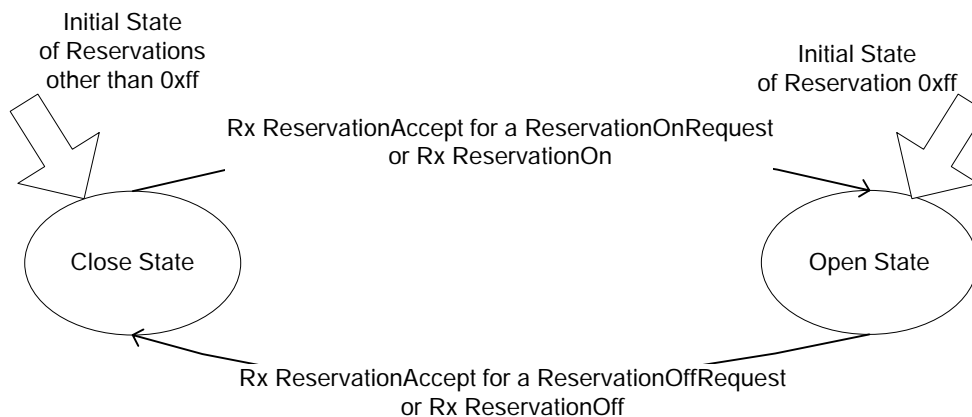
8 The ReservationLabel parameter of the Flow $NN$ ReservationFwd or Flow $NN$ ReservationRev attribute  
 9 indicates the higher layer flows associated with Link Flow  $NN$ . Each ReservationLabel shall be  
 10 associated with no more than one forward Link Flow. Each ReservationLabel shall be associated with  
 11 no more than one reverse Link Flow.

12 Each Reservation can be in one of the following two states:

- 13           ■ Close State  
 14           ■ Open State

15 The transmitter should transmit a higher layer packet using the Link Flow associated with the higher  
 16 layer flow if the associated Link Flow is activated and if the Reservation is in the Open state. The  
 17 transmitter should transmit a higher layer packet belonging to a higher layer flow that is not  
 18 associated with any Link Flow using the Link Flow with ReservationLabel 0xff. The transmitter may  
 19 transmit a higher layer packet belonging to a higher layer flow identified by a Reservation that is in  
 20 the Close state using the Link Flow with ReservationLabel 0xff. The transmitter may transmit a  
 21 higher layer packet belonging to a higher layer flow identified by a Reservation that is bound to a de-  
 22 activated Link Flow using the Link Flow with ReservationLabel 0xff.

23 Figure 35 and Figure 36 show the state transition diagram at the access terminal and the access  
 24 network. State transitions that may be caused by *IdleState.ConnectionOpened*,  
 25 *ConnectedState.ConnectionClosed*, and *ActiveSetManagement.ConnectionLost* indications are not  
 26 shown.

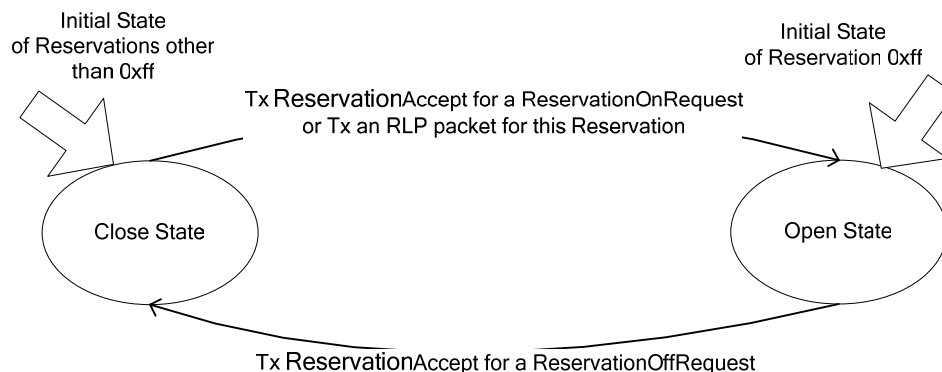


27

28

**Figure 35 Reservation state diagram (access terminal)**





1  
2 **Figure 36 Reservation state diagram (access network)**

3 **3.3.4.4.2.1.3.1 State independent requirements**

4 **3.3.4.4.2.1.3.1.1 Access terminal requirements**

5 Upon receiving a RevReservationOn message, the access terminal shall:

- 6
- 7 ■ Respond with a ReservationAccept message within the time period specified by
  - 8  $T_{RLPResponse}$  of receiving the RevReservationOn message.
  - 9 ■ Set the TransactionID field of the ReservationAccept message to that of the
  - 10 RevReservationOn message.

11 Upon receiving a RevReservationOff message, the access terminal shall:

- 12
- 13 ■ Respond with a ReservationAccept message within the time period specified by
  - 14  $T_{RLPResponse}$  of receiving the RevReservationOff message.
  - 15 ■ Set the TransactionID field of the ReservationAccept message to that of the
  - 16 RevReservationOff message.

17 Upon receiving a FwdReservationOn message, the access terminal shall:

- 18
- 19 ■ Respond with a FwdReservationAck message within the time period specified by
  - 20  $T_{RLPResponse}$  of reception of the FwdReservationOn message.
  - 21 ■ Set the TransactionID field of the FwdReservationAck message to that of the
  - 22 FwdReservationOn message.

23 Upon receiving a FwdReservationOff message, the access terminal shall

- 24
- 25 ■ Respond with a FwdReservationAck message within the time period specified by
  - 26  $T_{RLPResponse}$  of receiving the FwdReservationOff message.
  - 27 ■ Set the TransactionID field of the FwdReservationAck message to that of the
  - 28 FwdReservationOff message.

### 3.3.4.4.2.1.3.1.2 Access network requirements

The access network may re-send a FwdReservationOn message if it does not receive a FwdReservationAck message containing the same TransactionID within the time period specified by  $T_{RLPResponse}$  of sending the FwdReservationOn message.

The access network may re-send a FwdReservationOff message if it does not receive a FwdReservationAck message containing the same TransactionID within the time period specified by  $T_{RLPResponse}$  of sending the FwdReservationOff message.

The access network may send a RevReservationOn message to transition the state of the reverse link Reservation of the access terminal to the Open state. The access network may re-send a RevReservationOn message if it does not receive a ReservationAccept message containing the same TransactionID within the time period specified by  $T_{RLPResponse}$  of sending the RevReservationOn message.

The access network may send a RevReservationOff message to transition the state of the reverse link Reservation of the access terminal to the Close state. The access network may re-send a RevReservationOff message if it does not receive a ReservationAccept message containing the same TransactionID within the time period specified by  $T_{RLPResponse}$  of sending the RevReservationOff message.

If the access network receives a ReservationOnRequest message, it shall:

- Send either a ReservationAccept message or a ReservationReject message within the time period specified by  $T_{RLPResponse}$  of reception of the ReservationOnRequest message.
- Set the TransactionID field of the ReservationAccept or ReservationReject message to that of the ReservationOnRequest message.

If the access network receives a ReservationOffRequest message, it shall:

- Send a ReservationAccept or a ReservationReject message within the time period specified by  $T_{RLPResponse}$  of reception of the ReservationOffRequest message.
- Set the TransactionID field of the ReservationAccept or ReservationReject message to that of the ReservationOffRequest message.

### 3.3.4.4.2.1.3.2 Close state

#### 3.3.4.4.2.1.3.2.1 Access terminal requirements

The access terminal shall not transmit PDUs from higher layer flows belonging to this Reservation using any Link Flow other than the Link Flow associated with ReservationLabel 0xff.

The access terminal may send a ReservationOnRequest message to request transition of the Reservation to the Open state<sup>12</sup>. The access terminal may re-send a ReservationOnRequest message if it does not receive a corresponding ReservationAccept or ReservationReject message within the time

---

<sup>12</sup> Note that the ReservationOnRequest message supports requests for multiple Reservations on both the forward and reverse links. This arrangement allows requests for groups of Reservations (e.g., for bidirectional higher layer application flows) to be combined in the same ReservationOnRequest message.

1 period specified by  $T_{RLP_{Response}}$  of sending the ReservationOnRequest message. If the  
 2 ReservationOnRequest message contains a Reservation bound to a reverse Link Flow, then the  
 3 Reservation shall transition to the Open state when the access terminal receives the corresponding  
 4 ReservationAccept message.

5 Upon receiving a RevReservationOn message, the access terminal shall transition the Reservation to  
 6 the Open state. Upon receiving an *IdleState.ConnectionOpened* indication, the access terminal shall  
 7 transition the Reservations to the Open State whose corresponding ReservationKKIdleStateRev  
 8 attribute is 0x0002, where *KK* is the two-digit hexadecimal ReservationLabel in the range 0x00 to  
 9 0xff inclusive.

### 10 **3.3.4.4.2.1.3.2.2 Access network requirements**

11 If the Reservation entered this state as a result of any condition other than the following conditions,  
 12 then the access network shall send a FwdReservationOff message upon entering this state:

- 13 ■ The access network transmitted a ReservationAccept message in response to a  
 14 ReservationOffRequest message requesting to transition the Reservation to the Close  
 15 state, or
- 16 ■ ReservationKKIdleStateFwd attribute of the Reservation is 0x0001 or 0x0002, and the  
 17 Reservation transitioned to the Close state because the Connection was closed or lost.

18 Upon sending a ReservationAccept message for a Reservation Label bound to a forward Link Flow in  
 19 response to a ReservationOnRequest message, the access network shall transition the Reservation to  
 20 the Open state.

21 Upon sending a FwdReservationOn message, the access network shall transition the Reservation to  
 22 the Open state. Upon receiving an *IdleState.ConnectionOpened* indication, the access network shall  
 23 transition the Reservations to the Open state whose corresponding ReservationKKIdleStateFwd  
 24 attribute is 0x0002, where *KK* is the two-digit hexadecimal ReservationLabel in the range 0x00 to  
 25 0xff inclusive.

### 26 **3.3.4.4.2.1.3.3 Open state**

#### 27 **3.3.4.4.2.1.3.3.1 Access terminal requirements**

28 The access terminal may transmit PDUs from higher layer flows belonging to this Reservation using  
 29 the Link Flow to which the Reservation is bound.

30 The access terminal may send a ReservationOffRequest message to request the transition of a  
 31 Reservation to the Close state. The access terminal may re-send a ReservationOffRequest message if  
 32 it does not receive a ReservationAccept or ReservationReject message within the time period  
 33 specified by  $T_{RLP_{Response}}$  of sending the ReservationOffRequest message. If the ReservationOffRequest  
 34 message contains a Reservation bound to a reverse Link Flow, then the access terminal shall  
 35 transition the Reservation to the Close state when the access terminal receives a ReservationAccept  
 36 message.

37 Upon receiving a RevReservationOff message, the access terminal shall transition the Reservation to  
 38 the Close state.

1 Upon receiving a *ConnectedState.ConnectionClosed* or *ActiveSetManagement.ConnectionLost*  
 2 indication, the access terminal shall transition to the Close state Reservations whose corresponding  
 3 ReservationKKIdleStateRev attribute is 0x0001 or 0x0002, where *KK* is the two-digit hexadecimal  
 4 ReservationLabel.

### 5 **3.3.4.4.2.1.3.3.2 Access network requirements**

6 The access network may transmit PDUs from higher layer flows belonging to this Reservation using  
 7 the Link Flow to which the Reservation is bound.

8 Upon sending a ReservationAccept message for a ReservationLabel bound to a forward Link Flow in  
 9 response to a ReservationOffRequest message, the access network shall transition the Reservation to  
 10 the Close state.

11 Upon receiving a *ConnectedState.ConnectionClosed* or *ActiveSetManagement.ConnectionLost*  
 12 indication, the access network shall transition to the Close state Reservations whose corresponding  
 13 ReservationKKIdleStateFwd attribute is 0x0001 or 0x0002, where *KK* is the two-digit hexadecimal  
 14 ReservationLabel.

15 If, for any *KK*, all of the following conditions are true, the access network shall take action within  
 16  $T_{\text{Turnaround}}$ , where  $T_{\text{Turnaround}}$  is equal to 2 seconds, such that at least one of the following conditions  
 17 would no longer be true (e.g., by modifying the value of ReservationKKQoSUsedFwd or by  
 18 transitioning forward Reservation *KK* to the Close state):

- 19 ■ ReservationKKQoSListFwd is set to a non-default value.
- 20 ■ Forward Reservation *KK* is in the Open state.
- 21 ■ ReservationKKQoSUsedFwd is set to the default value, or the QoSAttributeSet\_ID field  
 22 in ReservationKKQoSUsedFwd is not equal to the value of any QoSAttributeSet\_ID field  
 23 in the corresponding ReservationKKQoSListFwd attribute.

24 If, for any *KK*, all of the following conditions are true, the access network shall take action within  
 25  $T_{\text{Turnaround}}$ , where  $T_{\text{Turnaround}}$  is equal to 2 seconds, such that at least one of the following conditions  
 26 would no longer be true (e.g., by modifying the value of ReservationKKQoSUsedRev or by  
 27 transitioning reverse Reservation *KK* to the Close state):

- 28 ■ ReservationKKQoSListRev is set to a non-default value.
- 29 ■ Reverse Reservation *KK* is in the Open state.
- 30 ■ ReservationKKQoSUsedRev is set to the default value or the QoSAttributeSet\_ID field in  
 31 ReservationKKQoSUsedRev is not equal to the value of any QoSAttributeSet\_ID field in  
 32 the corresponding ReservationKKQoSListRev attribute.

### 33 **3.3.4.4.2.2 RLP receive procedures**

34 If SecurityEnabled public data of the Security Protocol is set to '1', then the RLP receiver shall  
 35 discard any data unit received for which the IsSecure field of the Lower MAC header is set to '0'.

36 If FlowNNSequenceLengthFwd is 0x0000, then the access network will follow the procedures in  
 37 3.3.4.4.2.2.1 when receiving an RLP packet. If FlowNNSequenceLengthFwd is not 0x0000, then the  
 38 access network will follow the procedures in 3.3.4.4.2.2.2 when receiving an RLP packet.

1 If  $FlowNNSequenceLengthRev$  is 0x0000, then the access terminal will follow the procedures in  
 2 3.3.4.4.2.2.1 when receiving an RLP packet. If  $FlowNNSequenceLengthRev$  is not 0x0000, then the  
 3 access terminal will follow the procedures in 3.3.4.4.2.2.2 when receiving an RLP packet.

#### 4 3.3.4.4.2.2.1 Receive procedures for flows with a SequenceLength of zero

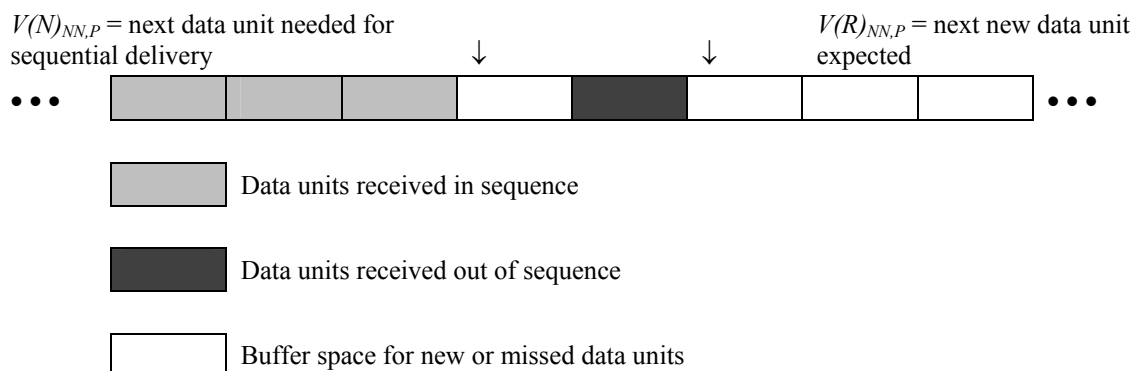
5 If the  $FlowNNSequenceLengthFwd$  or the  $FlowNNSequenceLengthRev$  is 0x0000, then the RLP  
 6 receiver shall perform the following:

- 7 ■ If the First and Last fields of the RLP header are '1', the RLP receiver shall pass the  
 8 complete Route Protocol packet to the Route Protocol layer.
- 9 ■ Otherwise, the RLP receiver shall discard the packet.

#### 10 3.3.4.4.2.2.2 Receive procedures for flows with a non-zero SequenceLength

11 The RLP receiver shall maintain two SequenceLength-bit variables for receiving,  $V(R)_{NN,P}$  and  
 12  $V(N)_{NN,P}$  (see Figure 37), where  $NN$  is the two-digit hexadecimal Link Flow number in the range 0x00  
 13 to  $N_{LinkFlowMax}-1$  inclusive, and  $P$  is the Route indicator that takes values of either A or B.  $V(R)_{NN,P}$   
 14 contains the sequence number of the next data unit expected to arrive.  $V(N)_{NN,P}$  contains the sequence  
 15 number of the first missing data unit, as described below.

16 In addition, the RLP receiver shall keep track of the status of each data unit in its resequencing buffer  
 17 indicating whether the data unit was received or not. Use of this status is implied in the following  
 18 procedures. The RLP receiver informs the RLP transmitter of the status of data units in its receive  
 19 buffer by sending a ReceiverStatus message. The ReceiverStatus message shall convey status of all  
 20 missing data from  $V(N)_{NN,P}$  onwards that has not been conveyed in a previous ReceiverStatus message  
 21 and  $V(R)_{NN,P}$ . The ReceiverStatus message may convey status of missing data that has been conveyed  
 22 in previous ReceiverStatus messages. The ReceiverStatus message shall not convey status of data  
 23 units with a sequence number less than  $V(N)_{NN,P}$ .



25 **Figure 37 RLP receive sequence number variables**

26 In the following,  $X$  denotes the sequence number of a received data unit. For each received data unit,  
 27 RLP shall perform the following procedures in order:

- 28 ■ If the RLP receiver is an access terminal and  $FlowNNAckNakEnableFwd$  is 0x0002, then  
 29 RLP shall send a ReceiverStatus message within AckTimer interval of receiving the data  
 30 unit.

- 1 ■ If the RLP receiver is an access network and Flow $_{NN}$ AckNakEnableRev is 0x0002, then  
2 RLP shall send a ReceiverStatus message within AckTimer interval of receiving the data  
3 unit.
- 4 ■ If the RLP receiver is an access terminal and Flow $_{NN}$ AckNakEnableFwd is 0x0003 and  
5 the Last field of the RLP header is '0', then RLP shall send a ReceiverStatus message  
6 within AckTimer interval of receiving the data unit.
- 7 ■ If the RLP receiver is an access network and Flow $_{NN}$ AckNakEnableRev is 0x0003 and  
8 the Last field of the RLP header is '0', then RLP shall send a ReceiverStatus message  
9 within AckTimer interval of receiving the data unit.
- 10 ■ If the RLP receiver is an access terminal and Flow $_{NN}$ AckNakEnableFwd is 0x0003 and  
11 the Last field of the RLP header is '1', then RLP shall send a ReceiverStatus message  
12 upon receiving the data unit.
- 13 ■ If the RLP receiver is an access network and Flow $_{NN}$ AckNakEnableRev is 0x0003 and  
14 the Last field of the RLP header is '1', then RLP shall send a ReceiverStatus message  
15 upon receiving the data unit.
- 16 ■ If  $X < V(N)_{NN,P}$ , the data unit shall be discarded as a duplicate.
- 17 ■ If  $V(N)_{NN,P} \leq X < V(R)_{NN,P}$ , and the data unit is not already stored in the resequencing  
18 buffer, then:
- 19 □ RLP shall store the received data unit in the resequencing buffer.
- 20 □ If  $X = V(N)_{NN,P}$  and if in-order delivery of Route Protocol packets is required, then  
21 RLP shall pass all contiguous complete Route Protocol packets in the resequencing  
22 buffer that have not been passed to the Route Protocol, from the beginning of the  
23 resequencing buffer upward to the Route Protocol. RLP shall then set  $V(N)_{NN,P}$  to  
24 (LAST+1) where LAST is the sequence number of the last contiguous data unit in the  
25 resequencing buffer.
- 26 □ If  $X = V(N)_{NN,P}$  and if in-order delivery of Route Protocol packets is not required, then  
27 RLP shall pass all complete Route Protocol packets in the resequencing buffer that  
28 have not been passed to the Route Protocol layer, from the beginning of the  
29 resequencing buffer upward to the Route Protocol. RLP shall then set  $V(N)_{NN,P}$  to  
30 (LAST+1) where LAST is the sequence number of the last contiguous data unit in the  
31 resequencing buffer.
- 32 ■ If  $V(N)_{NN,P} < X < V(R)_{NN,P}$  and the data unit has already been stored in the resequencing  
33 buffer, then the data unit shall be discarded as a duplicate.
- 34 ■ If  $X = V(R)_{NN,P}$ , then:
- 35 □ If  $V(R)_{NN,P} = V(N)_{NN,P}$ , then RLP shall increment  $V(N)_{NN,P}$  and  $V(R)_{NN,P}$  and shall pass  
36 all complete Route Protocol packets in the resequencing buffer that have not been  
37 passed to the Route Protocol, from the beginning of the resequencing buffer upward  
38 to the Route Protocol.
- 39 □ If  $V(R)_{NN,P} \neq V(N)_{NN,P}$ , then RLP shall increment  $V(R)_{NN,P}$  and shall store the data unit  
40 in the resequencing buffer. If in-order delivery of Route Protocol packets is not  
41 required, then RLP shall pass all complete Route Protocol packets in the  
42 resequencing buffer that have not been passed to the Route Protocol, from the  
43 beginning of the resequencing buffer upward to the Route Protocol.

- 1           ■ If  $X > V(R)_{NN,P}$ , then:
- 2           □ RLP shall store the data unit in the resequencing buffer.
- 3           □ If in-order delivery of Route Protocol packets is not required, then RLP shall pass all
- 4           complete Route Protocol packets in the resequencing buffer that have not been
- 5           passed to the Route Protocol, from the beginning of the resequencing buffer upward
- 6           to the Route Protocol.
- 7           □ If the RLP receiver is an access network, then RLP shall set an RLP abort timer to
- 8           AbortTimer, where AbortTimer is a parameter of the FlowNNTimersRev attribute,
- 9           for each missing RLP data unit from  $V(R)_{NN,P}$  to  $X-1$ , inclusive. If the RLP receiver is
- 10          an access terminal, then RLP shall set an RLP abort timer to AbortTimer where
- 11          AbortTimer is a parameter of the FlowNNTimersFwd attribute, for each missing RLP
- 12          data unit from  $V(R)_{NN,P}$  to  $X-1$ , inclusive.
- 13          □ RLP shall set  $V(R)_{NN,P}$  to  $X+1$ .
- 14          □ If the RLP receiver is an access terminal and if the FlowNNAckNakEnableFwd
- 15          attribute is not 0x0000, then RLP shall send a ReceiverStatus message. If the RLP
- 16          receiver is an access network and if the FlowNNAckNakEnableRev attribute is not
- 17          0x0000, then RLP shall send a ReceiverStatus message.

18          If a missing data unit has not arrived when its RLP abort timer expires and if in-order delivery of

19          Route Protocol packets is required, then RLP shall pass all complete Route Protocol packets that have

20          not been passed to the Route Protocol, from the beginning of the resequencing buffer upward up to

21          the next missing data unit to the Route Protocol. RLP may pass to the Route Protocol partially

22          received packets with an indication of partial packet delivery.

23          If the RLP receiver is the access network and if FlowNNAckNakEnableRev is not 0x0000, then the

24          access network may determine that a ReceiverStatus message or the retransmitted data units have

25          been lost if it does not receive the data units within an implementation-dependent time interval based

26          on an estimate of the round-trip delay and if other packets were received from the access terminal. If

27          the RLP receiver is the access network and if FlowNNAckNakEnableRev is not 0x0000 and the

28          access network determines that a ReceiverStatus message or the retransmitted data units were lost,

29          and the abort timer for the retransmitted data units has not expired, then the access network shall

30          transmit a ReceiverStatus message.

31          If the RLP receiver is the access terminal and if FlowNNAckNakEnableFwd is not 0x0000, then the

32          access terminal may determine that a ReceiverStatus message or the retransmitted data units have

33          been lost if it does not receive the data units within an implementation-dependent time interval based

34          on an estimate of the round-trip delay and if other packets were received from the access network. If

35          the RLP receiver is the access terminal and if FlowNNAckNakEnableFwd is not 0x0000 and the

36          access terminal determines that a ReceiverStatus message or the retransmitted data units were lost,

37          and the abort timer for the retransmitted data units has not expired, then the access terminal shall

38          transmit a ReceiverStatus message.

39          RLP shall set  $V(N)_{NN,P}$  to the sequence number of the next missing data unit, or to  $V(R)_{NN}$  if there are

40          no remaining missing data units. Further recovery is the responsibility of higher layer protocols.

### 3.3.4.4.3 In-band message transfer

The access network shall send the in-band messages in 3.3.4.6 on forward Link Flow  $N_{\text{LinkFlowMax}}-1$ .  
The access network shall not send an in-band message on forward Link Flows 0x00 to  $N_{\text{LinkFlowMax}}-2$ .

The access terminal shall send the in-band messages in 3.3.4.6 on reverse link Flow  $N_{\text{LinkFlowMax}}-1$ .  
The access terminal shall not send an in-band message on reverse Link Flows 0x00 to  $N_{\text{LinkFlowMax}}-2$ .

The access network and access terminal shall not send an in-band message using the Signaling Transport. The access network and access terminal shall discard an in-band message received from the Signaling Transport.

All in-band messages shall apply only to the instance of the Default Data Transport sending and receiving the message. All in-band messages shall apply only to the Route that the message is sent and received on.

### 3.3.4.4.4 RLP packet header

The RLP packet header, which precedes the RLP payload, has the following format:

Field	Length (bits)
LinkFlowNumber	4
Route	1
SequenceSpace	1
First	1
Last	1
SEQ	0, 8, 16, or 24

**LinkFlowNumber** The identifier for this Link Flow.

**Route** If this RLP packet is sent on Route A, then the sender shall set this field to '0'. Otherwise, the sender shall set this field to '1'.

**SequenceSpace** The sender shall set this flag to the value of the sequence space variable  $Q_{NN,TX}$ .

**First** If the payload of this RLP packet is the first segment of a Route Protocol packet, then the sender shall set this field to '1'. Otherwise, the sender shall set this field to '0'.

**Last** If the payload of this RLP packet is the last segment of a Route Protocol packet, then the sender shall set this field to '1'. Otherwise, the sender shall set this field to '0'.



1 SEQ The RLP sequence number of the first data unit in the RLP payload<sup>13</sup>. If this  
 2 RLP packet is being sent on the forward link, the length of this field is  
 3 indicated by the FlowNNSequenceLengthFwd attribute corresponding to this  
 4 flow. If this RLP packet is being sent on the reverse link, the length of this  
 5 field is indicated by the FlowNNSequenceLengthRev attribute corresponding  
 6 to this flow.

### 7 3.3.4.5 Message formats

8 The protocol uses the AttributeUpdateRequest, AttributeUpdateAccept, and AttributeUpdateReject  
 9 messages of the Generic Attribute Update Protocol in 10.9 to update configurable attributes.

#### 10 3.3.4.5.1 ReservationOnRequest

11 The access terminal sends this message to request transition of one or more Reservations to the Open  
 12 State.

Field	Length (bits)
MessageID	8
TransactionID	8
ReservationCount	8

ReservationCount occurrences of the following three fields:

{

Reserved	7
Link	1
ReservationLabel	8

}

14 MessageID The access terminal shall set this field to 0x04.

15 TransactionID The access terminal shall set this field according to 10.8 for each  
 16 ReservationOnRequest or ReservationOffRequest message sent.

17 ReservationCount The access terminal shall set this field to the number of ReservationLabel  
 18 fields in this message. The access terminal shall include ReservationCount  
 19 occurrences of the following three fields with the message.

20 Reserved The access terminal shall set this field to '0000000'. The access network  
 21 shall ignore this field.

22 Link If this request is for a forward Reservation, then the access terminal shall set  
 23 this field to '1'. If this request is for a reverse Reservation, then the access  
 24 terminal shall set this field to '0'.

<sup>13</sup> When data unit is set to RLP payload, the RLP packet contains one data unit.

1 ReservationLabel The access terminal shall set this field to the Reservation for which this  
 2 request is generated.

<b>Channels</b>	RTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Unicast	<b>Security</b>	Required

4

### 5 3.3.4.5.2 ReservationOffRequest

6 The access terminal sends this message to request transition of one or more Reservations to the Close  
 7 State.

8

Field	Length (bits)
MessageID	8
TransactionID	8
ReservationCount	8

ReservationCount occurrences of the following three fields:

Field	Length (bits)
Reserved	7
Link	1
ReservationLabel	8

9 MessageID The access terminal shall set this field to 0x05.

10 TransactionID The access terminal shall set this field according to 10.8 for each  
 11 ReservationOnRequest or ReservationOffRequest message sent.

12 ReservationCount The access terminal shall set this field to the number of ReservationLabel  
 13 fields in this message. The sender shall include ReservationCount  
 14 occurrences of the following three fields with the message.

15 Reserved The access terminal shall set this field to '0000000'. The access network  
 16 shall ignore this field.

17 Link If this request is for a forward Reservation, then the access terminal shall set  
 18 this field to '1'. If this request is for a reverse Reservation, then the access  
 19 terminal shall set this field to '0'.

20 ReservationLabel The access terminal shall set this field to the Reservation for which this  
 21 request is generated.

22

<b>Channels</b>	RTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Unicast	<b>Security</b>	Required

### 3.3.4.5.3 ReservationAccept

The access network sends this message to acknowledge reception of and allow the state transition requested by a ReservationOnRequest or ReservationOffRequest message. The access terminal sends this message to acknowledge reception of and accept the state transition requested by a RevReservationOn or RevReservationOff message.

Field	Length (bits)
MessageID	8
TransactionID	8

**MessageID** The sender shall set this field to 0x06.

**TransactionID** The access network shall set this field to the TransactionID field of the ReservationOnRequest or ReservationOffRequest message to which the access network is responding. The access terminal shall set this field to the TransactionID field of the RevReservationOn or RevReservationOff message to which the access terminal is responding.

<b>Channels</b>	FTC    RTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Unicast	<b>Security</b>	Required

### 3.3.4.5.4 ReservationReject

The access network sends this message to acknowledge reception of and deny the state transition requested by a ReservationOnRequest or ReservationOffRequest message.

Field	Length (bits)
MessageID	8
TransactionID	8
ReservationCount	8

ReservationCount occurrences of the following three fields:

{	
Reserved	7
AllowableLink	1
AllowableReservationLabel	8
}	

**MessageID** The access network shall set this field to 0x07.

**TransactionID** The access network shall set this field to the TransactionID field of the ReservationOnRequest or ReservationOffRequest message to which the access network is responding.

- 1 ReservationCount The access network shall set this field to the number of ReservationLabel  
2 fields in this message. The sender shall include ReservationCount  
3 occurrences of the following three fields with the message.
- 4 Reserved The access network shall set this field to '0000000'. The access terminal  
5 shall ignore this field.
- 6 AllowableLink If the Reservation for which the access network would have allowed the state  
7 transition requested in the ReservationOnRequest or ReservationOffRequest  
8 message is a forward Reservation, then the access network shall set this field  
9 to '1'. If the Reservation for which the access network would have allowed  
10 the state transition requested in the ReservationOnRequest or  
11 ReservationOffRequest message is a reverse Reservation, then the access  
12 network shall set this field to '0'.
- 13 AllowableReservationLabel  
14 The access network shall set this field to the Reservation for which the access  
15 network would have allowed the state transition requested in the  
16 ReservationOnRequest or ReservationOffRequest message.

<b>Channels</b>	FTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Unicast	<b>Security</b>	Required

18

### 19 3.3.4.5.5 RevReservationOn

20 The access network sends this message to transition an activated reverse Reservation to the Open  
21 state.

22

Field	Length (bits)
MessageID	8
TransactionID	8
ReservationCount	8

ReservationCount occurrences of the following field:

{

ReservationLabel	8
------------------	---

}

- 23 MessageID The access network shall set this field to 0x08.
- 24 TransactionID The access network shall set this field according to 10.8 for each  
25 RevReservationOn or RevReservationOff message sent..
- 26 ReservationCount The access network shall set this field to the number of ReservationLabel  
27 fields in this message. The sender shall include ReservationCount  
28 occurrences of the following field with the message.

1 ReservationLabel The access network shall set this field to the Reservation that is to be  
 2 transitioned to the Open state.

<b>Channels</b>	FTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Unicast	<b>Security</b>	Required

4

### 5 3.3.4.5.6 RevReservationOff

6 The access network sends this message to transition an activated reverse link Reservation to the Close  
 7 state.

8

Field	Length (bits)
MessageID	8
TransactionID	8
ReservationCount	8

ReservationCount occurrences of the following field:

{

ReservationLabel	8
------------------	---

}

9 MessageID The access network shall set this field to 0x09.

10 TransactionID The access network shall set this field according to 10.8 for each  
 11 RevReservationOn or RevReservationOff message sent. If this is the first  
 12 RevReservationOn or RevReservationOff message sent by the access  
 13 network, then the access network shall set this field to zero.

14 ReservationCount The access network shall set this field to the number of Reservation fields in  
 15 this message. The sender shall include ReservationCount occurrences of the  
 16 following field with the message.

17 ReservationLabel The access network shall set this field to the Reservation which is to be  
 18 transitioned to the Close state.

19

<b>Channels</b>	FTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Unicast	<b>Security</b>	Required

20

### 3.3.4.5.7 FwdReservationOn

The access network sends this message to inform the access terminal when a forward Reservation transitions to the Open state.

Field	Length (bits)
MessageID	8
TransactionID	8
ReservationCount	8

ReservationCount occurrences of the following field:

{	
ReservationLabel	8
}	

**MessageID** The access network shall set this field to 0x0a.

**TransactionID** The access network shall set this field according to 10.8 for each FwdReservationOn or FwdReservationOff message sent.

**ReservationCount** The access network shall set this field to the number of ReservationLabel fields in this message. The sender shall include ReservationCount occurrences of the following field with the message.

**ReservationLabel** The access network shall set this field to the Reservation which is to be transitioned to the Open state.

<b>Channels</b>	FTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Unicast	<b>Security</b>	Required

### 3.3.4.5.8 FwdReservationOff

The access network sends this message to inform the access terminal when a forward Reservation transitions to the Close state.

Field	Length (bits)
MessageID	8
TransactionID	8
ReservationCount	8

ReservationCount occurrences of the following field:

{	
ReservationLabel	8
}	

- 1 MessageID The access network shall set this field to 0x0b.
- 2 TransactionID The access network shall set this field according to 10.8 for each  
3 FwdReservationOn or FwdReservationOff message sent.
- 4 ReservationCount The access network shall set this field to the number of ReservationLabel  
5 fields in this message. The sender shall include ReservationCount  
6 occurrences of the following field with the message.
- 7 ReservationLabel The access network shall set this field to the Reservation which is to be  
8 transitioned to the Close state.

<b>Channels</b>	FTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Unicast	<b>Security</b>	Required

10

### 11 3.3.4.5.9 FwdReservationAck

- 12 The access terminal sends this message to acknowledge reception of the FwdReservationOn or the  
13 FwdReservationOff message and to accept the related state transition.

14

Field	Length (bits)
MessageID	8
TransactionID	8

- 15 MessageID The access network shall set this field to 0x0c.
- 16 TransactionID The access terminal shall set this field to the TransactionID field of the  
17 FwdReservationOn or FwdReservationOff message to which the access  
18 network is responding.

19

<b>Channels</b>	RTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Unicast	<b>Security</b>	Required

20

### 3.3.4.5.10 FlowQoSdetect

The access network sends this message to inform the access terminal that it should add a new Reservation or modify the Reservation *KK*QoSListFwd or Reservation *KK*QoSListRev attribute for an existing Reservation *KK*. The access network may include a packet filter specification in the message.

Field	Length (bits)
MessageID	8
Reserved1	6
Link	1
ReservationIncluded	1
ReservationLabel	0 or 8
ReservationPriority	8
QoSAttributeSetCount	8

QoSAttributeSetCount occurrences of the following record:

{	
QoSAttributeSetLength	8
QoSAttributeSet	QoSAttributeSetLength ×8
}	
FilterSpecCount	8

FilterSpecCount occurrences of the following record:

{	
FilterSpecType	8
FilterSpecLength	8
FilterSpec	FilterSpecLength × 8
}	

- 6 MessageID The access network shall set this field to 0x0d.
- 7 Reserved1 The access network shall set this field to '000000'. The access terminal shall  
8 ignore this field.
- 9 Link If this message is for a forward Reservation, then the access network shall set  
10 this field to '1'. If this message is for a reverse Reservation, then the access  
11 network shall set this field to '0'.
- 12 ReservationIncluded The access network shall set this field to '1' to modify an existing  
13 Reservation. The access network shall set this field to '0' to add a new  
14 Reservation.
- 15 ReservationLabel If ReservationIncluded is '0', then the access network shall omit this field.  
16 Otherwise, the access network shall set this field to the Reservation for which  
17 this message is generated.



- 1 ReservationPriority The access network shall set this field to indicate the priority to be assigned  
2 to the reservation. The value 0x00 indicates the highest priority and the value  
3 0xff indicates the lowest priority.
- 4 QoSAttributeSetCount The access network shall set this field to the number of QoSAttributeSets  
5 associated with this reservation. Each QoSAttributeSet contains one set of  
6 acceptable QoS parameters. If multiple QoS attribute sets are included, the  
7 sender shall include the QoS attribute sets in descending order of preference.  
8 The sender shall include QoSAttributeSetCount occurrences of the following  
9 two fields with the message.
- 10 QoSAttributeSetLength The access network shall set this field to the length, in octets, of the  
11 QoSAttributeSet.  
12
- 13 QoSAttributeSet The QoS parameters requested for the reservation. The access network shall  
14 set this record as defined in 3.3.6.2.9.1.
- 15 FilterSpecCount The access network shall set this field to the number of FilterSpecs  
16 associated with this reservation. The sender shall include FilterSpecCount  
17 occurrences of the following three fields with the message.
- 18 FilterSpecType The access network shall set this field to an identifier for the Filter  
19 Specification Type according to Table 13.
- 20 FilterSpecLength The access network shall set this field to the length of the FilterSpec field in  
21 units of octets.
- 22 FilterSpec If FilterSpecType is 0x01, then the access network shall set this record as  
23 defined in 3.3.6.2.13.1. If FilterSpecType is 0x02, then the sender shall set  
24 this record as defined in 3.3.6.2.13.2. Otherwise, the sender shall omit this  
25 record.

<b>Channels</b>	FTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Unicast	<b>Security</b>	Required

27

### 3.3.4.6 In-band message formats

#### 3.3.4.6.1 ResetRxRequest

The RLP receiver in the access terminal or the access network sends the ResetRxRequest message to reset its peer RLP transmitter.

Field	Length (bits)
MessageID	8
TransactionID	8
ResetAllFlows	1
LinkFlowNumber	4
SequenceSpace	1
Reserved	2

MessageID The sender shall set this field to 0xf0.

TransactionID The sender shall set this field according to 10.8.

ResetAllFlows The sender shall set this field to '1' to reset all Link Flows. Otherwise the sender shall set this field to '0'.

LinkFlowNumber The sender shall set this field to the Link Flow that is reset. The sender shall set this field to '0000' if the ResetAllFlows field is set to '1'. The receiver shall ignore this field if the ResetAllFlows field is set to '1'.

SequenceSpace The sender shall set this flag to the value of the sequence space variable  $Q_{NN,Rx}$ . The sender shall set this field to '0' if the ResetAllFlows field is set to '1'. The receiver shall ignore this field if the ResetAllFlows field is set to '1'.

Reserved The sender shall set this field to '00'. The receiver shall ignore this field.

#### 3.3.4.6.2 ResetRxAck

The RLP transmitter in the access terminal or the access network sends the ResetRxAck message in response to the ResetRxRequest message.

Field	Length (bits)
MessageID	8
TransactionID	8

MessageID The sender shall set this field to 0xf1.

TransactionID The sender shall set this field to the TransactionID of the associated ResetRxRequest message.

### 3.3.4.6.3 ResetTxIndication

The RLP transmitter in the access terminal or the access network sends the ResetTxIndication message to reset its peer RLP receiver.

Field	Length (bits)
MessageID	8
TransactionID	8
ResetAllFlows	1
LinkFlowNumber	4
SequenceSpace	1
Reserved	2

MessageID The sender shall set this field to 0xf2.

TransactionID The sender shall set this field according to 10.8.

ResetAllFlows The sender shall set this field to '1' to reset all Link Flows. Otherwise the sender shall set this field to '0'.

LinkFlowNumber The sender shall set this field to the Link Flow that is reset. The sender shall set this field to '0000' if the ResetAllFlows field is set to '1'. The receiver shall ignore this field if the ResetAllFlows field is set to '1'.

SequenceSpace The sender shall set this flag to the value of the sequence space variable  $Q_{NN,Tx}$ . The sender shall set this field to '0' if the ResetAllFlows field is set to '1'. The receiver shall ignore this field if the ResetAllFlows field is set to '1'.

Reserved The sender shall set this field to '00'. The receiver shall ignore this field.

### 3.3.4.6.4 ResetTxAck

The RLP receiver in the access terminal or the access network sends the ResetTxAck message in response to the ResetTxIndication message.

Field	Length (bits)
MessageID	8
TransactionID	8

MessageID The sender shall set this field to 0xf3.

TransactionID The sender shall set this field to the TransactionID of the associated ResetTxIndication message.

### 3.3.4.6.5 ReceiverStatus

The access terminal and the access network send the ReceiverStatus message to acknowledge the receipt of one or more RLP data units or to request the retransmission of one or more RLP data units.

Field	Length (bits)
MessageID	8
LinkFlowNumber	4
SequenceSpace	1
LatestDataUnit	1
Reserved0	1
TimeStampIncluded	1
Reserved1	0 or 4
TimeStamp	0 or 12
SequenceLength	8
ReportCount	8

ReportCount occurrences of the following record:

{	
FirstErasedDataUnit	SequenceLength
WindowLen	SequenceLength
}	
VR	SequenceLength

5	MessageID	The sender shall set this field to 0xf4.
6	LinkFlowNumber	The sender shall set this field to the Link Flow for which this ReceiverStatus is being sent.
7		
8	SequenceSpace	The sender shall set this flag to the value of the sequence space variable $Q_{NN,Rx}$ .
9		
10	Reserved0	The sender shall set this field to '0'. The receiver shall ignore this field.
11	LatestDataUnit	If the latest data unit in the receive buffer is the data unit with sequence number $V(R) - 1$ , then the sender shall set this field to '1'. Otherwise, the sender shall set this field to '0'.
12		
13		
14	TimeStampIncluded	If the value of the FlowNNAckNakEnableFwd attribute is 0x0002 or 0x0003, then the access terminal shall set this field to '1'. Otherwise, the access terminal shall set this field to '0'. If the value of the FlowNNAckNakEnableRev attribute is 0x0002 or 0x0003, then the access network shall set this field to '1'. Otherwise, the access network shall set this field to '0'. <i>NN</i> is the two-digit hexadecimal Link Flow number.
15		
16		
17		
18		
19		
20	Reserved1	If TimeStampIncluded is '0', then the sender shall omit this field. Otherwise, the sender shall set this field to '0000'. The receiver shall ignore this field.
21		

1	TimeStamp	If TimeStampIncluded is '0', then the sender shall omit this field. Otherwise, the sender shall set this field to the 12 least significant bits of the system time, in units of frames, when the latest data unit in the receive buffer was received.
2		
3		
4		
5	SequenceLength	The sender shall set this field to the length of the sequence number as indicated by the FlowNNSequenceLengthFwd or
6		FlowNNSequenceLengthRev attribute for forward or reverse Link Flow NN, respectively in units of bits.
7		
8		
9	ReportCount	The sender shall set this field to the number of Report records included in this message. The sender shall include ReportCount occurrences of the following two fields with the message.
10		
11		
12	FirstErasedDataUnit	The sender shall set this field to the sequence number of the first RLP data unit erased in a sequence of erased data units.
13		
14	WindowLen	The sender shall set this field to the length of the erased window in data units.
15		
16	VR	The sender shall set this field to $V(R)_{NN,P}$ .

### 17 3.3.4.7 Interface to other protocols

#### 18 3.3.4.7.1 Commands

19 This protocol issues the following command:

- 20 ■ *AirLinkManagement.OpenConnection*

#### 21 3.3.4.7.2 Indications

22 This protocol registers to receive the following indications:

- 23 ■ *IdleState.ConnectionOpened*
- 24 ■ *ReverseTrafficChannelMAC.ReverseTrafficPacketsMissed* along with parameters indicating the Link Flow number and missing octets.
- 25
- 26 ■ *ForwardTrafficChannelMAC.ForwardTrafficPacketsMissed* along with parameters indicating the Link Flow number and missing octets.
- 27
- 28 ■ *ConnectedState.ConnectionClosed*
- 29 ■ *ActiveSetManagement.ConnectionLost*

#### 30 3.3.4.8 RLP packet priorities

31 For a given Link Flow, the sender shall assign higher priority to packets containing retransmitted transport traffic than packets containing only first time transmissions. If  
 32 FlowNNTransmitAbortTimerRev is not set to 0x0000, then the access terminal should transmit a  
 33 higher layer data unit within FlowNNTransmitAbortTimerRev time of the higher layer data unit being  
 34 received. The access terminal may use the FlowNNTransmitAbortTimerRev attribute to determine the  
 35 priority of reverse RLP packets. If FlowNNTransmitAbortTimerFwd is not set to 0x0000, then the  
 36

1 access network should transmit a higher layer data unit within FlowNNTransmitAbortTimerFwd time  
 2 of the higher layer data unit being received. The access network may use the  
 3 FlowNNTransmitAbortTimerFwd attribute to determine the priority of forward RLP packets.

#### 4 3.3.4.9 Protocol numeric constants

Constant	Meaning	Value
$N_{\text{LinkFlowMax}}$	Maximum total number of activated and deactivated Link Flows.	16
$T_{\text{RLPResponse}}$	Time period within which the access network is to respond to ReservationOnRequest, ReservationOffRequest, ResetTxIndication and ResetRxRequest messages. Time period within which the access terminal is to respond to ResetTxIndication and ResetRxRequest messages.	1 second

### 7 3.3.5 Flow Control Protocol

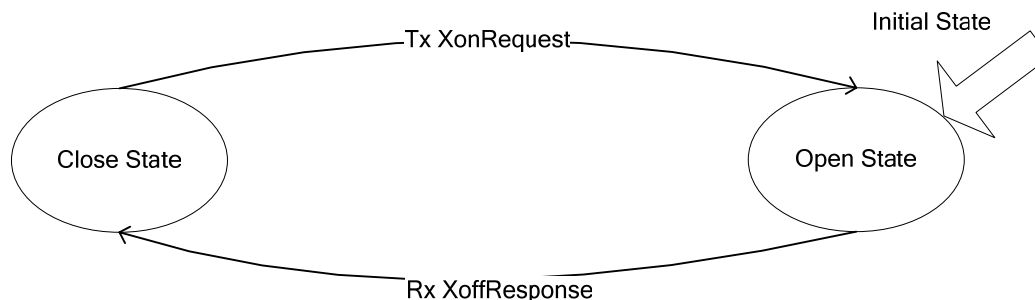
#### 8 3.3.5.1 Overview

9 The Flow Control Protocol provides procedures and messages used by the access terminal and the  
 10 access network to perform flow control for the forward link of the Default Data Transport.

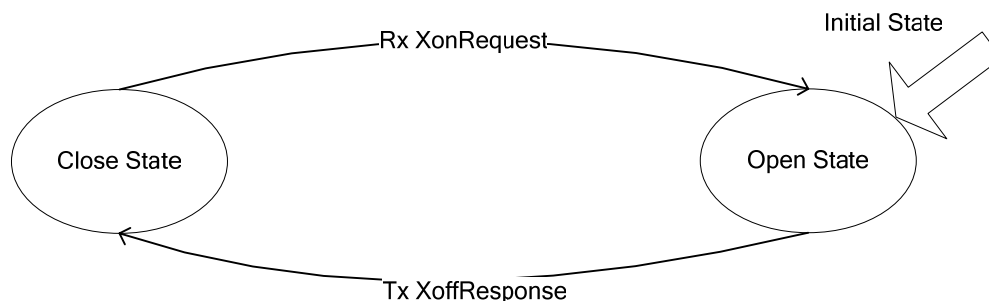
11 This protocol can be in one of the following states:

- 12 ■ Close State: in this state, the Default Data Transport for the access network does not send  
 13 any RLP packets.
- 14 ■ Open State: in this state, the Default Data Transport for the access network can send RLP  
 15 packets.

16 Figure 38 and Figure 39 show the state transition diagram at the access terminal and the access  
 17 network.



18  
 19 **Figure 38 Flow control protocol state diagram (access terminal)**



**Figure 39 Flow control protocol state diagram (access network)**

The flow control protocol is a protocol associated with the Default Data Transport.

### 3.3.5.2 Primitives

#### 3.3.5.2.1 Commands

This protocol does not define any commands.

#### 3.3.5.2.2 Return indications

This protocol does not return any indications.

### 3.3.5.3 Protocol data unit

The transmission unit of this protocol is a message. This is a control protocol and, therefore, it does not carry payload on behalf of other layers or protocols.

### 3.3.5.4 Procedures

All messages for the flow control protocol shall apply only to the instance of the Default Data Transport sending and receiving the message.

The flow control protocol makes use of the XOnRequest, XOnResponse, XOffRequest, and XOffResponse in-band messages as defined in 3.3.4.4.3.

#### 3.3.5.4.1 Transmission and processing of RestartNetworkInterface message

The access network may send a RestartNetworkInterface message to direct the access terminal to restart the interface between the Data Transport and the higher layer.

Upon receiving a RestartNetworkInterface message, the access terminal shall send a RestartNetworkInterfaceAck message within the time period specified by  $T_{FCResponse}$ , and shall restart the interface between the Data Transport and the higher layer. The access terminal may also restart higher layer protocols.

#### 3.3.5.4.2 Close state

In this state, the access network shall not send any RLP packets. In this state, the access network may send RLP in band messages.

#### 1 **3.3.5.4.2.1 Access terminal requirements**

2 The access terminal shall send an XonRequest message when it is ready to receive RLP packets from  
3 the access network.

4 The access terminal shall transition to the Open State when it sends an XonRequest message.

#### 5 **3.3.5.4.2.2 Access network requirements**

6 If the access network receives an XonRequest message, it shall:

- 7 ■ Send an XonResponse message within the time period specified by  $T_{FCResponse}$  after  
8 reception of the XonRequest message to acknowledge reception of the message.
- 9 ■ Transition to the Open State.

#### 10 **3.3.5.4.3 Open state**

11 In this state, the access terminal and the access network may send or receive any RLP packets.

#### 12 **3.3.5.4.3.1 Access terminal requirements**

13 The access terminal may re-send an XonRequest message if it does not receive an XonResponse  
14 message or an RLP packet (corresponding to this instance of the Default Data Transport) within the  
15 time period specified by  $T_{FCResponse}$  after sending the XonRequest message.

16 The access terminal may send an XoffRequest message to request the access network to stop sending  
17 RLP packets. The access terminal shall transition to the Close state when it receives an XoffResponse  
18 message with a TransactionID field equal to the TransactionID sent in the XoffRequest message.

19 The access terminal may re-send an XoffRequest message if it does not receive an XoffResponse  
20 message within the time period specified by  $T_{FCResponse}$  after sending the XoffRequest message.

#### 21 **3.3.5.4.3.2 Access network requirements**

22 If the access network receives an XoffRequest message, it shall

- 23 ■ Send an XoffResponse message within the time period specified by  $T_{FCResponse}$  after  
24 reception of the XoffRequest message to acknowledge reception of the message.
- 25 ■ Transition to the Close State.

26 If the access network receives an XonRequest message, it shall send an XonResponse message within  
27 the time period specified by  $T_{FCResponse}$  after reception of the XonRequest message to acknowledge  
28 reception of the message.



### 3.3.5.5 Message formats

#### 3.3.5.5.1 RestartNetworkInterface

The access network sends this message to request the access terminal to restart the network interface.

Field	Length (bits)
MessageID	8
TransactionID	8

MessageID            The access network shall set this field to 0x0e.

TransactionID        The access network shall set this field according to 10.8.

<b>Channels</b>	FTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Unicast	<b>Security</b>	Required

#### 3.3.5.5.2 RestartNetworkInterfaceAck

The access terminal sends this message to acknowledge reception of a RestartNetworkInterface message.

Field	Length (bits)
MessageID	8
TransactionID	8

MessageID            The access terminal shall set this field to 0x0f.

TransactionID        The access terminal shall set this value to the value of the TransactionID field of the corresponding RestartNetworkInterface message.

<b>Channels</b>	RTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Unicast	<b>Security</b>	Required

### 3.3.5.6 In-band message formats

#### 3.3.5.6.1 XonRequest

The access terminal sends this message to request transition to the Open State.

Field	Length (bits)
MessageID	8
TransactionID	8

MessageID The access terminal shall set this field to 0xf5.

TransactionID The access terminal shall set this field according to 10.8.

#### 3.3.5.6.2 XonResponse

The access network sends this message to acknowledge reception of the XonRequest message.

Field	Length (bits)
MessageID	8
TransactionID	8

MessageID The access network shall set this field to 0xf6.

TransactionID The access network shall set this field to the value of the TransactionID field of the corresponding XonRequest message.

#### 3.3.5.6.3 XoffRequest

The access terminal sends this message to request transition to the Close State.

Field	Length (bits)
MessageID	8
TransactionID	8

MessageID The access terminal shall set this field to 0xf7.

TransactionID The access terminal shall set this field according to 10.8.

### 3.3.5.6.4 XoffResponse

The access network sends this message to acknowledge reception of the XoffRequest message.

Field	Length (bits)
MessageID	8
TransactionID	8

MessageID            The access network shall set this field to 0xf8.

TransactionID        The access network shall set this field to the value of the TransactionID field of the corresponding XoffRequest message.

### 3.3.5.7 Interface to other protocols

#### 3.3.5.7.1 Commands

This protocol does not issue any commands.

#### 3.3.5.7.2 Indications

This protocol does not register to receive any indications.

### 3.3.5.8 Protocol numeric constants

Constant	Meaning	Value
$T_{FCResponse}$	Time period within which the access terminal and access network are to respond to flow control messages.	200 ms

### 3.3.6 Configuration attributes for the default data transport

The access terminal shall support default values of all attributes.

Unless specified otherwise, the access terminal and the access network shall use the Generic Attribute Update Protocol in 10.9 to update configurable attributes belonging to the Default Data Transport.

#### 3.3.6.1 Simple attributes

The negotiable simple attribute for this protocol is listed in Table 11. The access terminal and the access network shall use as defaults the values in Table 11 that are listed in *bold italics*.

**Table 11 Configurable values**

Attribute ID	Attribute	Values	Meaning
0xffff	MaxAbortTimer	<b>0x01f4</b>	Maximum abort timer defaults to 500 ms.
		0x0000 to 0x2710	Maximum abort timer in units of ms.
		All other values	Reserved
0xfe0f	Flow0FAckNakEnableFwd	<b>0x0000</b>	RLP receivers associated with forward Link Flow 0x0F do not transmit ReceiverStatus messages.
		All other values	Reserved.
0xfeNN NN is the two-digit hexadecimal Link Flow number of the forward Link Flow in the range 0x00 to N <sub>LinkFlowMax</sub> -2 inclusive.	FlowNNAckNakEnableFwd NN is the two-digit hexadecimal Link Flow number in the range 0x00 to N <sub>LinkFlowMax</sub> -2 inclusive, where hexadecimal digits A through F are specified in upper case letters.	0x0000	RLP receivers associated with forward Link Flow NN do not transmit ReceiverStatus messages.
		<b>0x0001</b>	RLP receivers associated with forward Link Flow NN transmit a ReceiverStatus message when missing data units are detected.
		0x0002	RLP receivers associated with forward Link Flow NN transmit a ReceiverStatus message when missing data units are detected. RLP receivers associated with forward Link Flow NN send ReceiverStatus messages within AckTimer interval of receiving a data unit.

Attribute ID	Attribute	Values	Meaning
		0x0003	<p>RLP receivers associated with forward Link Flow <i>NN</i> transmit a ReceiverStatus message when missing data units are detected.</p> <p>RLP receivers associated with forward Link Flow <i>NN</i> send ReceiverStatus messages within AckTimer interval of receiving a data unit.</p> <p>The receivers are required to send a ReceiverStatus message immediately upon receiving an RLP packet carrying the last segment of a higher layer packet.</p>
		All other values	Reserved.
0xfd0f	Flow0FAckNakEnableRev	0x0000	RLP receivers associated with reverse Link Flow 0x0F do not transmit ReceiverStatus messages.
		All other values	Reserved.
0xfd <i>NN</i> <i>NN</i> is the two-digit hexadecimal Link Flow number of the reverse Link Flow in the range 0x00 to $N_{\text{LinkFlowMax}}-2$ inclusive.	Flow <i>NN</i> AckNakEnableRev <i>NN</i> is the two-digit hexadecimal Link Flow number in the range 0x00 to $N_{\text{LinkFlowMax}}-2$ inclusive, where hexadecimal digits A through F are specified in upper case letters.	0x0000	RLP receivers associated with reverse Link Flow <i>NN</i> do not transmit ReceiverStatus messages.
		0x0001	RLP receivers associated with reverse Link Flow <i>NN</i> transmit a ReceiverStatus message when missing data units are detected.
		0x0002	<p>RLP receivers associated with reverse Link Flow <i>NN</i> transmit a ReceiverStatus message when missing data units are detected.</p> <p>RLP receivers associated with reverse Link Flow <i>NN</i> send ReceiverStatus messages within AckTimer interval of receiving a data unit.</p>
		0x0003	<p>RLP receivers associated with reverse Link Flow <i>NN</i> transmit a ReceiverStatus message when missing data units are detected.</p> <p>RLP receivers associated with reverse Link Flow <i>NN</i> send ReceiverStatus messages within AckTimer interval of receiving a data unit.</p> <p>The receivers are required to send a ReceiverStatus message immediately upon receiving an RLP packet carrying the last segment of a higher layer packet.</p>
		All other values	Reserved.

Attribute ID	Attribute	Values	Meaning
0xfc0f	Flow0FFTCMACNakEnableFwd	0x0000	RLP is to ignore <i>ForwardTrafficChannelMAC.ForwardTrafficPacketsMissed</i> indication.
		All other values	Reserved.
0xfcNN NN is the two-digit hexadecimal Link Flow number of the forward Link Flow in the range 0x00 to N <sub>LinkFlowMax</sub> -2 inclusive.	FlowNNFTCMACNakEnableFwd NN is the two-digit hexadecimal Link Flow number in the range 0x00 to N <sub>LinkFlowMax</sub> -2 inclusive, where hexadecimal digits A through F are specified in upper case letters.	0x0001	RLP is to retransmit data units when a <i>ForwardTrafficChannelMAC.ForwardTrafficPacketsMissed</i> indication is received.
		0x0000	RLP is to ignore <i>ForwardTrafficChannelMAC.ForwardTrafficPacketsMissed</i> indication.
		All other values	Reserved
0xfb0f	Flow0FRTCMACNakEnableRev	0x0000	RLP is to ignore <i>ReverseTrafficChannelMAC.ReverseTrafficPacketsMissed</i> indication.
		All other values	Reserved.
0xfbNN NN is the two-digit hexadecimal Link Flow number of the reverse Link Flow in the range 0x00 to N <sub>LinkFlowMax</sub> -2 inclusive.	FlowNNRTCMACNakEnableRev NN is the two-digit hexadecimal Link Flow number in the range 0x00 to N <sub>LinkFlowMax</sub> -2 inclusive, where hexadecimal digits A through F are specified in upper case letters.	0x0001	RLP is to retransmit data units when a <i>ReverseTrafficChannelMAC.ReverseTrafficPacketsMissed</i> indication is received.
		0x0000	RLP is to ignore <i>ReverseTrafficChannelMAC.ReverseTrafficPacketsMissed</i> indication.
		All other values	Reserved
0xfaKK KK is the two-digit hexadecimal ReservationLabel.	ReservationKKIdleStateFwd KK is the two-digit hexadecimal ReservationLabel.	0x0000	Reservation does not change states when a Connection is closed.
		0x0001	Reservation transitions to the Close state when a Connection is closed.
		0x0002	Reservation transitions to the Open state when a Connection is opened and transitions to the Close state when a Connection is closed.
		All other values	Reserved
0xf9KK KK is the two-digit hexadecimal ReservationLabel.	ReservationKKIdleStateRev KK is the two-digit hexadecimal ReservationLabel.	0x0000	Reservation does not change states when a Connection is closed.
		0x0001	Reservation transitions to the Close state when a Connection is closed.
		0x0002	Reservation transitions to the Open state when a Connection is opened and transitions to the Close state when a Connection is closed.
		All other values	Reserved

Attribute ID	Attribute	Values	Meaning
0xf800	Flow00ActivatedFwd	<b>0x0001</b>	Forward Link Flow 0x00 is activated.
		0x0000	Forward Link Flow 0x00 is not activated.
		All other values	Reserved
0xf80f	Flow0FActivatedFwd	<b>0x0001</b>	Forward Link Flow 0x0F is activated.
		All other values	Reserved
0xf8NN NN is the two-digit hexadecimal Link Flow number of the forward Link Flow in the range 0x01 to N <sub>LinkFlowMax</sub> -2 inclusive.	FlowNNActivatedFwd NN is the two-digit hexadecimal Link Flow number in the range 0x00 to N <sub>LinkFlowMax</sub> -2 inclusive, where hexadecimal digits A through F are specified in upper case letters.	<b>0x0000</b>	Forward Link Flow NN is not activated.
		0x0001	Forward Link Flow NN is activated.
		All other values	Reserved
0xf700	Flow00ActivatedRev	<b>0x0001</b>	Reverse Link Flow 0x00 is activated.
		0x0000	Reverse Link Flow 0x00 is not activated.
		All other values	Reserved
0xf70f	Flow0FActivatedRev	<b>0x0001</b>	Reverse Link Flow 0x0F is activated.
		All other values	Reserved
0xf7NN NN is the two-digit hexadecimal Link Flow number of the reverse Link Flow in the range 0x01 to N <sub>LinkFlowMax</sub> -2 inclusive.	FlowNNActivatedRev NN is the two-digit hexadecimal Link Flow number in the range 0x00 to N <sub>LinkFlowMax</sub> -2 inclusive, where hexadecimal digits A through F are specified in upper case letters.	<b>0x0000</b>	Reverse Link Flow NN is not activated.
		0x0001	Reverse Link Flow NN is activated.
		All other values	Reserved
0xf60f	Flow0FSequenceLengthFwd	<b>0x0001</b>	Forward Link Flow 0x0F has a 8-bit sequence number.
		All other values	Reserved
0xf6NN NN is the two-digit hexadecimal Link Flow number of the forward Link Flow in the range 0x00 to N <sub>LinkFlowMax</sub> -2 inclusive.	FlowNNSequenceLengthFwd NN is the two-digit hexadecimal Link Flow number in the range 0x00 to N <sub>LinkFlowMax</sub> -2 inclusive, where hexadecimal digits A through F are specified in upper case letters.	0x0000	Forward Link Flow NN has a 0-bit sequence number.
		0x0001	Forward Link Flow NN has a 8-bit sequence number.
		0x0002	Forward Link Flow NN has a 16-bit sequence number.
		<b>0x0003</b>	Forward Link Flow NN has a 24-bit sequence number.
		All other values	Reserved

Attribute ID	Attribute	Values	Meaning
0xf50f	Flow0FSequenceLengthRev	<b>0x0001</b>	Reverse Link Flow 0x0F has a 8-bit sequence number.
		All other values	Reserved
0xf5NN NN is the two-digit hexadecimal Link Flow number of the reverse Link Flow in the range 0x00 to $N_{\text{LinkFlowMax}}-2$ inclusive.	FlowNNSequenceLengthRev NN is the two-digit hexadecimal Link Flow number in the range 0x00 to $N_{\text{LinkFlowMax}}-2$ inclusive, where hexadecimal digits A through F are specified in upper case letters.	0x0000	Reverse Link Flow NN has a 0-bit sequence number.
		0x0001	Reverse Link Flow NN has a 8-bit sequence number.
		0x0002	Reverse Link Flow NN has a 16-bit sequence number.
		<b>0x0003</b>	Reverse Link Flow NN has a 24-bit sequence number.
		All other values	Reserved
0xf40f	Flow0FDataUnitFwd	<b>0x0001</b>	Data unit for forward Link Flow 0x0F is RLP packet payload.
		All other values	Reserved
0xf4NN NN is the two-digit hexadecimal Link Flow number of the forward Link Flow in the range 0x00 to $N_{\text{LinkFlowMax}}-2$ inclusive.	FlowNNDataUnitFwd NN is the two-digit hexadecimal Link Flow number in the range 0x00 to $N_{\text{LinkFlowMax}}-2$ inclusive, where hexadecimal digits A through F are specified in upper case letters.	<b>0x0000</b>	Data unit for forward Link Flow NN is octets.
		0x0001	Data unit for forward Link Flow NN is RLP packet payload.
		All other values	Reserved.
0xf30f	Flow0FDataUnitRev	<b>0x0001</b>	Data unit for reverse Link Flow 0x0F is RLP packet payload.
		All other values	Reserved
0xf3NN NN is the two-digit hexadecimal Link Flow number of the reverse Link Flow in the range 0x00 to $N_{\text{LinkFlowMax}}-2$ inclusive.	FlowNNDataUnitRev NN is the two-digit hexadecimal Link Flow number in the range 0x00 to $N_{\text{LinkFlowMax}}-2$ inclusive, where hexadecimal digits A through F are specified in upper case letters.	<b>0x0000</b>	Data unit for reverse Link Flow NN is octets.
		0x0001	Data unit for reverse Link Flow NN is RLP packet payload.
		All other values	Reserved.



Attribute ID	Attribute	Values	Meaning
0xf2NN NN is the two-digit hexadecimal Link Flow number of the forward Link Flow in the range 0x00 to N <sub>LinkFlowMax</sub> -1 inclusive.	FlowNNSimultaneousDeliveryOnBothRoutesFwd NN is the two-digit hexadecimal Link Flow number in the range 0x00 to N <sub>LinkFlowMax</sub> -1 inclusive, where hexadecimal digits A through F are specified in upper case letters.	0x0000	Forward Link Flow NN delivers Flow Protocol payload in-order.
		0x0001	Forward Link Flow NN may deliver Flow Protocol payload out-of-order.
		All other values	Reserved.
0xf10f	Flow0FOutOfOrderDeliveryToRouteProtocolFwd	0x0001	Each Route of forward Link Flow 0x0F may deliver Route Protocol payload out-of-order.
		All other values	Reserved
0xf1NN NN is the two-digit hexadecimal Link Flow number of the forward Link Flow in the range 0x00 to N <sub>LinkFlowMax</sub> -2 inclusive.	FlowNNOOutOfOrderDeliveryToRouteProtocolFwd NN is the two-digit hexadecimal Link Flow number in the range 0x00 to N <sub>LinkFlowMax</sub> -2 inclusive, where hexadecimal digits A through F are specified in upper case letters.	0x0000	Each Route of forward Link Flow NN delivers Route Protocol payload in-order.
		0x0001	Each Route of forward Link Flow NN may deliver Route Protocol payload out-of-order.
		All other values	Reserved.
0xf00f	Flow0FOutOfOrderDeliveryToRouteProtocolRev	0x0001	Each Route of reverse Link Flow 0x0F may deliver Route Protocol payload out-of-order.
		All other values	Reserved
0xf0NN NN is the two-digit hexadecimal Link Flow number of the forward Link Flow in the range 0x00 to N <sub>LinkFlowMax</sub> -2 inclusive.	FlowNNOOutOfOrderDeliveryToRouteProtocolRev NN is the two-digit hexadecimal Link Flow number in the range 0x00 to N <sub>LinkFlowMax</sub> -2 inclusive, where hexadecimal digits A through F are specified in upper case letters.	0x0000	Each Route of reverse Link Flow NN delivers Route Protocol payload in-order.
		0x0001	Each Route of reverse Link Flow NN may deliver Route Protocol payload out-of-order.
		All other values	Reserved.
0xefNN NN is the two-digit hexadecimal Link Flow number of the reverse Link Flow in the range 0x00 to N <sub>LinkFlowMax</sub> -1 inclusive.	FlowNNRequestLevelRev NN is the two-digit hexadecimal Link Flow number in the range 0x00 to N <sub>LinkFlowMax</sub> -1 inclusive, where hexadecimal digits A through F are specified in upper case letters.	0x0002	QoSFlow field is set to '10' for signaling requests in the R-REQCH for the reverse Link Flow NN.
		0x0000, 0x0001, 0x0003	Value of the QoSFlow field for signaling requests in the R-REQCH for the reverse Link Flow NN.
		All other values	Reserved.

Attribute ID	Attribute	Values	Meaning
0xeeNN NN is the two-digit hexadecimal Link Flow number of the forward Link Flow in the range 0x00 to N <sub>LinkFlowMax</sub> -1 inclusive.	FlowNNTransmitAbortTimerFwd NN is the two-digit hexadecimal Link Flow number in the range 0x00 to N <sub>LinkFlowMax</sub> -1 inclusive, where hexadecimal digits A through F are specified in upper case letters.	0x0000	Maximum delay for transmission of a higher layer data unit for forward Link Flow NN is not specified.
		0x0001 to 0x1388	Maximum delay for transmission of a higher layer data unit for forward Link Flow NN in units of ms.
		All other values	Reserved.
0xedNN NN is the two-digit hexadecimal Link Flow number of the reverse Link Flow in the range 0x00 to N <sub>LinkFlowMax</sub> -1 inclusive.	FlowNNTransmitAbortTimerRev NN is the two-digit hexadecimal Link Flow number in the range 0x00 to N <sub>LinkFlowMax</sub> -1 inclusive, where hexadecimal digits A through F are specified in upper case letters.	0x0000	Maximum delay for transmission of a higher layer data unit for reverse Link Flow NN is not specified.
		0x0001 to 0x1388	Maximum delay for transmission of a higher layer data unit for reverse Link Flow NN in units of ms.
		All other values	Reserved.

1

### 2 3.3.6.2 Complex attributes

3 The following complex attributes and default values are defined (see 10.3 for attribute record  
4 definition).

#### 5 3.3.6.2.1 FlowNNTimersFwd attribute

6 NN is the two-digit hexadecimal Link Flow number of the forward Link Flow in the range 0x00 to  
7 N<sub>LinkFlowMax</sub>-1, inclusive, where hexadecimal digits A through F are specified in upper case letters.

8

Field	Length (bits)	Default
Length	8	N/A
AttributeID	16	N/A
AbortTimer	16	0x01f4
FlushTimer	16	0x012c
AckTimer	16	0x0064

9 Length Length of the complex attribute in octets. The sender shall set this field to the  
10 length of the complex attribute excluding the Length field.

11 AttributeID The sender shall set this field to 0x01NN, where NN is the two-digit  
12 hexadecimal Link Flow number in the range 0x00 to N<sub>LinkFlowMax</sub>-1, inclusive.

13 AbortTimer The sender shall set this field to the value of the RLP abort timer for this  
14 forward Link Flow in units of ms. The sender shall not set this field to a  
15 value greater than MaxAbortTimer.

1 FlushTimer The sender shall set this field to the value of the RLP flush timer for this  
 2 forward Link Flow in units of ms. The value of the RLP flush timer shall be  
 3 less than or equal to that of the corresponding abort timer.

4 AckTimer The sender shall set this field to the value of the RLP Ack timer for this  
 5 forward Link Flow in units of ms.

### 6 3.3.6.2.2 FlowNNTimersRev attribute

7 *NN* is the two-digit hexadecimal Link Flow number of the reverse Link Flow in the range 0x00 to  
 8  $N_{\text{LinkFlowMax}}-1$ , inclusive, where hexadecimal digits A through F are specified in upper case letters.

9

Field	Length (bits)	Default
Length	8	N/A
AttributeID	16	N/A
AbortTimer	16	0x01f4
FlushTimer	16	0x012c
AckTimer	16	0x0064

10 Length Length of the complex attribute in octets. The sender shall set this field to the  
 11 length of the complex attribute excluding the Length field.

12 AttributeID The sender shall set this field to 0x02*NN*, where *NN* is the two-digit  
 13 hexadecimal Link Flow number in the range 0x00 to  $N_{\text{LinkFlowMax}}-1$ , inclusive.

14 AbortTimer The sender shall set this field to the value of the RLP abort timer for this  
 15 reverse Link Flow in units of ms. The sender shall not set this field to a value  
 16 greater than MaxAbortTimer.

17 FlushTimer The sender shall set this field to the value of the RLP flush timer for this  
 18 reverse Link Flow in units of ms. The value of the RLP flush timer shall be  
 19 less than or equal to that of the corresponding abort timer.

20 AckTimer The sender shall set this field to the value of the RLP Ack timer for this  
 21 reverse Link Flow in units of ms.

### 3.3.6.2.3 FlowNNReservationFwd attribute

NN is the two-digit hexadecimal Link Flow number of the forward Link Flow in the range 0x00 to  $N_{\text{LinkFlowMax}}-2$ , inclusive, where hexadecimal digits A through F are specified in upper case letters.

Field	Length (bits)	Default for NN = 0x00	Default for NN > 0x00
Length	8	N/A	N/A
AttributeID	16	N/A	N/A
ReservationCount	8	0x01	0x00

ReservationCount occurrences of the following field:

Field	Length (bits)	Default for NN = 0x00	Default for NN > 0x00
ReservationLabel	8	0xff	N/A

**Length** Length of the complex attribute in octets. The sender shall set this field to the length of the complex attribute excluding the Length field.

**AttributeID** The sender shall set this field to 0x03NN, where NN is the two-digit hexadecimal Link Flow number in the range 0x00 to  $N_{\text{LinkFlowMax}}-2$  inclusive.

**ReservationCount** The sender shall set this field to the number of reservations associated with this Link Flow. The sender shall include ReservationCount occurrences of the following field with the message.

**ReservationLabel** The sender shall set this field to the ReservationLabel of the reservation associated with this Link Flow.

### 3.3.6.2.4 FlowNNReservationRev attribute

NN is the two-digit hexadecimal Link Flow number of the reverse Link Flow in the range 0x00 to  $N_{\text{LinkFlowMax}}-2$ , inclusive, where hexadecimal digits A through F are specified in upper case letters.

Field	Length (bits)	Default for NN = 0x00	Default for NN >= 0x00
Length	8	N/A	N/A
AttributeID	16	N/A	N/A
ReservationCount	8	0x01	0x00

ReservationCount occurrences of the following field:

Field	Length (bits)	Default for NN = 0x00	Default for NN >= 0x00
ReservationLabel	8	0xff	N/A

**Length** Length of the complex attribute in octets. The sender shall set this field to the length of the complex attribute excluding the Length field.

1	AttributeID	The sender shall set this field to 0x04 <i>NN</i> , where <i>NN</i> is the two-digit hexadecimal Link Flow number in the range 0x00 to $N_{\text{LinkFlowMax}}-2$ inclusive.
2		
3	ReservationCount	The sender shall set this field to the number of reservations associated with this Link Flow. The sender shall include ReservationCount occurrences of the following field with the message.
4		
5		
6	ReservationLabel	The sender shall set this field to the ReservationLabel of the reservation associated with this Link Flow
7		

### 8 3.3.6.2.5 ATSupportedFlowProtocolParameters*PP* attribute

9 *PP* is the two-digit hexadecimal ProtocolID number for the Flow Protocol according to Table 14,  
10 where hexadecimal digits A through F are specified in upper case letters.

11

Field	Length (bits)	Default
Length	8	N/A
AttributeID	16	N/A
ProtocolSupported	8	0x01 for <i>PP</i> = 0x00 and 0x01. 0x00 otherwise.
SupportedProtocolParametersValuesLength	8	0x00
SupportedProtocolParametersValues	SupportedProtocolParametersValuesLength × 8	N/A

12	Length	Length of the complex attribute in octets. The sender shall set this field to the length of the complex attribute excluding the Length field.
13		
14	AttributeID	The sender shall set this field to 0x05 <i>PP</i> .
15	ProtocolSupported	The sender shall set this field to 0x00 if the Flow Protocol <i>PP</i> is not supported. Otherwise, the sender shall set this field to 0x01 if the Flow Protocol <i>PP</i> is supported. All other values are reserved.
16		
17		
18	SupportedProtocolParametersValuesLength	The sender shall set this field to the length of the SupportedProtocolParametersValues record in units of octets. If the ProtocolSupported field is set to 0x00, the sender shall set this field to 0x00. If the ProtocolSupported field is set to 0x01, the sender shall set this field to 0x00 for Flow Protocol ProtocolID 0x00, 0x01 and 0x03. If the ProtocolSupported field is set to 0x01, the sender shall set this field according to 3.3.6.2.5.1 for Route Protocol ProtocolID 0x02.
19		
20		
21		
22		
23		
24		
25		
26	SupportedProtocolParametersValues	If ProtocolID is 0x02 and ProtocolSupported is 0x01, then the sender shall set this record as defined in 3.3.6.2.5.1. Otherwise, the sender shall omit this record.
27		
28		
29		

### 3.3.6.2.5.1 Definition of SupportedProtocolParametersValues record when the Flow Protocol or Route Protocol is ROHC

Field	Length (bits)
MaxSupportedMaxCID	16
LargeCIDSupported	1
MaxSupportedMRRU	16
MaxSupportedDelayedDecompressionDepth	8
TimerBasedCompressionSupported	1
SupportedProfileCount	8

SupportedProfileCount occurrences of the following field:

{	
SupportedProfile	16
}	
Reserved	0 - 7 (as needed)

#### MaxSupportedMaxCID

The sender shall set this field to the maximum MAX\_CID parameter supported.

#### LargeCIDSupported

The sender shall set this field to '0' if large CID representation is not supported according to [11]. Otherwise, the sender shall set this field to '1' if large CID representation is supported.

#### MaxSupportedMRRU

The sender shall set this field to the MRRU supported by the decompressor according to [11]. Default value is 0x0000 (no segmentation).

#### MaxSupportedDelayedDecompressionDepth

The sender shall set this field to the maximum supported value of delayed decompression depth at the access terminal ROHC de-compressor.

#### TimerBasedCompressionSupported

The sender shall set this field to '1' if the compressor at the access terminal supports timer based compression mode. Otherwise, the sender shall set this field to '0'.

#### SupportedProfileCount

The sender shall set this field to the number of ROHC profiles supported. The sender shall include SupportedProfileCount occurrences of the following field with the message.

#### SupportedProfile

The sender shall set this field to the ROHC profile supported by the compressor and decompressor. IANA ROHC profile identifier definitions can be found at [15].

1 Reserved                    The sender shall add reserved bits to make the length of the entire record an  
 2 integer number of octets. The sender shall set these bits to '0'. The receiver  
 3 shall ignore this field.

#### 4 **3.3.6.2.6 ATSupportedRouteProtocolParametersPP attribute**

5 *PP* is the two-digit hexadecimal ProtocolID number for the Route Protocol according to Table 15,  
 6 where hexadecimal digits A through F are specified in upper case letters.

Field	Length (bits)	Default
Length	8	N/A
AttributeID	16	N/A
ProtocolSupported	8	0x01 for <i>PP</i> = 0x00. 0x00 otherwise.
SupportedProtocolParametersValuesLength	8	0x00
SupportedProtocolParametersValues	SupportedProtocolPar ametersValuesLength × 8	N/A

8 Length                    Length of the complex attribute in octets. The sender shall set this field to the  
 9 length of the complex attribute excluding the Length field.

10 AttributeID              The sender shall set this field to 0x06*PP*.

11 ProtocolSupported      The sender shall set this field to 0x00 if the Route Protocol *PP* is not  
 12 supported. Otherwise, the sender shall set this field to 0x01 if the Route  
 13 Protocol *PP* is supported. All other values are reserved

14 SupportedProtocolParametersValuesLength  
 15                              The sender shall set this field to the length of the  
 16 SupportedProtocolParametersValues record in units of octets. If the  
 17 ProtocolSupported field is set to 0x00, the sender shall set this field to 0x00.  
 18 If the ProtocolSupported field is set to 0x01, the sender shall set this field to  
 19 0x00 for Flow Protocol ProtocolID 0x00. If the ProtocolSupported field is set  
 20 to 0x01, the sender shall set this field according to 3.3.6.2.5.1 for Route  
 21 Protocol ProtocolID 0x02.

22 SupportedProtocolParametersValues  
 23                              If ProtocolID is 0x02 and ProtocolSupported is 0x01, then the sender shall  
 24 set this record as defined in 3.3.6.2.5.1. Otherwise, the sender shall omit this  
 25 record.

### 3.3.6.2.7 ATSupportedQoSProfiles attribute

Field	Length (bits)	Default
Length	8	N/A
AttributeID	16	N/A
QoSProfileCount	8	0x00

QoSProfileCount occurrences of the following record:

{		
FlowProfileID	FlowProfileIDType dependent	N/A
}		

**Length** Length of the complex attribute in octets. The sender shall set this field to the length of the complex attribute excluding the Length field.

**AttributeID** The sender shall set this field to 0x0700.

**QoSProfileCount** The sender shall set this field to the number of QoS profiles that are included in this message.

**FlowProfileID** The sender shall set this field to the profile according to 11.6.

### 3.3.6.2.8 ANSupportedQoSProfiles attribute

Field	Length (bits)	Default
Length	8	N/A
AttributeID	16	N/A
QoSProfileCount	8	0x00

QoSProfileCount occurrences of the following record:

{		
FlowProfileID	FlowProfileIDType dependent	N/A
}		

**Length** Length of the complex attribute in octets. The sender shall set this field to the length of the complex attribute excluding the Length field.

**AttributeID** The sender shall set this field to 0x0800.

**QoSProfileCount** The sender shall set this field to the number of QoS profiles that are included in this message.

**FlowProfileID** The sender shall set this field to the profile according to 11.6.



### 3.3.6.2.9 ReservationKKQoSListFwd attribute

KK is the two-digit hexadecimal ReservationLabel, where hexadecimal digits A through F are specified in upper case letters.

Field	Length (bits)	Default
Length	8	N/A
AttributeID	16	N/A
ReservationPriority	8	0xff
QoSAttributeSetCount	8	0x00

QoSAttributeSetCount occurrences of the following record:

{		
QoSAttributeSetLength	8	N/A
QoSAttributeSet	QoSAttributeSetLength × 8	N/A
}		

**Length** Length of the complex attribute in octets. The sender shall set this field to the length of the complex attribute excluding the Length field.

**AttributeID** The sender shall set this field to 0x09KK, where KK is the two-digit hexadecimal ReservationLabel.

**ReservationPriority** The sender shall set this field to indicate the priority to be assigned to the reservation. The value 0x00 indicates the highest priority and the value 0xff indicates the lowest priority.<sup>14</sup>

**QoSAttributeSetCount** The sender shall set this field to the number of QoSAttributeSets associated with this reservation. Each QoSAttributeSet contains one set of acceptable QoS parameters. If multiple QoS attribute sets are included, the sender shall include the QoS attribute sets in descending order of preference. The sender shall include QoSAttributeSetCount occurrences of the following two fields with the message.

**QoSAttributeSetLength** The sender shall set this field to the length, in octets, of the QoSAttributeSet.

**QoSAttributeSet** The QoS parameters requested for the reservation. The sender shall set this record as defined in 3.3.6.2.9.1. If the QoSAttributeSet specifies a FlowProfileID, then the sender shall set this field to a FlowProfileID that is included in the ANSupportedQoSProfiles attribute.

<sup>14</sup> The ReservationPriority field may be used by the access terminal to indicate the relative importance of reservations for purposes such as admission control.

### 3.3.6.2.9.1 Definition of QoSAttributeSet

Field	Length (bits)
QoSAttributeSet_ID	8
Reserved	7
Verbose	1
FlowProfileID	0 or FlowProfileIDType dependent
Traffic_Class	0 or 8
Peak_Rate	0 or 32
Bucket_Size	0 or 32
Token_Rate	0 or 32
Max_Latency	0 or 16
Max_Packet_Loss_Rate	0 or 8
Packet_Size	0 or 16
Max_Jitter	0 or 16

- 3 QoSAttributeSet\_ID The sender shall set this field to an identifier for the QoSAttributeSet. The  
4 sender shall not set this field to 0x00.
- 5 Reserved The sender shall set this field to '0000000'. The receiver shall ignore this  
6 field.
- 7 Verbose If the ProfileID field is included, the sender shall set this field to '1'.  
8 Otherwise, the sender shall set this field to '0'.
- 9 FlowProfileID If Verbose is '1', then the sender shall omit this field. Otherwise, the sender  
10 shall set this field to the profile according to 11.6.
- 11 Traffic\_Class If Verbose is '0', then the sender shall omit this field. Otherwise, the sender  
12 shall include this field to indicate the traffic class as specified in Table 12.

**Table 12 Traffic Class**

<b>Value</b>	<b>Description</b>
'000'	Unknown
'001'	Conversational <sup>15</sup>
'010'	Streaming <sup>16</sup>
'011'	Interactive <sup>17</sup>
'100'	Background <sup>18</sup>
'101'-'111'	Reserved

1

2	<b>Peak_Rate</b>	This field specifies the peak traffic rate as specified in [12], [13], and [14] in units of bytes per second. If Verbose is '0', then the sender shall omit this field. Otherwise, the sender shall set this field to the unsigned value n (range from 1 to 4294967295), where the peak rate = n bytes per second. If the sender doesn't want to specify the peak rate, the sender shall set this field to zero.
3		
4		
5		
6		
7		
8	<b>Bucket_Size</b>	This field specifies the token bucket size as specified in [12], [13], and [14] in units of bytes. If Verbose is '0', then the sender shall omit this field. Otherwise, the sender shall set this field to the unsigned value n (range from 1 to 4294967295), where the bucket size = n bytes. If the sender doesn't want to specify the token bucket size, the sender shall set this field to zero.
9		
10		
11		
12		
13	<b>Token_Rate</b>	This field specifies the token rate as specified in [12], [13], and [14] in units of bytes per second. If Verbose is '0', then the sender shall omit this field. Otherwise, the sender shall set this field to the unsigned value n (range from 1 to 4294967295), where the token rate = n bytes per second. If the sender doesn't want to specify the token rate, the sender shall set this field to zero.
14		
15		
16		
17		
18	<b>Max_Latency</b>	The maximum latency in units of milliseconds. The maximum latency specifies the maximum acceptable delay from the time that an octet of user data is submitted to the transmitter until the receiver receives the octet. It is measured between the sender and the receiver. If Verbose is '0', then the sender shall omit this field. Otherwise, the sender shall set this field to the unsigned value n (range from 1 to 65535), where the maximum latency = n milliseconds. If the sender doesn't want to specify the maximum latency, the sender shall set this field to zero.
19		
20		
21		
22		
23		
24		
25		

---

<sup>15</sup> Conversational traffic class has a low latency, medium error rate service requirement

<sup>16</sup> Streaming traffic class has a high latency, medium error rate service requirement

<sup>17</sup> Interactive traffic class has a low latency, low error rate service requirement

<sup>18</sup> Background traffic class has a high latency, low error rate service requirement

- 1 Max\_Packet\_Loss\_Rate  
 2 This field indicates the maximum higher layer packet loss rate. If Verbose is  
 3 '0', then the sender shall omit this field. Otherwise, the sender shall set this  
 4 field to an unsigned value n (range from 1 to 31), where the maximum packet  
 5 loss rate =  $10^{(-n/4)}$ . If the sender doesn't want to specify the maximum loss  
 6 rate, the sender shall set this field to zero. When Max\_Packet\_loss\_Rate is  
 7 used the Packet\_Size shall also be specified.
- 8 Packet\_Size: This field indicates the median packet size, in units of bytes. If Verbose is  
 9 '0', then the sender shall omit this field. Otherwise, the sender shall set this  
 10 field to the unsigned value n (range from 1 to 65535), where the median  
 11 packet size = n bytes. If the sender doesn't want to specify the median packet  
 12 size, the sender shall set this field to zero.
- 13 Max\_Jitter The maximum jitter in units of milliseconds. The maximum jitter specifies  
 14 the maximum acceptable latency variation from the time that a packet is  
 15 received until the time that the next packet is received as measured at the  
 16 receiver. If Verbose is '0', then the sender shall omit this field. Otherwise,  
 17 the sender shall set this field to the unsigned value n (range from 1 to 65535),  
 18 where the maximum jitter = n milliseconds. The sender shall set this field to  
 19 zero to indicate the traffic flow sensitivity to variation in delay is not  
 20 specified.

### 21 3.3.6.2.10 ReservationKKQoSListRev attribute

22 KK is the two-digit hexadecimal ReservationLabel, where hexadecimal digits A through F are  
 23 specified in upper case letters.

Field	Length (bits)	Default
Length	8	N/A
AttributeID	16	N/A
ReservationPriority	8	<i>0xff</i>
QoSAttributeSetCount	8	<i>0x00</i>

QoSAttributeSetCount occurrences of the following record:

{		
QoSAttributeSetLength	8	N/A
QoSAttributeSet	QoSAttributeSetLength × 8	N/A
}		

- 25 Length Length of the complex attribute in octets. The sender shall set this field to the  
 26 length of the complex attribute excluding the Length field.
- 27 AttributeID The sender shall set this field to 0x0aKK, where KK is the two-digit  
 28 hexadecimal ReservationLabel.
- 29 ReservationPriority The sender shall set this field to indicate the priority to be assigned to the  
 30 reservation. The value 0x00 indicates the highest priority and the value 0xff  
 31 indicates the lowest priority.

1 QoSAttributeSetCount The sender shall set this field to the number of QoSAttributeSets associated  
 2 with this reservation. Each QoSAttributeSet contains one set of acceptable  
 3 QoS parameters. If multiple QoS attribute sets are included, the sender shall  
 4 include the QoS attribute sets in descending order of preference. The sender  
 5 shall include QoSAttributeSetCount occurrences of the following two fields  
 6 with the message.

7 QoSAttributeSetLength  
 8 The sender shall set this field to the length, in octets, of the QoSAttributeSet.

9 QoSAttributeSet The QoS parameters requested for the reservation. The sender shall set this  
 10 record as defined in 3.3.6.2.9.1. If the QoSAttributeSet specifies a  
 11 FlowProfileID, then the sender shall set this field to a FlowProfileID that is  
 12 included in the ATSupportedQoSProfiles attribute.

### 13 3.3.6.2.11 ReservationKKQoSUsedFwd attribute

14 *KK* is the two-digit hexadecimal ReservationLabel, where hexadecimal digits A through F are  
 15 specified in upper case letters.

16

Field	Length (bits)	Default
Length	8	N/A
AttributeID	16	N/A
QoSAttributeSet_ID	8	0x00

17 Length Length of the complex attribute in octets. The sender shall set this field to the  
 18 length of the complex attribute excluding the Length field.

19 AttributeID The sender shall set this field to 0x0b*KK*, where *KK* is the two-digit  
 20 hexadecimal ReservationLabel.

21 QoSAttributeSet\_ID The sender may set this field to the identifier assigned by the corresponding  
 22 ReservationKKQoSListFwd message of the QoSAttributeSet that has been  
 23 granted; or the sender may set this field to 0x00 to indicate that requested  
 24 QoSAttributeSet is invalid.

### 25 3.3.6.2.12 ReservationKKQoSUsedRev attribute

26 *KK* is the two-digit hexadecimal ReservationLabel, where hexadecimal digits A through F are  
 27 specified in upper case letters.

28

Field	Length (bits)	Default
Length	8	N/A
AttributeID	16	N/A
QoSAttributeSet_ID	8	0x00

- 1 Length Length of the complex attribute in octets. The sender shall set this field to the  
2 length of the complex attribute excluding the Length field.
- 3 AttributeID The sender shall set this field to 0x0c*KK*, where *KK* is the two-digit  
4 hexadecimal ReservationLabel.
- 5 QoSAttributeSet\_ID The sender may set this field to the identifier assigned by the corresponding  
6 Reservation*KK*QoSListRev message of the QoSAttributeSet that has been  
7 granted; or the sender may set this field to 0x00 to indicate that requested  
8 QoSAttributeSet is invalid.

### 9 3.3.6.2.13 Reservation*KK*PacketFilterFwd attribute

10 *KK* is the two-digit hexadecimal ReservationLabel, where hexadecimal digits A through F are  
11 specified in upper case letters.

Field	Length (bits)	Default
Length	8	N/A
AttributeID	16	N/A
FilterPrecedence	16	0xff
FilterSpecCount	8	0x00

FilterSpecCount occurrences of the following record:

```
{
```

FilterSpecType	8	N/A
FilterSpecLength	8	N/A
FilterSpec	FilterSpecLength × 8	N/A

```
}
```

- 13 Length Length of the complex attribute in octets. The sender shall set this field to the  
14 length of the complex attribute excluding the Length field.
- 15 AttributeID The sender shall set this field to 0x0d*KK*, where *KK* is the two-digit  
16 hexadecimal ReservationLabel.
- 17 FilterPrecedence The sender shall set this field to indicate the precedence of the packet filter  
18 for Reservation*KK*PacketFilterFwd among all packet filters defined by the  
19 Reservation*KK*PacketFilterFwd attributes of all active forward Reservations  
20 associated with the access terminal. The evaluation precedence index is in  
21 the range of 0x0000 to 0xffff. The higher the value of the FilterPrecedence  
22 field, the lower the precedence of that packet filter. If a given packet matches  
23 more than one of the currently active packet filters, the packet is mapped to  
24 the Reservation corresponding to the packet filter of highest precedence. A  
25 given precedence level may be used only once per access terminal, except  
26 0xffff which is used as an indication of no precedence.
- 27 FilterSpecCount The sender shall set this field to the number of FilterSpecs associated with  
28 this reservation. The sender shall include FilterSpecCount occurrences of the  
29 following three fields with the message.

1 FilterSpecType The sender shall set this field to an identifier for the Filter Specification Type  
2 according to Table 13.

3 **Table 13 FilterSpecType for Packet Filter**

Value	FilterSpecType
0x00	IP version 4 [4]
0x01	IP version 6 [5]
0x02	Match all
All other values	Reserved

4 FilterSpecLength The sender shall set this field to the length of the FilterSpec field in units of  
5 octets.

6 FilterSpec If FilterSpecType is 0x00, then the sender shall set this record as defined in  
7 3.3.6.2.13.1. If FilterSpecType is 0x01, then the sender shall set this record  
8 as defined in 3.3.6.2.13.2. Otherwise, the sender shall omit this record.

9 **3.3.6.2.13.1 Definition of FilterSpec record for IPv4**

10

Field	Length (bits)
IPv4_Source_Address_Included	1
IPv4_Destination_Address_Included	1
Source_Port_Range_Included	1
Destination_Port_Range_Included	1
Packet_Length_Included	1
Protocol_Type_Included	1
Type_of_Service_Included	1
IPSec_SPI_Included	1
Protocol_Type	0 or 8
IPv4_Source_Address_Prefix_Length	0 or 8
IPv4_Destination_Address_Prefix_Length	0 or 8
IPv4_Source_Address	0 or 32
IPv4_Destination_Address	0 or 32
Source_Port_Lower	0 or 16
Source_Port_Upper	0 or 16
Destination_Port_Lower	0 or 16
Destination_Port_Upper	0 or 16
Packet_Length_Lower	0 or 16
Packet_Length_Upper	0 or 16
IPSec_SPI	0 or 32
Type_of_Service	0 or 8
Type_of_Service_Mask	0 or 8

1	IPv4_Source_Address_Included	
2		The sender shall set this field to '1' to match the value of the Source Address
3		field in the IP packet. Otherwise, the sender shall set this field to '0'.
4	IPv4_Destination_Address_Included	
5		The sender shall set this field to '1' to match the value of the Destination
6		Address field in the IP packet. Otherwise, the sender shall set this field to '0'.
7	Source_Port_Range_Included	
8		The sender shall set this field to '1' to match a range of Source Port numbers
9		in the IP packet. Otherwise, the sender shall set this field to '0'.
10	Destination_Port_Range_Included	
11		The sender shall set this field to '1' to match a range of Destination Port
12		numbers in the IP packet. Otherwise, the sender shall set this field to '0'.
13	Packet_Length_Included	
14		The sender shall set this field to '1' to match a range of IP packet lengths.
15		Otherwise, the sender shall set this field to '0'.
16	Protocol_Type_Included	
17		The sender shall set this field to '1' to match the value of the Protocol field in
18		the IP packet. Otherwise, the sender shall set this field to '0'.
19	Type_of_Service_Included	
20		The sender shall set this field to '1' to match the value of the Type of Service
21		field in the IP packet. Otherwise, the sender shall set this field to '0'.
22	IPSec_SPI_Included	
23		The sender shall set this field to '1' to match the value of the IPSec Security
24		Parameter Index (SPI) field in the IP packet. Otherwise, the sender shall set
		this field to '0'.
25	Protocol_Type	
26		If Protocol_Type_Included is '0', then the sender shall omit this field.
27		Otherwise, the sender shall set this field to the value of the Protocol field to
28		match in the IP packet. The sender shall set this field in the range from 0x00
		to 0xff.
29	IPv4_Source_Address_Prefix_Length	
30		The IPv4_Source_Address up to the IPv4_Source_Address_Prefix_Length is
31		matched against the Source Address in the IP packet. If
32		IPv4_Source_Address_Included is '0', then the sender shall omit this field.
33		Otherwise, the sender shall set this field in the range from 0x01 to 0x10.
34	IPv4_Destination_Address_Prefix_Length	
35		The IPv4_Destination_Address up to the
36		IPv4_Destination_Address_Prefix_Length is matched against the Destination
37		Address in the IP packet. If IPv4_Destination_Address_Included is '0', then
38		the sender shall omit this field. Otherwise, the sender shall set this field in the
39		range from 0x01 to 0x10.



1	IPv4_Source_Address	If IPv4_Source_Address_Included is '0', then the sender shall omit this field.
2		Otherwise, the sender shall set this field to the value of the Source Address
3		field to match in the IP packet.
4	IPv4_Destination_Address	
5		If IPv4_Destination_Address_Included is '0', then the sender shall omit this
6		field. Otherwise, the sender shall set this field to the value of the Destination
7		Address field to match in the IP packet.
8	Source_Port_Lower	If Source_Port_Range_Included is '0', then the sender shall omit this field.
9		Otherwise, the sender shall set this field to the lowest value of the Source
10		Port Number to match in the IP packet. The sender shall set this field in the
11		range from 0x0000 to 0xffff.
12	Source_Port_Upper	If Source_Port_Range_Included is '0', then the sender shall omit this field.
13		Otherwise, the sender shall set this field to the highest value of the Source
14		Port Number to match in the IP packet. The sender shall set this field in the
15		range from Source_Port_Lower to 0xffff.
16	Destination_Port_Lower	
17		If Destination_Port_Range_Included is '0', then the sender shall omit this
18		field. Otherwise, the sender shall set this field to the lowest value of the
19		Destination Port Number to match in the IP packet. The sender shall set this
20		field in the range from 0x0000 to 0xffff.
21	Destination_Port_Upper	
22		If Destination_Port_Range_Included is '0', then the sender shall omit this
23		field. Otherwise, the sender shall set this field to the highest value of the
24		Destination Port Number to match in the IP packet. The sender shall set this
25		field in the range from Destination_Port_Lower to 0xffff.
26	Packet_Length_Lower	If Packet_Length_Included is '0', then the sender shall omit this field.
27		Otherwise, the sender shall set this field to the shortest packet length IP
28		packet to match. The sender shall set this field in the range from 0x0000 to
29		0xffff.
30	Packet_Length_Upper	If Packet_Length_Included is '0', then the sender shall omit this field.
31		Otherwise, the sender shall set this field to the highest packet length IP
32		packet to match. The sender shall set this field in the range from
33		Packet_Length_Lower to 0xffff.
34	IPSec_SPI	If IPSec_SPI_Included is '0', then the sender shall omit this field. Otherwise,
35		the sender shall set this field to the value of the IPSec Security Parameter
36		Index (SPI) to match in the IP packet.
37	Type_of_Service	If Type_of_Service_Included is '0', then the sender shall omit this field.
38		Otherwise, the sender shall set this field to the value of the Type of Service
39		field to match in the IP packet.

1 Type\_of\_Service\_Mask

2 If Type\_of\_Service\_Included is '0', then the sender shall omit this field.  
 3 Otherwise, the sender shall set this field to the bits of the Type\_of\_Service  
 4 field to match against the actual value of the corresponding field in the IP  
 5 packets. The mask contains ones in the bit positions to be used in the  
 6 matching operation.

7 **3.3.6.2.13.2 Definition of FilterSpec record for IPv6**

8

Field	Length (bits)
IPv6_Source_Address_Included	1
IPv6_Destination_Address_Included	1
Source_Port_Range_Included	1
Destination_Port_Range_Included	1
Packet_Length_Included	1
Traffic_Class_Included	1
Flow_Label_Included	1
IPSec_SPI_Included	1
IPv6_Source_Address_Prefix_Length	0 or 8
IPv6_Destination_Address_Prefix_Length	0 or 8
IPv6_Source_Address	0 or 128
IPv6_Destination_Address	0 or 128
Source_Port_Lower	0 or 16
Source_Port_Upper	0 or 16
Destination_Port_Lower	0 or 16
Destination_Port_Upper	0 or 16
Packet_Length_Lower	0 or 16
Packet_Length_Upper	0 or 16
IPSec_SPI	0 or 32
Traffic_Class	0 or 8
Traffic_Class_Mask	0 or 8
Reserved	0 or 4
Flow_Label	0 or 20

9 IPv6\_Source\_Address\_Included

10 The sender shall set this field to '1' to match the value of the Source Address  
 11 field in the IP packet. Otherwise, the sender shall set this field to '0'.

12 IPv6\_Destination\_Address\_Included

13 The sender shall set this field to '1' to match the value of the Destination  
 14 Address field in the IP packet. Otherwise, the sender shall set this field to '0'.

1	Source_Port_Range_Included	
2		The sender shall set this field to '1' to match a range of Source Port numbers
3		in the IP packet. Otherwise, the sender shall set this field to '0'.
4	Destination_Port_Range_Included	
5		The sender shall set this field to '1' to match a range of Destination Port
6		numbers in the IP packet. Otherwise, the sender shall set this field to '0'.
7	Packet_Length_Included	
8		The sender shall set this field to '1' to match a range of IP packet lengths.
9		Otherwise, the sender shall set this field to '0'.
10	Traffic_Class_Included	
11		The sender shall set this field to '1' to match the value of the Traffic Class
12		field in the IP packet. Otherwise, the sender shall set this field to '0'.
13	Flow_Label_Included	
14		The sender shall set this field to '1' to match the value of the Flow Label
		field in the IP packet. Otherwise, the sender shall set this field to '0'.
15	IPSec_SPI_Included	
16		The sender shall set this field to '1' to match the value of the IPSec Security
17		Parameter Index (SPI) field in the IP packet. Otherwise, the sender shall set
		this field to '0'.
18	IPv6_Source_Address_Prefix_Length	
19		If IPv6_Source_Address_Included is '0', then the sender shall omit this field.
20		Otherwise, the IPv6_Source_Address up to the
21		IPv6_Source_Address_Prefix_Length is matched against the Source Address
22		in the IP packet. The sender shall set this field in the range from 0x01 to
23		0x80.
24	IPv6_Destination_Address_Prefix_Length	
25		If IPv6_Destination_Address_Included is '0', then the sender shall omit this
26		field. Otherwise, the IPv6_Destination_Address up to the
27		IPv6_Destination_Address_Prefix_Length is matched against the Destination
28		Address in the IP packet. The sender shall set this field in the range from
29		0x01 to 0x80.
30	IPv6_Source_Address	
31		If IPv6_Source_Address_Included is '0', then the sender shall omit this
32		field. Otherwise, the sender shall set this field to the value of the Source
		Address field to match in the IP packet.
33	IPv6_Destination_Address	
34		If IPv6_Destination_Address_Included is '0', then the sender shall omit this
35		field. Otherwise, the sender shall set this field to the value of the Destination
36		Address field to match in the IP packet.
37	Source_Port_Lower	
38		If Source_Port_Range_Included is '0', then the sender shall omit this field.
39		Otherwise, the sender shall set this field to the lowest value of the Source
40		Port Number to match in the IP packet. The sender shall set this field in the
		range from 0x0000 to 0xffff.

1	Source_Port_Upper	If Source_Port_Range_Included is '0', then the sender shall omit this field.
2		Otherwise, the sender shall set this field to the highest value of the Source
3		Port Number to match in the IP packet. The sender shall set this field in the
4		range from Source_Port_Lower to 0xffff.
5	Destination_Port_Lower	
6		If Destination_Port_Range_Included is '0', then the sender shall omit this
7		field. Otherwise, the sender shall set this field to the lowest value of the
8		Destination Port Number to match in the IP packet. The sender shall set this
9		field in the range from 0x0000 to 0xffff.
10	Destination_Port_Upper	
11		If Destination_Port_Range_Included is '0', then the sender shall omit this
12		field. Otherwise, the sender shall set this field to the highest value of the
13		Destination Port Number to match in the IP packet. The sender shall set this
14		field in the range from Destination_Port_Lower to 0xffff.
15	Packet_Length_Lower	If Packet_Length_Included is '0', then the sender shall omit this field.
16		Otherwise, the sender shall set this field to the shortest packet length IP
17		packet to match. The sender shall set this field in the range from 0x0000 to
18		0xffff.
19	Packet_Length_Upper	If Packet_Length_Included is '0', then the sender shall omit this field.
20		Otherwise, the sender shall set this field to the highest packet length IP
21		packet to match. The sender shall set this field in the range from
22		Packet_Length_Lower to 0xffff.
23	IPSec_SPI	If IPSec_SPI_Included is '0', then the sender shall omit this field. Otherwise,
24		the sender shall set this field to the value of the IPSec Security Parameter
25		Index (SPI) to match in the IP packet.
26	Traffic_Class	If Traffic_Class_Included is '0', then the sender shall omit this field.
27		Otherwise, the sender shall set this field to the value of the Traffic Class field
28		to match in the IP packet. The sender shall set this field in the range from
29		0x00 to 0xff.
30	Traffic_Class_Mask	If Traffic_Class_Included is '0', then the sender shall omit this field.
31		Otherwise, the sender shall set this field to the bits of the Traffic_Class field
32		to match against the actual value of the corresponding field in the IP packets.
33		The mask contains ones in the bit positions to be used in the matching
34		operation.
35	Reserved	If Flow_Label_Included is '0', then the sender shall omit this field.
36		Otherwise, the sender shall set this field to '0000'. The receiver shall ignore
37		this field.
38	Flow_Label	If Flow_Label_Included is '0', then the sender shall omit this field.
39		Otherwise, the sender shall set this field to the value of the Flow Label field
40		to match in the IP packet.

### 3.3.6.2.14 ReservationKKPacketFilterRev attribute

KK is the two-digit hexadecimal ReservationLabel, where hexadecimal digits A through F are specified in upper case letters.

Field	Length (bits)	Default
Length	8	N/A
AttributeID	16	N/A
FilterPrecedence	16	0xff
FilterSpecCount	8	0x00

FilterSpecCount occurrences of the following record:

Field	Length (bits)	Default
FilterSpecType	8	N/A
FilterSpecLength	8	N/A
FilterSpec	FilterSpecLength × 8	N/A

**Length** Length of the complex attribute in octets. The sender shall set this field to the length of the complex attribute excluding the Length field.

**AttributeID** The sender shall set this field to 0x0eKK, where KK is the two-digit hexadecimal ReservationLabel.

**FilterPrecedence** The sender shall set this field to indicate the precedence of the packet filter for ReservationKKPacketFilterRev among all packet filters defined by the ReservationKKPacketFilterRev attributes of all active reverse Reservations associated with the access terminal. The evaluation precedence index is in the range of 0x0000 to 0xffff. The higher the value of the FilterPrecedence field, the lower the precedence of that packet filter. If a given packet matches more than one of the currently active packet filters, the packet is mapped to the Reservation corresponding to the packet filter of highest precedence. A given precedence level may be used only once per access terminal, except 0xffff which is used as an indication of no precedence.

**FilterSpecCount** The sender shall set this field to the number of FilterSpecs associated with this reservation. The sender shall include FilterSpecCount occurrences of the following three fields with the message.

**FilterSpecType** The sender shall set this field to an identifier for the Filter Specification Type according to Table 13.

**FilterSpecLength** The sender shall set this field to the length of the FilterSpec field in units of octets.

**FilterSpec** If FilterSpecType is 0x00, then the sender shall set this record as defined in 3.3.6.2.13.1. If FilterSpecType is 0x01, then the sender shall set this record as defined in 3.3.6.2.13.2. Otherwise, the sender shall omit this record.

### 3.3.6.2.15 FlowNNFlowProtocolParametersFwd attribute

NN is the two-digit hexadecimal forward Link Flow identifier, where hexadecimal digits A through F are specified in upper case letters.

Field	Length (bits)	Default
Length	8	N/A
AttributeID	16	N/A
ProtocolID	8	<b>0x01</b>
ProtocolParametersLength	8	<b>0x00</b>
ProtocolParameters	ProtocolParametersLength × 8	N/A

**Length** Length of the complex attribute in octets. The sender shall set this field to the length of the complex attribute excluding the Length field.

**AttributeID** The sender shall set this field to 0x0fNN, where NN is the two-digit hexadecimal forward Link Flow number.

**ProtocolID** The sender shall set this field to an identifier for the Flow Protocol according to Table 14.

**Table 14 ProtocolID for Flow Protocol**

Value	Protocol
0x00	NULL
0x01	Internet Protocol (IP) version 4 [4] and version 6 [5]
0x02	Robust Header Compression (ROHC) [11]
0x03	EAP encapsulation over Layer 2 [22]
0x04	IEEE 802.3 / DIX Ethernet [17]
All other values	Reserved

**ProtocolParametersLength** The sender shall set this field to the length of the ProtocolParameters field in units of octets.

**ProtocolParameters** If ProtocolID is 0x02, then the sender shall set this record as defined in 3.3.6.2.15.1. Otherwise, the sender shall omit this record.

### 3.3.6.2.15.1 Definition of ProtocolParameters record when the Flow Protocol or Route Protocol is ROHC

Field	Length (bits)
MaxCID	16
LargeCIDs	1
FeedbackForIncluded	1
FeedbackFor	0 or 5
MRRU	16
DelayedDecompressionDepth	8
ProfileCount	8

ProfileCount occurrences of the following field:

{	
Profile	16
}	
Reserved	0 – 7 (as needed)

- 4 MaxCID The sender shall set this field to the MAX\_CID parameter for this ROHC  
5 Channel. The sender shall not set this field to a value greater than  
6 MaxSupportedMaxCID.
- 7 LargeCIDs If the LARGE\_CIDS parameter for this ROHC Channel is false, then the  
8 sender shall set this field to '0'. Otherwise, the sender shall set this field to  
9 '1'. The sender shall not set this field to '1' if LargeCIDsSupported is not set  
10 to '1'.
- 11 FeedbackForIncluded If ROHC feedback associated with another Link flow (ROHC channel) is  
12 sent on this Link flow (ROHC channel), then this field shall be set to '1'.  
13 Otherwise, this field shall be set to '0'.
- 14 FeedbackFor If FeedbackForIncluded is set to '0', then the sender shall omit this field.  
15 Otherwise, the sender shall set this field to the Link flow number (ROHC  
16 channel) to which ROHC feedback sent on this Link flow (ROHC channel)  
17 refers.
- 18 MRRU The sender shall set this field to the MRRU parameter for this ROHC  
19 channel. The sender shall not set this field to a value larger than  
20 MaxSupportedMRRU.
- 21 DelayedDecompressionDepth  
22 The sender shall set this field to the maximum number of packets that can be  
23 buffered and thus possibly be delayed decompressed by the decompressor  
24 according to [11] for this ROHC channel. If the value of this field is 0x00,  
25 then delayed decompression shall not be enabled. The sender shall not set  
26 this field to a value greater than  
27 MaxSupportedDelayedDecompressionDepth.

1	ProfileCount	The sender shall set this field to the number of ROHC profiles supported by the decompressor. The sender shall include ProfileCount occurrences of the following field with the message.
2		
3		
4	Profile	The sender shall set this field to the ROHC profile supported by the decompressor according to [11]. The sender shall not set this field to a value that is not included in the list of supported Profiles.
5		
6		
7	Reserved	The sender shall add reserved bits to make the length of the entire record an integer number of octets. The sender shall set these bits to '0'. The receiver shall ignore this field.
8		
9		

### 10 3.3.6.2.16 FlowNNFlowProtocolParametersRev attribute

11 NN is the two-digit hexadecimal forward Link Flow identifier, where hexadecimal digits A through F  
12 are specified in upper case letters.

13

Field	Length (bits)	Default
Length	8	N/A
AttributeID	16	N/A
ProtocolID	8	<i>0x01</i>
ProtocolParametersLength	8	<i>0x00</i>
ProtocolParameters	ProtocolParametersLength × 8	N/A

14	Length	Length of the complex attribute in octets. The sender shall set this field to the length of the complex attribute excluding the Length field.
15		
16	AttributeID	The sender shall set this field to 0x10NN, where NN is the two-digit hexadecimal forward Link Flow number.
17		
18	ProtocolID	The sender shall set this field to an identifier for the Flow Protocol according to Table 14.
19		
20	ProtocolParametersLength	The sender shall set this field to the length of the ProtocolParameters field in units of octets.
21		
22		
23	ProtocolParameters	If ProtocolID is 0x02, then the sender shall set this record as defined in 3.3.6.2.16.1. Otherwise, the sender shall omit this record.
24		



### 3.3.6.2.16.1 Definition of ProtocolParameters record when the Flow Protocol or Route Protocol is ROHC

Field	Length (bits)
MaxCID	16
LargeCIDs	1
FeedbackForIncluded	1
FeedbackFor	0 or 5
MRRU	16
TimerBasedCompression	1
ProfileCount	8

ProfileCount occurrences of the following field:

{	
Profile	16
}	
Reserved	0 – 7 (as needed)

- 4 MaxCID The sender shall set this field to the MAX\_CID parameter for this ROHC  
5 Channel. The sender shall not set this field to a value greater than  
6 MaxSupportedMaxCID.
- 7 LargeCIDs If the LARGE\_CIDS parameter for this ROHC Channel is false, then the  
8 sender shall set this field to '0'. Otherwise, the sender shall set this field to  
9 '1'. The sender shall not set this field to '1' if LargeCIDsSupported is not set  
10 to '1'.
- 11 FeedbackForIncluded If ROHC feedback associated with another Link flow (ROHC channel) is  
12 sent on this Link flow (ROHC channel), then this field shall be set to '1'.  
13 Otherwise, this field shall be set to '0'.
- 14 FeedbackFor If FeedbackForIncluded is set to '0', then the sender shall omit this field.  
15 Otherwise, the sender shall set this field to the Link flow number (ROHC  
16 channel) to which ROHC feedback sent on this Link flow (ROHC channel)  
17 refers.
- 18 MRRU The sender shall set this field to the MRRU parameter for this ROHC  
19 channel. The sender shall not set this field to a value larger than  
20 MaxSupportedMRRU.
- 21 TimerBasedCompression  
22 The sender shall set this field to '0' if timer based compression according to  
23 [11] is not enabled for this ROHC channel. The sender shall set this field to  
24 '1' if timer based compression according to [11] is enabled for this ROHC  
25 channel. If TimerBasedCompressionSupported is set to '0', then the sender  
26 shall not set this field to '1'.

1	ProfileCount	The sender shall set this field to the number of ROHC profiles supported by the decompressor. The sender shall include ProfileCount occurrences of the following field with the message.
2		
3		
4	Profile	The sender shall set this field to the ROHC profile supported by the decompressor according to [11]. The sender shall not set this field to a value that is not included in the list of supported Profiles.
5		
6		
7	Reserved	The sender shall add reserved bits to make the length of the entire record an integer number of octets. The sender shall set these bits to '0'. The receiver shall ignore this field.
8		
9		

### 10 3.3.6.2.17 FlowNNRouteProtocolParametersFwd attribute

11 NN is the two-digit hexadecimal forward Link Flow number, where hexadecimal digits A through F  
12 are specified in upper case letters.

13

Field	Length (bits)	Default
Length	8	N/A
AttributeID	16	N/A
ProtocolID	8	0x00
ProtocolParametersLength	8	0x00
ProtocolParameters	ProtocolParametersLength × 8	N/A

14	Length	Length of the complex attribute in octets. The sender shall set this field to the length of the complex attribute excluding the Length field.
15		
16	AttributeID	The sender shall set this field to 0x11NN, where NN is the two-digit hexadecimal forward Link Flow number.
17		
18	ProtocolID	The sender shall set this field to an identifier for the Route Protocol according to Table 15.
19		

20

**Table 15 ProtocolID for Route Protocol**

Value	Protocol
0x00	NULL
0x02	The Route Protocol is Robust Header Compression (ROHC) [11]
All other values	Reserved

1 ProtocolParametersLength  
 2 The sender shall set this field to the length of the ProtocolParameters field in  
 3 units of octets.

4 ProtocolParameters If ProtocolID is 0x02, then the sender shall set this record as defined in  
 5 3.3.6.2.15.1. Otherwise, the sender shall omit this record.

### 6 3.3.6.2.18 FlowNNRouteProtocolParametersRev attribute

7 *NN* is the two-digit hexadecimal forward Link Flow number, where hexadecimal digits A through F  
 8 are specified in upper case letters.

9

Field	Length (bits)	Default
Length	8	N/A
AttributeID	16	N/A
ProtocolID	8	<b>0x00</b>
ProtocolParametersLength	8	<b>0x00</b>
ProtocolParameters	ProtocolParametersLength × 8	N/A

10 Length Length of the complex attribute in octets. The sender shall set this field to the  
 11 length of the complex attribute excluding the Length field.

12 AttributeID The sender shall set this field to 0x12*NN*, where *NN* is the two-digit  
 13 hexadecimal forward Link Flow number.

14 ProtocolID The sender shall set this field to field to an identifier for the Route Protocol  
 15 according to Table 15.

16 ProtocolParametersLength  
 17 The sender shall set this field to the length of the ProtocolParameters field in  
 18 units of octets.

19 ProtocolParameters If ProtocolID is 0x02, then the sender shall set this record as defined in  
 20 3.3.6.2.16.1. Otherwise, the sender shall omit this record

### 21 3.3.7 Session state information

22 The Session State Information record (see 10.10) consists of parameter records.

23 This transport defines the following parameter records in addition to the configuration attributes for  
 24 this transport.

### 3.3.7.1 FlowControlState parameter

**Table 16 Format of the parameter record for the FlowControlState parameter**

Field	Length (bits)
ParameterType	8
Length	8
FlowControlState	8

ParameterType This field shall be set to 0x01 for this parameter record.

Length This field shall be set to the length of this parameter record in units of octets excluding the Length field.

FlowControlState This field shall be set to 0x00 if the state of the Flow Control Protocol associated with the access terminal's session is Close. Otherwise, this field shall be set to 0x01. All of the other values for this field are reserved.

### 3.3.7.2 ReservationState parameter

**Table 17 Format of the parameter record for the ReservationState parameter**

Field	Length (bits)
ParameterType	8
Length	8
OpenReservationCount	8

OpenReservationCount occurrences of the following record:

{

Reserved	7
Link	1
ReservationLabel	8

}

ParameterType This field shall be set to 0x02 for this parameter record.

Length This field shall be set to the length of this parameter record in units of octets excluding the Length field.

OpenReservationCount This field shall be set to the number of Reservations that are in the Open state. The sender shall include OpenReservationCount occurrences of the following three fields with the message.

Reserved This field shall be set to '0000000'. The receiver shall ignore this field.

Link This field shall be set to '1' for a forward link Reservation, and to '0' for a reverse link Reservation.

1 ReservationLabel This field shall be set to the ReservationLabel.

### 2 3.3.7.3 RouteState parameter

3 **Table 18 Format of the parameter record for the RouteState parameter**

Field	Length (bits)
ParameterType	8
Length	8
RouteSelectionProtocolState	2
NextRouteSelectTransactionID	8
NextActivateRouteTransactionID	8
Reserved	6

4 ParameterType This field shall be set to 0x03 for this parameter record.

5 Length This field shall be set to the length of this parameter record in units of octets  
6 excluding the Length field.

7 RouteSelectionProtocolState  
8 This field shall be set to indicate the state of Route Selection Protocol  
9 according to Table 19.

10 NextRouteSelectTransactionID  
11 This field shall be set to the TransactionID field of the next RouteSelect  
12 message that will be sent.

13 NextActivateRouteTransactionID  
14 This field shall be set to the TransactionID field of the next ActivateRoute  
15 message that will be sent.

16 Reserved This field shall be set to '000000'. The receiver shall ignore this field.

17 **Table 19 RouteSelectionProtocolState encoding**

State	Value
A Open B Draining	'00'
A Open B Activating	'01'
A Draining B Open	'10'
A Activating B Open	'11'

18

### 3.3.7.4 RadioLinkState parameter

**Table 20 The format of the parameter record for the RadioLinkNNState parameter**

Field	Length (bits)
ParameterType	8
Length	8
QTxStateVector	$N_{\text{LinkFlowMax}}$
QRxStateVector	$N_{\text{LinkFlowMax}}$
Reserved	0 – 7 (as needed)

ParameterType	This field shall be set to 0x04 for this parameter record.
Length	This field shall be set to the length of this parameter record in units of octets excluding the Length field.
QTxStateVector	This field shall be set to the vector of binary values of the sequence state variables $[Q_{00,Tx}, Q_{01,Tx}, \dots, Q_{LFMax,Tx}]$ , where $LFMax$ is equal to $N_{\text{LinkFlowMax}}-1$ .
QRxStateVector	This field shall be set to the vector of binary values of the sequence state variables $[Q_{00,Rx}, Q_{01,Rx}, \dots, Q_{LFMax,Rx}]$ , where $LFMax$ is equal to $N_{\text{LinkFlowMax}}-1$ .
Reserved	The sender shall add reserved bits to make the length of the entire record an integer number of octets. The sender shall set these bits to '0'. The receiver shall ignore this field.

## 3.4 Default Packet Consolidation Protocol

### 3.4.1 Overview

The Default Packet Consolidation Protocol provides the following functions:

- Multiplexing of transports for one access terminal. Each transport maps to a Transport in the Packet Consolidation Protocol. Transport 0 is always assigned to the Signaling Transport. The other Transports can be assigned to transports with different Quality of Service (QoS) requirements, or other types of transports.
- Provision of configuration messages that map transports to Transports.
- Packet consolidation on the transmit side and packet de-multiplexing on the receive side.
- Prioritization of the transmission of packets.

1 Table 21 specifies the values of Transport Subtypes for transports defined in this specification.

2 **Table 21 Transport subtypes for transports defined in this specification**

Value	Meaning
0x0000	Default Signaling Transport.
0x0001	Default Packet Transport.
0xffff	Transport not used
All other values are reserved.	

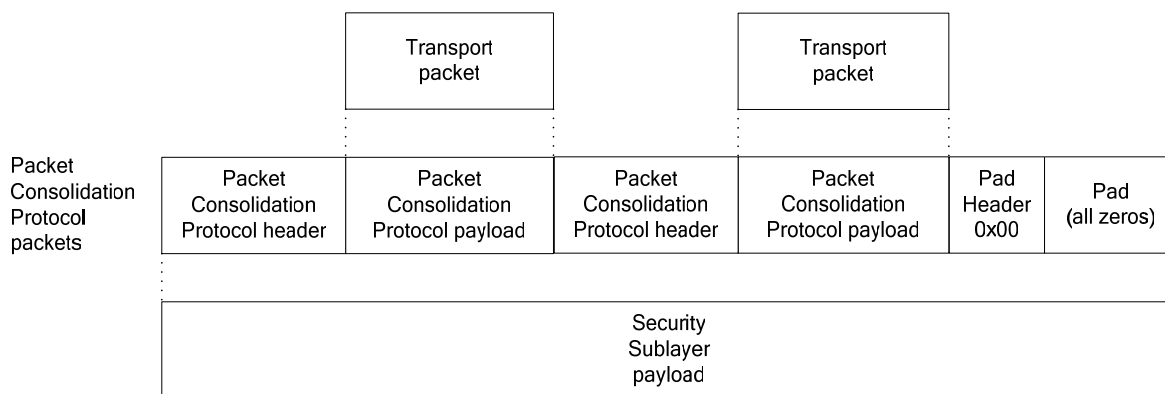
3 The Default Packet Consolidation Protocol provides the ability to multiplex up to 8 transports using  
4 the Transport field in the Packet Consolidation Protocol header. Transport 0 is always reserved for a  
5 Signaling Transport.

6 This protocol uses the Generic Attribute Update Protocol in 10.9 to map transports to Transports.

7 Packet Consolidation Protocol packets contain one or more transport packets. The protocol places the  
8 Packet Consolidation Protocol header defined in 3.4.8 in front of each transport packet and enough  
9 padding to create a maximum length packet. The header added by this protocol for a consolidated  
10 packet is 16 bits in length per transport packet and 8 bits in length for padding.

### 11 3.4.2 Data encapsulation

12 Figure 40 illustrates the relationship between a transport packet, a Packet Consolidation Protocol  
13 packet, and a Security Sublayer payload for a Packet Consolidation packet containing two transport  
14 packets and padding.



15  
16 **Figure 40 Packet Consolidation Protocol encapsulation**

### 17 3.4.3 Primitives

#### 18 3.4.3.1 Commands

19 This protocol does not define any commands.

### 1 **3.4.3.2 Return indications**

2 This protocol does not return any indications.

## 3 **3.4.4 Public data**

### 4 **3.4.4.1 Static public data**

5 This protocol does not define any static public data.

### 6 **3.4.4.2 Dynamic public data**

- 7 ■ Subtype for this protocol

## 8 **3.4.5 Protocol data unit**

9 The Protocol Data Unit for this protocol is a Packet Consolidation Protocol packet. Packet  
10 Consolidation Protocol packets contain transport packets destined to or from the same access terminal  
11 address.

## 12 **3.4.6 Protocol initialization and swap procedures**

### 13 **3.4.6.1 Protocol initialization**

14 Upon creation, the value of the attributes for this protocol instance in the access terminal and access  
15 network shall be set to the default values specified for each attribute.

### 16 **3.4.6.2 Protocol swap**

17 This protocol defines an empty swap procedure.

## 18 **3.4.7 Procedures**

19 This protocol receives transport packets for transmission from up to 8 different transports. All  
20 transmitted packets are forwarded to the Security Sublayer. All Packet Consolidation Protocol packets  
21 forwarded to the Security Sublayer shall be octet aligned.

22 The protocol receives Packet Consolidation Protocol packets from the Security Sublayer and removes  
23 the Packet Consolidation Protocol header. The transport packet obtained in this manner is forwarded  
24 to the transport indicated by the Transport field of the Packet Consolidation Protocol header.

25 The maximum size transport packet the protocol can encapsulate depends on the Physical Layer  
26 channel on which this packet will be transmitted and on the specific security protocols negotiated.

27 The access terminal and the access network may use the Generic Attribute Update Protocol messages  
28 in 10.9 to map a transport to a Transport that is not already assigned to another Transport.

29 The access terminal and the access network shall not use the Generic Attribute Update Protocol in  
30 10.9 to map a transport to a Transport that is already assigned to another Transport.



1 Once the access terminal and the access network agree upon the mapping of a new transport to a  
 2 Transport, the access terminal and access network shall create an instance of the agreed upon  
 3 transport and add the instance of the transport to that Transport.

4 This protocol receives the following information with every transmitted transport packet:

- 5 ■ Destination channel: Forward Unicast Traffic Channel, Forward Broadcast Traffic  
 6 Channel, or Reverse Traffic Channel.
- 7 ■ Priority number of the transport packet. This field is determined by the  
 8 FlowNNRequestLevelRev public data of the Data Transport and the constant  
 9  $N_{SLPRequestLevelRev}$  of the Signaling Transport. In this protocol, the use of the priority level is  
 10 defined only for access terminal transmissions.
- 11 ■ Forced Single Encapsulation: Whether or not the transport packet can be encapsulated  
 12 with other transport packets in the same Packet Consolidation Protocol packet.  
 13 (Applicable only on the Forward Data Channel)

#### 14 **3.4.7.1 Destination channels**

15 Associated with a transport packet received by this protocol there shall be a parameter indicating the  
 16 destination channel on which the packet is to be transmitted.

17 Associated with a transport packet received by this protocol there may be a parameter indicating a  
 18 transmission deadline.

#### 19 **3.4.7.2 Priority order**

20 The priority used by the access network to derive Packet Consolidation Protocol packets from  
 21 transport packets is beyond the scope of this specification.

22 The priority used by the access terminal to derive Packet Consolidation Protocol packets from  
 23 transport packets shall follow the following rules.

- 24 ■ Packets with lower priority number, as determined by the FlowNNRequestLevelRev  
 25 public data of the Packet Transport or the constant  $N_{SLPRequestLevelRev}$  of the Signaling  
 26 Transport shall have higher priority for transmission.
- 27 ■ For packets with the same priority number, the packet that was received first by the  
 28 protocol shall have higher priority for transmission.

29 Transmission of packets that have higher priority shall take precedence over transmission of packets  
 30 with lower priority within the constraints imposed by lower layer protocols.

#### 31 **3.4.7.3 Forced single encapsulation**

32 If a Forward Traffic Channel Transport packet is marked as Forced Single Encapsulation, the access  
 33 network shall encapsulate it without any other transport packets in a Packet Consolidation Protocol  
 34 packet. The Packet Consolidation Protocol shall also pass an indication down to the physical layer  
 35 with the Packet Consolidation Protocol packet, instructing the physical layer to ensure that the  
 36 Physical Layer packet containing this packet does not contain any other Packet Consolidation  
 37 Protocol packet. Forced Single Encapsulation applies only to the Forward Traffic Channel MAC  
 38 Layer packets.

1 Forced Single Encapsulation is used for test services that require a one to one mapping between  
2 transport packets and Physical Layer packets.

### 3 **3.4.7.4 Transmit procedures**

4 The transmitter shall create a Packet Consolidation Protocol packet by adding the Packet  
5 Consolidation Protocol header, defined in 3.4.8 in front of every transport packet, concatenating the  
6 result and adding enough padding to fill the Security Sublayer payload. The resulting packet length  
7 shall not exceed the maximum payload that can be carried on the Physical Layer Channel, given the  
8 transmission rate that will be used to transmit the packet and the headers added by the lower layers.

9 The transmitter shall forward the Packet Consolidation Protocol packet for transmission to the  
10 Security Sublayer.

#### 11 **3.4.7.4.1 Pad**

12 When creating a Packet Consolidation Protocol packet, the access network and the access terminal  
13 shall add sufficient padding so that the packet fills the Security Sublayer payload and set the padding  
14 bits to '0'. When receiving a Packet Consolidation Protocol packet, the access network and the access  
15 terminal shall ignore the padding bits.

### 16 **3.4.7.5 Access network procedures**

#### 17 **3.4.7.5.1 Control channel**

18 This protocol does not transmit over the Control Channel.

#### 19 **3.4.7.5.2 Broadcast forward traffic channel**

20 All transport packets sent in a Packet Consolidation Protocol packet should be destined to all MAC  
21 IDs.

#### 22 **3.4.7.5.3 Unicast forward traffic channel**

23 All transport packets sent in a Packet Consolidation Protocol packet should be destined to one MAC  
24 ID.

### 25 **3.4.8 Packet Consolidation Protocol header**

26 The sender adds the following header in front of every transport packet encapsulated in a Packet  
27 Consolidation Protocol packet:

Field	Length (bits)
IsTransport	1
Transport	3
Length	12

- 1 IsTransport This field shall be set to 1.
- 2 Transport The sender shall set this field to the Transport number associated with the  
3 transport sending the transport packet following the header.
- 4 Length This field shall be set to the length of the transport packet in octets.
- 5 The transport packet shall be at least one byte. The header value 0x00 shall indicate the beginning of  
6 the Pad, and the receiver shall ignore a Packet Consolidation Protocol packet beyond the 0x00 header.  
7 The Packet Consolidation Protocol packet format is described in Figure 40. In case the transport  
8 packets together with the Packet Consolidation Protocol headers fill the entire available payload, the  
9 pad and pad header shall be omitted.

### 10 **3.4.9 Message formats**

11 The protocol uses the AttributeUpdateRequest, AttributeUpdateAccept, and AttributeUpdateReject  
12 messages of the Generic Attribute Update Protocol in 10.9 to update configurable attributes.

### 13 **3.4.10 Interface to other protocols**

#### 14 **3.4.10.1 Commands**

15 This protocol does not issue any commands.

#### 16 **3.4.10.2 Indications**

17 This protocol does not register to receive any indications.

### 18 **3.4.11 Configuration attributes**

19 The following complex attribute and default values are defined (see 10.3 for attribute record  
20 definition).

21 Unless specified otherwise, the access terminal and the access network shall use the Generic Attribute  
22 Update Protocol in 10.9 to update configurable attributes belonging to the Default Data Transport.

### 3.4.11.1 TransportConfiguration attribute

Field	Length (bits)	Default
Length	8	N/A
AttributeID	8	N/A
Transport0	16	0x0000
Transport1	16	0xffff
Transport2	16	0xffff
Transport3	16	0xffff
Transport4	16	0xffff
Transport5	16	0xffff
Transport6	16	0xffff
Transport7	16	0xffff

**Length** Length of the complex attribute in octets. The sender shall set this field to the length of the complex attribute excluding the Length field.

**AttributeID** The sender shall set this field to 0x00.

**Transport0** The sender shall set this field to the subtype of the transport used over Transport 0.

**Transport1** The sender shall set this field to the subtype of the transport used over Transport 1.

**Transport2** The sender shall set this field to the subtype of the transport used over Transport 2.

**Transport3** The sender shall set this field to the subtype of the transport used over Transport 3.

**Transport4** The sender shall set this field to the subtype of the transport used over Transport 4.

**Transport5** The sender shall set this field to the subtype of the transport used over Transport 5.

**Transport6** The sender shall set this field to the subtype of the transport used over Transport 6.

**Transport7** The sender shall set this field to the subtype of the transport used over Transport 7.

The sender shall set the Transport*N* fields to one of the non-reserved values for the Transport Subtype as specified in Table 21.

### 3.4.12 Protocol numeric constants

Constant	Meaning	Value
N <sub>PCPType</sub>	Type field for this protocol.	Table 9
N <sub>PCPDefault</sub>	Subtype field for this protocol	0x0000

### 3.4.13 Session state information

The Session State Information record (see 10.10) consists of parameter records.

The parameter records for this protocol consist of only the configuration attributes of this protocol.

## 4 Security Control Sublayer

### 4.1 Introduction

#### 4.1.1 General overview

The Security Control sublayer provides the following functions:

- **Key Exchange:** Provides the procedures followed by the access network and by the access terminal to exchange security keys for authentication and encryption.

The Security Control Sublayer uses the Key Exchange Protocol to provide these functions.

### 4.2 Default Key Exchange Protocol

#### 4.2.1 Overview

The Default Key Exchange Protocol provides a method for simultaneous generation of the session key at the access terminal and the access network. The session key is derived from a PairwiseMasterKey (PMK) that is negotiated by higher layer protocols and assumed available at the access terminal and the access network. This protocol supports cases where there may be multiple PairwiseMasterKeys. The procedure for deriving the PairwiseMasterKey is considered to be out of scope for this document.

The session key is used to derive the MIC Key, Authentication Key and Encryption Key. The MIC key is used to verify the four way exchange messages of this protocol. The Authentication Key may be used to authenticate packets (see the Authentication Protocol for details), and the Encryption Key may be used to encrypt packets (see the Encryption Protocol for details).

This protocol also provides methods and messages to change session (security) key after a session has been established.

#### 4.2.2 Primitives

##### 4.2.2.1 Commands

This protocol does not define any commands.

##### 4.2.2.2 Return indications

- *FirstKeyComplete*

#### 4.2.3 Public data

##### 4.2.3.1 Static public data

This protocol does not define any static public data.

### 4.2.3.2 Dynamic public data

- Subtype for this protocol
- FLAuthKey and its length
- RLAuthKey and its length
- FLEncKey and its length
- RLEncKey and its length
- KeyChangeInitiated

### 4.2.4 Protocol data unit

The transmission unit of this protocol is a message. This is a control protocol and, therefore, it does not carry payload on behalf of other layers or protocols.

This protocol uses the Signaling Application to transmit and receive messages.

### 4.2.5 Protocol initialization and swap

#### 4.2.5.1 Protocol initialization

Upon initialization, the value of the attributes for this protocol instance in the access terminal and the access network shall be set to the following default values specified for each attribute.

- Set SKey[i] to zero and its length to 384, for values of i from 0 through 7.
- Set FLAuthKey[i] to zero and its length to 128, for values of i from 0 through 7.
- Set RLAuthKey[i] to zero and its length to 128, for values of i from 0 through 7.
- Set FLEncKey[i] to zero and its length to 128, for values of i from 0 through 7.
- Set RLEncKey[i] to zero and its length to 128, for values of i from 0 through 7.
- Set ATNonce to NULL.
- Set ANNonce to NULL.
- Set LastValidTransactionID to 255.
- Set KeyChangeInitiated to '0'.

#### 4.2.5.2 Protocol swap

- Set LastValidTransactionID to 255 upon protocol swap.

### 4.2.6 Procedures

The Default Key Exchange Protocol uses the KeyRequest, KeyResponse, ANKeyComplete, and ATKeyComplete messages to derive secret session keys, verify that the access terminal and the access network have derived the same session keys, and to exchange security capabilities.

This protocol is able to swap the current session key that is in use with another key that has already been derived from the PMK. This is done using the KeyChange bit included in the MAC header, as well as KeyChangeRequest and KeyChangeAck messages.

## 4.2.6.1 Access terminal requirements

### 4.2.6.1.1 Processing the KeyRequest message

Upon receiving the KeyRequest message, the access terminal shall perform the following:

- The access terminal shall declare the message to be valid if the TransactionID of the message does not match the TransactionID of any outstanding KeyRequest message.
- If the KeyRequest message is not valid, then the access terminal shall send an ATKeyComplete message with ResultCode set to 'Transaction ID Invalid', declare failure, and stop performing the rest of the key exchange procedure.
- The access terminal shall identify the PairwiseMasterKey that satisfies  $\text{PairwiseMasterKeyID} = \text{HMAC-SHA256-128}(\text{PairwiseMasterKey}, \text{"PMK\_Name"} \mid \text{SessionSeed})$ .
  - PairwiseMasterKeyID is a field of the received KeyRequest message, "PMK\_Name" is the ASCII encoded value of the string.
  - HMAC-SHA256-128 function is specified in 4.2.6.5.
  - SessionSeed is public data of the Session Management Protocol
  - The notation "|" implies concatenation.
- If the access terminal cannot identify a valid PairwiseMasterKey that satisfies the above Equation, then the access terminal shall declare failure and shall send an ATKeyComplete message with ResultCode set to 0x03, declare failure, and stop performing the rest of the key exchange procedure.
- The access terminal should set ATNonce to  $\text{PRF}(\text{Random number}, \text{"Init\_Counter"}, \text{PhyFrameIndex64}, 256)$ ,
  - Random number is a 256-bit random number. This number may be generated according to the pseudorandom number generator specified in 10.6. If the procedure of 10.6 is used and a physical random number  $\chi$  is available, a fresh initialization should be used each time the random number is generated.
  - "Init\_Counter" is the ASCII encoded value of the string.
  - PhyFrameIndex64 is the 64-bit representation of the PHY Frame Index defined in the Lower MAC Sublayer.
  - PRF function is specified in 4.2.6.4.
- The access terminal shall compute SKey[i] the session key in the following way:
 

$\text{SKey}[i] = \text{PRF}(\text{PairwiseMasterKey}, \text{"Pairwise\_Key\_Expansion"}, \text{SessionSeed} \mid \text{Nonce1} \mid \text{Nonce2}, 384)$ .

  - Where i is the SessionKeyIndex field of the corresponding KeyRequest message,  $\text{Nonce1} = \text{Min}(\text{ATNonce}, \text{ANNonce})$ ,  $\text{Nonce2} = \text{Max}(\text{ATNonce}, \text{ANNonce})$ ,
  - ANNonce is the ANNonce field of the received KeyRequest message, PairwiseMasterKey is the key associated with the PairwiseMasterKeyID field of the KeyRequest message.
  - "Pairwise\_Key\_Expansion" is the ASCII encoded value of the string.



- 1           □ SessionSeed is public data of the Session Management Protocol.
- 2           □ PRF function is specified in 4.2.6.4.
- 3           ■ The access terminal shall generate MIC Key, Authentication Key and Encryption Key as
- 4           specified in 4.2.6.3.
- 5           ■ The access terminal shall send a KeyResponse message.

#### 6   **4.2.6.1.2 Processing the ANKeyComplete message**

7   After receiving an ANKeyComplete message with a TransactionID field that matches the  
8   TransactionID field of the associated KeyRequest message, the access terminal shall perform the  
9   following:

- 10          ■ The access terminal shall generate a MessageIntegrityCode as HMAC-SHA256-  
11          128(MICKey[i], *Message*), where *Message* is the received ANKeyComplete message  
12          with the MessageIntegrityCode field set to zero, *i* is the SessionKeyIndex field of the  
13          corresponding KeyRequest message, and the HMAC-SHA256-128 function is specified  
14          in [10].
- 15          ■ The access terminal shall set LastValidTransactionID to the TransactionID field of the  
16          ANKeyComplete message and send an ATKeyComplete message with the ResultCode  
17          field set to 0x00, unless one of the following conditions holds. In that case, the access  
18          terminal shall declare failure and send an ATKeyComplete message with the appropriate  
19          ResultCode.
  - 20              □ If the MessageIntegrityCode computed in the previous step does not match the  
21              MessageIntegrityCode field of the ANKeyComplete message In this case, the access  
22              terminal shall use the ResultCode corresponding to “Message integrity code failed”)
  - 23              □ Otherwise, if the supported tokens sent by the access network in the  
24              ANKeyComplete message include a token that the access terminal supports and  
25              prefers to use to the token currently in use (SessionConfigurationToken in the public  
26              data of the Session Configuration Protocol). In this case, the access terminal shall use  
27              the ResultCode corresponding to “Message integrity code successful, but capabilities  
28              verification failed”.
- 29          ■ If the access terminal sends a ATKeyComplete message with ResultCode field set to  
30          0x00, and the key exchange was performed for SessionKeyIndex set to the configuration  
31          attribute SessionKeyIndexInUse, the access terminal shall generate a *FirstKeyComplete*  
32          indication.

#### 33   **4.2.6.1.3 Processing the KeyChangeRequest message**

34   Only the access network is permitted to initiate a key change by sending a KeyChangeRequest  
35   message. The access terminal may request initiation of a key change by sending a  
36   KeyChangeInitiateRequest message with the appropriate SessionKeyIndexRequested field.

1 Upon receipt of a KeyChangeRequest message, the access terminal shall verify that  
 2 KeyChangeInitiated='0'. If not, the access terminal shall abort any key exchange in progress by  
 3 setting SessionKeyIndexPending to 0xff.

4 If the access terminal proceeds with the key change process, it shall

- 5 ■ Set the SessionKeyIndexPending to the value received in the KeyChangeRequest  
 6 message
- 7 ■ Respond with a KeyChangeAck message and set KeyChangeInitiated='1'.
  - 8 □ In the KeyChangeAck message, the access terminal shall copy TransactionID and  
 9 KeyIndexPending from the KeyChangeRequest message that caused the generation  
 10 of this KeyChangeAck message.

#### 11 4.2.6.1.4 Processing the KeyChange bit

12 If the access terminal receives a packet with the KeyChange bit in the Forward Traffic Channel MAC  
 13 header toggled the access terminal shall verify that KeyChangeInitiated is equal to '1'. If not, the  
 14 access terminal shall abort any key exchange in progress and set SessionKeyIndexPending to 0xff.

15 If KeyChangeInitiated is equal to '1', the access terminal shall

- 16 ■ Set SessionKeyIndexInUse to the value of SessionKeyIndexPending and set  
 17 SessionKeyIndexPending to 0xff.
- 18 ■ Update the values of MIC Key, FLAuthKey, RLAuthKey, FLEncKey and RLEncKey  
 19 following the procedure specified in 4.2.6.3 before the received packet is processed by  
 20 the Security Sublayer.
- 21 ■ Set KeyChangeInitiated to '0' (completing the key change process).
- 22 ■ Toggle the value of the KeyChange bit in the Reverse Traffic Channel MAC header for  
 23 subsequent transmissions.

#### 24 4.2.6.2 Access network requirements

25 The access network shall initiate the key exchange by sending a KeyRequest message. The access  
 26 network shall choose a nonce, ANNonce as follows:

- 27 ■ The access network should set ANNonce to
  - 28 □ PRF(Random number, "Init\_Counter", AP SectorID | PhyFrameIndex64, 256).
  - 29 □ Random number is a 256-bit random number. This number may be generated  
 30 according to the pseudorandom number generator specified in 10.6. If the procedure  
 31 of 10.6 is used and a physical random number  $\chi$  is available, a fresh initialization  
 32 should be used each time the random number is generated.
  - 33 □ "Init\_Counter" is the ASCII encoded value of the string, and PhyFrameIndex64 is the  
 34 64-bit representation of the Phy Frame Index defined in the Lower MAC Sublayer.
  - 35 □ PRF function is specified in 4.2.6.4.

#### 4.2.6.2.1 Processing the KeyInitiateRequest message

Upon receiving the KeyInitiateRequest message, the access network shall perform the following:

- The access network shall identify the PairwiseMasterKey that satisfies  $\text{PairwiseMasterKeyID} = \text{HMAC-SHA256-128}(\text{PairwiseMasterKey}, \text{"PMK\_Name"} \mid \text{SessionSeed})$ .
  - PairwiseMasterKeyID is a field of the received KeyRequest message, "PMK\_Name" is the ASCII encoded value of the string.
  - SessionSeed is public data of the Session Management Protocol.
  - HMAC-SHA256-128 function is specified in [10].
  - The notation "|" implies concatenation.

If the access network can identify a valid PairwiseMasterKey that satisfies the above equation, then the access network may initiate a session key exchange by sending a KeyRequest message.

#### 4.2.6.2.2 Processing the KeyResponse message

After receiving a KeyResponse message with a TransactionID field that matches the TransactionID field of the associated KeyRequest message, the access network shall perform the following:

- The access network shall compute  $\text{SKey}[i]$ , the session key as follows:
  - $\text{SKey}[i] = \text{PRF}(\text{PairwiseMasterKey}, \text{"Pairwise\_Key\_Expansion"}, \text{SessionSeed} \mid \text{Nonce1} \mid \text{Nonce2}, 384)$ .
  - Where  $i$  is the SessionKeyIndex field of the corresponding KeyRequest message,
  - "Pairwise\_Key\_Expansion" is the ASCII encoded value of the string,  $\text{Nonce1} = \text{Min}(\text{ATNonce}, \text{ANNonce})$ ,  $\text{Nonce2} = \text{Max}(\text{ATNonce}, \text{ANNonce})$ .
  - ATNonce is the ATNonce field of the KeyResponse message, and the PRF function is specified in 4.2.6.4.
  - SessionSeed is public data of the Session Management Protocol.
- The access network shall generate MIC Key, Authentication Key and Encryption Key as specified in 4.2.6.3.
- The access network shall generate a MessageIntegrityCode as HMAC-SHA256-128 ( $\text{MICKey}[i]$ , *Message*), where *Message* is the received KeyResponse message with the MessageIntegrityCode field set to zero,  $i$  is the SessionKeyIndex field of the corresponding KeyRequest message, and the HMAC-SHA256-128 function is specified in [10].
- The access network shall send an ANKeyComplete message and increment LastValidTransactionID unless one of the following conditions holds. In that case, the access network shall declare failure and send an ANKeyComplete message with the appropriate ResultCode.
  - If the MessageIntegrityCode computed in the previous step does not match the MessageIntegrityCode field of KeyResponse message.

- 1           □ If the supported tokens sent by the access terminal in the KeyResponse message
- 2           contain a token that the access network supports and prefers to use to the token
- 3           currently in use (SessionConfigurationToken in the public data of the Session
- 4           Configuration Protocol).

#### 5   **4.2.6.2.3 Processing the ATKeyComplete message**

6   If the access network receives an ATKeyComplete message with ResultCode field set to a value other  
7   than 0x00, the access network shall declare failure and stop performing the rest of the key exchange  
8   procedure.

9   If the access network receives a ATKeyComplete message with ResultCode field set to 0x00, and the  
10   key exchange was performed for SessionKeyIndex set to the configuration attribute  
11   SessionKeyIndexInUse, the access network shall generate a *FirstKeyComplete* indication.

#### 12   **4.2.6.2.4 Transmitting the KeyChangeRequest message**

13   Upon receipt of KeyChangeInitiateRequest message, the access network may initiate a key change.

14   The access network shall initiate the key change by sending a KeyChangeRequest message only if no  
15   key change request initiated by the access network is in progress, i.e. if the key change timer is  
16   inactive and the KeyChangeInitiated bit is set to '0'.

17   Upon sending a KeyChangeRequest message, the access network shall

- 18           ■ start a key change timer for  $T_{KEPTimer}$  and abort the key change process by setting
- 19           SessionKeyIndexPending to 0xff if the timer expires.
- 20           ■ set SessionKeyIndexPending to the proposed SessionKeyIndex.

#### 21   **4.2.6.2.5 Processing the KeyChangeAck message**

22   The access network shall ignore the KeyChangeAck message if KeyChangeInitiated='1'. Otherwise,  
23   upon receipt of a KeyChangeAck message the access network shall

- 24           ■ Set KeyChangeInitiated to '1'.
- 25           ■ Disable the key change timer.
- 26           ■ Set SessionKeyIndexInUse to the value of SessionKeyIndexPending.
- 27           ■ Toggle the KeyChange bit in the Forward Traffic Channel MAC header for subsequent  
28           packet transmissions.
- 29           ■ Update the values of MIC Key, FLAuthKey and FLEncKey following the procedure  
30           specified in 4.2.6.3 before the next packet is processed for transmission.

#### 31   **4.2.6.2.6 Processing the KeyChange bit**

32   If KeyChangeInitiated is equal to '1' and the access network receives a packet with the KeyChange  
33   bit in the Reverse Traffic Channel MAC header toggled, the access network shall

- 34           ■ Set SessionKeyIndexPending to 0xff.
- 35           ■ Update the values of MIC Key, RLAAuthKey, and RLEncKey following the procedure  
36           specified in 4.2.6.3 before the received packet is processed by the Security Sublayer.

- 1           ■ Set KeyChangeInitiated to '0' (completing the key change process).

### 2   **4.2.6.3 MIC Key, Authentication Key, and Encryption Key generation**

3   The keys used for message integrity code, authentication and encryption are generated from the  
4   session key using the procedures specified in this section.

5   The access network and the access terminal shall compute and store a MIC Key, Authentication Key,  
6   and Encryption Key derived from each session key. The keys derived from SKey[*i*] are referred to by  
7   the subscript *i*. The Encryption and Authentication Protocols at the access network and the access  
8   terminal shall use the Authentication Key and Encryption Key derived from the SKey with index *i* set  
9   to the SessionKeyIndexInUse.

10   The MIC Key, Authentication Key and Encryption Key attributes are computed in the following way.

- 11           ■ The access network and the access terminal shall set the MICKey[*i*] to SKey[*i*][127:0],  
12           where *i* is the session key index.
- 13           ■ The access network and the access terminal shall set FLAuthKey[*i*] and RLAuthKey[*i*] to  
14           SKey[*i*][255:128], where *i* is the session key index.
- 15           ■ The access network and the access terminal shall set FLEncKey[*i*], and RLEncKey[*i*] to  
16           SKey[*i*][383:256], where *i* is the session key index.

### 17   **4.2.6.4 Pseudorandom function, PRF(K, A, B, Len)**

18   A pseudorandom function (PRF) is used in a number of places in this document.

19   Len shall be no greater than 255\*160.

20   The output of the pseudorandom function is obtained by executing the following pseudo-code:

- 21           ■ R = NULL
- 22           ■ **for** *i* = 0 **to** (Len+159)/160 **do**
  - 23           □ R = R | HMAC-SHA256-160(K, A | Y | B | *i*),
  - 24           □ *Y* is a single octet containing the value zero.
  - 25           □ *i* is a single octet containing the parameter.
  - 26           □ HMAC-SHA256-160 function is specified in [10].

27   The output of the PRF function shall be set to the *Len* most significant bits of R.

### 28   **4.2.6.5 HMAC-SHA256(K, Message)**

29   The HMAC-SHA1 procedure as specified in [10], shall be performed with SHA-256 [1] as the  
30   message digest algorithm.

31   The output of the HMAC-SHA256-128 function shall be set to the 128 Most Significant Bits of the  
32   output of the HMAC-SHA1 procedure.

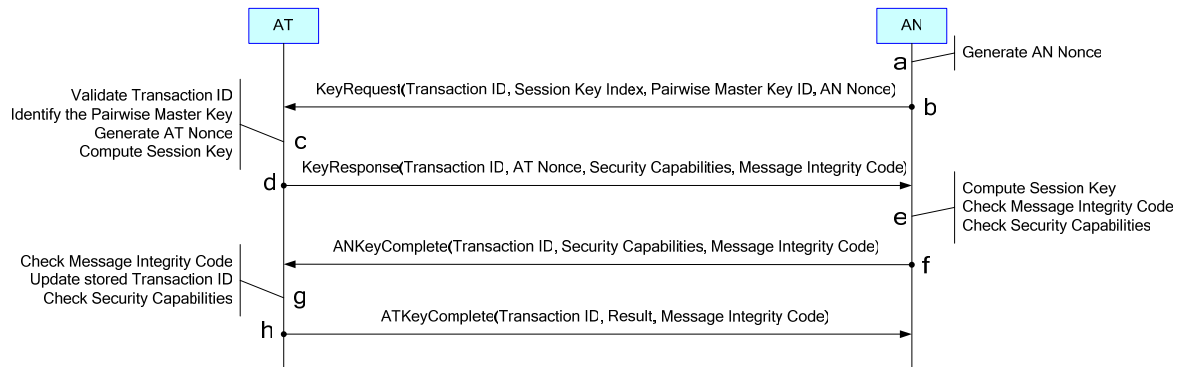
33   The output of the HMAC-SHA256-160 function shall be set to the 160 Most Significant Bits of the  
34   output of the HMAC-SHA1 procedure.

1 **4.2.7 Message format and flows**

2 **4.2.7.1 Message flows**

3 **4.2.7.1.1 Message flow for Default Key Exchange Protocol**

4 This section describes the message flow for the Default Key Exchange Protocol. Figure 41 shows the  
 5 message exchanges for the Default Key Exchange Protocol.

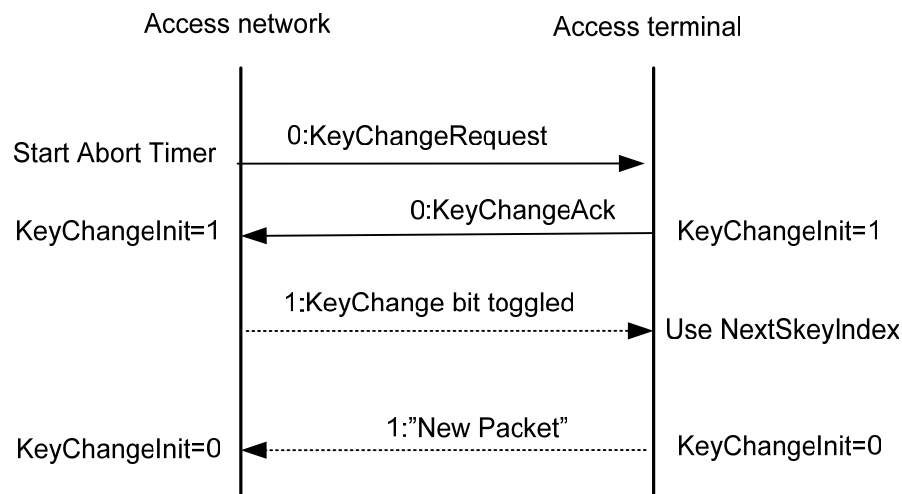


6  
7 **Figure 41 Default Key Exchange Protocol message flow**

8 **4.2.7.1.2 Message flow for Security Key Change Protocol**

9 This section describes the message flow for the Key Change Protocol.

10 Message flow needed to execute key change is shown in Figure 42. Key change may be negotiated by  
 11 the access terminal or the access network. Solid lines in Figure 42 indicate the messages exchanged  
 12 between the communicating peers. The value of the KeyChange bit is assumed to be '0'. The dotted  
 13 lines do not indicate messages related to the Key Exchange Protocol. They show the exchange of  
 14 regular packets after the KeyChange bit is toggled and actual key change is executed.



15  
16 **Figure 42 Security Key Change Protocol [??Figure updated above]**

## 4.2.7.2 Message formats

The protocol uses the AttributeUpdateRequest, AttributeUpdateAccept, and AttributeUpdateReject messages of the Generic Attribute Update Protocol in 10.9 to update configurable attributes.

### 4.2.7.2.1 KeyInitiateRequest

The access terminal may send the KeyInitiateRequest message to request the access network to initiate a session key exchange. The access network may or may not initiate a key exchange in response to this message.

Field	Length (bits)
MessageID	8
SessionKeyIndex	8
PairwiseMaskterKeyID	128

**MessageID** The access terminal shall set this field to 0x00.

**SessionKeyIndex** The access terminal shall set this field to the ID of the SKey for which this key exchange is being initiated. The values 0x08 to 0xff are reserved.

**PairwiseMasterKeyID** The access terminal shall set this field to HMAC-SHA256-128 (PMK, "PMK\_Name" | SessionSeed), where "PMK\_Name" is the ASCII encoded value of the string.

<b>Channels</b>	RTC	<b>SLP</b>	Reliable
<b>Addressing</b>	Unicast	<b>Security</b>	Required

### 4.2.7.2.2 KeyRequest

The access network sends the KeyRequest message to initiate the session key exchange.

Field	Length (bits)
MessageID	8
TransactionID	8
SessionKeyIndex	8
PairwiseMaskterKeyID	128
ANNonce	256

**MessageID** The access network shall set this field to 0x01.

**TransactionID** The access network shall set this field according to 10.8.

**SessionKeyIndex** The access network shall set this field to the ID of the SKey for which this key exchange is being initiated. The values 0x08 to 0xff are reserved.

- 1 Reserved The access network shall set this field to '00000'. The access terminal shall  
2 ignore this field.
- 3 PairwiseMasterKeyID The access network shall set this field to HMAC-SHA256-128 (PMK,  
4 "PMK\_Name" | SessionSeed), where "PMK\_Name" is the ASCII encoded  
5 value of the string.
- 6 ANNonce The access network shall set this field to the nonce chosen by the access  
7 network.

<b>Channels</b>	FTC	<b>SLP</b>	Reliable
<b>Addressing</b>	Unicast	<b>Security</b>	Required

9

### 10 4.2.7.2.3 KeyResponse

11 The access terminal sends the KeyResponse message in response to the KeyRequest message.

12

Field	Length (bits)
MessageID	8
TransactionID	8
ANNonce	256
TokenCount	8
TokenCount occurrences of the following field:	
SupportedToken	16
MessageIntegrityCode	128

- 13 MessageID The access terminal shall set this field to 0x02.
- 14 TransactionID The access terminal shall set this field to the value of the TransactionID field  
15 of the corresponding KeyRequest message.
- 16 ANNonce The access terminal shall set this field to the nonce chosen by the access  
17 terminal.
- 18 TokenCount The access terminal shall set this field to the number of tokens supported by  
19 the access terminal.
- 20 SupportedToken The access terminal shall set this field to a token supported by the access  
21 terminal.
- 22 MessageIntegrityCode The access terminal shall set this field to HMAC-SHA256-128(MICKey[i],  
23 *Message*), where *Message* is set to all fields of this message with this field  
24 set to zero, and *i* is SessionKeyIndex field of the corresponding KeyRequest  
25 message.

26



<b>Channels</b>	RTC	<b>SLP</b>	Reliable
<b>Addressing</b>	Unicast	<b>Security</b>	Optional

#### 4.2.7.2.4 ANKeyComplete

The access network sends the ANKeyComplete message in response to the KeyResponse message.

Field	Length (bits)
MessageID	8
TransactionID	8
ResultCode	8
TokenCount	8
TokenCount occurrences of the following field:	
SupportedToken	16
MessageIntegrityCode	128

**MessageID** The access network shall set this field to 0x03.

**TransactionID** The access network shall set this field to the value of the TransactionID field of the corresponding KeyRequest message.

**ResultCode** The access network shall set this field according to Table 22.

**Table 22 Definition of result field**

Value	Meaning
0x00	Security capabilities verification and message integrity code successful
0x01	Message integrity code failed
0x02	Message integrity code successful, but capabilities verification failed.
0x03	Pairwise MasterKey not found.
0x04	Transaction ID invalid.
All other values	Reserved

**TokenCount** The access network sets this field to the number of tokens that the access network supports and includes in this message.

**SupportedToken** The access network shall set this field to a token supported by the access network.

**MessageIntegrityCode** The access network shall set this field to HMAC-SHA256-128(MICKey[i], Message), where Message is set to all fields of this message with this field set to zero, and *i* is the SessionKeyIndex field of the corresponding KeyRequest message.

<b>Channels</b>	FTC	<b>SLP</b>	Reliable
<b>Addressing</b>	Unicast	<b>Security</b>	Required

#### 4.2.7.2.5 ATKeyComplete

The access terminal sends the ATKeyComplete message in response to the ANKeyComplete message.

Field	Length (bits)
MessageID	8
TransactionID	8
ResultCode	8
MessageIntegrityCode	0 or 128
LastTransactionID	0 or 8

**MessageID** The access terminal shall set this field to 0x04.

**TransactionID** The access terminal shall set this field to the value of the TransactionID field of the corresponding KeyRequest message.

**ResultCode** The access terminal shall set this field according to Table 23.

**Table 23 Definition of result field**

Value	Meaning
0x00	Security capabilities verification and message integrity code successful
0x01	Message integrity code failed
0x02	Message integrity code successful, but capabilities or token verification failed.
0x03	Pairwise MasterKey not found.
0x04	Transaction ID invalid.
0x05	Key exchange procedures not supported.
All other values	Reserved

**MessageIntegrityCode** If the Result field is 0x01 or 0x03, the access terminal shall omit this field. Otherwise, the access terminal shall set this field to HMAC-SHA256-128(MICKey[i], *Message*), where *Message* is set to all fields of this message with this field set to zero, and *i* is the SessionKeyIndex field of the corresponding KeyRequest message.

**LastTransactionID** If the MessageIntegrityCode field is set to '0x04', then the access terminal shall set this field to the value of the LastValidTransactionID parameter. Otherwise, the access terminal shall omit this field.

<b>Channels</b>	RTC	<b>SLP</b>	Reliable
<b>Addressing</b>	Unicast	<b>Security</b>	Required

#### 4.2.7.2.6 KeyChangeInitiateRequest

The KeyChangeInitiateRequest message is sent by the access terminal to request the access network to initiate a key change.

Field	Length (bits)
MessageID	8
SessionKeyIndexRequested	8

**MessageID** The sender shall set this field to 0x05.

**SessionKeyIndexRequested** The sender shall set this value to the index of requested session key. The values 0x08 to 0xff are reserved.

<b>Channels</b>	RTC	<b>SLP</b>	Reliable
<b>Addressing</b>	Unicast	<b>Security</b>	Required

#### 4.2.7.2.7 KeyChangeRequest

The KeyChangeRequest message is sent the by the access network to initiate a key change.

Field	Length (bits)
MessageID	8
TransactionID	8
SessionKeyIndexPending	8

**MessageID** The sender shall set this field to 0x06.

**TransactionID** The sender shall set this field according to 10.8.

**SessionKeyIndexPending** The sender shall set this value to the index of proposed session key. The values 0x08 to 0xff are reserved.

<b>Channels</b>	FTC	<b>SLP</b>	Reliable
<b>Addressing</b>	Unicast	<b>Security</b>	Required

### 4.2.7.2.8 KeyChangeAck

The KeyChangeAck message is sent by the access terminal to acknowledge the receipt of a KeyChangeRequest message.

Field	Length (bits)
MessageID	8
TransactionID	8
SessionKeyIndexPending	8

**MessageID** The sender shall set this field to 0x07.

**TransactionID** The sender shall set this value to the TransactionID field of the corresponding KeyChangeRequest message.

**SessionKeyIndexPending** The sender shall set this value to the index of the proposed session key. The values 0x08 to 0xff are reserved.

<b>Channels</b>	RTC	<b>SLP</b>	Reliable
<b>Addressing</b>	Unicast	<b>Security</b>	Required

## 4.2.8 Interface to other protocols

### 4.2.8.1 Commands

This protocol does not issue any commands.

### 4.2.8.2 Indications

This protocol does not register to receive any indications.

## 4.2.9 Configuration attributes

The configurable, simple attributes for this protocol are listed in Table 24. The access terminal shall use as defaults the values in Table 24 that are listed in *bold italics*.

Unless specified otherwise, the access terminal and the access network shall use the Generic Attribute Update Protocol in 10.9 to update configurable attributes belonging to the Default Key Exchange Protocol. The access terminal or access network shall not use the Generic Attribute Update Protocol in 10.9 to update the SessionKeyIndexInUse attribute.

**Table 24 Configurable values**

Attribute ID	Attribute	Values	Meaning
0x00	SessionKeyIndexInUse	<i>0x00</i>	SKey <sub>0</sub> is used
		0x01	SKey <sub>1</sub> is used
		0x02	SKey <sub>2</sub> is used
		0x03	SKey <sub>3</sub> is used
		0x04	SKey <sub>4</sub> is used
		0x05	SKey <sub>5</sub> is used
		0x06	SKey <sub>6</sub> is used
		0x07	SKey <sub>7</sub> is used
		0x08 – 0xff	Reserved
0x01	SessionKeyIndexPending	0x00	SKey <sub>0</sub> is pending.
		<i>0x01</i>	SKey <sub>1</sub> is pending
		0x02	SKey <sub>2</sub> is pending
		0x03	SKey <sub>3</sub> is pending
		0x04	SKey <sub>4</sub> is pending
		0x05	SKey <sub>5</sub> is pending
		0x06	SKey <sub>6</sub> is pending
		0x07	SKey <sub>7</sub> is pending
		0x08 – 0xfe	Reserved
0xff	Pending Skey is not defined		

## 4.2.10 Protocol numeric constants

Constant	Meaning	Value
N <sub>KEPT</sub> <sub>type</sub>	Type field for this protocol	Table 9
N <sub>KEPG</sub>	Subtype field for this protocol	0x0001
T <sub>KEPT</sub> <sub>timer</sub>	Timer duration for response to KeyExchangeRequest message	500 ms

## 4.2.11 Session state information

The Session State Information record (see 10.10) consists of parameter records.

This protocol defines the following parameter record in addition to the configuration attributes for this protocol.

### 4.2.11.1 SKey parameter

**Table 25 Format of the parameter record for the SKey parameter**

Field	Length (bits)
ParameterType	8
Length	8
SKey <sub>0</sub> Included	1
SKey <sub>1</sub> Included	1
SKey <sub>2</sub> Included	1
SKey <sub>3</sub> Included	1
SKey <sub>4</sub> Included	1
SKey <sub>5</sub> Included	1
SKey <sub>6</sub> Included	1
SKey <sub>7</sub> Included	1
SKey <sub>0</sub>	0 or 384
SKey <sub>1</sub>	0 or 384
SKey <sub>2</sub>	0 or 384
SKey <sub>3</sub>	0 or 384
SKey <sub>4</sub>	0 or 384
SKey <sub>5</sub>	0 or 384
SKey <sub>6</sub>	0 or 384
SKey <sub>7</sub>	0 or 384

ParameterType This field shall be set to 0x01 for this parameter record.

Length This field shall be set to the length of this parameter record in units of octets excluding the Length field.

SKey<sub>0</sub>Included If SKey<sub>0</sub> is zero, then this field shall be set to '0'. Otherwise, this field shall be set to '1'.

SKey<sub>1</sub>Included If SKey<sub>1</sub> is zero, then this field shall be set to '0'. Otherwise, this field shall be set to '1'.

SKey<sub>2</sub>Included If SKey<sub>2</sub> is zero, then this field shall be set to '0'. Otherwise, this field shall be set to '1'.

1	SKey <sub>3</sub> Included	If SKey <sub>3</sub> is zero, then this field shall be set to '0'. Otherwise, this field shall
2		be set to '1'.
3	SKey <sub>4</sub> Included	If SKey <sub>4</sub> is zero, then this field shall be set to '0'. Otherwise, this field shall
4		be set to '1'.
5	SKey <sub>5</sub> Included	If SKey <sub>5</sub> is zero, then this field shall be set to '0'. Otherwise, this field shall
6		be set to '1'.
7	SKey <sub>6</sub> Included	If SKey <sub>6</sub> is zero, then this field shall be set to '0'. Otherwise, this field shall
8		be set to '1'.
9	SKey <sub>7</sub> Included	If SKey <sub>7</sub> is zero, then this field shall be set to '0'. Otherwise, this field shall
10		be set to '1'.
11	SKey <sub>0</sub>	If SKey <sub>0</sub> Included is '0', then this field shall be omitted. Otherwise, this field
12		shall be set to the value of the session key with key index 0x00.
13	SKey <sub>1</sub>	If SKey <sub>1</sub> Included is '0', then this field shall be omitted. Otherwise, this field
14		shall be set to the value of the session key with key index 0x01.
15	SKey <sub>2</sub>	If SKey <sub>2</sub> Included is '0', then this field shall be omitted. Otherwise, this field
16		shall be set to the value of the session key with key index 0x02.
17	SKey <sub>3</sub>	If SKey <sub>3</sub> Included is '0', then this field shall be omitted. Otherwise, this field
18		shall be set to the value of the session key with key index 0x03.
19	SKey <sub>4</sub>	If SKey <sub>4</sub> Included is '0', then this field shall be omitted. Otherwise, this field
20		shall be set to the value of the session key with key index 0x04.
21	SKey <sub>5</sub>	If SKey <sub>5</sub> Included is '0', then this field shall be omitted. Otherwise, this field
22		shall be set to the value of the session key with key index 0x05.
23	SKey <sub>6</sub>	If SKey <sub>6</sub> Included is '0', then this field shall be omitted. Otherwise, this field
24		shall be set to the value of the session key with key index 0x06.
25	SKey <sub>7</sub>	If SKey <sub>7</sub> Included is '0', then this field shall be omitted. Otherwise, this field
26		shall be set to the value of the session key with key index 0x07.

### 4.2.11.2 Nonce parameter

**Table 26 Format of the parameter record for the Nonce parameter**

Field	Length (bits)
ParameterType	8
Length	8
NULLATNonce	1
NULLANNonce	1
Reserved	6
ATNonce	0 or 256
ANNonce	0 or 256

ParameterType	This field shall be set to 0x02 for this parameter record.
Length	This field shall be set to the length of this parameter record in units of octets excluding the Length field.
NULLATNonce	If ATNonce is NULL, then this field shall be set to '1'. Otherwise, this field shall be set to '0'.
NULLANNonce	If ANNonce is NULL, then this field shall be set to '1'. Otherwise, this field shall be set to '0'.
Reserved	This field shall be set to '000000'. The receiver shall ignore this field.
ATNonce	If NULLATNonce is '1', then this field shall be omitted. Otherwise, this field shall be set to the value of the ATNonce.
ANNonce	If NULLANNonce is '1', then this field shall be omitted. This field shall be set to the value of the ANNonce.

### 4.2.11.3 LastValidTransactionID parameter

**Table 27 Format of the parameter record for the LastValidTransactionID parameter**

Field	Length (bits)
ParameterType	8
Length	8
LastValidTransactionID	8

ParameterType	This field shall be set to 0x03 for this parameter record.
Length	This field shall be set to the length of this parameter record in units of octets, excluding the Length field.
LastValidTransactionID	This field shall be set to the value of the LastValidTransactionID parameter.



#### 4.2.11.4 PMK parameter

**Table 28 Format of the parameter record for the PMK parameter**

Field	Length (bits)
ParameterType	8
Length	8
PMKCount	8
PMKCount occurrences of the following two fields:	
PMKLength	8
PMK	PMKLength × 8

**ParameterType** This field shall be set to 0x04 for this parameter record.

**Length** This field shall be set to the length of this parameter record in units of octets excluding the Length field.

**PMKCount** This field shall be set to the number of occurrences of the PMK field in this parameter record.

**PMKLength** This field shall be set to the length of the PMK field in units of octets.

**PMK** This field shall be set to a PairwiseMasterKey.

## 5 Security Sublayer

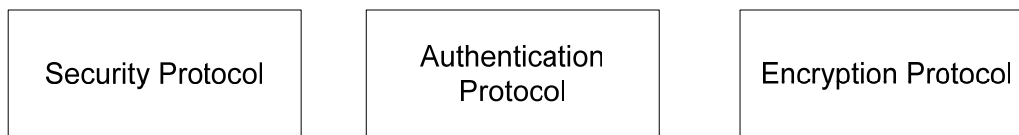
### 5.1 Introduction

#### 5.1.1 General overview

The Security Sublayer provides the following functions:

- **Cryptosync Generation:** Provides a cryptosync for use by the Authentication and Encryption protocols in the Security Sublayer
- **Authentication:** Provides the procedures followed by the access network and the access terminal for authenticating traffic.
- **Encryption:** Provides the procedures followed by the access network and the access terminal for encrypting traffic.

The Security Sublayer uses the Authentication Protocol, Encryption Protocol, and Security Protocol to provide these functions. In particular, the Security Protocol provides the cryptosync needed by the authentication and encryption protocols, the Authentication Protocol provides authentication, and the Encryption Protocol provides encryption. Figure 43 shows the protocols within the Security Sublayer.



**Figure 43 Security Sublayer protocols**

### 5.2 Packet encapsulation for the protocol instances

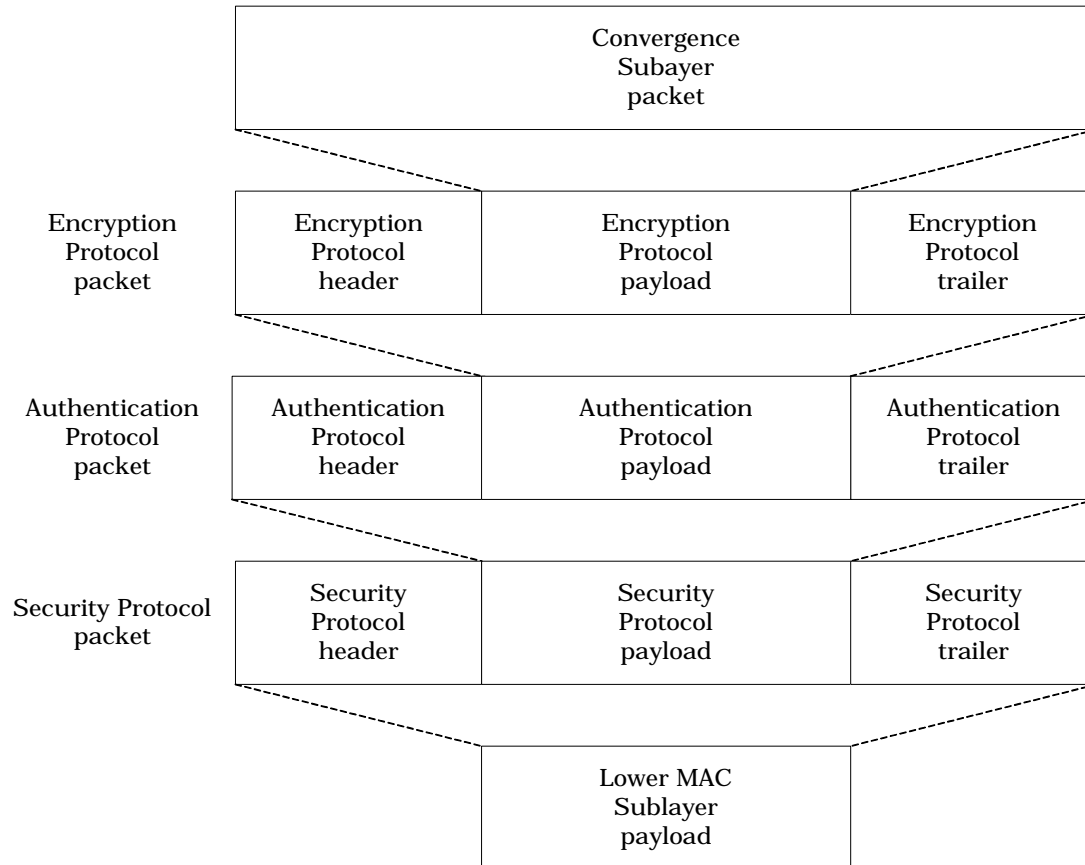
In the transmit direction, the Security Sublayer receives a Convergence Sublayer Packet, accompanied by a IsSecure field. The Security Sublayer processes this packet and delivers a Lower MAC Payload to the Lower MAC Sublayer, accompanied by the IsSecure field.

In the receive direction, the Security Sublayer receives a Lower MAC Sublayer Packet, accompanied by a IsSecure field. The Security Sublayer processes this packet and delivers a Convergence Sublayer Packet to the Convergence Sublayer, accompanied by a IsSecure field.

Packet encapsulation for the Security Sublayer operates in a different way for the secure and unsecure packets, as described in next.

## 5.2.1 Packet encapsulation with IsSecure set

When the IsSecure field is set to '1', Figure 44 illustrates the relationship between a Convergence Sublayer packet, an Encryption Protocol packet, an Authentication Protocol packet, a Security Sublayer packet, and the Lower MAC Sublayer payload. The order of Authentication and Encryption is such that it can avoid unnecessary decryption when authentication fails.



**Figure 44 Security Sublayer data encapsulation for IsSecure=1**

The Security Sublayer headers or trailers may or may not be present (or equivalently, have a size of zero) if the SessionConfigurationToken specifies the Default Security Protocol or if the configured Security Protocols do not require a header or trailer.

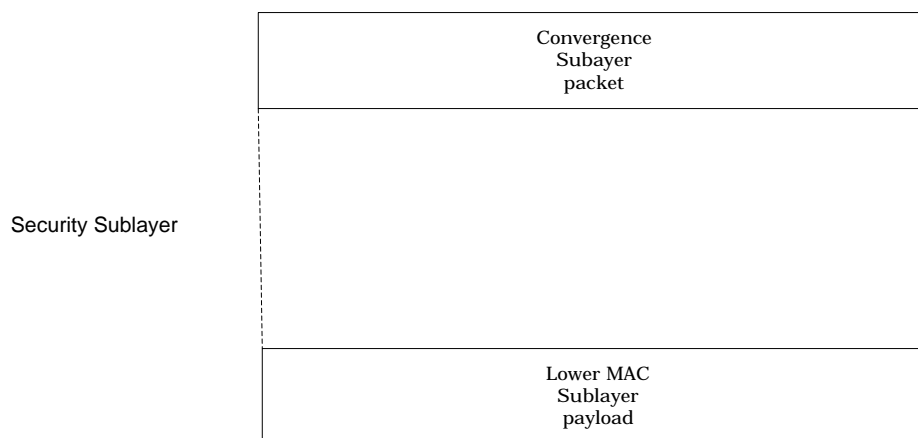
The Encryption Protocol may add a trailer to hide the actual length of the plaintext or padding to be used by the encryption algorithm. The Encryption Protocol Header may contain variables such as an initialization vector (IV) to be used by the Encryption Protocol.

The Authentication Protocol header or trailer may contain the Message Authentication Code that is used to authenticate the portion of the Authentication Protocol Packet that is authenticated.

The Security Protocol header or trailer may contain variables needed by the authentication and encryption protocols (e.g., cryptosync, time-stamp, etc.).

## 5.2.2 Packet encapsulation with IsSecure not set

If the IsSecure field is set to zero, the relation between a Convergence Sublayer Packet and a Lower MAC Sublayer payload is as shown in Figure 45. The packet does not pass through the Security Protocol, Authentication Protocol and Encryption Protocol.



**Figure 45 Security Sublayer data encapsulation for IsSecure=0**

## 5.2.3 Security Sublayer data transmit operation overview

When a Convergence Sublayer Packet with IsSecure=0 is delivered to the Security Sublayer, the Security Sublayer sets the Lower MAC Sublayer payload to the Convergence Sublayer Packet.

When a Convergence Sublayer packet with IsSecure=1 is delivered to the Security Sublayer, the following steps are performed by the protocols in the Security Sublayer in the order specified in the following:

- The Security Protocol generates a cryptosync for the channel for which the Convergence Sublayer packet is destined. This value is called TheCryptosync for referencing it in the following steps.
- The Convergence Sublayer packet and TheCryptosync are delivered to the Encryption Protocol.
- The Encryption Protocol uses TheCryptosync, the encryption key, and other parameters specified by the Encryption Protocol (if any) to encrypt the Convergence Sublayer packet and construct the Encryption Protocol packet.
- The Encryption Protocol delivers the Encryption Protocol packet and TheCryptosync to the Authentication Protocol.
- The Authentication Protocol uses TheCryptosync, authentication key, and other parameters specified by the Authentication Protocol to construct the Authentication Protocol packet.
- The Authentication Protocol delivers the Security Sublayer packet to the Security Protocol.
- The Security Protocol delivers the Security Sublayer packet and other specified parameters (if any) to the Lower MAC Sublayer.

## 5.2.4 Security Sublayer data receive operation overview

When a Lower MAC Sublayer payload with IsSecure=0 is delivered to the Security Sublayer, the Security Sublayer sets the Convergence Sublayer Packet to the Lower MAC Sublayer payload.

When the Security Sublayer receives a Lower MAC Sublayer Packet payload with IsSecure=1, the following steps are performed by the protocols in the Security Sublayer in the order specified below:

- The Security Protocol constructs the cryptosync using information from the Lower MAC Sublayer and the Security Sublayer Protocol header and trailer (if any). For the purpose for referencing this value of cryptosync in the following steps, denote this value as TheCryptosync.
- The Security Protocol removes the Security Protocol header and trailer (if any) and delivers TheCryptosync and the Security Protocol payload to the Authentication Protocol.
- The Authentication Protocol uses TheCryptosync, authentication key, Authentication Protocol payload, Authentication Protocol header and trailer, and other parameters specified by the Authentication Protocol (if any) to verify the authentication signature. If the authentication signature passes, the Authentication Protocol delivers the Authentication Protocol payload to the Encryption Protocol; otherwise, the Authentication Protocol Packet is discarded.
- The Encryption Protocol uses TheCryptosync and the encryption key to decrypt the Encryption Protocol packet. The decrypted payload is then delivered to the Convergence Sublayer.

## 5.3 Default Encryption Protocol

### 5.3.1 Overview

The Default Encryption Protocol does not alter the Convergence Sublayer packet payload and does not add an Encryption Protocol header or trailer. It transfers packets between the Authentication Protocol and the Security Protocol.

### 5.3.2 Primitives

#### 5.3.2.1 Commands

This protocol does not define any commands.

#### 5.3.2.2 Return indications

This protocol does not return any indications.

### 5.3.3 Public data

#### 5.3.3.1 Static public data

This protocol does not define any static public data.

#### 5.3.3.2 Dynamic public data

- Subtype for this protocol

### 1 **5.3.4 Protocol data unit**

2 The protocol data unit for this protocol is an Encryption Protocol Packet.

### 3 **5.3.5 Protocol initialization and swap**

#### 4 **5.3.5.1 Protocol initialization**

5 Upon initialization, the value of the attributes for this protocol instance in the access terminal and the  
6 access network shall be set to the default values specified for each attribute.

#### 7 **5.3.5.2 Protocol swap**

8 This protocol defines an empty swap procedure.

### 9 **5.3.6 Procedures**

10 On the transmit side, this protocol shall receive a Convergence Sublayer packet, and it shall forward  
11 the packet to Authentication Protocol.

12 On the receive side, this protocol shall receive an Authentication Protocol packet, and it shall forward  
13 the packet to Convergence Sublayer.

### 14 **5.3.7 Default Encryption Protocol header and trailer**

15 The Default Encryption Protocol does not add a header or a trailer.

### 16 **5.3.8 Message formats**

17 No messages are defined for this protocol.

### 18 **5.3.9 Interface to other protocols**

#### 19 **5.3.9.1 Commands**

20 This protocol does not issue any commands.

#### 21 **5.3.9.2 Indications**

22 This protocol does not register to receive any indications.

### 23 **5.3.10 Configuration attributes**

24 No configuration attributes are defined for this protocol.

### 25 **5.3.11 Protocol numeric constants**

26

Constant	Meaning	Value
N <sub>EPT</sub> Type	Type field for this protocol	Table 9
N <sub>EP</sub> Default	Subtype field for this protocol	0x0000

### 1 **5.3.12 Session state information**

2 The Session State Information record (see 10.10) consists of parameter records.

3 The parameter records for this protocol consist of the configuration attributes of this protocol.

## 4 **5.4 Default Security Protocol**

### 5 **5.4.1 Overview**

6 The Default Security protocol performs the following tasks:

- 7 ■ Procedures to enter a secure mode of operation, where the access network and access  
8 terminal may secure all unicast air interface packets.
- 9 ■ On the transmission side,
  - 10 □ This protocol generates the cryptosync based on information provided by the Lower  
11 MAC Sublayer and makes the cryptosync publicly available. The cryptosync may be  
12 used by the negotiated Authentication Protocol and Encryption Protocol.
  - 13 □ This protocol transfers packets from the Authentication Protocol to the Lower MAC  
14 Sublayer.
- 15 ■ On the receiving side,
  - 16 □ This protocol generates the cryptosync based on information provided by the Lower  
17 MAC Sublayer and makes the cryptosync publicly available. The cryptosync may be  
18 used by the negotiated Authentication Protocol and Encryption Protocol.
  - 19 □ This protocol transfers packets from the Lower MAC Sublayer to the Authentication  
20 Protocol

### 21 **5.4.2 Primitives**

#### 22 **5.4.2.1 Commands**

23 This protocol does not define any commands.

#### 24 **5.4.2.2 Return indications**

25 This protocol does not return any indications.

### 26 **5.4.3 Public data**

#### 27 **5.4.3.1 Static public data**

28 This protocol does not define any static public data.

#### 29 **5.4.3.2 Dynamic public data**

- 30 ■ Subtype for this protocol
- 31 ■ Cryptosync for Security Sublayer packets associated with the FL

- 1           ■ Cryptosync for Security Sublayer packets associated with the RL
- 2           ■ SecurityEnabled

### 3   **5.4.4 Protocol data unit**

4   The protocol data unit for this protocol is a Security Sublayer packet.

### 5   **5.4.5 Protocol initialization and swap**

#### 6   **5.4.5.1 Protocol Initialization**

7   Upon initialization, the value of the attributes for this protocol instance in the access terminal and the  
8   access network shall be set to the default values specified for each attribute.

#### 9   **5.4.5.2 Protocol swap**

10   This protocol defines an empty swap procedure.

### 11   **5.4.6 Procedures**

#### 12   **5.4.6.1 Secure State Procedures**

13   This protocol shall set the SecurityEnabled public data field to the SecurityEnabled configuration  
14   attribute.

15   This protocol shall be said to be in a secure mode of operation (SecurityEnabled mode) if the  
16   SecurityEnabled public data is set to 1.

17   In the SecurityEnabled mode at the access terminal

- 18           ■ This protocol shall set the IsSecure bit to ‘1’ on all transmitted packets

19   In the SecurityEnabled mode at the access network

- 20           ■ This protocol shall set the IsSecure bit to ‘1’ on all unicast packets transmitted to the  
21           access terminal.

#### 22   **5.4.6.2 Generation of the Cryptosync**

23   The Security Protocol shall compute the Cryptosync for the channel on which the Security Sublayer  
24   packet is to be sent, or on the channel on which the Security Sublayer packet is received as shown in  
25   Table 29.

26                   **Table 29 Subfield of the Cryptosync**

Subfield	Length (bits)
MACID	12
PilotPN	12
ConnectCount	16
CryptoAttribute	16
PhyFrameIndex	40



- 1    MACID                      This field shall be set to the MACID of the sending sector in the FL and  
2                                      target sector in the RL with zero padding on the MSB side if needed. A  
3                                      MACID larger than 12 bits is not supported.
- 4    PilotPN                      This field shall be set to the PilotPN of the sending sector in the FL or, target  
5                                      sector in the RL with zero padding on the MSB side if needed.
- 6    ConnectCount                This field shall be set to the current value of ConnectCount, as defined in the  
7                                      public data of the Idle State Protocol.
- 8    CryptoAttribute              This field is encoded as specified in Table 30.

9                                      **Table 30 Encoding of the CryptoAttribute Field**

Bit location	Name	Meaning
0 (LSB)	ISFL	IsFL='1' implies FL; IsFL='0' implies RL
1	ISSticky	IsSticky='1' implies channel assignment is sticky; IsSticky='0' implies assignment is non-sticky.
Others	Reserved	Not defined

- 10   PHY Frame Index            This field shall be set to the value of the PHY Frame Index as defined in the  
11                                      Overview chapter (“Definitions” section), with bits of zero padding on the  
12                                      MSB side if necessary. The PHY frame index shall be measured at the  
13                                      beginning of the packet transmission, with respect to the sector that is  
14                                      receiving or transmitting the packet.

### 15    5.4.6.3 Transmit procedures

16    When this protocol receives an Authentication Protocol packet from the Authentication Protocol,

- 17        ■ It shall construct a Security Sublayer packet by adding a Security Protocol header and  
18                                      trailer (if any)
- 19        ■ Compute the Cryptosync associated with the Security Sublayer packet as shown in  
20                                      Table 30.
- 21        ■ Deliver the packet for transmission to the Lower MAC Sublayer

### 22    5.4.6.4 Receive procedures

23    When this protocol receives a Security Sublayer packet from the Lower MAC Sublayer, the protocol  
24    shall

- 25        ■ Construct an Authentication Protocol packet by removing the Security Protocol header  
26                                      and trailer (if any),
- 27        ■ Compute the Cryptosync associated with the Lower MAC Sublayer packet as shown in  
28                                      Table 30.
- 29        ■ Deliver the Authentication Protocol packet together with the computed value of the  
30                                      cryptosync to the Authentication Protocol.

## 5.4.7 Header and trailer

The Default Security Protocol does not add a header or a trailer.

## 5.4.8 Message formats

No messages are defined for this protocol.

## 5.4.9 Interface to other protocols

### 5.4.9.1 Commands

This protocol does not issue any commands.

### 5.4.9.2 Indications

This protocol registers to receive the following indications

- *KeyExchange.FirstKeyComplete*

### 5.4.10 Configuration attributes

This protocol defines the following configuration attributes.

The access terminal and the access network shall not use the Generic Attribute Update Protocol in 10.9 to update configurable attributes belonging to this protocol. The SecurityEnabled attribute shall be defined by the SessionConfigurationToken of the Session Configuration Protocol.

**Table 31 Configurable values**

Attribute ID	Attribute	Values	Meaning
0x00	SecurityEnabled	0x00	SecurityEnabled mode is off
		0x01	SecurityEnabled mode is on.
		0x02 – 0xff	Reserved

### 5.4.11 Protocol numeric constants

Constant	Meaning	Value
N <sub>SPT</sub> Type	Type field for this protocol	Table 9
N <sub>SPG</sub>	Subtype field for this protocol	0x0000
T <sub>SPSecurityConfirmWait</sub>	Duration the access network waits before retransmitting the EnableSecurityAssignment.	1 s
T <sub>SPSecurityResponseWait</sub>	Duration the initiator waits for a EnableSecurityResponse message	1 s

## 5.4.12 Session state information

The Session State Information record (see 10.10) consists of parameter records.

The parameter records for this protocol consist of the configuration attributes of this protocol.

## 5.5 Default Authentication Protocol

### 5.5.1 Overview

The Default Authentication Protocol provides a method for authentication of packets by applying (on the transmit side) and checking (on the receive side) the HMAC-SHA256 message authentication function to *message bits* that are composed of the Authentication Protocol payload, CryptoSync, together with FLAuthKey or RLAuthKey, as appropriate. The HMAC-SHA256 function is defined in RFC 2104 [10] with SHA-256 [1] as the message digest algorithm.

### 5.5.2 Primitives

#### 5.5.2.1 Commands

This protocol does not define any commands.

#### 5.5.2.2 Return indications

- *Failed*

### 5.5.3 Public data

#### 5.5.3.1 Static public data

This protocol does not define any static public data.

#### 5.5.3.2 Dynamic public data

- Subtype for this protocol

### 5.5.4 Protocol data unit

The protocol data unit for this protocol is an Authentication Protocol packet.

### 5.5.5 Protocol initialization and swap

#### 5.5.5.1 Protocol initialization

Upon initialization, the value of the attributes for this protocol instance in the access terminal and the access network shall be set to the default values specified for each attribute.

#### 5.5.5.2 Protocol swap

This protocol defines an empty swap procedure.

## 5.5.6 Procedures

On the transmit side: When this protocol receives an Encryption Protocol Packet, it shall add the Authentication Protocol Header defined in 5.5.6.2 in front of the Encryption Protocol Packet and shall forward the newly generated Authentication Protocol Packet to the Security Protocol.

On the receive side: When this protocol receives a Security Sublayer packet from the Security Protocol, it shall check the message authentication code in the Authentication Protocol Header. If the message authentication code passes, this protocol removes the Authentication Protocol Header and shall forward the newly generated Authentication Protocol Packet to the Encryption Protocol.

### 5.5.6.1 Access terminal requirements

#### 5.5.6.1.1 Transmit Procedures

Upon reception of an Encryption Protocol packet destined for transmission the access terminal shall compute the packet authentication code (PAC) as follows:

- The access terminal shall construct the AuthKey as follows:
  - If the Key Exchange Protocol does not define RLAAuthKey as public data, this protocol shall discard the packet.
  - Otherwise, this protocol shall perform the following:
    - If the length of RLAAuthKey is equal to the length of AuthKey, then AuthKey shall be RLAAuthKey.
    - Otherwise, if the length of RLAAuthKey is greater than the length of AuthKey, then AuthKey shall be the  $N_{\text{APAuthKeyLength}}$  least significant bits of RLAAuthKey.
    - Otherwise, if the length of RLAAuthKey is less than the length of AuthKey this protocol shall discard the packet.
- The access terminal shall construct the cryptosync as described by the Security Protocol.
- The access terminal shall construct the *message bits* for computing the PAC as shown in Table 32.

**Table 32 Message bits for AT PAC computation**

Field	Length(bits)
Authentication Protocol Payload	Variable
Cryptosync	96

- The access terminal shall compute the *message digest* as HMAC-SHA256(AuthKey, *message bits*). The PAC field shall be set to the  $N_{\text{APMessageAuthCodeLength}}$  least significant bits of the *message digest*.

### 5.5.6.1.2 Receive Procedures

Upon reception of an Authentication Protocol packet, the access terminal shall compute and verify the Lower MAC Sublayer packet authentication code (PAC) given in the authentication protocol header as follows:

- The access terminal shall construct the AuthKey as follows:
  - If the Key Exchange Protocol does not define FLAuthKey as public data, this protocol shall discard the packet.
  - Otherwise, the access terminal shall perform the following:
    - If the length of FLAuthKey is equal to the length of AuthKey, then AuthKey shall be FLAuthKey.
    - Otherwise, if the length of FLAuthKey is greater than the length of AuthKey, then AuthKey shall be the  $N_{APAuthKeyLength}$  least significant bits of FLAuthKey.
    - Otherwise, if the length of FLAuthKey is less than the length of AuthKey, this protocol shall discard the packet.
  - The access terminal shall use the cryptosync provided by the Security Protocol.
- The access terminal shall construct the message bits for computing the PAC as shown in Table 33.
- The access terminal shall compute the *message digest* as HMAC-SHA256(AuthKey, *message bits*). The PAC field shall be set to the  $N_{APMessageAuthCodeLength}$  least significant bits of the *message digest*.

If the PAC computed in the previous step matches the PAC field in the Protocol Header, then the Protocol shall deliver the Authentication Sublayer Payload to the Encryption Protocol. Otherwise, the Protocol shall issue a *Failed* indication and shall discard the Security Sublayer packet.

### 5.5.6.2 Access network requirements

#### 5.5.6.2.1 Transmit Procedures

Upon reception of an Encryption Protocol packet destined for transmission the access network shall compute the packet authentication code (PAC) as follows:

- The access terminal shall construct AuthKey as follows:
  - If the Key Exchange Protocol does not define FLAuthKey as public data, the access terminal shall discard the packet.
  - Otherwise, the access network shall perform the following:
    - If the length of FLAuthKey is equal to the length of AuthKey, then AuthKey shall be FLAuthKey.
    - Otherwise, if the length of FLAuthKey is greater than the length of AuthKey, then AuthKey shall be the  $N_{APAuthKeyLength}$  least significant bits of FLAuthKey.
    - Otherwise, if the length of FLAuthKey is less than the length of AuthKey, then the access network shall discard the packet.

- 1 ■ The access network shall construct the *message bits* for computing the PAC as shown in  
2 Table 33.

3 **Table 33 Message bits for AN PAC computation**

Field	Length(bits)
Authentication Protocol Payload	Variable
Cryptosync	96

- 4
- 5 ■ The access network shall compute the *message digest* as HMAC-SHA256(AuthKey,  
6 *message bits*). The PAC field shall be set to the  $N_{APMessageAuthCodeLength}$  least significant bits  
7 of the *message digest*.

### 8 5.5.6.2.2 Receive Procedures

9 Upon reception of an Authentication Protocol packet the access network shall compute and verify the  
10 Lower MAC Sublayer Packet Authentication Code (PAC) given in the authentication protocol header  
11 as follows:

- 12 ■ The access network shall construct the AuthKey as follows:
- 13 □ If the Key Exchange Protocol does not define RLAAuthKey as public data, the access  
14 network shall discard the packet.
- 15 □ Otherwise, the access network shall perform the following:
- 16 – If the length of RLAAuthKey is equal to the length of AuthKey, then AuthKey  
17 shall be RLAAuthKey.
- 18 – Otherwise, if the length of RLAAuthKey is greater than the length of AuthKey,  
19 then AuthKey shall be the  $N_{APAuthKeyLength}$  least significant bits of RLAAuthKey.
- 20 – Otherwise, if the length of RLAAuthKey is less than the length of AuthKey, then  
21 the access network shall discard the packet.
- 22 □ The access network shall use the cryptosync provided by the Security Protocol.
- 23 ■ The access network shall construct the *message bits* for computing PAC as shown in  
24 Table 32.
- 25 ■ The access network shall compute the *message digest* as HMAC-SHA256(AuthKey,  
26 *message bits*). The PAC shall be set to the  $N_{APMessageAuthCodeLength}$  least significant bits of  
27 the *message digest*.

28 If the PAC computed in the previous step matches the PAC field in the Protocol Header, then the  
29 Protocol shall deliver the Authentication Protocol Payload to the Encryption Protocol. Otherwise, the  
30 Protocol shall issue a *Failed* indication and shall discard the Security Sublayer packet.

## 5.5.7 Header and trailer

### 5.5.7.1 Header

The Default Authentication Protocol header is defined as follows:

Field	Length (bits)
PAC	0 or $N_{APMessageAuthCodeLength}$

PAC Packet Authentication Code. This field shall be computed as specified in 5.5.6.1. This field shall be included if UATIInfoIncluded bit associated with the packet is '1' or the configuration attribute AuthenticationMode is equal to '1'.

### 5.5.7.2 Trailer

The Default Authentication Protocol does not add a trailer.

## 5.5.8 Message formats

No messages are defined for this protocol.

## 5.5.9 Interface to other protocols

### 5.5.9.1 Commands

This protocol does not issue any commands.

### 5.5.9.2 Indications

This protocol does not register to receive any indications.

## 5.5.10 Configuration attributes

This protocol defines the following configuration attributes.

The access terminal and the access network shall not use the Generic Attribute Update Protocol in 10.9 to update configurable attributes belonging to this protocol. The AuthenticationMode attribute shall be defined by the SessionConfigurationToken of the Session Configuration Protocol.

**Table 34 Configurable values**

Attribute ID	Attribute	Values	Meaning
0x00	AuthenticationMode	0x00	Only packets with UATIInfoIncluded=1 and IsSecure=1 in the Lower MAC header are authenticated.
		0x01	All packets with IsSecure=1 in the Lower MAC header are authenticated
		0x02 – 0xff	Reserved

### 5.5.11 Protocol numeric constants

Constant	Meaning	Value
N <sub>APType</sub>	Type field for this protocol	Table 9 (0x06)
N <sub>APG</sub>	Subtype field for this protocol	0x0000
N <sub>APMessageAuthCodeLength</sub>	Number of bits in the message authentication code	0x0060
N <sub>APAAuthKeyLength</sub>	Length of the authentication key	0x00A0

### 5.5.12 Session state information

The Session State Information record (see 10.10) consists of parameter records.

The parameter records for this protocol consist of the configuration attributes of this protocol.

## 5.6 Generic Encryption Protocol

### 5.6.1 Overview

The Generic Encryption Protocol uses the AES (or, Rijndael) procedures defined in [16] in order to encrypt the Convergence Sublayer packets and decrypt the Authentication Protocol packets.

### 5.6.2 Primitives

#### 5.6.2.1 Commands

This protocol does not define any commands.

#### 5.6.2.2 Return indications

This protocol does not return any indications.



### 1 5.6.3 Public data

#### 2 5.6.3.1 Static public data

3 This protocol does not define any static public data.

#### 4 5.6.3.2 Dynamic public data

- 5 ■ Subtype for this protocol

### 6 5.6.4 Protocol data unit

7 The protocol data unit for this protocol is an Encryption Protocol Packet.

### 8 5.6.5 Protocol initialization and swap

#### 9 5.6.5.1 Protocol initialization

10 Upon initialization, the value of the attributes for this protocol instance in the access terminal and the  
11 access network shall be set to the default values specified for each attribute.

#### 12 5.6.5.2 Protocol swap

13 This protocol defines an empty swap procedure.

### 14 5.6.6 Procedures

15 On the transmit side: When this protocol receives a Convergence Sublayer packet and the cryptosync  
16 from the Security Protocol, it shall follow the transmit procedures specified in 5.6.6.3.

17 On the receive side: When this protocol receives a Security Sublayer packet and the cryptosync from  
18 the Security Protocol, it shall follow the receive procedures specified in 5.6.6.4.

#### 19 5.6.6.1 Constructing the encryption key for the FL

20 The Generic Encryption Protocol shall construct the encryption keys for the FL as follows:

- 21 ■ the protocol shall construct the encryption key for the Forward Link Channels,  
22 FLEncryptionKey, as follows:
  - 23 □ If the Key Exchange Protocol does not define FLEncKey as public data, this protocol  
24 shall discard all packets.
  - 25 □ Otherwise, this protocol shall perform the following:
    - 26 – If the length of FLEncKey is equal to  $N_{\text{EPencKeyLength}}$ , then FLEncryptionKey shall  
27 be set to FLEncKey.
    - 28 – Otherwise, if the length of FLEncKey is greater than  $N_{\text{EPencKeyLength}}$ , then  
29 FLEncryptionKey shall be the  $N_{\text{EPencKeyLength}}$  most significant bits of FLEncKey.
    - 30 – Otherwise, if the length of FLEncKey is less than  $N_{\text{EPencKeyLength}}$ , this protocol  
31 shall discard all packets.

### 1 5.6.6.2 Constructing the encryption key for the RL

2 The Generic Encryption Protocol shall construct the encryption keys for the RL as follows:

- 3 ■ The protocol shall construct the encryption key for the Reverse Link Channels,  
4 RLEncryptionKey, as follows:
  - 5 □ If the Key Exchange Protocol does not define RLEncKey as public data, this protocol  
6 shall discard all packets.
  - 7 □ Otherwise, the protocol shall perform the following:
    - 8 – If the length of RLEncKey is equal to  $N_{\text{EPEncKeyLength}}$ , then RLEncryptionKey shall  
9 be set to RLEncKey.
    - 10 – Otherwise, if the length of RLEncKey is greater than  $N_{\text{EPEncKeyLength}}$ , then  
11 RLEncryptionKey shall be the  $N_{\text{EPEncKeyLength}}$  most significant bits of RLEncKey.
    - 12 – Otherwise, if the length of RLEncKey is less than  $N_{\text{EPEncKeyLength}}$ , this protocol  
13 shall discard all packets.

### 14 5.6.6.3 Transmit procedures

15 The protocol shall construct the Encryption Protocol packet from the Convergence Sublayer packet  
16 that is destined for FL or RL by performing the following for each of the channels:

- 17 ■ The protocol shall call the ESP\_AES procedure specified in [16] with its inputs set as  
18 follows:
  - 19 □ Set *key* to the EncryptionKey for the channel under consideration (e.g.,  
20 FLEncryptionKey).
  - 21 □ Set *fresh* to the value of the cryptosync provided by the Security Protocol.
  - 22 □ Set *freshsize* to 96.
  - 23 □ Set *buf* to the address of the beginning of the memory space that contains the  
24 Convergence Sublayer packet.
  - 25 □ Set *bit\_offset* to zero.
  - 26 □ Set *bit\_count* to the length of the Convergence Sublayer Packet in bits.
- 27 ■ After the ESP\_AES procedure is returned, the protocol shall set the Encryption Protocol  
28 packet to the output of the ESP\_AES procedure that starts at the memory space specified  
29 by *buf* and is of the same size as the Convergence Sublayer packet.

### 30 5.6.6.4 Receive procedures

31 If the Encryption Protocol packet is received on the FL or RL, then the receiver shall construct the  
32 Convergence Sublayer packet from the Encryption Protocol packet by performing the following for  
33 each of the channels:

- 34 ■ The protocol shall call the ESP\_AES procedure specified in [16] with its inputs set as  
35 follows:
  - 36 □ Set *key* to the EncryptionKey for the channel under consideration (e.g.,  
37 FLEncryptionKey).
  - 38 □ Set *fresh* to the value of the cryptosync provided by the Security Protocol.

- 1           □ Set *freshsize* to 96.
- 2           □ Set *buf* to the address of the beginning of the memory space that contains the
- 3           Encryption Protocol packet.
- 4           □ Set *bit\_offset* to zero.
- 5           □ Set *bit\_count* to the length of the Encryption Protocol Packet in bits.
- 6           ■ After the ESP\_AES procedure is returned, the protocol shall set the Convergence
- 7           Sublayer packet to the output of the ESP\_AES procedure which starts at the memory
- 8           space specified by *buf* and is of the same size as the Encryption Protocol packet.

### 9   **5.6.7 Generic Encryption Protocol header and trailer**

10 The Generic Encryption Protocol does not add a header or a trailer.

### 11   **5.6.8 Message formats**

12 No messages are defined for this protocol.

### 13   **5.6.9 Interface to other protocols**

#### 14   **5.6.9.1 Commands**

15 This protocol does not issue any commands.

#### 16   **5.6.9.2 Indications**

17 This protocol does not register to receive any indications.

### 18   **5.6.10 Configuration attributes**

19 No configuration attributes are defined for this protocol.

### 20   **5.6.11 Protocol numeric constants**

Constant	Meaning	Value
N <sub>EPT</sub> Type	Type field for this protocol	<b>Table 9</b>
N <sub>EPG</sub>	Subtype field for this protocol	0x0001
N <sub>EP</sub> EncKeyLength	Length of the encryption key	0x00A0

### 23   **5.6.12 Session state information**

24 The Session State Information record (see 10.10) consists of parameter records.

25 The parameter records for this protocol consist of the configuration attributes of this protocol.

## 6 Lower MAC Control Sublayer

### 6.1 Introduction

#### 6.1.1 General overview

The Lower MAC Control sublayer controls the state of the air-link by managing the states of individual Lower MAC sublayer protocols, and by providing individual MAC Sublayer protocols with operating parameters. The protocols in this sublayer are control protocols, and do not carry data on the behalf of other protocols. The protocols in this sublayer use the Signaling Transport to transmit and receive messages (with the exception of the Overhead Messages Protocol that sends some information blocks using the Lower MAC Sublayer).

The access terminal and the access network maintain a connection whose state dictates the form in which communications between these entities can take place. The connection can be either of the following three states:

- *Closed Connection and no assigned MAC ID*: The access terminal is not assigned any dedicated air-link resources. Communications between the access terminal and the access network are conducted over the R-ACH, F-pBCH0, F-pBCH1, F-SSCH, and F-DCH physical layer channels.
- *Closed Connection and assigned MAC ID*: This is an intermediate state between a closed connection and an open connection. The access terminal is assigned a MAC ID and associated resources on the forward and reverse links. These resources are used by the access terminal to request a connection, and by the access network to indicate grant or rejection of a connection, possibly based on the identity of the access terminal. This state corresponds to the BindUATI state of the Idle State Protocol.
- *Open Connection and assigned MAC ID*: Opening of the connection indicates that the access network has granted dedicated resources on the forward and reverse links based on the identity of the access terminal. This state corresponds to the Open State of the Connected State Protocol.

The Lower MAC Control Sublayer provides the following connection-related functions:

- Manages initial acquisition of the network.
- Manages opening and closing of connections.
- Maintains an approximate access terminal location in either connection states.
- Manages the radio link between the access terminal and the access network when a connection is open.
- Performs supervision at the access terminal both when the connection is open and when it is closed.

1 The Lower MAC Control Sublayer performs these functions through the following protocols:

- 2 ■ *Air Link Management Protocol*: This protocol maintains the overall connection state in  
3 the access terminal and the access network. The protocol can be in one of three states,  
4 corresponding to whether the access terminal has yet to acquire the network  
5 (Initialization State), has acquired the network but the connection is closed (Idle State), or  
6 has an open connection with the access network (Connected State). This protocol  
7 activates one of the following three protocols as a function of its current state.
- 8 ■ *Initialization State Protocol*: This protocol performs the actions associated with acquiring  
9 an access network.
- 10 ■ *Idle State Protocol*: This protocol performs the actions associated with an access terminal  
11 that has acquired the network, but does not have an open connection. Mainly, these are  
12 keeping track of the access terminal's approximate location in support of efficient Paging  
13 (using the Active Set Management protocol), the procedures leading to the opening of a  
14 connection, and support of access terminal power conservation.
- 15 ■ *Connected State Protocol*: This protocol performs the actions associated with an access  
16 terminal that has an open connection. These actions primarily include managing the radio  
17 link between the access terminal and the access network, including the management of  
18 tune away and selected interlace operation, and the procedures leading to the close of the  
19 connection.

20 In addition to the above protocols, which deal with the state of the connection, the Lower MAC  
21 Control sublayer also contains the following protocols:

- 22 ■ *Active Set Management Protocol*: This protocol performs the actions associated with  
23 keeping track of an access terminal's location and maintaining the radio link between the  
24 access terminal and the access network.
- 25 ■ *Overhead Messages Protocol*: This protocol broadcasts and receives essential parameters  
26 over the Control Channel MAC and the Forward Traffic Channel MAC. This protocol  
27 also performs supervision on the parameters, and generates SupervisionFailed indications  
28 when overhead parameters are not current.

### 29 **6.1.2 Data encapsulation**

30 This sublayer does not encapsulate data on the behalf of other layers or protocols.

## 31 **6.2 Default Air Link Management Protocol**

### 32 **6.2.1 Overview**

33 The Default Air Link Management Protocol provides the following functions:

- 34 ■ General state machine and state-transition rules to be followed by an access terminal and  
35 an access network for the Lower MAC Control Sublayer.
- 36 ■ Activation and deactivation of Lower MAC Control Sublayer protocols applicable to  
37 each protocol state.

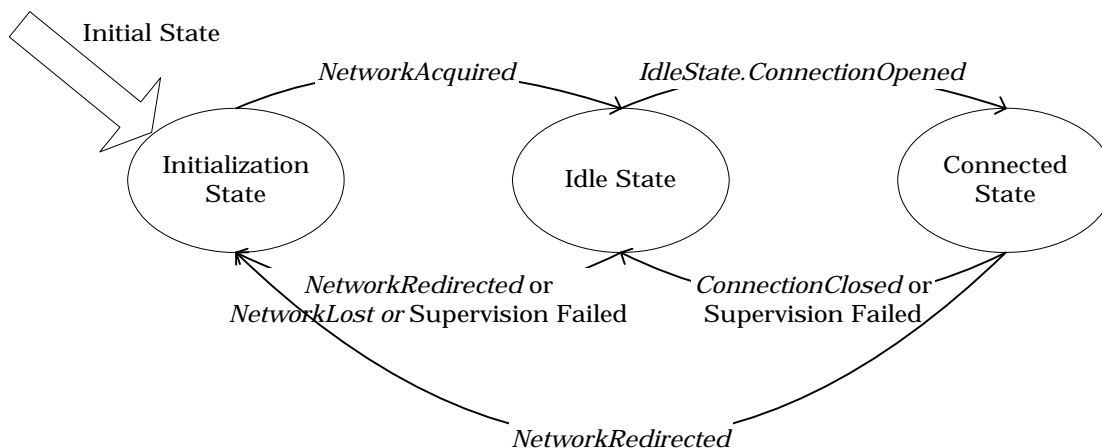
- 1           ■ Responding to supervision failures indications from other Protocols, and associated state  
2 transitions of Lower MAC Sublayer Protocols and Lower MAC Control Sublayer  
3 Protocols.
- 4           ■ Mechanism through which the access network can redirect the access terminal to another  
5 network.

6 The actual behavior and message exchange in each state is mainly governed by protocols that are  
7 activated by the Default Air Link Management Protocol. These protocols return indications which  
8 trigger the state transitions of this protocol. These protocols also share data with each other in a  
9 controlled fashion, by making that data public.

10 This protocol can be in one of three states:

- 11           ■ *Initialization State*: In this state the access terminal acquires an access network. The  
12 protocol activates the Initialization State Protocol to execute the procedures relevant to  
13 this state. The access network does not support this state.
- 14           ■ *Idle State*: In this state the connection is closed. The protocol activates the Idle State  
15 Protocol to execute the procedures relevant to this state.
- 16           ■ *Connected State*: In this state the connection is open. The protocol activates the  
17 Connected State Protocol to execute the procedures relevant to this state.

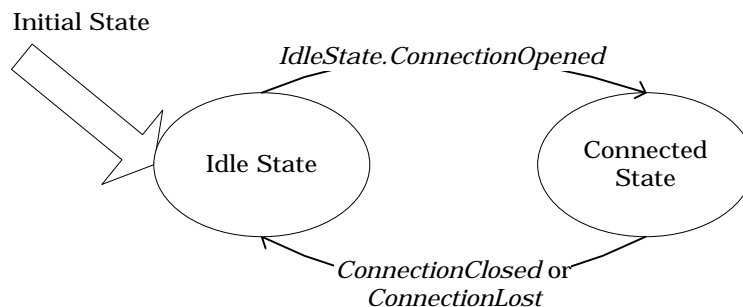
18 Figure 46 provides an overview of the access terminal states and state transitions. All transitions are  
19 caused by indications returned from protocols activated by the Default Air Link Management  
20 Protocol.



21

22 **Figure 46 Air Link Management Protocol state diagram (access terminal)**

1 Figure 47 provides an overview of the access network states and state transitions.



2

3 **Figure 47 Air Link Management Protocol state diagram (access network)**

4 Table 35 provides a summary of the Lower MAC Control Sublayer and Lower MAC Sublayer  
5 protocols that are active in each state.

6

**Table 35 Active protocols per Air Link Management Protocol state**

Initialization State <sup>19</sup>	Idle State	Connected State
Overhead Messages Protocol	Overhead Messages Protocol	Overhead Messages Protocol
Initialization State Protocol	Idle State Protocol	Connected State Protocol
Control Channel MAC Protocol <sup>20</sup>	Active Set Management Protocol	Active Set Management Protocol
	Control Channel MAC Protocol	Control Channel MAC Protocol
	Shared Signaling MAC Protocol	Shared Signaling MAC Protocol
	Forward Traffic Channel MAC Protocol	Forward Traffic Channel MAC Protocol
	Access Channel MAC Protocol <sup>21</sup>	Access Channel MAC Protocol <sup>22</sup>
	Reverse Traffic Channel MAC Protocol <sup>23</sup>	Reverse Traffic Channel MAC Protocol
	Reverse Control Channel MAC Protocol <sup>24</sup>	Reverse Control Channel MAC Protocol

<sup>19</sup> Applicable only to access terminal

<sup>20</sup> Activated by the Initialization State Protocol

<sup>21</sup> Used by access terminal only during connection setup

<sup>22</sup> Used by access terminal only during handoff to an asynchronous sector

<sup>23</sup> Used by access terminal only during connection setup

<sup>24</sup> Used by access terminal only during connection setup

## 6.2.2 Primitives

### 6.2.2.1 Commands

This protocol defines the following commands:

- *OpenConnection*
- *CloseConnection*

### 6.2.2.2 Return indications

This protocol does not return any indications.

## 6.2.3 Public data

### 6.2.3.1 Static public data

This protocol does not define any static public data.

### 6.2.3.2 Dynamic public data

- Subtype for this protocol

## 6.2.4 Protocol initialization and swap procedures

### 6.2.4.1 Protocol initialization

Upon initialization at the access terminal,

- The values of the attributes for this protocol instance shall be set to the default values specified for each attribute.
- The protocol shall enter the Initialization State.

Upon initialization at the access network,

- The values of the attributes for this protocol instance shall be set to the default values specified for each attribute.
- The protocol shall enter the Idle State.

### 6.2.4.2 Protocol swap

Upon swap at the access terminal,

- The protocol shall enter the Initialization State.

Upon swap at the access network,

- The protocol shall enter the Idle State.

Upon creation of the InUse instance of this protocol, the access network shall have a single InUse instance of this protocol operating in the Initialization State at the access network, serving all access terminals.



## 1 6.2.5 Procedures

### 2 6.2.5.1 Command processing

#### 3 6.2.5.1.1 OpenConnection

4 If the protocol receives the *OpenConnection* command in the Initialization State, the access terminal  
5 shall queue the command and execute it when the access terminal enters the Idle State.

6 The access network shall ignore the command in the Initialization State.

7 If the protocol receives this command in the Idle State:

- 8 ■ Access terminal shall issue an *IdleState.OpenConnection* command.
- 9 ■ Access network shall issue an *IdleState.OpenConnection* command.

10 If the protocol receives this command in the Connected State the command shall be ignored.

#### 11 6.2.5.1.2 CloseConnection

12 If the protocol receives the *CloseConnection* command in the Connected State:

- 13 ■ Access terminal shall issue a *ConnectedState.CloseConnection* command.
- 14 ■ Access network shall issue a *ConnectedState.CloseConnection* command.

15 If the protocol receives this command in any other state it shall be ignored.

### 16 6.2.5.2 Initialization state

17 This state is not applicable to the access network. In the Initialization State the access terminal has no  
18 information about the serving access network. In this state the access terminal selects a serving access  
19 network and obtains time synchronization from the access network.

#### 20 6.2.5.2.1 Access terminal requirements

21 The access terminal shall enter the Initialization State when the Default Air Link Management  
22 Protocol is instantiated. This may happen on events such as network redirection and initial power-on.  
23 A comprehensive list of events causing the Default Air Link Management Protocol to enter the  
24 Initialization State is beyond the scope of this specification.

25 The access terminal shall issue an *InitializationState.Activate* command upon entering this state. If the  
26 access terminal entered this state because the protocol received a Redirect message and a  
27 ChannelBand Record was received with the message, the access terminal shall provide the  
28 ChannelBand Record with the command.

29 If the protocol receives an *InitializationState.NetworkAcquired* indication the access terminal shall  
30 issue an *InitializationState.Deactivate* command<sup>25</sup> and transition to the Idle State.

---

<sup>25</sup> Some of the *Deactivate* commands issued by this protocol are superfluous (because the commanded protocol already put itself in the Inactive State) but are specified here for completeness.

### 1 6.2.5.2.2 Access network requirements

2 This state is not defined for the access network.

### 3 6.2.5.3 Idle state

4 In this state the access terminal has acquired the access network but does not have an open connection  
5 with the access network.

#### 6 6.2.5.3.1 Access terminal requirements

7 The access terminal shall issue the following commands upon entering this state:

- 8 ■ *IdleState.Activate*
- 9 ■ *ActiveSetManagement.Activate*

10 If the access terminal had a queued *OpenConnection* command, it shall issue an  
11 *IdleState.OpenConnection* command.

12 If the protocol receives an *IdleState.ConnectionOpened* indication, the access terminal shall:

- 13 ■ Issue a *IdleState.Deactivate* command
- 14 ■ Transition to the Connected State

15 If the protocol receives an *IdleState.ConnectionFailed*, a  
16 *ForwardTrafficChannelMAC.SupervisionFailed*, or a *ReverseTrafficChannelMAC.SupervisionFailed*  
17 indication, the access terminal shall:

- 18 ■ Issue a *IdleState.Close* command
- 19 ■ Issue a *ActiveSetManagement.Close* command
- 20 ■ Issue a *ReverseTrafficChannelMAC.Deactivate* command
- 21 ■ Issue a *ReverseControlChannelMAC.Deactivate* command

22 If the protocol receives a Redirect message, a *ActiveSetManagement.NetworkLost*, an  
23 *OverheadMessages.SupervisionFailed*, or a *ControlChannelMAC.SupervisionFailed* indication, the  
24 access terminal shall:

- 25 ■ Issue a *ActiveSetManagement.Deactivate* command
- 26 ■ Issue a *ReverseTrafficChannelMAC.Deactivate* command
- 27 ■ Issue a *ReverseControlChannelMAC.Deactivate* command
- 28 ■ Issue a *ForwardTrafficChannelMAC.Deactivate* command
- 29 ■ Issue an *OverheadMessages.Deactivate* command
- 30 ■ Issue a *ControlChannelMAC.Deactivate* command
- 31 ■ Issue a *IdleState.Deactivate* command
- 32 ■ Issue a *AccessChannelMAC.Deactivate* command
- 33 ■ Transition to the Initialization State

### 6.2.5.3.2 Access network requirements

The access network shall issue the following commands upon entering this state:

- *IdleState.Activate*
- *ActiveSetManagement.Activate*

If the protocol receives an *IdleState.ConnectionFailed* indication, or a *ReverseTrafficChannelMAC.SupervisionFailed* indication, the access network shall:

- Issue an *IdleState.Close* command
- Issue a *ReverseTrafficChannel.Deactivate* command
- Issue a *ReverseControlChannelMAC.Deactivate* command
- Issue a *ActiveSetManagement.Close* command
- Issue a *IdleState.Deactivate* command

If the protocol receives an *IdleState.ConnectionOpened* indication, the access network shall:

- Issue a *IdleState.Deactivate* command
- Transition to the Connected State

The access network may send the access terminal a Redirect message to redirect it from the current serving network and optionally, provide it with information directing it to another network. If the access network sends a Redirect message it shall:

- Issue a *ReverseTrafficChannel.Deactivate* command
- Issue a *ReverseControlChannel.Deactivate* command
- Issue a *ActiveSetManagement.Deactivate* command
- Issue a *IdleState.Deactivate* command.
- Transition to the Idle State

### 6.2.5.4 Connected state

In the Connected State, the access terminal and the access network have an open connection.

#### 6.2.5.4.1 Access terminal requirements

##### 6.2.5.4.1.1 General requirements

The access terminal shall issue the following commands upon entering this state:

- *ConnectedState.Activate*
- *ActiveSetManagement.Open*

1 If the protocol receives a *ConnectedState.ConnectionClosed*, an  
 2 *OverheadMessages.SupervisionFailed*, a *ControlChannelMAC.SupervisionFailed*, a  
 3 *ActiveSetManagement.AssignmentRejected*, or a *ForwardTrafficChannelMAC.SupervisionFailed*  
 4 indication, the access terminal shall:

- 5 ■ Issue a *ActiveSetManagement.Close* command
- 6 ■ Perform the cleanup procedure defined in 6.2.5.4.1.2
- 7 ■ Transition to the Idle State

8 If the protocol receives a Redirect message, the access terminal shall:

- 9 ■ Issue a *ActiveSetManagement.Deactivate* command
- 10 ■ Issue an *OverheadMessages.Deactivate* command
- 11 ■ *ForwardTrafficChannelMAC.Deactivate*
- 12 ■ *SharedSignalingMAC.Deactivate*
- 13 ■ Perform the cleanup procedure defined in 6.2.5.4.1.2
- 14 ■ Transition to the Initialization State

#### 15 **6.2.5.4.1.2 Connected state cleanup procedures**

16 The access terminal shall issue the following commands when it exits this state:

- 17 ■ *ReverseTrafficChannel.Deactivate*
- 18 ■ *ReverseControlChannelMAC.Deactivate*
- 19 ■ *ActiveSetManagement.Close*
- 20 ■ *ConnectedState.Deactivate*

#### 21 **6.2.5.4.2 Access network requirements**

##### 22 **6.2.5.4.2.1 General requirements**

23 The access network shall issue the following commands upon entering this state:

- 24 ■ *ConnectedState.Activate*
- 25 ■ *ActiveSetManagement.Open*

26 If the protocol receives a *ConnectedState.ConnectionClosed*, or  
 27 *ActiveSetManagement.ConnectionLost* indication, the access network shall:

- 28 ■ Issue a *ActiveSetManagement.Close* command
- 29 ■ Perform the cleanup procedures defined in 6.2.5.4.2.2
- 30 ■ Transition to the Idle State

1 The access network may send the access terminal a Redirect message to redirect it from the current  
 2 serving network and optionally, provide it with information directing it to another network. If the  
 3 access network sends a Redirect message it shall:

- 4 ■ Issue a *ActiveSetManagement.Deactivate* command
- 5 ■ Perform the cleanup procedures defined in 6.2.5.4.2.2
- 6 ■ Transition to the Idle State

#### 7 **6.2.5.4.2.2 Connected state cleanup procedures**

8 The access network shall issue the following commands when it exits this state:

- 9 ■ *ReverseTrafficChannel.Deactivate*
- 10 ■ *ReverseControlChannel.Deactivate*
- 11 ■ *ActiveSetManagement.Close*
- 12 ■ *ConnectedState.Deactivate*

### 13 **6.2.6 Message formats**

#### 14 **6.2.6.1 Redirect**

15 The access network sends the Redirect message to redirect the access terminal(s) away from the  
 16 current network; and, optionally, the access network provides it with information directing it to one of  
 17 a set of different networks.

Field	Length (bits)
MessageID	8
StayAwayDuration	16
NumChannelBands	8
RedirectReason	8
NumChannelBands instances of the following two fields	
ChannelBandRecord	ChannelBandRecord Type Dependent
TechnologyType	8

19 MessageID The access network shall set this field to 0x00.

20 StayAwayDuration The access network shall set this field to the duration, in units of seconds, for  
 21 which the access terminal shall not make an access attempt at the sector  
 22 sending this message.

23 RedirectReason The sender shall set this field to reflect the redirect reason, as shown in  
 24 Table 36.

**Table 36 Encoding of the RedirectReason field**

Field value	Description
0x00	Reserved
0x01	Network Busy
0x02	Authentication or billing failure
0x03	Desired QoS unavailable
0x04	No route to host
0x05	Network Maintenance
0xff	General Failure
All other values are reserved	

NumChannelBands      The access network shall set this field to the number of ChannelBand records it is including in this message.

ChannelBandRecord    This field shall be set to the channel band that the access terminal should reacquire. The channel band shall be specified using the standard ChannelBand Record definition, see 10.1.

TechnologyType        This field shall be set to the type of technology, and shall be interpreted as defined in Table 107. This field shall not take the value 0x00.

<b>Channels</b>	FTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Broadcast      Unicast	<b>Security</b>	Required

## 6.2.7 Interface to other protocols

### 6.2.7.1 Commands

This protocol issues the following commands:

- *InitializationState.Activate*
- *InitializationState.Deactivate*
- *IdleState.Activate*
- *IdleState.Deactivate*
- *IdleState.Close*
- *IdleState.OpenConnection*
- *ConnectedState.Activate*
- *ConnectedState.Deactivate*
- *ConnectedState.CloseConnection*
- *ActiveSetManagement.Activate*
- *ActiveSetManagement.Deactivate*

- 1       ■ *ActiveSetManagement.Close*
- 2       ■ *ActiveSetManagement.Open*
- 3       ■ *OverheadMessages.Deactivate*
- 4       ■ *ControlChannelMAC.Deactivate*
- 5       ■ *AccessChannelMAC.Deactivate*
- 6       ■ *ReverseTrafficChannelMAC.Deactivate*
- 7       ■ *ReverseControlChannelMAC.Deactivate*
- 8       ■ *SharedSignalingMAC.Deactivate*

### 9       **6.2.7.2 Indications**

10      This protocol registers to receive the following indications:

- 11       ■ *InitializationState.NetworkAcquired*
- 12       ■ *IdleState.ConnectionOpened*
- 13       ■ *IdleState.ConnectionFailed*
- 14       ■ *ConnectedState.ConnectionClosed*
- 15       ■ *ActiveSetManagement.ConnectionLost*
- 16       ■ *ActiveSetManagement.NetworkLost*
- 17       ■ *ActiveSetAssignment.AssignmentRejected*
- 18       ■ *OverheadMessages.SupervisionFailed*
- 19       ■ *ControlChannelMAC.SupervisionFailed*
- 20       ■ *ReverseTrafficChannelMAC.SupervisionFailed*
- 21       ■ *ForwardTrafficChannelMAC.SupervisionFailed*

### 22      **6.2.8 Configuration attributes**

23      No configuration attributes are defined for this protocol.

### 24      **6.2.9 Protocol numeric constants**

Constant	Meaning	Value
$N_{ALMPType}$	Type field for this protocol	Table 9
$N_{ALMPDefault}$	Subtype field for this protocol	0x0000

### 27      **6.2.10 Session state information**

28      The Session State Information record (see 10.10) consists of parameter records.

29      The parameter records for this protocol consist of the configuration attributes of this protocol.

## 6.3 Default Idle State Protocol

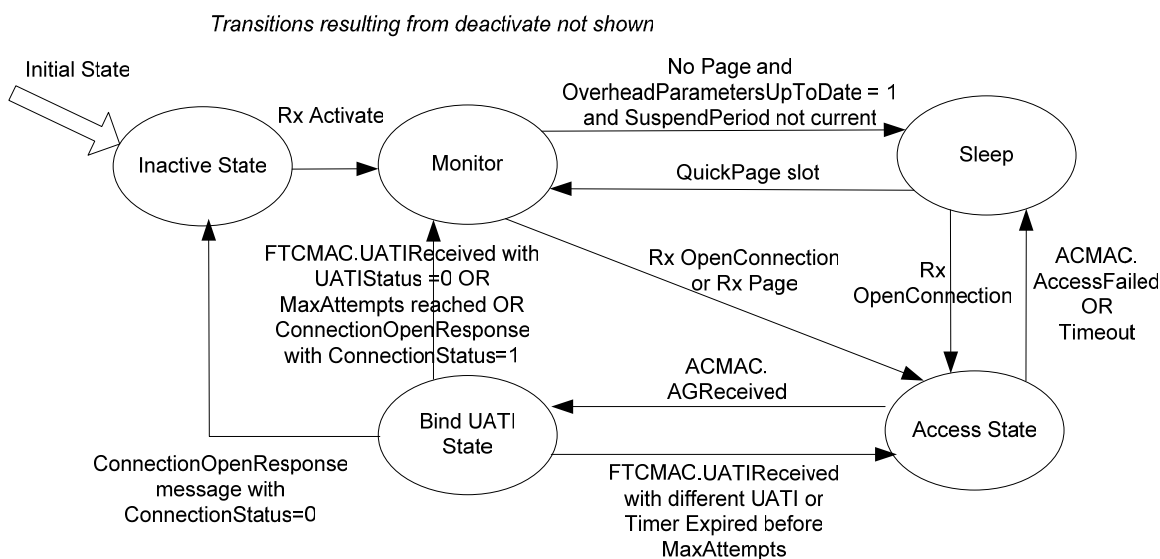
### 6.3.1 Overview

The Default Idle State Protocol provides the procedures and messages used by the access terminal and the access network when the access terminal has acquired a network and a connection is not open.

This protocol operates in one of the following four states:

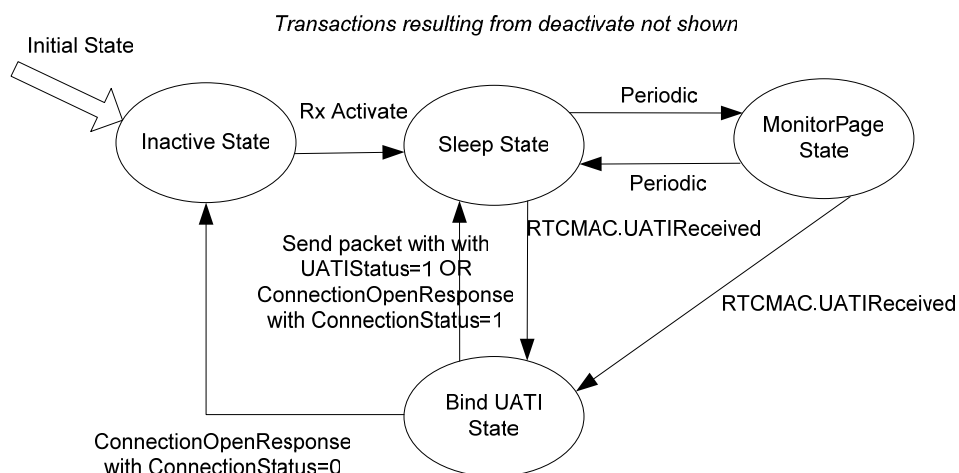
- *Inactive State*: In this state the protocol waits for an *Activate* command.
- *Sleep State*: In this state the access terminal may shut down part of its subsystems to conserve power. The access terminal does not monitor any Forward Channel, and the access network is not allowed to transmit unicast packets to it.
- *Monitor State*: In this state the access terminal listens for Pages and QuickPages and if necessary, updates the parameters received from the Overhead Messages Protocol. The access network may transmit unicast packets to the access terminal in this state.
- *Access State*: In this state the access terminal sends access preambles to the access network and receives an access grant from the network. This state is not defined for the access network.
- *BindUATI State*: In this state the access terminal sends an ATIdentifier (UATI or SessionSeed) to the access network and waits for an acknowledgement in the form of a *ForwardTrafficChanneMAC.UATIReceived* indication or a *ConnectionOpenResponse* message. In this state, the access network sends a packet with the MAC header field *UATIInfoIncluded* set to '1' to the access terminal and may send a *ConnectionGrant* message to the access terminal.

Protocol states and events causing the transition between the states are shown in Figure 48 and Figure 49.



**Figure 48 Default Idle State Protocol (access terminal)**





**Figure 49 Default Idle State Protocol state (access network)**

This protocol supports periodic network monitoring by the access terminal, allowing for significant power savings. The following access terminal operation modes are supported:

- Continuous operation, in which the access terminal continuously monitors the Control Channel.
- Suspended mode operation, in which the access terminal monitors the Control Channel continuously for a period of time and then proceeds to operate in the slotted mode. Suspended mode follows operation in the Air Link Management Protocol Connected State and allows for quick network-initiated reconnection.
- Slotted mode operation, in which the access terminal monitors only selected superframes and sleeps in between. The slotted mode supports staggered operation, where the time interval between the superframes monitored by the terminal increases with time. For the first WakeCount1 sleep instances, the sleep period may be Period1 superframes, and for the next WakeCount2-WakeCount1 sleep instances, the sleep period may be Period2 superframes, and for subsequent sleep instances, the sleep period may be Period3 superframes.
- This protocol supports connection set up: this procedure is always performed at the initiative of the access terminal.<sup>26</sup> It consists of the following steps:
  - Access terminal sending an access preamble on R-ACH.
  - Access network sends an access grant on F-SSCH.
  - Access terminal sends a packet on the R-DCH. This packet contains the UATI or SessionSeed in the MAC header and ConnectionOpenRequest message in the payload.
  - Access network sends a response packet on the F-DCH. The response packet contains the UATI or SessionSeed in the MAC header and ConnectionOpenResponse message in the payload.

<sup>26</sup> The access network may transmit a Page message to the access terminal directing it to initiate the procedure.

## 6.3.2 Primitives

### 6.3.2.1 Commands

This protocol defines the following commands:

- *Activate*
- *Deactivate*
- *OpenConnection*
- *Close*

### 6.3.2.2 Return indications

This protocol returns the following indications:

- *ConnectionOpened*
- *ConnectionFailed*
- *RegistrationRadiusUpdated*

## 6.3.3 Public data

### 6.3.3.1 Static public data

- PageTimes array
- ConnectCount
- RQuickPage

### 6.3.3.2 Dynamic public data

- Subtype for this protocol

## 6.3.4 Protocol initialization and swap procedures

### 6.3.4.1 Protocol initialization

Upon initialization at the access terminal and the access network:

- The values of the attributes for this protocol instance shall be set to the default values specified for each attribute.
- The protocol shall enter the Inactive State.
- The protocol shall set ConnectCount to zero.

### 6.3.4.2 Protocol swap

Upon swap at the access terminal and access network:

- The protocol shall enter the Inactive State.

## 6.3.5 Procedures

### 6.3.5.1 Command processing

#### 6.3.5.1.1 Activate

When the protocol at the access terminal or the access network receives an *Activate* command in the Inactive State:

- If the multi-carrier mode is set to MultiCarrierOn, the access terminal shall select its operating carrier as follows. If PreferredPagingCarrierEnabled is set to '1', the access terminal and the access network shall set  $C = \text{PreferredPagingCarrier}$ . Otherwise, the access terminal and access network shall calculate  $C$  as the result of applying the hash function (see 10.4) using the following parameters:
  - Key = SessionSeed
  - Decorrelate =  $6 \times \text{SessionSeed}[11:0]$
  - N = NumCarriers

where SessionSeed is given as public data of the Session Management Protocol. The access terminal shall set its carrier to  $\text{CarrierID} = \text{mod}(C, \text{NumCarriers})$ , where NumCarriers is the number of carriers in the public data of the Overhead Messages Protocol. The access network should use this CarrierID to communicate with the access terminal.

When the protocol at the access terminal receives an *Activate* command in the Inactive State:

- If the access terminal entered the Idle State upon sending a ConnectionClose message with CloseReason set to "Deregistration Request", the access terminal shall transition to the Sleep State
- Otherwise, the access terminal shall transition to the Monitor State.

When the protocol at the access network receives an *Activate* command in the Inactive State:

- The access network shall transition to the Sleep State.<sup>27</sup>

If the protocol receives this command in any other state the command shall be ignored.

#### 6.3.5.1.2 Deactivate

When the protocol receives a *Deactivate* command in the Inactive State it shall be ignored.

When the protocol receives this command in any other state:

- The access terminal shall transition to the Inactive State.
- The access network shall transition to the Inactive State.

---

<sup>27</sup> Since the transitions happen asynchronously, this requirement guarantees that the access network will not transmit unicast packets to the access terminal over the Control Channel when the access terminal is not monitoring the channel.

### 1 **6.3.5.1.3 OpenConnection**

2 When the protocol receives an *OpenConnection* command in the Inactive State, the Access State or  
3 the BindUATI State, the command shall be ignored.

4 When the protocol receives this command in the Sleep State:

- 5 ■ The access terminal shall transition to the Access State.
- 6 ■ The access network shall queue the command and execute it when it is in the Monitor  
7 State.

8 When the protocol receives this command in the Monitor State:

- 9 ■ The access terminal shall transition to the Access State.
- 10 ■ The access network shall send a QuickPage and a Page.

### 11 **6.3.5.1.4 Close**

12 When the protocol receives a *Close* command in the Inactive State it shall be ignored.

13 When the protocol receives a *Close* command in any other state:

- 14 ■ The access terminal shall transition to the Monitor State.
- 15 ■ The access network shall transition to the Sleep State.

### 16 **6.3.5.2 General overview of paging**

17 Paging is implemented using the F-pBCH1 and F-DCH physical layer channels. The F-pBCH1  
18 channel occurs at the beginning of a superframe (in superframes with even superframe numbers), and  
19 contains a QuickPaging block. This QuickPaging block may have one of two possible formats.

- 20 1. QuickPage block with full ATI results in a one step page.
- 21 2. QuickPage block with LSBs of the ATI results in a two step page.

22 In case the QuickPaging block has the full ATI of the access terminal, a  
23 *ControlChannelMAC.PageReceived* indication is generated, completing a one step page process.

24 In case the QuickPaging block has the LSBs of the ATI of the access terminal, a  
25 *ControlChannelMAC.QuickPageReceived* indication is generated, and a QuickPage is considered to  
26 be received.

27 Upon receipt of a QuickPage, the access terminal monitors the F-DCH for the full ATI, and if the ATI  
28 is detected, a *ForwardTrafficChannelMAC.PageReceived* indication is generated, completing a two-  
29 step page process.

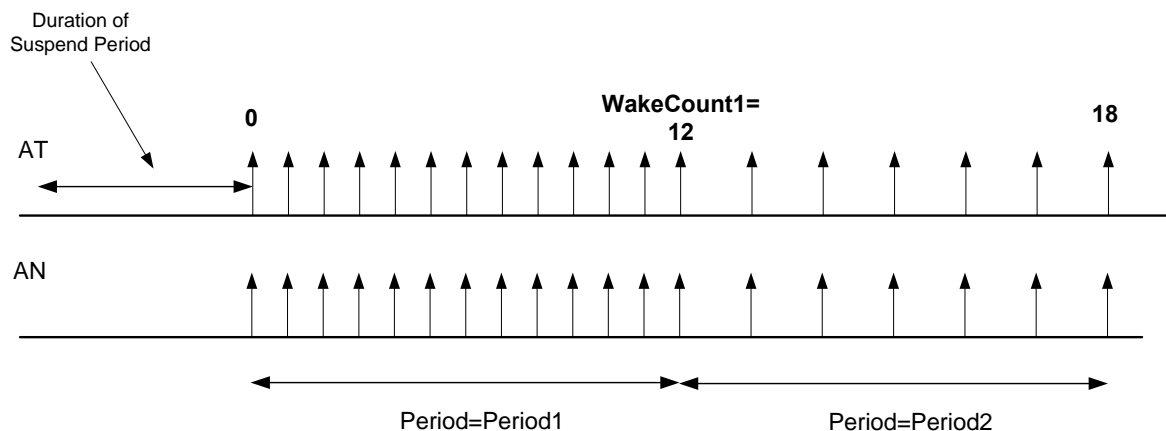
### 30 **6.3.5.3 General overview of sleep cycle**

31 This section is for informative purposes, and describes which superframes may be used for sending  
32 pages by the access network, and receiving pages by the access terminal.

1 The Default Idle State Protocol allows the implementation of staggered mode sleep, where the  
 2 terminal sleep period increases with time, from a sleep period of Period1 superframes for the first  
 3 WakeCount1 sleep periods to Period2 superframes for the next WakeCount2 sleep periods, and  
 4 finally to Period3 superframes for the remaining sleep periods. Staggered sleep presents the design  
 5 problem that the access terminal and access network may be unable to synchronize timing for  
 6 staggering, and therefore may increase the sleep period at different times. This may result in missed  
 7 pages, where the access network is in monitor state (because it is using a smaller sleep period) while  
 8 the access terminal is in sleep state (because it is using a larger sleep period). To prevent missed  
 9 pages, this protocol makes the following design choices:

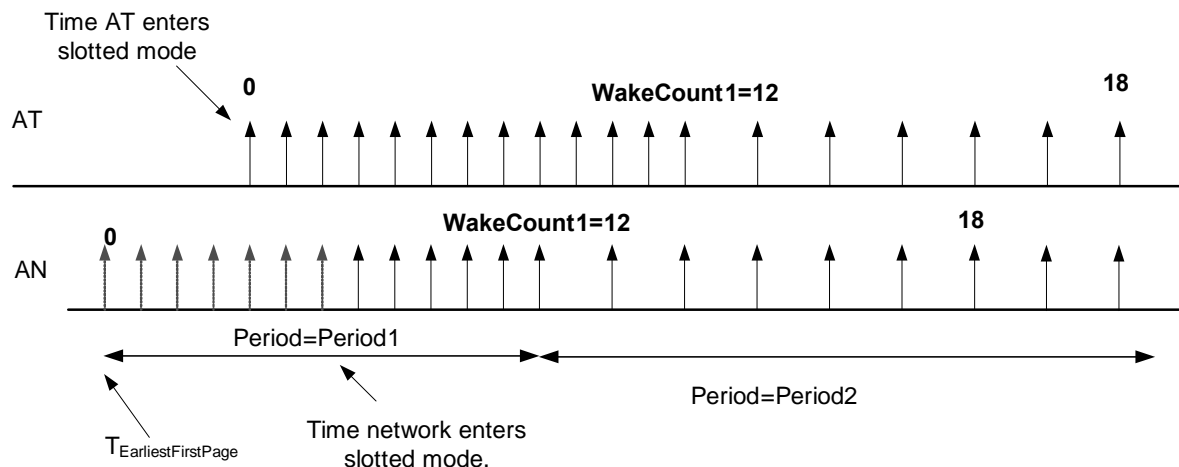
- 10 ■ The sleep times Period2 and Period3 shall be some multiple of a power of two of  
 11 Period1. For example, if Period1 is 6 superframes, Period2 may be 24 superframes and  
 12 Period3 may be 96 superframes. Such setting of the Periods guarantees that the access  
 13 network is in the monitor state only when the access terminal is also in the monitor state,  
 14 as shown below.
- 15 ■ The access network initializes with a staggered sleep cycle at a time that is conservative.

16 Staggered sleep operation is discussed in more detail below. In case the access terminal has  
 17 advertised a suspend period in a ConnectionClose message, and the access network has received this  
 18 message, the access network can synchronize slotted mode operation with the access terminal, i.e., the  
 19 access network and access terminal can enter the monitor state in the same superframe. This  
 20 synchronous operation is shown in Figure 50.



21  
 22 **Figure 50 Slotted mode operation when access terminal and access network are**  
 23 **synchronized**

24 In case the access network has not received a suspend period from the access terminal, the access  
 25 network may begin slotted operation at a time that is different from the time the access terminal enters  
 26 slotted mode. The access network begins to count pages starting at time EarliestFirstPage, where the  
 27 access network has determined that EarliestFirstPageTime is the earliest time the access terminal  
 28 could have entered slotted mode.



1  
2 **Figure 51 Slotted mode operation when access terminal and access network are not**  
3 **synchronized**

#### 4 **6.3.5.4 General procedures**

##### 5 **6.3.5.4.1 Paging cycle offset**

6 *R* (the paging cycle offset) shall be obtained as follows. Since the paging cycle occurs only once  
7 every two superframes, the following procedure is designed to return even values.

- 8 ■ If PreferredPageOffsetEnabled is equal to '0', then *R* is the result of applying the hash  
9 function (see 10.4) using the following parameters, and multiplying the result by 2:
- 10 □ Key = SessionSeed
  - 11 □ Decorrelate =  $6 \times \text{SessionSeed}[11:0]$
  - 12 □  $N = \text{Max}(\text{Period1}, \text{Period2}, \text{Period3}, 1)$
- 13 where SessionSeed is given as public data of the Session Management Protocol.
- 14 ■ If PreferredPageOffsetEnabled is equal to '1', then *R* is set to twice the  
15 PreferredPageOffset.

##### 16 **6.3.5.4.2 RQuickPage calculation**

17 The parameter RQuickPage is used by the Lower MAC Sublayer to determine if a quick page is  
18 received, and RQuickPage shall be decided as follows.

- 19 ■ If PreferredQuickPageEnabled is equal to '0', then *RQuickPage* is the result of applying  
20 the hash function (see 10.4) using the following parameters:
- 21 □ Key = SessionSeed
  - 22 □ Decorrelate =  $6 \times \text{SessionSeed}[23:12]$
  - 23 □  $N = 2^{\text{N}_{\text{QP\_BLK}}}$
- 24 where SessionSeed is given as public data of the Session Management Protocol and  
25  $\text{N}_{\text{QP\_BLK}}$  is a numeric constant of the Control Channel MAC Protocol.

If PreferredQuickPageEnabled is equal to '1', then RQuickPage is set to PreferredRQuickPage.

#### 6.3.5.4.3 Paging period calculation

The access network and the access terminal shall compute  $Period_i$  according to Table 37.

**Table 37 Computation of  $Period_i$  from  $SlotCycle_i$**

<b>SlotCycle<math>_i</math></b>	<b>Period<math>_i</math></b>
0x00 to 0x1f	SlotCycleBase * $2^{SlotCycle_i}$ superframes
0x20 to 0xff	Reserved

#### 6.3.5.4.4 Procedure for page time calculation

Given a value for StartTime (an argument to this procedure), this procedure shall return an array PageTimes[j].

The following formula shall be called the QuickPage formula, and shall be used to determine the PageTimes array:

$$[T+R] \bmod \text{Period} = 0,$$

where R is defined in 6.3.5.4, and values for Period are given in the steps below.

The PageTimes array shall be calculated by the following procedure, using a temporary variable x. This procedure follows the general description in 6.3.5.2.

1. Set x as the minimum value of T that is greater than StartTime and satisfies the QuickPage formula with Period=Period1.
2. For  $0 \leq j < \text{WakeCount1}$ , set  $\text{PageTimes}[j] = x + j * \text{Period1}$ .
3. Set x as the minimum value of T that is greater than  $\text{PageTimes}[\text{WakeCount1}-1]$  and satisfies the QuickPage formula with Period=Period2.
4. For  $\text{WakeCount1} \leq j < \text{WakeCount2}$ , set  $\text{PageTimes}[j] = x + (j - \text{WakeCount1}) * \text{Period2}$ .
5. Set x as the minimum value of T that is greater than  $\text{PageTimes}[\text{WakeCount2}-1]$  and satisfies the QuickPage formula with Period=Period2.
6. For  $\text{WakeCount2} \leq j$ , set  $\text{PageTimes}[j] = x + (j - \text{WakeCount2}) * \text{Period3}$ .

The access network transmits pages for the access terminal in superframe index T, where T takes values PageTimes[j],  $j=0,1,2, \dots$ . The access network may also transmit the QuickPages for the access terminal in superframes that occur FastRepageInterval superframes after values in the PageTimes array.

### 1 **6.3.5.5 Inactive state**

2 When the protocol is in the Inactive State it waits for an *Activate* command and at the access terminal,  
3 sets internal variable NumAccessAttempts to zero.

### 4 **6.3.5.6 Sleep state**

5 When the access terminal is in the Sleep State it may stop monitoring the access network by issuing  
6 the following commands:

- 7 ■ *ControlChannelMAC.Deactivate*
- 8 ■ *SharedSignalingMAC.Deactivate*
- 9 ■ *ForwardTrafficChannelMAC.Deactivate*
- 10 ■ *OverheadMessages.Deactivate*

11 The access terminal may shut down processing resources to reduce power consumption.

### 12 **6.3.5.6.1 Access terminal requirements**

13 If the access terminal has a queued *OpenConnection* command, it shall transition to the Access State.

14 If the access terminal entered the Idle State as upon sending a *ConnectionClose* message with  
15 *CloseReason* set to “Deregistration Request”, this specification does not specify rules for entering the  
16 Monitor State. Otherwise, the access terminal shall transition to the monitor state for any one of the  
17 following reasons:

- 18 ■ *Update of overhead messages*: To determine the time of transition to the monitor state,  
19 the access terminal may rely on the *ExtendedChannelInfoExpiryTime*, that is public data  
20 of the Overhead Messages Protocol. The exact algorithm used by the access terminal to  
21 update overhead messages is beyond the scope of this specification.
- 22 ■ *To receive pages*: The access terminal shall transition to the monitor state in superframes  
23 specified by the *PageTimes* array.
- 24 ■ *To receive fast repages*: If there was a paging error on the last page read by the access  
25 terminal (say in superframe T), and T is an entry in the *PageTimes* array, then the access  
26 terminal shall transition to the monitor state in superframe T+*FastRepageInterval*.

### 27 **6.3.5.6.2 Access network requirements**

28 Upon entering the Sleep State from the Inactive State, the access network shall:

- 29 ■ Invoke the procedure in 6.3.5.4.4 to determine the *PageTimes* array. While invoking this  
30 procedure, the access network shall set *StartTime* such that the access terminal does not  
31 miss any pages.

32 In order to set *StartTime* above, the access network may use the following procedure. If the access  
33 network has received a suspend period from the access terminal, it may set *StartTime* to the  
34 superframe number that contains the last part of the suspend period. If the access network did not  
35 receive a suspend period, it may set *StartTime* to the superframe number where the last packet was  
36 received from the access terminal.



1 If the access network entered the Idle State as upon receiving a *ConnectionClose* message with  
2 *CloseReason* set to “Deregistration Request”, the access network shall not enter the Monitor State.

3 Otherwise, the access network shall enter the monitor state at one of the following two times:

- 4 ■ The smallest entry in the *PageTimes* array that is greater than the current superframe  
5 number (for routine pages).
- 6 ■ The sum of *FastRepageInterval* and the largest entry in the *PageTimes* array that is less  
7 than or equal to the current superframe number (for fast repages).

8 The setting of which one of the two times above is selected is beyond the scope of this specification,  
9 and may depend on the implementation of the fast repage mechanism at the access network.

10 When the access network is in the Sleep State, it is prohibited from sending unicast packets to the  
11 access terminal.

12 If the access network receives a *ReverseTrafficChannelMAC.UATIReceived* indication in the Sleep  
13 State, it shall transition to the BindUATI State.

#### 14 **6.3.5.7 Monitor state**

15 The access terminal shall enter the Monitor State either to receive a Page, QuickPage or other  
16 messages from the access network.

17 When the access network is in the Monitor State, it may send unicast packets to the access terminal.

##### 18 **6.3.5.7.1 Access terminal requirements**

19 Upon entering the Monitor State, the access terminal shall:

- 20 ■ Issue *ControlChannelMAC.Activate*
- 21 ■ Issue *ForwardTrafficChannelMAC.Activate*
- 22 ■ Issue *SharedSignalingMAC.Activate*
- 23 ■ Issue *OverheadMessages.Activate*
- 24 ■ Set internal variable *NumAccessAttempts* to zero

25 Upon entering the Monitor State from the Inactive State, or if the BindUATI state has been entered  
26 since the last visit to the Monitor State, the access terminal shall:

- 27 ■ Invoke the procedure specified in 6.3.5.4.4 to determine the *PageTimes* array. While  
28 invoking this procedure, the access terminal shall set *StartTime* to the superframe number  
29 when the access terminal entered the monitor state.

30 The access terminal shall comply with the following requirements when in the Monitor State:

- 31 ■ If the current superframe number is in the *PageTimes* array the Idle State Protocol shall
  - 32 □ Determine if there is a paging error in the current superframe, where the paging error  
33 event is defined in the Control Channel MAC.

1 The access terminal shall transition to the Access State if any of the following conditions are met:

- 2 ■ The access terminal receives a *ForwardTrafficChannelMAC.PageReceived* indication
- 3 ■ The access terminal receives a *ControlChannelMAC.PageReceived* indication
- 4 ■ The access terminal receives a PageUATI message where the UATI field matches the  
5 UATI public data field of the Address Management Protocol
- 6 ■ The access terminal has a queued *OpenConnection* command

7 The access terminal may transition to the Sleep State if all of the following conditions are met:

- 8 ■ OverheadParametersUpToDate=1
- 9 ■ Transitioning to sleep state will not cause the access terminal to miss a page in the current  
10 superframe.
- 11 ■ Access terminal has not advertised a suspend period that is current (see 6.4.5.2.4.1). The  
12 suspend period is current if the time advertised in the associated ConnectionClose  
13 message is greater than the current system time.<sup>28</sup>

#### 14 **6.3.5.7.2 Access network requirements**

15 The access network shall comply with the following requirements in the Monitor State:

- 16 ■ If the access network receives a *ReverseTrafficChannelMAC.UATIReceived* indication, it  
17 shall transition to the BindUATI State. This requirement shall take precedence over other  
18 requirements applicable to this state.
- 19 ■ If the access network has a queued *OpenConnection* command, the access network shall
  - 20 □ Send the access terminal a Page. (Procedures for sending a Page are defined in the  
21 Control Channel MAC and the Forward Traffic Channel MAC. The Forward Traffic  
22 Channel MAC is used to send the page only if the page does not fit in the Control  
23 Channel MAC due to resource limitations.) If a page is sent over the Forward Traffic  
24 Channel MAC, the page packet shall begin transmission in PHY Frame index 1  
25 through 7 of the superframe after the superframe where the QuickPage was sent  
26 (where the first PHY Frame of the superframe has index 0).
  - 27 □ If the access terminal has sleep period greater than one superframe, and the page is  
28 sent over the Forward Traffic Channel MAC, then the access network shall send the  
29 page in the superframe after the superframe where the QuickPage was sent.
  - 30 □ After the page is sent, transition to the Sleep State
- 31 ■ If the access network does not have a queued *OpenConnection* command, the access  
32 terminal shall
  - 33 □ Transition to the Sleep State.

---

<sup>28</sup> The access terminal monitors the Control Channel continuously during a suspend period thus avoiding the delay in opening access-network-initiated connections due to the sleep period.

### 6.3.5.8 Access state

The access terminal and the access network use the Access State to provide a MAC ID to the access terminal.

#### 6.3.5.8.1 Access terminal requirements

The access terminal shall comply with the following requirements.

- Upon entering the Access State the access terminal shall:
  - Issue a *ControlChannelMAC.Activate* command
  - Issue a *SharedSignalingMAC.Activate* command
  - Issue a *AccessChannelMAC.Activate* command
  - Issue a *ForwardTrafficChannelMAC.Activate* command
  - Issue a *OverheadMessages.Activate* command
  - Issue a *AccessChannelMAC.AttemptAccess* command
  - Issue a *ActiveSetManagement.SendPilotReport* command
  - Set a state timer for  $T_{IDPATSetup}$  seconds
  - If NumAccessAttempts is '0', generate a ConnectionOpenRequest message
- If the state timer expires or the protocol receives an *AccessChannelMAC.AccessFailed* indication, the access terminal shall set internal variable NumAccessAttempts to 0, return a *ConnectionFailed* indication and transition to the Sleep State.
- If the access terminal receives a *AccessChannelMAC.AccessGrantReceived* indication, it shall:
  - Increment public data ConnectCount by 1
  - Issue a *ReverseTrafficChannelMAC.Activate* command
  - Issue a *ReverseControlChannelMAC.Activate* command
  - Transition to the BindUATI State

#### 6.3.5.8.2 Access network requirements

The Access State is not applicable to the Idle State Protocol at the Access Network.

### 6.3.5.9 BindUATI state

#### 6.3.5.9.1 Access terminal requirements

The access terminal shall comply with the following requirements.

- Upon entering the BindUATI State the access terminal shall:
  - Start a state timer for  $T_{IDSTABind}$  seconds.
  - Increment internal variable NumAccessAttempts by 1.
  - Transmit a ConnectionOpenRequest message with the RegistrationRadiusFlag field set to the public data RegistrationRadiusFlag of the Active Set Management Protocol.

- 1           ■ If the state timer expires, and NumAccessAttempts is less than MaxAccessAttempts, the  
2           access terminal shall transition to the Access State.
- 3           ■ If the state timer expires and NumAccessAttempts is greater than or equal to  
4           MaxAccessAttempts the access terminal shall execute the cleanup procedures given in  
5           6.3.5.9.1.3.

#### 6   **6.3.5.9.1.1 Processing the ForwardTrafficChannelMAC.UATIReceived indication**

7   The access terminal shall process this indication according to the following rules.

8   The access terminal shall declare the indication valid if the ATIdentifier that accompanies the  
9   indication matches the ATIdentifier last transmitted by the access terminal.

- 10          ■ If the indication is not valid, the access terminal shall perform the following:
- 11           □ If NumAccessAttempts is less than MaxAccessAttempts, the access terminal shall  
12           transition to the Access State, otherwise
- 13           □ The access terminal shall execute the cleanup procedures given in 6.3.5.9.1.3.
- 14          ■ If the access terminal receives a valid indication with UATISStatus equal to 0x1, it shall  
15           execute the cleanup procedures given in 6.3.5.9.1.3.
- 16          ■ If the access terminal receives a valid indication with UATISStatus equal to 0x0 it shall  
17           reset the state timer.

#### 18   **6.3.5.9.1.2 Processing the ConnectionOpenResponse message**

19   On receiving a ConnectionOpenResponse message and a *ForwardTrafficChannelMAC.UATIReceived*  
20   indication in the same packet, the access terminal shall declare the ConnectionOpenResponse  
21   message to be invalid if the UATISStatus field with the indication is set to 0x2.

22   Otherwise, it shall declare the ConnectionOpenResponse message to be valid.

23   On receiving a valid ConnectionOpenResponse message with ConnectionStatus set to 0x0, the access  
24   terminal shall:

- 25          ■ Return a *ConnectionOpened* indication
- 26          ■ Set NumAccessAttempts to 0
- 27          ■ Transition to the Inactive State

28   On receiving a valid ConnectionOpenResponse message with ConnectionStatus set to 0x1, the access  
29   terminal shall:

- 30          ■ Set NumAccessAttempts to 0.
- 31          ■ Generate a *RegistrationRadiusFlagUpdated* indication accompanied by the  
32          RegistrationRadius field of the ConnectionOpenResponse message.
- 33          ■ Transition to the Monitor State.

### 6.3.5.9.1.3 Cleanup procedures for the BindUATI state

The access terminal shall do the following when this procedure is invoked:

- Set NumAccessAttempts to 0
- Issue a *ReverseTrafficChannelMAC.Deactivate* command
- Issue a *ReverseControlChannelMAC.Deactivate* command
- Issue a *AccessChannelMAC.Deactivate* command
- Return a *ConnectionFailed* indication
- Transition to the Sleep State.

### 6.3.5.9.2 Access network requirements

Upon entering the BindUATI State, the access network shall:

- Send a packet with the header field UATIInfoIncluded set to '1'. The setting of UATIStatus in the MAC header is beyond the scope of this specification.
- If the access network sets UATIStatus to 1 (does not accept the connection), it shall:
  - Return a *ConnectionFailed* indication
  - Issue a *ReverseTrafficChannelMAC.Deactivate* command
  - Issue a *ReverseControlChannelMAC.Deactivate* command
  - Transition to the Sleep State.
- If the access network accepts the connection request (sends a *ConnectionOpenResponse* message with *ConnectionStatus* set to 0), the Idle State Protocol shall return a *ConnectionOpened* Indication and transition to the Inactive State. The access network should set the *RegistrationRadiusFlag* field of the *ConnectionOpenResponse* message to '0' only if the *RegistrationRadiusFlag* field of the *ConnectionOpenRequest* message was set to 0.

Note that activation of the Reverse Traffic Channel MAC and the Reverse Control Channel MAC at the access network is performed by the Access Channel MAC Protocol in the Lower MAC Sublayer.

### 6.3.6 Message formats

The protocol uses the *AttributeUpdateRequest*, *AttributeUpdateAccept*, and *AttributeUpdateReject* messages of the Generic Attribute Update Protocol in 10.9 to update configurable attributes.

### 6.3.6.1 ConnectionOpenRequest

The access terminal sends the ConnectionRequest message to the access network to request the opening of a connection.

Field	Length (bits)
MessageID	8
ConnectRequestReason	2
RegistrationRadiusFlag	1
Reserved	5

**MessageID** The access network shall set this field to 0x00.

**ConnectRequestReason** The access terminal shall set this field according to Table 38.

**Table 38 Encoding of the ConnectRequestReason field**

Field value	Description
0x0	Response to page
1	Registration Attempt
2	Terminal initiated data transfer
All other values are reserved	

**RegistrationRadiusFlag** The access terminal shall set this field based on RegistrationRadiusFlag that is public data of the Active Set Management Protocol.

**Reserved** This field shall be ignored by the receiver.

<b>Channels</b>	RTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Unicast	<b>Security</b>	Required

### 6.3.6.2 ConnectionOpenResponse

The access network sends the ConnectionOpenResponse message to the access terminal in response to a ConnectionOpenRequest message.

Field	Length (bits)
MessageID	8
ConnectionStatus	1
RegistrationRadiusFlag	1
Reserved	6

- 1 MessageID The access network shall set this field to 0x01.
- 2 ConnectionStatus The access terminal shall set this field according to Table 39.

**Table 39 Encoding of the ConnectionStatus field**

Field value	Description
0	Connection opened
1	Registration successful, connection closed
All other values are reserved	

- 4 RegistrationRadiusFlag The setting of this field at the access network is beyond the scope of this specification.
- 5
- 6
- 7 Reserved This field shall be ignored by the receiver.
- 8

<b>Channels</b>	FTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Unicast	<b>Security</b>	Required

9

**6.3.6.3 PageUATI**

11 The access network may send the PageUATI message to direct the access terminal to request a connection.

12

Field	Length (bits)
MessageID	8
UATI	128

- 14 MessageID The access network shall set this field to 0x02.
- 15 UATI The access network shall set this field to the UATI of the access terminal, where the UATI is public data of the Address Management Protocol.
- 16
- 17

<b>Channels</b>	FTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Broadcast Unicast	<b>Security</b>	Required

18

### 6.3.6.4 PreferredChannelRequest

The access terminal sends this message to request a connection on a channel that is different from the channel where the access preamble was sent. Sending the PreferredChannelRequest message during connection initiation is optional.

Field	Length (bits)
MessageID	8
MessageSequence	8
PreferredChannelBandCount	8
PreferredChannelBandCount occurrences of the following field:	
PreferredChannelBand	ChannelBandRecordType Dependent

**MessageID** The access terminal shall set this field to 0x03.

**MessageSequence** The access terminal shall increment this field modulo 256 for each new PreferredChannelRequest message sent. If this is the first PreferredChannelRequest message sent by the access terminal, the access terminal shall set this field to zero.

**PreferredChannelBandCount** The access terminal shall set this field to the number of occurrences of the PreferredChannelBand field in this message.

**PreferredChannelBand** The access terminal shall set this field to the ChannelBand record specification for the channel on which the access terminal prefers to be assigned a Traffic Channel.

<b>Channels</b>	RTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Unicast	<b>Security</b>	Required

## 6.3.7 Interface to other protocols

### 6.3.7.1 Commands

This protocol issues the following commands:

- *ActiveSetManagement.SendPilotReport*
- *OverheadMessages.Activate*
- *OverheadMessages.Deactivate*
- *ControlChannelMAC.Activate*
- *ControlChannelMAC.Deactivate*
- *ForwardTrafficChannelMAC.Activate*



- 1       ■ *ReverseTrafficChannelMAC.Activate*
- 2       ■ *ReverseControlChannelMAC.Activate*
- 3       ■ *AccessChannelMAC.AttemptAccess*
- 4       ■ *SharedSignalingMAC.Activate*
- 5       ■ *SharedSignalingMAC.Deactivate*

### 6.3.7.2 Indications

7 This protocol registers to receive the following indications:

- 8       ■ *AccessChannelMAC.AccessGrantReceived*
- 9       ■ *AccessChannelMAC.AccessFailed*
- 10      ■ *ReverseTrafficChannelMAC.UATIReceived*
- 11      ■ *ForwardTrafficChannelMAC.UATIReceived*
- 12      ■ *ControlChannelMAC.PageReceived*
- 13      ■ *ControlChannelMAC.QuickPageReceived*
- 14      ■ *ForwardTrafficChannelMAC.PageReceived*

### 6.3.8 Configuration attributes

16 The following complex attributes and default values are defined (see 10.3 for attribute record  
17 definition).

18 Unless specified otherwise, the access terminal and the access network shall use the Generic Attribute  
19 Update Protocol in 10.9 to update configurable attributes belonging to the Default Idle State Protocol.

#### 6.3.8.1 Preferred paging attribute

Field	Length (bits)	Default
Length	8	N/A
AttributeID	8	N/A
PreferredPageOffsetEnabled	1	'0'
PreferredPageOffset	0 or 15	N/A
PreferredQuickPageEnabled	1	0
PreferredRQuickPage	0 or 15	N/A
PreferredPagingCarrierEnabled	1	0
PreferredPagingCarrier	7	N/A
Reserved	7 or 0	N/A

22 Length                      Length of the complex attribute in octets. The sender shall set this field to the  
23 length of the complex attribute excluding the Length field.

24 AttributeID                This field shall be set to 0x00.

- 1 PreferredPageOffsetEnabled  
 2 This field shall be set to '1' if PreferredPageOffset field is included in this  
 3 attribute; otherwise, the sender shall set this field to '0'.
- 4 PreferredPageOffset  
 5 If PreferredPageOffsetEnabled is set to '1', this field shall specify the  
 6 superframe in which the access terminal transitions out of the Sleep State  
 7 (see 6.3.5.6) in order to monitor the Control Channel. This field shall be  
 8 omitted if PreferredPageOffsetEnabled is set to '0'.
- 9 PreferredQuickPageEnabled  
 10 This field shall be set to '1' if the PreferredQuickPageCycle field is included  
 11 in this attribute; otherwise, this field shall be set field to '0'.
- 12 PreferredRQuickPage PreferredQuickPageEnabled is set to '1', this field shall be set to specify the  
 13 response of the access terminal to a QuickPage packet (see 6.3.5.4.2). This  
 14 field shall be omitted this field if PreferredQuickPageEnabled is set to '0'.
- 15 Reserved The length of this field shall be such that the attribute value is octet-aligned.  
 16 This field shall be set to zero.

### 17 6.3.8.2 SlottedMode attribute

Field	Length (bits)	Default
Length	8	N/A
AttributeID	8	N/A
SlotCycleBase	8	0x2
SlotCycle1	8	0x9
SlotCycle2	8	0x9
SlotCycle3	8	0x9
WakeCount1	8	0x0
WakeCount2	8	0x0
Reserved	0	N/A

- 19 Length Length of the complex attribute in octets. The sender shall set this field to the  
 20 length of the complex attribute excluding the Length field.
- 21 AttributeID The sender shall set this field to 0x01.
- 22 SlotCycleBase The sender shall set this field to the SlotCycleBase that is used in calculating  
 23 Period $j$ . SlotCycleBase shall take only even values.
- 24 SlotCycle1 The sender shall set this field to SlotCycle1.
- 25 SlotCycle2 The sender shall set this field to SlotCycle2. SlotCycle2 shall be greater than  
 26 or equal to SlotCycle1.

- 1 SlotCycle3 The sender shall set this field to SlotCycle3. SlotCycle3 shall be greater than  
2 or equal to SlotCycle2.
- 3 WakeCount1 The sender shall set this field to WakeCount1.
- 4 WakeCount2 The sender shall set this field to WakeCount2. WakeCount2 shall be greater  
5 or equal to than WakeCount1.
- 6 Reserved The sender shall set this field to '0000'. The receiver shall ignore this field.

### 7 6.3.8.3 FastRepage attribute

Field	Length (bits)	Default
Length	8	N/A
AttributeID	8	N/A
FastRepageEnabled	8	0
FastRepageInterval	16	0

- 9 Length Length of the complex attribute in octets. The sender shall set this field to the  
10 length of the complex attribute excluding the Length field.
- 11 AttributeID The sender shall set this field to 0x03.
- 12 FastRepageEnabled The sender shall set this field to 0x01 if FastRepage is enabled
- 13 FastRepageInterval This sender shall set this field to zero if FastRepageEnabled is not equal to  
14 0x01. Otherwise, the sender shall set this field to the interval at which the  
15 access network pages the access terminal when the access network receives  
16 no response to the page. The unit for this field shall be superframes. This  
17 field shall not take odd values. A fast repage is performed only once for each  
18 missed page.

### 19 6.3.8.4 MaxAccessAttempts attribute

Field	Length (bits)	Default
Length	8	N/A
AttributeID	8	N/A
MaxAccessAttempts	8	0x03

- 21 MaxAccessAttempts The sender shall set this field to the maximum number of visits to the Access  
22 State before the access attempt is stopped.

### 6.3.9 Protocol numeric constants

Constant	Meaning	Value	Comments
N <sub>IDPType</sub>	Type field for this protocol	<b>Table 9</b>	
N <sub>IDPDefault</sub>	Subtype field for this protocol	0x0000	
T <sub>IDSTABind</sub>	Maximum access terminal time in the BindUATI State	2.5 seconds	

### 6.3.10 Session state information

The Session State Information record (see 10.10) consists of the PageTimes array and the parameter records.

The parameter records for this protocol consist of the configuration attributes of this protocol.

## 6.4 Default Connected State Protocol

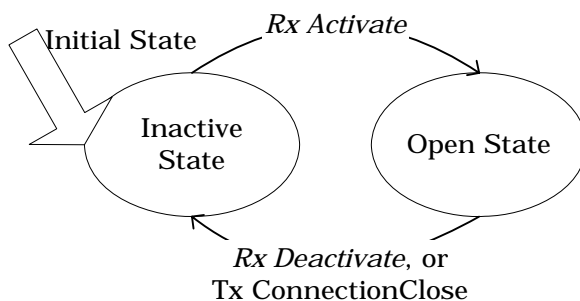
### 6.4.1 Overview

The Default Connected State Protocol provides procedures and messages used by the access terminal and the access network while a connection is open.

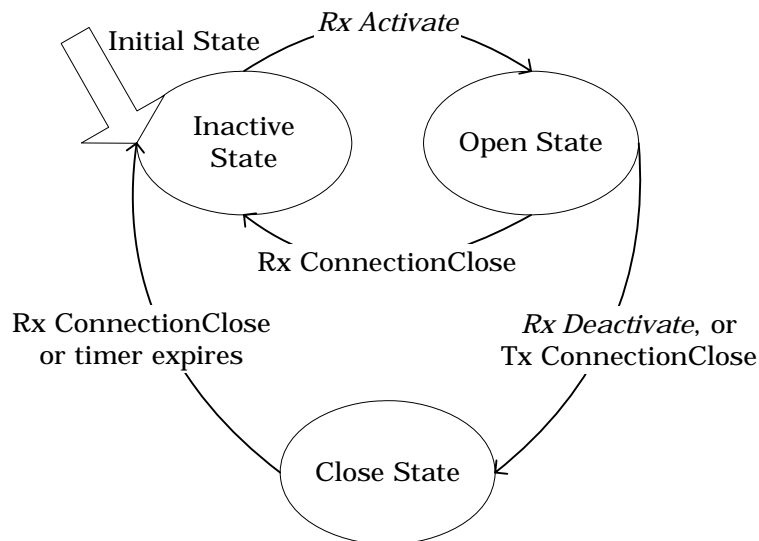
This protocol can be in one of three states:

- *Inactive State*: In this state the protocol waits for an *Activate* command.
- *Open State*: In this state the access terminal can use the Reverse Traffic Channel and the access network can use the Forward Traffic Channel and Control Channel to send application traffic to each other.
- *Close State*: This state is associated only with the access network. In this state the access network waits for connection resources to be safely released.

Figure 52 and Figure 53 show the state transition diagrams at the access terminal and the access network respectively.



**Figure 52 Default Connected State Protocol state diagram (access terminal)**



1  
2 **Figure 53 Default Connected State Protocol state diagram (access network)**

3 **6.4.2 Primitives**

4 **6.4.2.1 Commands**

5 This protocol defines the following commands:

- 6 ■ *Activate*
- 7 ■ *Deactivate*
- 8 ■ *CloseConnection*<sup>29</sup>

9 **6.4.2.2 Return indications**

10 This protocol returns the following indications:

- 11 ■ *ConnectionClosed*
- 12 ■ *RegistrationRadiusUpdated*
- 13 ■ *TunedAway*
- 14 ■ *TunedBack*

15 **6.4.3 Public data**

16 **6.4.3.1 Static public data**

17 This protocol does not define any static public data

---

<sup>29</sup> The *CloseConnection* command performs the same function as the *Deactivate* command and is provided for clarity in the specification.

### 6.4.3.2 Dynamic public data

- Subtype for this protocol
- RLImplicitDeassignEnabled (configuration attribute)
- FLImplicitDeassignEnabled (configuration attribute)
- TuneAwayStatus
- SelectedInterlaceMode
- SelectedInterlaceAssignment

## 6.4.4 Protocol initialization and swap procedures

### 6.4.4.1 Protocol initialization

Upon initialization at the access terminal and the access network:

- The value for each attribute for this protocol instance shall be set to the default value for that attribute.
- The protocol shall enter the Inactive State.

### 6.4.4.2 Protocol swap

Upon swap at the access terminal and access network:

- The protocol shall enter the Inactive State.

## 6.4.5 Procedures

### 6.4.5.1 Command processing

#### 6.4.5.1.1 Activate

When the protocol receives an *Activate* command in the Inactive State:

- The access terminal shall transition to the Open State.
- The access network shall transition to the Open State.

When the protocol receives this command in any other state it shall be ignored.

#### 6.4.5.1.2 Deactivate

When the protocol receives a *Deactivate* command in the Inactive State it shall return a *ConnectionClosed* indication.

When the protocol receives a *Deactivate* command in the Close State the command shall be ignored.

1 When the protocol receives this command in the Open State:

- 2 ■ Access terminal shall send a ConnectionClose message to the access network and  
3 perform the cleanup procedures defined in 6.4.5.2.4.2.
- 4 ■ Access network shall send a ConnectionClose message to the access terminal and  
5 transition to the Close State.

### 6 6.4.5.1.3 CloseConnection

7 The access terminal and the access network shall process the *CloseConnection* command following  
8 the same procedures used for the *Deactivate* command, see 6.4.5.1.2.

### 9 6.4.5.2 Open state

10 In the Open State, the access terminal and the access network maintain a connection and can use it to  
11 exchange application traffic on the Reverse Traffic Channel, Forward Traffic Channel, and Control  
12 Channel.

#### 13 6.4.5.2.1 TuneAway procedures

14 Tune away defines a repetitive set of time periods during which the access terminal and access  
15 network do not exchange any transmission. TuneAway is used by the access terminal and access  
16 network to allow the access terminal to measure the availability of other ChannelBands or other  
17 technologies. Additionally, TuneAway is used by the access terminal when the some overhead  
18 parameters are not up to date.

19 The beginning and end of tune away is determined by the TuneAway attribute and the  
20 TuneAwayRequest and TuneAwayResponse messages. Further, the access terminal and access  
21 network may operate on multiple tune away schedules. Each tune away schedule is specified by a  
22 separate TuneAway attribute, but may share the same TuneAwayRequest and TuneAwayResponse  
23 messages.

24 The tune away operation is controlled through a variable TuneAwayStatus, that is public data of the  
25 protocol.

26 If the TuneAwayStatus is set to '1' at the access terminal:

- 27 ■ The access terminal may stop monitoring the forward channels and shall stop transmitting  
28 on the reverse channels.

29 If the TuneAwayStatus is set to '1' at the access network:

- 30 ■ The access network may stop monitoring the reverse channels and shall stop transmitting  
31 to this access terminal on the forward channels.

#### 32 6.4.5.2.1.1 TuneAway time calculations

33 The following formulas determine the beginning and end of tune away periods for a tune away  
34 schedule as a function of the TuneAway attribute and the TuneAwayRequest message. In the  
35 following calculations, the current TuneAway attribute is used, and the TuneAwayRequest message  
36 with MessageSequence matching the last received TuneAwayResponse message is used.

1 Consider tune away schedule N and consider a given serving sector. Let  $t_0$  be the system time at the  
 2 beginning of StartSuperframeNumberN. Then, tuneaway period number n for a sector with given  
 3 SectorOffset begins at

$$4 \quad \text{TuneAwayTime}_{N_n} = t_0 + \text{StartSuperframeOffset}_N + \text{SectorOffset} + (n-1) * \text{TuneAwayPeriodicity}_N$$

5 and ends at

$$6 \quad \text{TuneBackTime}_{N_n} = \text{TuneAwayTime}_{N_n} + \text{TuneAwayDuration}_N$$

7 Since a TuneAwayTime and a TuneBackTime may be misaligned with PHY Frame boundaries,  
 8 actual tune away and tune back operations obey the following parameters.

9 TuneAwayFrame $N_n$  shall be set to the PHY Frame number of the reverse link PHY Frame that  
 10 contains time instance TuneAwayTime $N_n$ . In case TuneAwayTime $N_n$  lies on a boundary of PHY  
 11 Frames j and j+1, TuneAwayFrame $N_n$  shall be set to j.

12 TuneBackFrame $N_n$  shall be set to the PHY Frame number of the reverse link PHY Frame that  
 13 contains time instance TuneBackTime $N_n$ . In case TuneBackTime $N_n$  lies on a PHY Frame boundary of  
 14 PHY Frames j and j+1, TuneBackFrame $N_n$  shall be set to j+1.

15 For each tune away schedule, the access terminal and access network may be in one of three states:

- 16 ■ *Disabled State*: When a tune away schedule is in this state, the access terminal does not  
 17 tune away for that tune away schedule.
- 18 ■ *Camped State*: When a tune away schedule is in this state, it does not require the access  
 19 terminal to tune away.
- 20 ■ *TunedAway State*: When a tune away schedule is in this state, it requires the access  
 21 terminal to tune away.

22 In addition to the above per schedule state, the access terminal shall also perform State transitions for  
 23 a tune away schedule.

#### 24 **Access Terminal**

25 The access terminal shall enter the disabled state for a tune away schedule N if

- 26 ■ No valid TuneAwayResponse message has been received
- 27 ■ The access terminal sends a TuneAwayRequest message with the TuneAwayEnabled $N$   
 28 field set to '0'.

29 The access terminal shall enter the Camped state for a tune away schedule N if

- 30 ■ If the access terminal receives a TuneAwayResponse message with the  
 31 TuneAwayEnabled $N$  field set to '1'.
- 32 ■ At the beginning of PHY Frame number TuneBackFrame $N_n$

33 The access terminal shall enter the TunedAway state for a tune away schedule N

- 34 ■ At the end of PHY Frame number TuneAwayFrame $N_n - 1$



## 1 Access Network

2 The access network shall enter the disabled state for a tune away schedule N if

- 3 ■ No TuneAwayResponse message for tune away schedule N has been sent
- 4 ■ The access network sends a TuneAwayResponse message with the TuneAwayEnabledN
- 5 field set to '0'.

6 The access network shall enter the Camped state for a tune away schedule N if

- 7 ■ If the access network sends a TuneAwayResponse message with the TuneAwayEnabledN
- 8 field set to '1'.
- 9 ■ At the beginning of PHY Frame number TuneBackFrameN<sub>n</sub>

10 The access network shall enter the TunedAway state for a tune away schedule N

- 11 ■ At the end of PHY Frame number TuneAwayFrameN<sub>n</sub> - 1

### 12 6.4.5.2.1.2 Procedures for setting TuneAwayStatus

13 The access terminal shall set the public data TuneAwayStatus as follows:

- 14 ■ If any of the tune away schedules is in TuneAway state, this protocol shall set
- 15 TuneAwayStatus to '1'.
- 16 ■ If QuickChannelInfoUptoDate is '0' or ExtendedChannelInfoUptoDate is '0' in the
- 17 public data of the Overhead Messages Protocol, the access terminal shall set
- 18 TuneAwayStatus to '1'.
- 19 ■ Otherwise, the access terminal shall set TuneAwayStatus to '0'.

20 The access network shall set the public data TuneAwayStatus as follows:

- 21 ■ If any of the tune away schedules is in TuneAway state, this protocol shall set
- 22 TuneAwayStatus to '1'.
- 23 ■ Otherwise, the access terminal shall set TuneAwayStatus to '0'.

24 This protocol shall generate the following indications:

- 25 ■ If TuneAwayStatus changes from '1' to '0', this protocol shall generate a TunedBack
- 26 indication.
- 27 ■ If TuneAwayStatus changes from '0' to '1', this protocol shall generate a TunedAway
- 28 indication.

### 29 6.4.5.2.2 SelectedInterlace operation procedures

30 The access terminal and access network may operate in one of two modes: SelectedInterlaceOn or

31 SelectedInterlaceOff.

- 32 ■ *SelectedInterlaceOn* mode: In this mode, the access network sends certain SSCH blocks
- 33 to the access terminal only on a set of interlaces called the SelectedInterlaceSet. Details
- 34 may be found in the Lower MAC Sublayer.
- 35 ■ *SelectedInterlaceOff* mode: In this mode, no restrictions are placed on the access network
- 36 and access terminal.

#### 6.4.5.2.2.1 State transitions for selected interlace operation

The access network shall enter the SelectedInterlaceOn mode after

- Sending a SelectedInterlaceAck message with SelectedInterlaceEnabled set to '1'.

The access terminal shall enter the SelectedInterlaceOn mode after

- Receiving a SelectedInterlaceAck message with SelectedInterlaceEnabled set to '1'.

The access terminal shall enter the SelectedInterlaceOff mode after

- Receiving a SelectedInterlacesAssignment message with SelectedInterlacesEnabled equal to '0'.
- When the desired serving sector is not the same as the serving sector

The access network shall enter the SelectedInterlaceOff mode after

- Receiving a SelectedInterlaceAck message with the SelectedInterlacesEnabled field equal to '0'
- The serving sector for the access terminal changes

If the access terminal receives a SelectedInterlaceAssignment message with SelectedInterlacesEnabled equal to '0', the access terminal shall respond with a SelectedInterlaceAck message.

To change the selected interlace assignment to an access terminal, the access network should first disable selected interlace mode, and then send a SelectedInterlaceAssignment message.

#### 6.4.5.2.2.2 Procedures in selected interlace states

On entering the SelectedInterlaceOn state

- The access terminal and access network shall set public data SelectedInterlaceMode to '1'.
- Place the most recent SelectedInterlaceAssignment message in the public data.

On entering the SelectedInterlaceOff state

- The access terminal and access network shall set public data SelectedInterlaceMode to '0'.

#### 6.4.5.2.3 Channel measurement procedures

The access network may obtain channel measurement reports from the access terminal by sending a ChannelMeasurementReportRequest message.

If an access terminal receives a ChannelMeasurementReportRequest message, the access terminal may respond with a ChannelMeasurementReport message. Channel measurements are based on the F-CPICH.

#### 1 6.4.5.2.4 Access terminal requirements

##### 2 6.4.5.2.4.1 General requirements

3 The access terminal shall comply with the following requirements when in the Open State:

- 4 ■ The access terminal shall receive the Control Channel and the Forward Traffic Channel.
- 5 ■ The access terminal may request a MIMO mode on the Forward Traffic Channel by  
6 sending a MIMORequest message.
- 7 ■ The access terminal shall monitor the overhead messages as specified in the Overhead  
8 Messages Protocol (see 6.5.5.4.2).
- 9 ■ If the access terminal receives a ConnectionClose message, it shall generate a  
10 *RegistrationRadiusUpdated* indication accompanied by the RegistrationRadiusFlag  
11 contained in the message.
- 12 ■ If the access terminal receives a ConnectionClose message, it shall send a  
13 ConnectionClose message with CloseReason set to “Close Reply” and execute the  
14 cleanup procedures defined in 6.4.5.2.4.2.
- 15 ■ If the access terminal sends a ConnectionClose message, it may advertise, as part of the  
16 ConnectionClose message, that it shall be monitoring the Control Channel continuously,  
17 until a certain time following the closure of the connection. This period is called a  
18 suspend period, and can be used by the access network to accelerate the process of  
19 sending a unicast packet (and specifically, a Page message or ActiveSetAssignment  
20 message) to the access terminal. The suspend period shall be said to be current from the  
21 time the access terminal sends the ConnectionClose message to the time given in the  
22 SuspendTime field of the ConnectionClose message.

##### 23 6.4.5.2.4.2 Cleanup procedures

24 If the access terminal executes cleanup procedures it shall:

- 25 ■ Return a *ConnectionClosed* indication.
- 26 ■ Transition to the Inactive State.

#### 27 6.4.5.2.5 Access network requirements

##### 28 6.4.5.2.5.1 General requirements

29 The access network shall comply with the following requirements when in the Open State:

- 30 ■ Access network shall receive the Reverse Traffic Channel and may transmit on the  
31 Forward Traffic Channel.
- 32 ■ If access network receives a ConnectionClose message, it shall consider the connection  
33 closed, and it should execute the cleanup procedures defined in 6.4.5.2.5.2.
- 34 ■ If access network requires closing the connection, it shall transmit a ConnectionClose  
35 message, and transition to the Close State.

### 6.4.5.2.5.2 Cleanup procedures

When the access network performs cleanup procedures it shall:

- Return a *ConnectionClosed* indication.
- Transition to the Inactive State.

### 6.4.5.3 Close state

The Close State is associated only with the access network. In this state the access network waits for a replying *ConnectionClose* message from the access terminal or for the expiration of the “CSP Close Timer” defined below.

Upon entering this state, the access network shall set a “CSP Close Timer” for  $T_{\text{CSPClose}}$  seconds. If the access network receives a *ConnectionClose* message in this state, or if the timer expires, it shall execute the cleanup procedures defined in 6.4.5.2.5.2, it may close all connection-related resources assigned to the access terminal, and it should transition to the Inactive State.

### 6.4.6 Message formats

The protocol uses the *AttributeUpdateRequest*, *AttributeUpdateAccept*, and *AttributeUpdateReject* messages of the Generic Attribute Update Protocol in 10.9 to update configurable attributes.

#### 6.4.6.1 ConnectionClose

The access terminal and the access network send the *ConnectionClose* message to close the connection.

Field	Length (bits)
MessageID	8
CloseReason	3
SuspendEnable	1
SuspendTime	0 or 34
RegistrationRadiusFlag	1
Reserved	variable

MessageID                      The sender shall set this field to 0x00.

CloseReason                    The sender shall set this field to reflect the close reason, as shown in Table 40.

**Table 40 Encoding of the CloseReason field**

Field value	Description
'000'	Normal Close; Reason Unspecified
'001'	Close Reply
'010'	Connection Error
'011'	Deregistration Request
'100'	Normal close requested by access terminal because the connection was opened for registration.
All other values are reserved	

**SuspendEnable** The access terminal shall set this field to '1' if it will enable a suspend period following the close of the connection. The access terminal shall set this field to '0' if the CloseReason field is set to "Deregistration Request". The access network shall set this field to '0'.

**SuspendTime** Suspend period end time. This field is included only if the SuspendEnable field is set to '1'. The access terminal shall set this field to the absolute system time of the end of its suspend period in units of superframes.

**RegistrationRadiusFlag** This field shall be set by the access terminal to RegistrationRadiusFlag that is public data of the Active Set Management Protocol.

**Reserved** The length of this field shall be such that the entire message is octet-aligned. The sender shall set this field to zero. The receiver shall ignore this field.

<b>Channels</b>	FTC	RTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Unicast		<b>Security</b>	Required

#### 6.4.6.2 MIMORquest

This message shall be sent by the access terminal to the access network to indicate its MIMO capabilities.

Field	Length (bits)
MessageID	8
SupportedMIMOMode	2
Reserved	6

**MessageID** The access terminal shall set this field to 0x01.

## SupportedMIMOMode

The access terminal shall set this field to indicate the MIMO modes it supports, as shown in Table 41.

**Table 41 Encoding of the SupportedMIMOMode field**

Field value	Description
'00'	MIMO not supported
'01'	Space Time Transmit Diversity (STTD)
'10'	Single Code Word (SCW)
'11'	Multiple Code Word (MCW)

## Reserved

The access terminal shall set this field to all zeros.

<b>Channels</b>	RTC	<b>SLP</b>	Reliable
<b>Addressing</b>	Unicast	<b>Security</b>	Required

## 6.4.6.3 SelectedInterlaceRequest

This message shall be sent by the access terminal to request a selected interlace mode with a particular sector.

Field	Length (bits)
MessageID	8
PilotPN	12
InterlacesRequested	4

## MessageID

This field shall be set to 0x02

## PilotPN

This field shall be set to the PilotPN of the sector to which this message is directed. The access network shall ignore this message if the PilotPN does not match the PilotPN of the sector that received the message.

## InterlacesRequested

The access terminal shall set this field to indicate a requested number of interlaces requested.

<b>Channels</b>	RTC	<b>SLP</b>	Reliable
<b>Addressing</b>	Unicast	<b>Security</b>	Required

#### 6.4.6.4 SelectedInterlaceAssignment

This message shall be sent by the access network to assign a selected interlace mode to an access terminal.

Field	Length (bits)
MessageID	8
PilotPN	12
SelectedInterlacesEnabled	1
NumAssignedInterlaces	4
NumAssignedInterlaces instances of the following field	
InterlaceID	3
Reserved	Variable

**MessageID** This field shall be set to 0x03.

**PilotPN** This field shall be set to the PilotPN of the sector that sent this message.

**SelectedInterlacesEnabled**  
If this field is set to '1' the access terminal shall operate in SelectedInterlace mode. If this field is set to '0' the access terminal shall not operate in SelectedInterlace mode.

**NumAssignedInterlaces**  
The access network shall set this field to the number of assigned interlaces, or to 0 if the SelectedInterlacesAssigned field is set to '0'.

**InterlaceID** This field shall be set to an interlace assigned to the access terminal for SelectedInterlace operation.

**Reserved** The length of this field shall be such that the entire message is octet-aligned. The sender shall set this field to zero. The receiver shall ignore this field.

<b>Channels</b>	FTC
-----------------	-----

<b>SLP</b>	Reliable
------------	----------

<b>Addressing</b>	Unicast
-------------------	---------

<b>Security</b>	Required
-----------------	----------

### 6.4.6.5 SelectedInterlaceAck

This message shall be sent by the access terminal to acknowledge transition to SelectedInterlacesOff state.

Field	Length (bits)
MessageID	8
PilotPN	12
SelectedInterlaceEnabled	1
Reserved	3

**MessageID** This field shall be set to 0x04.

**PilotPN** This field shall be set to the PilotPN of the sector to which this message is directed. The access network shall ignore this message if the PilotPN does not match the PilotPN of the sector that received the message.

**SelectedInterlaceEnabled** The access terminal shall set this field to '1' if it has selected interlace mode enabled, and to '0' otherwise.

**Reserved** The sender shall set this field to zero. The receiver shall ignore this field.

<b>Channels</b>	RTC	<b>SLP</b>	Reliable
<b>Addressing</b>	Unicast	<b>Security</b>	Required

### 6.4.6.6 TuneAwayRequest

This message shall be sent by the access terminal to control tune away operations.

Field	Length (bits)
MessageID	8
MessageSequence	16
TuneAwayEnabledMap	$N_{CSPTuneAwayMaxSched}$
NumPilots	3
NumPilots instances of the following field	
ActiveSetIndex	3
SectorOffset	24
Reserved	Variable



- 1 MessageID This field shall be set to 0x05.
- 2 MessageSequence The access terminal shall set this field to the sequence number of this  
3 message. The sequence number of this message is 1 more than the sequence  
4 number of the last TuneAwayRequest message (modulo 65536) sent by this  
5 access terminal. If this is the first TuneAway message sent by the access  
6 terminal, it shall set this field to 0x00.
- 7 TuneAwayEnabledMap Bit position  $N$  of this field shall be set to TuneAwayEnabled $N$ .  
8 TuneAwayEnabled $N$  shall be set to '1' if the terminal will tune away at  
9 periodic intervals corresponding to tune away schedule  $N$ .  
10 TuneAwayEnabled $N$  shall be set to '0' if the terminal will not tune away  
11 corresponding to tune away schedule  $N$ .
- 12 NumPilots This field shall be set to the number of pilots included in the message.
- 13 ActiveSetIndex This field shall be used to identify Active Set members, as indexed in the  
14 ActiveSetAssignment message of the Active Set Management Protocol.
- 15 SectorOffset This field shall be set to the time, in units of 1 microsecond, that the terminal  
16 adds to the StartSuperframeOffset attribute when this Active Set member is  
17 the serving sector. The access terminal should determine this field based on  
18 the timing offset it measures between different sectors.
- 19 Reserved The length of this field shall be such that the entire message is octet-aligned.  
20 The sender shall set this field to zero.

<b>Channels</b>	RTC	<b>SLP</b>	Reliable
<b>Addressing</b>	Unicast	<b>Security</b>	Required

22

### 23 6.4.6.7 TuneAwayResponse

24 This message shall be sent by the access network to control tune away operations.

25

Field	Length (bits)
MessageID	8
MessageSequence	16
TuneAwayEnabledMap	$N_{CSPTuneAwayMaxSched}$
Reserved	Variable

- 26 MessageID This field shall be set to 0x06.
- 27 MessageSequence The access network shall set this field to the last received TuneAwayRequest  
28 message sent to this access terminal.

## 1 TuneAwayEnabledMap

2 Bit position  $N$  of this field shall be set to TuneAwayEnabled $N$ .

3 TuneAwayEnabled $N$  shall be set to '1' if the access network will tune away  
4 at periodic intervals corresponding to tune away schedule  $N$ .

5 TuneAwayEnabled $N$  shall be set to '0' if the access network will not tune  
6 away corresponding to tune away schedule  $N$ .

## 7 Reserved

The length of this field shall be such that the entire message is octet-aligned.

8 The sender shall set this field to zero. The receiver shall ignore this field.

<b>Channels</b>	FTC	<b>SLP</b>	Reliable
<b>Addressing</b>	Unicast	<b>Security</b>	Required

## 11 6.4.6.8 ChannelMeasurementReportRequest

12 The access network sends this message to request a ChannelMeasurementReport from one or more  
13 access terminals.

Field	Length (bits)
MessageID	8
PilotPN	12
CarrierID	2
StartPHYFrame	40
NumChannels	3
MeasurementsPerMessage	8
NumMeasurementsRequested	8
Reserved	4

### 14 MessageID

This field shall be set to 0x07.

### 15 PilotPN

16 This field shall be set to the PilotPN of the sector requesting the  
measurement report.

### 17 CarrierID

18 This field shall be set to the carrier on which the measurements are  
requested.

### 19 StartPHYFrame

20 This field shall be set to the PHY Frame number of the PHYFrame where  
access terminals are required to begin measurements.

### 21 NumChannels

22 This field shall be set to the number of channels to be measured by the access  
terminal. Each measured channel corresponds to a different transmit antenna  
23 at the sector being measured.

### 24 MeasurementsPerMessage

25 This field shall determine the number of measurements (in terms of PHY  
26 Frames measured) to be included in one ChannelMeasurementReport  
27 message.

1 NumMeasurementsRequested

2 This field shall determine the total number of measurements to be made by  
3 the access terminal.

4 Reserved

This field shall be set to zero. The receiver shall ignore this field.

<b>Channels</b>	FTC	<b>SLP</b>	Reliable
<b>Addressing</b>	Broadcast Unicast	<b>Security</b>	Required

#### 6.4.6.9 ChannelMeasurementReport

8 The access terminal sends this message to report channel measurements.

Field	Length (bits)
MessageID	8
PilotPN	12
CarrierID	2
StartPHYFrameNumber	40
MeasurementInterval	8
NumMeasurements	8
NumMeasurements instances of the following record {	
NumChannels	3
NumChannels instances of the following record {	
NumTaps	3
NumTaps instances of the following record {	
TapOffset	5
RealGain	8
ImagGain	8
}}}	
Reserved	Variable

10 MessageID This field shall be set to 0x08.

11 PilotPN This field shall be set to the PilotPN of the sector for which the measurement  
12 was performed.

13 CarrierID This field shall be set to the carrier on which the measurements are  
14 performed.

15 StartPHYFrameNumber

16 This field shall be set to the PHY Frame number of the PHY Frame where  
17 access terminal made the first measurement reported in this message.

1	MeasurementInterval	This field shall determine the number of PHY Frames between
2		measurements made by the access terminal.
3	NumMeasurements	This field shall be set to the number of measurements included in this
4		message. Each measurement corresponds to a different PHY Frame.
5	NumChannels	This field determines the number of channels measured by the access
6		terminal. The access terminal shall set this field to equal to the NumChannels
7		field in the received ChannelMeasurementReportRequest message.
8	NumTaps	This field shall be set to the number of taps being reported.
9	TapOffset	This field shall be set to a offset for which the channel was measured.
10	RealGain	This field shall be set to the real component of the measured channel gain on
11		the corresponding TapOffset.
12	ImagGain	This field shall be set to the imaginary component of the measured channel
13		gain on the corresponding TapOffset.
14	Reserved	The length of this field shall be such that the entire message is octet-aligned.
15		The sender shall set this field to zero. The receiver shall ignore this field.

<b>Channels</b>	RTC
-----------------	-----

<b>SLP</b>	Reliable
------------	----------

<b>Addressing</b>	Unicast
-------------------	---------

<b>Security</b>	Required
-----------------	----------

17

## 6.4.7 Interface to other protocols

18

### 6.4.7.1 Commands

19

This protocol does not issue any commands.

20

### 6.4.7.2 Indications

21

This protocol does not register to receive any indications.

22

## 6.4.8 Configuration attributes

The following configuration attributes are defined for this protocol.

Unless specified otherwise, the access terminal and the access network shall use the Generic Attribute Update Protocol in 10.9 to update configurable attributes belonging to the Default Connected State Protocol.

### 6.4.8.1 Simple attributes

The negotiable simple attribute for this protocol is listed in Table 42. The access terminal and the access network shall use as defaults the values in Table 42 that are listed in *bold italics*.

**Table 42 Configurable values**

Attribute ID	Attribute	Values	Meaning
0x00	RLImplicitDeassignEnabled	<i>0x00</i>	Reverse link assignments are expired at the beginning of tune away
		0x01	Reverse link assignments are not expired at the beginning of tune away
		0x02-0xff	Reserved
0x01	FLImplicitDeassignEnabled	<i>0x00</i>	Forward link assignments are expired at the beginning of tune away
		0x01	Forward link assignments are not expired at the beginning of tune away
		0x02-0xff	Reserved

### 6.4.8.2 Complex attributes

The following complex attributes and default values are defined (see 10.3 for attribute record definition).

#### 6.4.8.3 TuneAwayScheduleN attribute

N takes values from 0 through  $N_{\text{CSPTuneAwayMaxSched}} - 1$ .

This complex attribute shall determine the periodicity, duration and offset of tuneaways that the access terminal may perform. Such tuneaways may be used for handoff candidate search or alternate technology page reception.

Field	Length (bits)	Default
Length	8	N/A
AttributeID	8	N/A
StartSuperframeNumber	34	0
StartSuperframeOffset	16	0
TuneAwayDuration	24	0
TuneAwayPeriodicity	24	0x989680
Reserved	6	0

1	Length	Length of the complex attribute in octets. The sender shall set this field to the
2		length of the complex attribute excluding the Length field.
3	AttributeID	This field shall set this field to $0x0(N+2)$ , where N takes values from 0
4		through $N_{\text{CSPTuneAwayMaxSched}} - 1$ .
5	StartSuperframeNumber	
6		To compute the tuneaway cycles, it shall be assumed that the first tuneaway
7		occurred in this superframe.
8	StartSuperframeOffset	This field is a measure of time in units of 1 micro second. To compute the
9		tuneaway cycles, it shall be assumed that the first tuneaway begins
10		StartSuperframeOffset time after the beginning of superframe number
11		StartSuperframeNumber.
12	TuneAwayDuration	This field determines the duration of the tune away in units of 1 micro
13		second.
14	TuneAwayPeriod	This field determines the time between the start of successive tuneaways in
15		units of 1 microsecond.
16	Reserved	This field shall be set to all zeros.
17		

#### 6.4.9 Protocol numeric constants

Constant	Meaning	Value	Comments
$N_{\text{CSPTType}}$	Type field for this protocol	Table 9	
$N_{\text{CSPDefault}}$	Subtype field for this protocol	0x0000	
$N_{\text{CSPTuneAwayMaxSched}}$	Maximum number of tune away schedules	0x04	
$T_{\text{CSPClose}}$	Access network timer waiting for a responding ConnectionClose message	1.5 seconds	

20

## 6.4.10 Session state information

The Session State Information record (see 10.10) consists of the parameter records.

This protocol defines the following parameter record in addition to the configuration attributes for this protocol.

### 6.4.10.1 ConnectedState parameter

The following parameter shall be included in the Session State Information record only if the Session State Information is being transferred while the connection is open.

**Table 43 The Format of the parameter record for the ActiveSetManagement parameter**

Field	Length (bits)
ParameterType	8
Length	8
SelectedInterlaceAssignmentMessageLength	8
SelectedInterlaceAssignmentMessage	Variable
TuneAwayResponseMessageLength	8
TuneAwayResponseMessage	Variable

**ParameterType** This field shall be set to 0x01 for this parameter record.

**Length** This field shall be set to the length of this parameter record in units of octets excluding the Length field.

**SelectedInterlaceAssignmentMessageLength**  
This field shall be set to the length of the last SelectedInterlaceAssignment message that was sent by the source access network.

**SelectedInterlaceAssignmentMessage**  
Last SelectedInterlaceAssignment message that was sent by the source access network.

**TuneAwayResponseMessageLength**  
This field shall be set to the length of the last TuneAwayResponse message that was sent by the source access network.

**TuneAwayResponseMessage**  
Last TuneAwayResponse message that was sent by the source access network

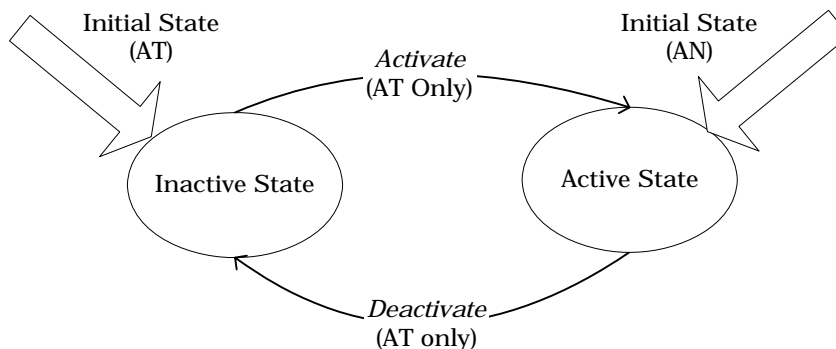
## 6.5 Overhead Messages Protocol

### 6.5.1 Overview

The Overhead Messages Protocol is responsible for the transmission, reception and supervision of the SystemInfo block, the QuickChannelInfo block, the ExtendedChannelInfo message and the SectorParameters message. The SystemInfo and QuickChannelInfo blocks are broadcast by the access network directly over the Control Channel MAC Protocol. The ExtendedChannelInfo and SectorParameters messages are broadcast using the Signaling Transport.

This protocol can be in one of two states:

- *Inactive State*: In this state, the protocol waits for an *Activate* command. This state corresponds only to the access terminal and occurs when the access terminal has not acquired an access network or is not required to receive overhead messages.
- *Active State*: In this state the access network transmits and the access terminal receives overhead messages.



**Figure 54 Overhead Messages Protocol state diagram**

### 6.5.2 Primitives

#### 6.5.2.1 Commands

This protocol defines the following commands:

- *Activate*
- *Deactivate*

#### 6.5.2.2 Return indications

This protocol returns the following indications:

- *SupervisionFailed*
- *QuickChannelInfoUpdated*
- *ExtendedChannelInfoUpdated*



- 1           ■ *SectorParametersUpdated*
- 2           ■ *OverheadMessagesUpdated*

### 3   **6.5.3 Public data**

#### 4   **6.5.3.1 Static public data**

5   This protocol defines the following static public data:

- 6           ■ An OverheadParameterList that shall contain for each PilotPN in the active set, the
  - 7           following entries. When multi-carrier mode is set to MultiCarrierOn in the public data of
  - 8           the Physical Layer Protocol, each of the following entries shall be maintained
  - 9           independently for each carrier, indexed by CarrierID. Other protocols may refer to a field
  - 10          of a block or message as FieldName(CarrierID). For example, FLChannelTreeIndex(2)
  - 11          refers to the FLChannelTreeIndex on carrier 2.
    - 12           □ Received SystemInfo block,
    - 13           □ Received QuickChannelInfo block with associated QuickChannelInfoExpiryTime,
    - 14           □ Received ExtendedChannelInfo message with associated
    - 15           ExtendedChannelInfoExpiryTime,
- 16          ■ Currently valid SectorParameters messages indexed by the sector's PilotPN
- 17          ■ QuickChannelInfoUpToDate
- 18          ■ OverheadParametersUpToDate
- 19          ■ ExtendedChannelInfoUpToDate
- 20          ■ SectorParametersUpToDate

#### 21   **6.5.3.2 Dynamic public data**

- 22          ■ Subtype for this protocol

### 23   **6.5.4 Protocol initialization and swap procedures**

#### 24   **6.5.4.1 Protocol initialization**

25   Upon initialization at the access terminal:

- 26          ■ The value for each attribute for this protocol instance shall be set to the default value for
  - 27           that attribute.
- 28          ■ The protocol shall enter the Inactive State

29   Upon initialization at the access network:

- 30          ■ The value for each attribute for this protocol instance shall be set to the default value for
  - 31           that attribute.
- 32          ■ The protocol shall enter the Active State

## 1 **6.5.4.2 Protocol swap**

2 Upon swap at the access terminal the protocol shall enter the Inactive State.

3 Upon swap at the access network the protocol shall enter the Active State.

## 4 **6.5.5 Procedures**

### 5 **6.5.5.1 Extensibility requirements**

6 Further revisions of the access network may add new overhead messages.

7 The access terminal shall discard overhead messages with a MessageID field it does not recognize.

8 Further revisions of the access network may add new fields to existing overhead messages. These  
9 fields shall be added to the end of the message, prior to the Reserved field, if such a field is defined.

10 The access terminal shall ignore fields it does not recognize.

### 11 **6.5.5.2 Command processing**

12 The access network shall ignore all commands.

#### 13 **6.5.5.2.1 Activate**

14 If this protocol receives an *Activate* command in the Inactive State:

- 15 ■ The access terminal shall transition to the Active State.
- 16 ■ The access network shall ignore it.

17 If this protocol receives the command in the Active State, it shall be ignored.

#### 18 **6.5.5.2.2 Deactivate**

19 If this protocol receives a *Deactivate* command in the Inactive State, it shall be ignored.

20 If this protocol receives the command in the Active State:

- 21 ■ Access terminal shall transition to the Inactive State.
- 22 ■ Access network shall ignore it.

### 23 **6.5.5.3 Inactive state**

24 This state corresponds only to the access terminal and occurs when the access terminal has not  
25 acquired an access network or is not required to receive overhead messages. In this state, the protocol  
26 waits for an *Activate* command.

## 1 6.5.5.4 Active state

### 2 6.5.5.4.1 Access network requirements

3 If the access network is ready to provide service, it shall broadcast the SystemInfo block,  
4 QuickChannelInfo block, ExtendedChannelInfo message and SectorParameters message as specified  
5 below. The SystemInfo block, QuickChannelInfo block, ExtendedChannelInfo message and the  
6 SectorParameters message shall be public data of the Overhead Messages Protocol.

#### 7 6.5.5.4.1.1 Procedure for transmission of the SystemInfo block

8 The SystemInfo block shall be transmitted every  $N_{\text{pBCH0\_Period}}$  superframes. The SystemInfo block  
9 shall be carried by the Control Channel MAC Protocol over the pBCH0 physical channel, and shall  
10 not pass through the Signaling Transport.  $N_{\text{pBCH0\_Period}}$  is defined in the Physical Layer Protocol. If the  
11 multi-carrier mode is MultiCarrierOn, the SystemInfo block shall be transmitted on each carrier, and  
12 all contents of the SystemInfo block except the CarrierID, FLReservedInterlaces and  
13 NumFLReservedSubbands shall be identical for all carriers. The multi-carrier mode is public data of  
14 the physical layer protocol.

#### 15 6.5.5.4.1.2 Procedure for transmission of the QuickChannelInfo block

16 The QuickChannelInfo block shall be transmitted in every superframe with an odd superframe index.  
17 The QuickChannelInfo block shall be carried by the Control Channel MAC Protocol, over the pBCH1  
18 physical channel, and shall not pass through the Signaling Transport.. The information carried in the  
19 QuickChannelInfo block transmitted in superframe  $m=2k+1$  describes the structure of

- 20 ■ All PHY Frames except the first PHY Frame of superframe  $2k+1$ , and
- 21 ■ All PHY Frames of superframe  $2k+2$
- 22 ■ The first PHY Frame in superframe  $2k+3$ .

23 The access network should change the contents of the QuickChannelInfo block in accordance with  
24 the QuickChannelInfoValidity field of the block.

25 If the multi-carrier mode is MultiCarrierOn, the QuickChannelInfo block shall be transmitted on each  
26 carrier. The multi-carrier mode is public data of the physical layer protocol.

#### 27 6.5.5.4.1.3 Procedure for transmission of the ExtendedChannelInfo message

28 The ExtendedChannelInfo message shall be broadcast over the Forward Traffic Channel MAC. The  
29 message shall begin transmission in superframes with superframe numbers divisible by  
30  $N_{\text{OMPEExtendedChannelInfo}}$ . The ExtendedChannelInfo message may be delivered in one superframe, or in a  
31 set of consecutive superframes. If transmission of an ExtendedChannelInfo message begins in  
32 superframe  $n$  and spans  $k$  superframe,

- 33 ■ The ExtendedChannelInfo message should describe the structure of superframes  $n+k$   
34 through  $n+k+\text{ValidityPeriod}$ , where ValidityPeriod is a field of the ExtendedChannelInfo  
35 message.
- 36 ■ The structure of superframes  $n$  through  $n+k-1$  shall be described by the last  
37 ExtendedChannelInfo message transmitted before superframe  $n$ .

1 If the multi-carrier mode is MultiCarrierOn, the ExtendedChannelInfo message shall be transmitted  
2 on each carrier. The multi-carrier mode is public data of the physical layer protocol.

#### 3 **6.5.5.4.1.4 Procedure for transmission of the SectorParameters message**

4 The access network should send a SectorParameters message over the Forward Traffic Channel MAC  
5 in superframe numbers that are divisible by  $N_{\text{OMP SectorParameters}}$ . The access network shall set the  
6 SectorSignature field of the ExtendedChannelInfo message to the SectorSignature field of the next  
7 SectorParameters message.

8 If the multi-carrier mode is MultiCarrierOn, the SectorParameters message shall be transmitted on  
9 each carrier. The multi-carrier mode is public data of the physical layer protocol.

#### 10 **6.5.5.4.2 Access terminal requirements**

11 Upon entering the Active State, the access terminal shall invoke the procedure for determining if the  
12 OverheadMessages are up-to-date, as specified in 6.5.5.4.2.5, and the procedure for generating  
13 TriggerCode based pilot reports, as specified in 6.5.5.4.2.6,. When in the Active State, the access  
14 terminal shall perform supervision on the QuickChannelInfo, ExtendedChannelInfo and the  
15 SectorParameters messages as specified in 6.5.5.4.3.1, 6.5.5.4.3.2 and 6.5.5.4.3.3, respectively.

16 If the access terminal receives a *ActiveSetManagement.IdleHO* indication or if it receives a  
17 *ConnectedState.ConnectionClosed* indication, the access terminal shall invoke the procedures for  
18 determining if the OverheadMessages are up-to-date, as specified in 6.5.5.4.2.5.

19 When the access terminal receives a ExtendedChannelInfo message from a sector, it shall perform the  
20 procedures in 6.5.5.4.2.1.

21 When the access terminal receives a SectorParameters message from a sector, it shall perform the  
22 procedures in 6.5.5.4.2.4.

#### 23 **6.5.5.4.2.1 Procedure for processing SystemInfo block**

24 The access terminal shall place the received SystemInfo block, indexed by PilotPN and CarrierID, in  
25 the public data.

#### 26 **6.5.5.4.2.2 Procedure for processing the QuickChannelInfo block and** 27 **EncapsulatedQuickChannelInfo message**

28 The access terminal shall place the received QuickChannelInfo block, indexed by PilotPN and  
29 CarrierID in the public data.

1 When the access terminal receives a QuickChannelInfo block from a sector, it shall perform the  
2 following:

- 3 ■ If the QuickChannelInfo block is received in superframe n, or if a  
4 EncapsulatedQuickChannelInfo block is received with the SuperframeNumber field set to  
5 n, then at the first end of the first PHY Frame (frame 0) of superframe n, the access  
6 terminal shall:
  - 7 □ If the received QuickChannelInfo block differs from the stored block in the public  
8 data in any field except the QuickChannelInfoValidity field, the access terminal shall  
9 generate a *QuickChannelInfoUpdated* indication.
  - 10 □ Store the block, indexed by PilotPN and CarrierID, in the public data
  - 11 □ If the QuickChannelInfoValidity field is set to m, the access terminal shall set  
12 QuickChannelInfoExpiryTime to the end of the first PHY Frame of superframe  
13  $2 \times 4^m \left\lceil n / (2 \cdot 4^m) \right\rceil + 1$ .

#### 14 6.5.5.4.2.3 Procedure for processing the ExtendedChannelInfo message

15 When the access terminal receives a ExtendedChannelInfo message from a sector, it shall perform the  
16 following:

- 17 ■ The access terminal shall determine the superframe number n when the access network  
18 started ExtendedChannelInfo transmission, and the superframe number n+k-1 when the  
19 access network ended (or will end) ExtendedChannelInfo message transmission. For  
20 example, if transmission of the ExtendedChannelInfo message spans superframes 16 and  
21 17, then n=16 and k=2.
- 22 ■ At the beginning of superframe n+k, the access terminal shall perform the following  
23 operations:
  - 24 □ If the received ExtendedChannelInfo message differs from the ExtendedChannelInfo  
25 message in the public data (for the same PilotPN and CarrierID) in any fields except  
26 the SystemTime, SectorParametersSignature or ValidityPeriod, the access terminal  
27 shall generate an *ExtendedChannelInfoUpdated* indication.
  - 28 □ Store the ExtendedChannelInfo message, indexed by PilotPN and CarrierID in the  
29 public data.
  - 30 □ Set the ExtendedChannelInfoExpiryTime for the message in the public data to  
31 n+ValidityPeriod
- 32 ■ When the access terminal adds a ExtendedChannelInfo message to the public data, it  
33 shall process the stored SectorParameters messages according to the following rules:
  - 34 □ If the public data contains a SectorParameters message with the same PilotPN as the  
35 sector that transmitted the ExtendedChannelInfo message, the access terminal shall  
36 compare the SectorParametersSignature in the ExtendedChannelInfo message with  
37 the SectorParametersSignature in the stored SectorParameters message. If the  
38 signatures do not match, the access terminal shall purge the SectorParameters  
39 message from the public data

#### 1 6.5.5.4.2.4 Procedure for processing the SectorParameters message

2 When the access terminal receives a SectorParameters message, it shall perform the following:

- 3 ■ If the public data contains a SectorParameters message with the same SectorID as the  
4 received message, the access terminal shall compare the SectorParametersSignature of  
5 the received message with the SectorParametersSignature in the stored SectorParameters  
6 message. If the signatures do not match, the access terminal shall:
  - 7 □ Replace the SectorParameters message in the public data with the received  
8 SectorParameters message.
  - 9 □ If the sector is a member of the Active Set, return a *SectorParametersUpdated* and  
10 *OverheadMessagesUpdated* indication.
- 11 ■ If the public data does not contain a SectorParameters message with the SectorID of the  
12 received message, the access terminal shall:
  - 13 □ Add the received SectorParameters message to the public data.
  - 14 □ If the sector is a member of the Active Set, return a *SectorParametersUpdated* and  
15 *OverheadMessagesUpdated* indication.
- 16 ■ If necessary, the access terminal may delete old SectorParameters messages  
17 corresponding to sectors not in the Active Set.

#### 18 6.5.5.4.2.5 Procedure for checking if parameters are up-to-date

19 When this set of procedures is invoked, the access terminal determines QuickChannelInfoUpToDate,  
20 ExtendedChannelInfoUpToDate and SectorParametersUpToDate as follows:

21 QuickChannelInfoUpToDate shall be set to '1' if all of the following conditions are satisfied for the  
22 following members of the Active Set: RLSS, FLSS, DRLSS, DFLSS (as indicated by the public data  
23 of the Reverse Control Channel MAC Protocol). This field shall be set to '0' otherwise.

- 24 ■ A QuickChannelInfo block for this member of the Active Set is available in the public  
25 data.
- 26 ■ The QuickChannelInfo block that is currently in the public data of the protocol has a  
27 validity time that is greater than or equal to the current time.

28 ExtendedChannelInfoUpToDate shall be set to '1' if all of the following conditions are satisfied for  
29 the following members of the Active Set: RLSS, FLSS, DRLSS, DFLSS (as indicated by the public  
30 data of the Reverse Control Channel MAC Protocol). This field shall be set to '0' otherwise.

- 31 ■ An ExtendedChannelInfo message for this member of the Active Set is available in the  
32 public data.
- 33 ■ The ExtendedChannelInfo message that is currently in the public data of the protocol has  
34 a validity time that is greater than or equal to the current time.

1 SectorParametersUpToDate shall be set to ‘1’ if all of the following conditions are satisfied for all  
2 members of the Active Set, and shall be set to ‘0’ otherwise.

- 3 ■ A SectorParameters message with the same PilotPN as this member of the Active Set is  
4 available in the public data.
- 5 ■ An ExtendedChannelInfo block for this member of the Active Set is available in the  
6 public data.
- 7 ■ The SectorParametersSignature in the last received ExtendedChannelInfo block is the  
8 same as the SectorParametersSignature in the stored SectorParameters message.

9 OverheadParametersUpToDate shall be set to the logical “and” of SectorParametersUpToDate and  
10 ExtendedChannelInfoUpToDate. The OverheadParametersUpToDate field is used by the Idle State  
11 Protocol.

12 The access terminal uses tune away procedures of the ConnectedState protocol when  
13 QuickChannelInfoUpToDate or ExtendedChannelInfoUpToDate is set to one.

#### 14 **6.5.5.4.2.6 Procedure for ZoneCode-based registration**

15 The access terminal shall store a list of RegistrationZoneCodes associated with subnets visited by the  
16 access terminal for future comparisons and for future use. This list is called the  
17 RegistrationZoneCodeList. Each entry in the RegistrationZoneCodeList shall include the subnet and  
18 the RegistrationZoneCode. Other protocols may cache information keyed by (Subnet,  
19 RegistrationZoneCode) pairs. If other protocols cache information keyed by (Subnet,  
20 RegistrationZoneCode) pairs, then these protocols shall delete such information when the (Subnet,  
21 RegistrationZoneCode) pair is deleted from the RegistrationZoneCodeList.

22 The access terminal shall be capable of storing exactly  $N_{OMPMinZoneSignatureListSize}$  entries in the  
23 RegistrationZoneCodeList. If the (Subnet, RegistrationZoneCode) pair from the SectorParameters  
24 message from some sector in the Active Set is not included in the RegistrationZoneCodeList, then the  
25 access terminal shall add the entry to the RegistrationZoneCodeList. The access terminal shall  
26 generate a *ActiveSetManagement.SendPilotReport* command when it adds an entry to the  
27 RegistrationZoneCodeList. If there are more entries in the RegistrationZoneCodeList than the  
28 supported size of the RegistrationZoneCodeList, the access terminal shall delete the oldest entries  
29 first. The access terminal shall delete an entry from the RegistrationZoneCodeList when the entry has  
30 stayed in the RegistrationZoneCodeList for  $2^{(RegistrationZoneMaxAge + 3)} \times 1.28$  seconds.

#### 31 **6.5.5.4.3 Supervision procedures**

##### 32 **6.5.5.4.3.1 Supervision of QuickChannelInfo block**

33 The access terminal shall use the following procedure to supervise the QuickChannelInfo block:

- 34 ■ The access terminal shall set a QuickChannelInfo supervision timer for  $T_{OMPQCISupervision}$ .
- 35 ■ If QuickChannelInfoUpToDate becomes ‘1’ while the timer is active, the access terminal  
36 shall disable the timer. If QuickChannelInfoUpToDate becomes ‘0’ while the timer is  
37 inactive, the access terminal shall start the timer.
- 38 ■ If the timer expires, the access terminal shall return a *SupervisionFailed* indication and  
39 disable the timer.

1 Delayed reception of the QuickChannelInfo block may also cause a supervision failure at the Lower  
2 MAC Sublayer.

### 3 **6.5.5.4.3.2 Supervision of ExtendedChannelInfo message**

4 The access terminal shall use the following procedure to supervise the ExtendedChannelInfo  
5 message:

- 6 ■ The access terminal shall set a ExtendedChannelInfo supervision timer for  
7  $T_{\text{OMPECISupervision}}$ .
- 8 ■ If ExtendedChannelInfoUpToDate becomes '1' while the timer is active, the access  
9 terminal shall disable the timer. If ExtendedChannelInfoUpToDate becomes '0' while the  
10 timer is inactive, the access terminal shall start the timer.
- 11 ■ If the timer expires, the access terminal shall return a *SupervisionFailed* indication and  
12 disable the timer.

### 13 **6.5.5.4.3.3 Supervision of SectorParameters message**

14 Upon entering the Active State, the access terminal shall start the following procedure to supervise  
15 the SectorParameters message:

- 16 ■ The access terminal shall set a SectorParameters supervision timer for  $T_{\text{OMPSPSupervision}}$ .
- 17 ■ If SectorParametersUpToDate becomes '1' while the timer is active, the access terminal  
18 shall disable the timer. If SectorParametersUpToDate becomes '0' while the timer is  
19 inactive, the access terminal shall start the timer.
- 20 ■ If the timer expires, the access terminal shall return a *SupervisionFailed* indication and  
21 disable the timer.

## 22 **6.5.6 Message and block formats**

23 In the interpretation of these messages, the symbol 'n' is used to denote the value of a bit field. For  
24 example, the field CPLength is assigned two bits, and the parameter CPLength takes values  
25  $N_{\text{FFT}} \cdot (1+n)/16$ , where n is the 2 bit field that takes the value 0, 1, 2 or 3.



### 6.5.6.1 SystemInfo block

The SystemInfo block shall be transmitted directly by the Control Channel MAC Protocol over the pBCH0 channel, and shall not pass through other sublayers. The SystemInfo block shall have the following format.

Field	Length (bits)
MaximumRevision	4
MinimumRevision	4
CarrierID	2
NumCarriers	2
SystemTimeLSB	12
CPLength	2
NumGuardSubcarriers	3
BlockHoppingEnabled	1
N_FLBurst	2
N_RLBurst	2
FLReservedInterlaces	4
NumFLReservedSubbands	4

- 6 MinimumRevision This field shall be set to the minimum revision number that the sector can  
7 support.
- 8 MaximumRevision This field shall be set to the maximum revision number that the sector can  
9 support.
- 10 CarrierID This field shall be set to the CarrierID of the carrier this block is transmitted  
11 on.
- 12 NumCarriers This field shall determine the number of carriers available at this sector. This  
13 parameter shall take the value (n+1).
- 14 SystemTimeLSB This field shall be set to the twelve lower bits of the superframe number at  
15 the time the SystemInfo block starts transmission.
- 16 CPLength This field shall determine the cyclic prefix length in units of chips. This  
17 parameter shall take the value  $N_{\text{FFT}} \cdot (1+n)/16$ , where n is equal to the 2 bit  
18 field that takes the value 0, 1, 2 or 3.
- 19 NumGuardSubcarriers This parameter shall take the value  $N_{\text{GUARD,PR}} - 32 \cdot n$ . Here  $N_{\text{GUARD,PR}}$  is a  
20 numeric constant of the Physical Layer Protocol.
- 21 BlockHoppingEnabled This field shall be set to '1' if block hopping is enabled. This field shall be  
22 set to '0' if symbol rate hopping is enabled.

- 1 N\_FLBurst This field shall determine the number of forward link PHY Frames that  
 2 comprise a forward link burst in TDD mode. This parameter shall take the  
 3 value (n+1). This field shall be ignored in FDD mode.
- 4 N\_RLBurst This field shall determine the number of reverse link PHY Frames that  
 5 comprise a reverse link burst in TDD mode. This parameter shall take the  
 6 value (n+1). This field shall be ignored in FDD mode.
- 7 FLReservedInterlaces This field shall determine what interlaces contain reserved bandwidth on the  
 8 forward link.

9 **Table 44 Interpretation of FLReservedInterlaces**

Value	Interpretation: Reserved FL bandwidth on the following interlaces
0000	None
0001	0
0010	0, 1
0011	0, 1, 2
0100	0, 1, 2, 3
0101	0, 1, 2, 3, 4
0110	0, 1, 2, 3, 4, 5
0111	0, 1, 2, 3, 4, 5, 6
1000	0, 1, 2, 3, 4, 5, 6, 7
1001	0, 1, 2, 3, 4, 5, 6, 7, 8
1010	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
1011	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
1100	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11
1101	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12
1110	0, 3
1111	0, 6

- 10 NumFLReservedSubbands  
 11 The interpretation of this field is used by the Physical Layer to govern FL  
 12 PHY Frame Modulation.

### 6.5.6.2 QuickChannelInfo block

The QuickChannelInfo block shall be transmitted directly by the Control Channel MAC Protocol over the pBCH1 channel, and shall not pass through other sublayers. The QuickChannelInfo block shall have the following format:

Field	Length (bits)
QuickChannelInfoValidity	3
FLFirstRestrictedSetSubband	$2 + \log_2 (N_{\text{CARRIER\_SIZE}}/512)$
FLNumRestrictedSetSubbands	2
FLChannelTreeIndex	4
FLSectorHopSeed	4
FLIntraCellCommonHopping	1
FLDPISectorOffset	2
FLDPISectorScramble	1
FLNumSDMADimensions	2
FLNumSubbands	1
FLDiversityHoppingMode	1
NumPilots	1
EffectiveNumAntennas	3
NumCommonPilotTransmitAntennas	1
EnableCommonPilotStaggering	1
EnableAuxPilotStaggering	1
SSCHNumHopports	3
SSCHNumBlocks	3
SSCHModulationSymbolsPerBlock	2

#### QuickChannelInfoValidity

If this field takes the value  $m$  and the current superframe number is  $n$ , the access network should keep all fields of the QuickChannelInfo block (except the QuickChannelInfoValidity field) unchanged from superframes  $m$  till superframe  $2 \cdot 4^m \lceil n / (2 \cdot 4^m) \rceil$ .

#### FLFirstRestrictedSetSubband

This field shall be set to the index of the first restricted subband on the forward link.

#### FLNumRestrictedSetSubbands

This field shall be set to the number of restricted subbands on the forward link. This field shall be set to 0 if no subbands are restricted. Otherwise, subbands FLFirstRestrictedSetSubband through FLFistRestrictedSetSubband+FLNumRestrictedSetSubbands-1 shall be considered to be restricted subbands, with possible rollover at subband zero.

1	FLChannelTreeIndex	This field shall be used by the Lower MAC Sublayer.
2	FLSectorHopSeed	This field shall be used by the PHY Layer to determine the hopping pattern.
3	FLIntraCellCommonHopping	
4		This field shall be used by the PHY Layer to determine the hopping pattern.
5	FLDPISectorOffset	This field shall be set to the relative offset of pilots as defined in the
6		F-DPICH section in the Physical Layer.
7	FLDPISectorScramble	This field shall determine the scrambling of pilots as defined in the sector
8		and cell specific scrambling sections in the Physical Layer.
9	FLNumSDMADimensions	
10		This field shall determine the number of spatial dimensions on the forward
11		link. This parameter shall take the value n+1.
12	FLNumSubbands	This field shall determine the number of subbands on the forward link. If
13		n=0, this parameter shall be set to $N_{\text{CARRIER\_SIZE}}/128$ and if n=1, this
14		parameter shall be set to $N_{\text{CARRIER\_SIZE}}/256$ .
15	FLDiversityHoppingMode	
16		This field shall be used by the Physical Layer to determine the hop pattern
17		for the sector. This field shall be set to '1' if DiversityHoppingMode is On,
18		and to '0' if DiversityHoppingMode is Off.
19	NumPilots	This field shall determine the nominal number of pilots in F-CPICH as being
20		$N_{\text{CARRIER\_SIZE}}/16$ or $N_{\text{CARRIER\_SIZE}}/8$ , depending on whether the field is set to
21		'0' or '1', respectively.
22	EffectiveNumAntennas	
23		This field shall determine the effective number of antennas, and this
24		parameter shall take the value n+1. The access network shall set
25		EffectiveNumAntennas to four or below when the BlockHoppingEnabled
26		field of the SystemInfo block is set to '0'.
27	NumCommonPilotTransmitAntennas	
28		This field shall determine the number of common pilot transmit antennas,
29		and this parameter shall take the value n+1.
30	EnableCommonPilotStaggering	
31		This field is set to '1' if common pilot staggering is enabled. This field is set
32		to '0' if common pilot staggering is not enabled.
33	EnableAuxPilotStaggering	
34		This field is set to '1' if auxiliary pilot staggering is enabled. This field is set
35		to '0' if auxiliary pilot staggering is not enabled.

1 SSCHNumHopports This field shall determine the number of hop-ports allocated to F-SSCH. This  
2 field shall be interpreted as follows:

3 **Table 45 Interpretation of SSCHNumHopports**

Value	Interpretation when MultiCarrierOn	Interpretation when MultiCarrierOff
000	48	$48 \times N_{\text{CARRIER\_SIZE}}/512$
001	64	$64 \times N_{\text{CARRIER\_SIZE}}/512$
010	80	$80 \times N_{\text{CARRIER\_SIZE}}/512$
011	96	$96 \times N_{\text{CARRIER\_SIZE}}/512$
100	128	$128 \times N_{\text{CARRIER\_SIZE}}/512$
101	160	$160 \times N_{\text{CARRIER\_SIZE}}/512$
110	208	$208 \times N_{\text{CARRIER\_SIZE}}/512$
111	256	$256 \times N_{\text{CARRIER\_SIZE}}/512$

4 SSCHNumBlocks This field shall determine the number of blocks carried by the F-SSCH. This  
5 parameter shall take the value  $2^{*(n+1)}$ .

6 SSCHModulationSymbolsPerBlock

7 This field shall determine the number of modulation symbols for each block  
8 carried by the F-SSCH.

9 **Table 46 Interpretation of SSCHModulationSymbolsPerBlock**

Value	Interpretation
00	45
01	60
10	90
11	180

### 11 6.5.6.3 ExtendedChannelInfo

12 The ExtendedChannelInfo message provides the configuration for the Physical Layer and Lower  
13 MAC Sublayer operation. The message consists of several groups, and a group consists of fields, as  
14 defined in the following:

Field	Length (bits)
MessageID	8
ValidityPeriod	16
SectorInformation Group	See 6.5.6.3.1
PowerControl Group	See 6.5.6.3.2
AccessParameters Group	See 6.5.6.3.3

- 1 MessageID This field shall be set to 0x00.
- 2 SectorInformation Group  
3 This field is defined in 6.5.6.3.1.
- 4 ValidityPeriod This field determines the time till when the parameters in the  
5 ExtendedChannelInfo message are valid . This parameter shall take the value  
6  $(1+n) \cdot N_{\text{OMPExtendedChannelInfo}}$  superframes.
- 7 PowerControl Group This field is defined in 6.5.6.3.2.
- 8 AccessParameters Group  
9 This field is defined in 6.5.6.3.3.

10

<b>Channels</b>	FTC
-----------------	-----

<b>SLP</b>	Best Effort
------------	-------------

<b>Addressing</b>	Broadcast    Unicast
-------------------	----------------------

<b>Security</b>	Optional
-----------------	----------

11

### 6.5.6.3.1 SectorInformation group

The SectorInformation group shall consist of the following fields.

**Table 47 SectorInformation group**

Field	Length (bits)
PilotPN	12
CarrierID	2
SystemTime	34
SectorParametersSignature	16
RLChannelTreeIndex	4
RLSectorHopSeed	4
RLIntraCellCommonHopping	1
BFCHBeamCodeBookIndex	4
RLDPISectorOffset	2
RLDPISectorScramble	1
RLNumSDMADimensions	2
RLRestrictedSetBitmap	16
RLNumSubbands	1
RLDiversityHoppingMode	1
NumRLControlSubbands	3
RACKBandwidthFactor	2
HalfDuplexModeSupported	1
ReverseLinkSilenceDuration	4
ReverseLinkSilencePeriod	4
TransmitPower	6
CommonPilotPower	4
AuxPilotPower	4
PreamblePilotPower	4

**PilotPN** This field shall be set to the PilotPN of the sector this message refers to. The sector this message refers to may be different from the sector transmitting this message.

**CarrierID** This field shall be set to the CarrierID field transmitted on the SystemInfo block on this carrier.

**SystemTime** This field shall be set to the time at the sector this message refers to at the beginning of the superframe in which the ExtendedChannelInfo block started transmission.

1	<b>SectorParametersSignature</b>	
2		The access network shall set this field to the SectorParametersSignature of
3		the next SectorParameters message to be transmitted by the access network.
4	<b>RLChannelTreeIndex</b>	This field shall be used by the Lower MAC Sublayer .
5	<b>RLSectorHopSeed</b>	This field shall be used by the PHY Layer to determine the hopping pattern.
6	<b>RLIntraCellCommonHopping</b>	
7		This field shall be used by the PHY Layer to determine the hopping pattern.
8	<b>BFCHBeamCodeBookIndex</b>	
9		This field shall refer to the code book index, the code book comprising of
10		transmit weights for SDMA and precoding.
11	<b>RLDPISectorOffset</b>	This field shall be set to the relative offset of pilots as defined in the
12		R-DPICH section in the Physical Layer.
13	<b>RLDPISectorScramble</b>	This field shall be set to the scrambling of pilots as defined in the sector and
14		cell specific scrambling sections of the Physical Layer.
15	<b>RLNumSDMADimensions</b>	
16		This field shall determine the number of spatial dimensions on the reverse
17		link. This parameter shall take the value n+1.
18	<b>RLRestrictedSetBitmap</b>	
19		Bit position $j$ in this bit field shall be set to 1 if subband $j$ is restricted on the
20		reverse link.
21	<b>RLNumSubbands</b>	This field shall determine the number of subbands on the reverse link. If n=0,
22		this parameter shall take the value $N_{\text{CARRIER\_SIZE}}/128$ and if n=1, this
23		parameter shall take the value $N_{\text{CARRIER\_SIZE}}/256$ .
24	<b>RLDiversityHoppingMode</b>	
25		This field shall be used by the Physical Layer to determine the hop pattern
26		for the sector. This field shall be set to '1' if DiversityHoppingMode is on,
27		and to '0' if DiversityHoppingMode is off.
28	<b>NumRLControlSubbands</b>	
29		This field shall determine the number of control subbands on the reverse link
30		and this parameter shall take the value (n+1).
31	<b>RACKBandwidthFactor</b>	
32		This field shall determine the bandwidth reduction on the R-ACKCH. This
33		parameter shall take the value n+1.



1	<b>HalfDuplexModeSupported</b>	
2		This field shall be set to '1' if the access network supports half duplex
3		terminals, and shall be set to '0' otherwise. If half-duplex terminals are
4		supported, the access network should assign MAC IDs and channel
5		assignments in a manner that enables half-duplex terminal operation. A half-
6		duplex access terminal is not required to monitor forward link transmissions
7		on a PHY Frame where it is scheduled to make a reverse link transmission.
8	<b>ReverseLinkSilenceDuration</b>	
9		The access network shall set this field to specify the duration of the Reverse
10		Link Silence Interval. This parameter shall take the value $2^n$ PHY Frames. In
11		a region with asynchronous sectors, the access network should set this field
12		to a value larger than the timing offset between sectors.
13	<b>ReverseLinkSilencePeriod</b>	
14		The access network shall set this field to specify the periodicity of
15		occurrence the Reverse Link Silence Interval. This parameter shall take the
16		value
17		$\text{ReverseLinkSilencePeriod} = (1+n)*144000$
18		The Reverse Link Silence Interval is defined as the time interval of duration
19		ReverseLinkSilenceDuration RL PHY Frames that starts at superframe index
20		m that satisfies the following equation:
21		$m \bmod (\text{ReverseLinkSilencePeriod}) = 0$
22	<b>TransmitPower</b>	This field shall be set to the transmit power of the sector in units of dBm.
23	<b>CommonPilotPower</b>	The field shall be determine the power spectral density of the F-CPICH
24		during the FL PHY frame relative to the F-ACQCH. This parameter shall
25		take the value $(-4 + n*0.5)$ dB.
26	<b>AuxPilotPower</b>	The field shall determine the power spectral density of the F-AuxPICH
27		relative to the F-ACQCH. This parameter shall take the value $(-4 + n*0.5)$
28		dB.
29	<b>PreamblePilotPower</b>	The field shall determine the power spectral density of the F-CPICH during
30		the superframe preamble relative to the F-ACQCH. This parameter shall take
31		the value $(-4 + n*0.5)$ dB.

### 6.5.6.3.2 PowerControl group

Field	Length (bits)
MACIDRange	3
FLPCReportInterval	4
RLCtrlPCMode	1
FastOSIEnabled	1
CtrlAccessOffset	3
BFCHPowerOffset	4
SFCHPowerOffset	4
PICHPowerOffset	4
REQChannelGain0	4
REQChannelGain1	4
REQChannelGain2	4
REQChannelGain3	4
ErasureGain0	4
ErasureGain1	4
ErasureGain2	4
ErasureGain3	4

#### MACIDRange

This field shall be set to indicate the range of assigned MACID values in the sector. For example, a MACIDRange of 63 indicates that the sector has not assigned MACID values 64 and above. The field shall be interpreted as follows.

**Table 48 Interpretation of MACIDRange**

Value	Interpretation
000	63
001	127
010	255
011	511
100	1023
101	2047
110 to 111	Reserved

#### FLPCReportInterval

This field shall determine the periodicity at which power control commands are sent to the access terminal. This parameter shall take the value (n+1) PHY Frames.

#### RLCtrlPCMode

This field shall determine the closed loop power control mode of the sector, with values '0' corresponding to 'ErasureBased' and '1' corresponding to 'UpDown'.

1	FastOSIEnabled	This field shall be set to '1' if the F-SSCH transmitted by this sector contains a Fast OSI Segment. This field shall be set to '0' if the F-SSCH transmitted by this sector does not contain a Fast OSI Segment..
2		
3		
4	CtrlAccessOffset	This field determines the initial gain of the R-CQICH over the R-ACH. The value of this parameter shall be in units of dB in 2's complement notation.
5		
6	BFCHPowerOffset	This field shall determine the power offset of the R-BFCH relative to the R-CQICH. This parameter shall be in units of dB in 2's complement notation.
7		
8	SFCHPowerOffset	This field shall determine the power offset of the R-SFCH relative to the R-CQICH. This parameter shall be in units of dB in 2's complement notation.
9		
10	PICHPowerOffset	This field is determines the gain of the R-PICH over the R-CQICH. This parameter shall be in units of dB in 2's complement notation.
11		
12	REQChannelGainj	This field determines the gain of the R-REQCH over the R-CQICH. This parameter shall be in units of dB in 2's complement notation.
13		
14	ErasureGainj	This field determines the transmit power of erasure sequences for different assignment sizes, and this parameter shall take the value n-4 dB.
15		

### 6.5.6.3.3 AccessParameters group

Field	Length (bits)
AccessCycleDuration	2
AccessSequencePartition	5
MaxProbesPerSequence	4
ProbeRampUpStepSize	4
RDCHInitialPacketFormat	6
PilotThreshold1	3
PilotThreshold2	3
OpenLoopAdjust	8

$N_{ACMPCClass}$  values of the following field

AccessRetryPersistence	3
------------------------	---

18	AccessCycleDuration	This field shall be determine the duration of the access cycle in units of Control Segment Periods (as defined by the Physical Layer). The AccessCycleDuration parameter shall be set according to the value of the field as follows.
19		
20		
21		

**Table 49 Interpretation of AccessCycleDuration**

Value	Interpretation in units of Control Segment Periods
00	1
01	2
10	3
11	4

- 2 **AccessSequencePartition**  
3 This field shall indicate the partition of the access sequence space to allow  
4 the access terminal to signal pilot power and buffer status information with  
5 the access sequence. The interpretation of this field is in the Access Channel  
6 MAC Protocol.
- 7 **MaxProbesPerSequence**  
8 This field shall determine the maximum number of probe sequences that can  
9 be part of one access sequence. This parameter shall take the value  $n+1$ .
- 10 **ProbeRampUpStepSize** This field shall determine the power ramp up used for probes within a probe  
11 sequence. This parameter shall take the value  $0.5*(1+n)$  dB.
- 12 **RDCHInitialPacketFormat**  
13 This field shall be set to the packet format that is used on the first  
14 transmission the access terminal makes on the R-DCH after getting an access  
15 grant.
- 16 **PilotThreshold1** This field shall determine PilotThreshold1 used by the Access Channel MAC  
17 Protocol. This parameter shall take the value  $-2n$  dB.
- 18 **PilotThreshold2** This field shall determine PilotThreshold2 used by the Access Channel MAC  
19 Protocol. This parameter shall take the value  $-2n$  dB.
- 20 **AccessRetryPersistence**  
21 This field shall determine the persistence probability for determining access  
22 sequence backoff. If this field is set to  $n$ , the access terminal shall use  $2^{-n}$  as  
23 the retry persistence.
- 24 **OpenLoopAdjust** This field shall determine the nominal power to be used by access terminal in  
25 the open loop power estimate. The value of this field shall be  $70+n$  dB.

### 6.5.6.4 SectorParameters

The SectorParameters message is used to convey sector specific information to the access terminals.

Field	Length (bits)
MessageID	8
CountryCode	12
SectorID	128
SubnetMask	8
SectorParametersSignature	16
ChannelBand	ChannelBandRecord Type Dependent
Latitude	22
Longitude	23
RegistrationRadius	11
LeapSeconds	8
LocalTimeOffset	11
RegistrationZoneCodeIncluded	1
RegistrationZoneCode	12
RegistrationZoneMaxAge	4
GloballySynchronous	1
SynchronousWithNextPilot	1
ChannelBandCount	5
ChannelBandCount occurrences of the following record{	
ChannelBand	ChannelBandRecord Type Dependent
NeighborCount	5
NeighborCount occurrences of the following record{	
NeighborPilotPN	12
NeighborCarrierID	2
TransmitPower	6
GloballySynchronous	1
SynchronousWithNextPilot	1
}}	
NumOtherTechnologies	4
NumOtherTechnologies occurrences of the following fields	
TechnologyType	8
TechnologyNeighborListLength	8
TechnologyNeighborList	TechnologyNeighbor ListLength x 8
Reserved	Variable

1	MessageID	The access network shall set this field to 0x01.
2	CountryCode	The access network shall set this field to the three-digit BCD (binary coded decimal) encoded representation of the Mobile Country Code (as specified in [6]) associated with this sector.
3		
4		
5	SectorID	Sector Address Identifier. The access network shall set the value of the SectorID according to the rules specified in 10.11. The access terminal shall not assume anything about the format of the SectorID. The SectorID shall uniquely identify a sector.
6		
7		
8		
9	SubnetMask	Sector Subnet identifier. The access network shall set this field to the number of consecutive 1's in the subnet mask of the subnet to which this sector belongs.
10		
11		
12	SectorParametersSignature	SectorParameters message signature. The access network shall change this field if the contents of the SectorParameters message changes.
13		
14		
15	ChannelBand	ChannelBand record specification for each channel. See 10.1 for the ChannelBand record format.
16		
17	Latitude	The latitude of the sector. The access network shall set this field to this sector's latitude in units of 0.25 second, expressed as a two's complement signed number with positive numbers signifying North latitudes. The access network shall set this field to a value in the range -1296000 to 1296000 inclusive (corresponding to a range of -90° to +90°).
18		
19		
20		
21		
22	Longitude	The longitude of the sector. The access network shall set this field to this sector's longitude in units of 0.25 second, expressed as a two's complement signed number with positive numbers signifying East longitude. The access network shall set this field to a value in the range -2592000 to 2592000 inclusive (corresponding to a range of -180° to +180°).
23		
24		
25		
26		
27	RegistrationRadius	If access terminals are to perform distance based registration, the access network shall set this field to the non-zero "distance" beyond which the access terminal is to send a new PilotReport message (see Default Active Set Management Protocol, 6.6). If access terminals are not to perform distance based registration, the access network shall set this field to 0.
28		
29		
30		
31		
32	LeapSeconds	The number of leap seconds that have occurred since the start of system time.
33	LocalTimeOffset	The access network shall set this field to the offset of the local time from System Time. This value will be in units of minutes, expressed as a two's complement signed number.
34		
35		
36	RegistrationZoneIncluded	The access network shall set this field to '1' if the RegistrationZoneCode and RegistrationZoneMaxAge are included in the message.
37		
38		

1	RegistrationZoneCode	The access network shall include this field if RegistrationZoneIncluded is set to '1'.
2		
3	RegistrationZoneMaxAge	The access network shall include this field if RegistrationZoneIncluded is set to '1'.
4		
5		
6	SynchronousSystem	The access network shall set this field to '1' if all sectors in the deployment are synchronous. The access network shall set this field to '0' otherwise.
7		
8	ChannelBandCount	The access network shall set this field to the number of neighbor channels available to the access terminal on this sector.
9		
10	ChannelBand	ChannelBand record specification for each channel. See 10.1 for the ChannelBand record format. If the optional parameters of this ChannelBand record are different from the parameters for the sector transmitting this message, the access network shall include the optional parameters in the ChannelBand record.
11		
12		
13		
14		
15	NeighborCount	The access network shall set this field to the number of records specifying neighboring sectors information included in this message for this channel.
16		
17	NeighborPilotPN	The access network shall set this field to the PilotPN of a neighboring sector that the access terminal should add to its Neighbor Set.
18		
19	NeighborCarrierID	The access network shall set this field to the CarrierID of the neighboring pilot.
20		
21	TransmitPower	This field shall be set to the transmit power of the sector in units of dBm.
22	GloballySynchronous	The access network shall set this field according to the following rule: If any two sectors have the GloballySynchronous field set to '1', then the two sectors are synchronous with each other.
23		
24		
25	SynchronousWithNextPilot	The access network shall set this field to '1' if this sector is synchronous with the next sector listed in the message. The access network shall set this field to '0' if this sector is the last sector listed in the message, or if this sector is not synchronous with the next sector listed in the message. Rules for determining if two sectors are synchronous are given in the synchronization and timing section of the physical layer, 9.4.2.
26		
27		
28		
29		
30		
31		
32	NumOtherTechnologies	This field shall be set to the number of other technologies included in the message. Other technology neighbors are included in this message to assist the access terminal in inter-technology handoff.
33		
34		
35		
36	TechnologyType	This field shall be set to the type of technology, and shall be interpreted as defined in Table 107.
37		

- 1 TechnologyNeighborListLength  
 2 This field shall be set the length, in bytes, of the neighbor list information for  
 3 the other technology.
- 4 TechnologyNeighborList  
 5 This field shall be set to the neighbor list information for the other  
 6 technology. The interpretation of this field is beyond the scope of this  
 7 specification.
- 8 Reserved  
 9 The number of bits in this field is equal to the number needed to make the  
 10 message length an integer number of octets. The access network shall set this  
 11 field to zero. The access terminal shall ignore this field.

<b>Channels</b>	FTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Broadcast Unicast	<b>Security</b>	Optional

12

### 13 6.5.6.5 EncapsulatedQuickChannelInfo

- 14 This message may be used to deliver the QuickChannelInfo block over the Forward Traffic Channel  
 15 MAC. This message may also be used to deliver the QuickChannelInfo block of one sector over the  
 16 Forward Traffic Channel of another sector.

17

Field	Length (bits)
MessageID	8
SuperframeNumber	34
PilotPN	12
SectorIDIncluded	1
SectorID	0 or 128
CarrierID	2
QuickChannelInfoLength	8
QuickChannelInfoBlock	Variable
Reserved	Variable

- 18 MessageID  
 19 This field shall be set to 0x02.
- 20 SuperframeNumber  
 21 This field shall be set to the superframe number at the sector this message  
 22 refers to, when that sector transmitted the QuickChannelBlock included in  
 23 this message over the Control Channel MAC.
- 24 PilotPN  
 This field shall be set to the PilotPN of the sector this message refers to. The  
 sector this message refers to may be different from the sector transmitting  
 this message.



1	SectorIDIncluded	This field shall be set to '1' if the SectorID of the sector this message refers to is included in the message. The access network shall set this field to '1' if the PilotPN is not sufficient to uniquely identify the sector this message refers to.
2		
3		
4		
5	SectorID	This field shall be included if SectorIDIncluded is equal to '1', and shall be set to the SectorID of the sector this message refers to.
6		
7	CarrierID	This field shall be set to the CarrierID field transmitted on the SystemInfo block on this carrier.
8		
9	QuickChannelInfoLength	This field shall be set to the length, in bits, of the following QuickChannelInfoBlock.
10		
11		
12	QuickChannelInfoBlock	This field shall be set to the QuickChannelBlock.
13		
14	Reserved	The number of bits in this field is equal to the number needed to make the message length an integer number of octets. The access network shall set this field to zero. The access terminal shall ignore this field.
15		
16		
17		

<b>Channels</b>	FTC		<b>SLP</b>	Best Effort
<b>Addressing</b>	Broadcast	Unicast	<b>Security</b>	Optional

18

## 19 6.5.7 Interface to other protocols

### 20 6.5.7.1 Commands

21 This protocol sends no commands.

### 22 6.5.7.2 Indications

23 This protocol registers to receive the following indications:

- 24 ■ *ActiveSetManagement.IdleHO*
- 25 ■ *ConnectedState.ConnectionClosed*

### 26 6.5.8 Configuration attributes

27 No configuration attributes are defined for this protocol.

## 6.5.9 Protocol numeric constants

Constant	Meaning	Value
NOMPType	Type field for this protocol	Table 9
NOMPDefault	Subtype field for this protocol	0x0000
TOMPQCISupervision	QuickChannelInfo supervision timer	0.5 s
TOMPECISupervision	ExtendedChannelInfo supervision timer	1 s
TOMPSPSupervision	SectorParameters supervision timer	4 s
NOMPEExtendedChannelInfo	The recommended time between two consecutive ExtendedChannelInfo message transmissions	16 superframes
NOMPSectorParameters	The recommended time between two consecutive SectorParameters message transmissions	64 superframes
NOMPMinZoneSignatureListSize	Minimum number of entries supported by the access terminal in the RegistrationZoneCodeList	8

## 6.5.10 Session state information

The Session State Information record (see 10.10) consists of parameter records.

The parameter records for this protocol consist of the configuration attributes of this protocol.

## 6.6 Default Active Set Management Protocol

### 6.6.1 Overview

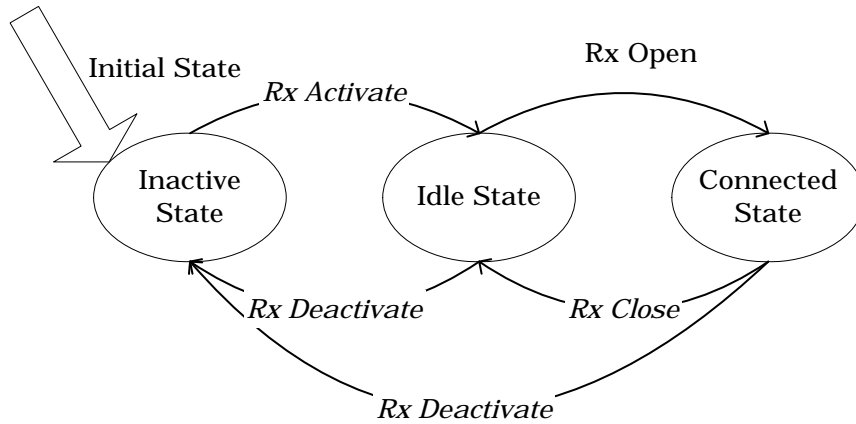
The Default Active Set Management Protocol provides the procedures and messages used by the access terminal and the access network to keep track of the access terminal's approximate location and to maintain the radio link as the access terminal moves between the coverage areas of different sectors.

This protocol can be in one of three states:

- *Inactive State*: In this state, the protocol waits for an *Activate* command.
- *Idle State*: This state corresponds to the Air Link Management Protocol Idle State. In this state, the access terminal autonomously maintains the Active Set. PilotReport messages from the access terminal to the access network are generated based on the distance between the access terminal's current serving sector and the serving sector at the time the access terminal last sent an update. The generation of such PilotReport messages causes transition from the Idle State to the Connected State.
- *Connected State*: In this state, the access network dictates the access terminal's Active Set. PilotReport messages from the access terminal to the access network are based on changing radio link conditions.

Transitions between states are driven by commands received from other Lower MAC Control sublayer protocols and the transmission and reception of the ActiveSetAssignment message.

1 The protocol states, messages and commands causing the transition between the states are shown in  
 2 Figure 55.



3  
 4 **Figure 55 Default Active Set Management Protocol state diagram**

5 This protocol uses parameters that are provided as public data by the Overhead Messages Protocol,  
 6 configured attributes, or protocol constants.

7 Table 50 lists all of the protocol parameters obtained from the public data of the Overhead Messages  
 8 Protocol.

9 **Table 50 Active Set Management Protocol parameters that are public data of the**  
 10 **Overhead Messages Protocol**

Parameter	Comment
Latitude	Latitude of sector in units of 0.25 second.
Longitude	Longitude of sector in units of 0.25 second.
RegistrationRadius	Distance between the serving sector and the sector in which location was last reported that triggers a new report. If this field is set to zero, then distance triggered reporting is disabled.
NumNeighbors	Number of neighbors specified in the message.
NeighborPN	PilotPN of each neighbor.
NeighborChannelBandIncluded	Set to '1' if a ChannelBand record is included for the neighbor.
NeighborChannelBand	Neighbor ChannelBand record specifying network type and frequency.

11

## 6.6.2 Primitives

### 6.6.2.1 Commands

This protocol defines the following commands:

- *Activate*
- *Deactivate*
- *Open*
- *Close*
- *SendPilotReport*

### 6.6.2.2 Return indications

This protocol returns the following indications:

- *ConnectionLost* (access network only)
- *NetworkLost*
- *IdleHO*
- *ActiveSetUpdated*
- *AssignmentRejected*

## 6.6.3 Public data

### 6.6.3.1 Static public data

This protocol defines the following static public data:

- Active Set
- Current ActiveSetAssignment message
- PilotIncrement specified in the PilotSearch Configuration Attribute
- A list of measured PilotStrengths, and RxPowers indexed by PilotPN
- RegistrationRadiusFlag

### 6.6.3.2 Dynamic public data

- Subtype for this protocol

## 6.6.4 Protocol initialization and swap procedures

### 6.6.4.1 Protocol initialization

Upon initialization at the access terminal and the access network

- The value for each attribute for this protocol instance shall be set to the default value for that attribute.
- The protocol shall enter the inactive state.

## 1 6.6.4.2 Protocol swap

- 2 ■ The protocol shall enter the Inactive State.

## 3 6.6.5 Procedures

### 4 6.6.5.1 Command processing

#### 5 6.6.5.1.1 Activate

6 If the protocol receives an *Activate* command in the Inactive State, the access network shall transition  
7 to the Idle State.

8 If the protocol at the access terminal receives an *Activate* command in the Inactive State, the protocol  
9 shall transition to the Idle State, and generate a PilotReport message.

10 If this command is received in any other state, it shall be ignored.

#### 11 6.6.5.1.2 Deactivate

12 If the protocol receives a *Deactivate* command in the Inactive State, it shall be ignored.

13 If the protocol receives this command in any other state, the access terminal and the access network  
14 shall transition to the Inactive State.

#### 15 6.6.5.1.3 Open

16 If the protocol receives an *Open* command in the Idle State,

- 17 ■ The access terminal shall transition to the Connected State.
- 18 ■ The access network shall transition to the Connected State.

19 If this command is received in any other state, it shall be ignored.

#### 20 6.6.5.1.4 Close

21 If the protocol receives a *Close* command in the Connected State, the access terminal and the access  
22 network shall transition to the Idle State.

23 If this command is received in any other state, it shall be ignored.

### 24 6.6.5.2 Processing the RegistrationRadiusUpdated indications

25 The RegistrationRadiusFlag determines the Idle State registration procedure at the access terminal.  
26 When the flag value is 1 registration is performed when the access terminal travels a distance more  
27 than RegistrationRadius distance. When flag value is 0 is registration is performed when the access  
28 terminal moves to a sector with a different latitude and longitude. The value of the flag at the access  
29 terminal is set based on ConnectionOpenResponse and ConnectionClose messages received by the  
30 access terminal.

1 Upon receiving a *IdleState.RegistrationRadiusUpdated* indication or a  
 2 *ConnectedState.RegistrationRadiusUpdated* indication, the Active Set Management Protocol at the  
 3 access terminal shall set the RegistrationRadiusFlag public data to the RegistrationRadiusFlag value  
 4 provided with the indication.

### 5 **6.6.5.3 Pilots and pilot sets**

6 The access terminal estimates the strength of the Forward Channel transmitted by each sector in its  
 7 neighborhood. This estimate is based on measuring the strength of the Forward Pilot Channel  
 8 (specified by the pilot's PilotPN and the pilot's ChannelBand), henceforth referred to as the pilot.

9 When this protocol is in the Connected State, the access terminal uses pilot strengths to decide when  
 10 to generate PilotReport messages.

11 When this protocol is in the Idle State, the access terminal uses pilot strengths to decide which  
 12 sector's Control Channel it monitors.

13 The following pilot sets are defined to support the Active Set Management process:<sup>30</sup>

- 14 ■ *Active Set*: The set of pilots associated with the sectors currently serving the access  
 15 terminal. When a connection is open, a sector is considered to be serving an access  
 16 terminal when there is a MAC ID assigned to the access terminal in that sector. When a  
 17 connection is not open, a sector is considered to be serving the access terminal when the  
 18 access terminal is monitoring that sector's control channel for page reception and  
 19 distance based registration. Parameters for Active Set member sectors in connected state  
 20 are sent as part of the ActiveSetAssignment message, and placed in the public data of the  
 21 Active Set Management Protocol.
- 22 ■ *Candidate Set*: The pilots that are not in the Active Set, but are received by the access  
 23 terminal with sufficient strength to indicate that the sectors transmitting them are good  
 24 candidates for inclusion in the Active Set.
- 25 ■ *Neighbor Set*: The set of pilots that are not in either one of the two previous sets, but are  
 26 likely candidates for inclusion in the Active Set.
- 27 ■ *Remaining Set*: The set of all possible pilots on the current channel assignment, excluding  
 28 the pilots that are in any of the three previous sets.

29 At any given instant a pilot in the current ChannelBand is a member of exactly one set.

30 The access terminal maintains all four sets. The access network maintains only the Active Set.

31 The access terminal complies with the following rules when searching for pilots, estimating the  
 32 strength of a given pilot, and moving pilots between sets.

---

<sup>30</sup> In this context, a pilot identifies a sector.

### 6.6.5.3.1 Pilot search

The access terminal shall continually search for pilots in the Connected State and whenever it is monitoring any channel in the Idle State. The access terminal shall search for pilots in all pilot sets.

The access terminal should use the same search priority for pilots in the Active Set and Candidate Set. In descending order of search rate, the access terminal shall search, most often, the pilots in the Active Set and Candidate Set, then shall search the pilots in the Neighbor Set, and lastly shall search the pilots in the Remaining Set.

### 6.6.5.3.2 Pilot strength measurement

The access terminal shall measure the strength of every pilot it searches. The strength estimate formed by the access terminal shall be computed as the ratio of received pilot energy per chip,  $E_c$ , to total received spectral density,  $I_0$  (signal and noise). The access terminal shall place the measured ratios in its public data.

In addition, the access terminal shall also measure the received power (RxPower in dBm) from each acquisition pilot at the antenna input and place the result in the public data. If the access terminal has more than one receive antenna, the power shall be averaged across all antennas. If a pilot is not being measured, the RxPower shall be assumed to be negative infinity. The access terminal should update the measured power every superframe.

### 6.6.5.3.3 Pilot drop timer maintenance

For each pilot, the access terminal shall maintain a pilot drop timer.

If DynamicThresholds is equal to '0', the access terminal shall perform the following:

- The access terminal shall start a pilot drop timer for each pilot in the Candidate Set or the Active Set whenever the strength becomes less than the value specified by PilotDrop. The access terminal shall consider the timer to be expired after the time specified by PilotDropTimer.
- The access terminal shall reset and disable the timer whenever the strength of the pilot becomes greater than the value specified by PilotDrop.

If DynamicThresholds is equal to '1', the access terminal shall perform the following:

- The access terminal shall start a pilot drop timer for each pilot in the Candidate Set whenever the strength of the pilot becomes less than the value specified by PilotDrop. The access terminal shall consider the timer value to be expired after the time specified by PilotDropTimer. The access terminal shall reset and disable the timer if the strength of the pilot becomes greater than the value specified by PilotDrop.

- 1           ■ For each pilot in the Active Set, the access terminal shall sort pilots in the Active Set in  
2 order of increasing strengths, i.e.,  $PS_1 < PS_2 < PS_3 < \dots < PS_{N_A}$ , where  $N_A$  is the number  
3 of the pilots in the Active Set. The access terminal shall start the timer whenever the  
4 strength  $PS_i$  satisfies the following inequality:

$$5 \quad 10 \times \log_{10} PS_i < \max \left( \frac{\text{SoftSlope}}{8} \times 10 \times \log_{10} PS_{N_A} + \frac{\text{DropIntercept}}{2}, -\frac{\text{PilotDrop}}{2} \right)$$

6            $i = 1, 2, \dots, N_A - 1$

6           The access terminal shall reset and disable the timer whenever the above inequality is not  
7 satisfied for the corresponding pilot.

8 Sections 6.6.5.3.5 and 6.6.5.7.3 specify the actions the access terminal takes when the pilot drop timer  
9 expires.

### 10 **6.6.5.3.4 Active set management**

11 The access terminal shall support a maximum Active Set size of  $N_{\text{ASMPActive}}$  pilots.

12 Rules for maintaining the Active Set are specific to each protocol state (see 6.6.5.6.1 and 6.6.5.7.1).

### 13 **6.6.5.3.5 Candidate set management**

14 The access terminal shall support a maximum Candidate Set size of  $N_{\text{ASMPCandidate}}$  pilots.

15 The access terminal shall add a pilot to the Candidate Set if one of the following conditions is met:

- 16           ■ Pilot is not already in the Active Set or Candidate Set and the strength of the pilot  
17 exceeds the value specified by PilotAdd.
- 18           ■ Pilot is deleted from the Active Set, its pilot drop timer has expired, DynamicThresholds  
19 is equal to '1', and the pilot strength is above the threshold specified by PilotDrop.
- 20           ■ Pilot is deleted from the Active Set but its pilot drop timer has not expired.

21 The access terminal shall delete a pilot from the Candidate Set if one of the following conditions is  
22 met:

- 23           ■ Pilot is added to the Active Set.
- 24           ■ Pilot's drop timer has expired.
- 25           ■ Pilot is added to the Candidate Set; and, as a consequence, the size of the Candidate Set  
26 exceeds  $N_{\text{ASMPCandidate}}$ . In this case, the access terminal shall delete the weakest pilot in the  
27 set. Pilot A is considered weaker than pilot B:
- 28           □ If pilot A has an active drop timer but pilot B does not.
- 29           □ If both pilots have an active drop timer and pilot A's drop timer is closer to  
30 expiration than pilot B's.
- 31           □ If neither of the pilots has an active drop timer and pilot A's strength is less than pilot  
32 B's.



### 6.6.5.3.6 Neighbor set management

The access terminal shall support a minimum Neighbor Set size of  $N_{ASMPNeighbor}$  pilots.

- The access terminal shall maintain a counter, AGE, for each pilot in the Neighbor Set as follows.

The access terminal shall perform the following in the order specified:

- If a pilot is added to the Active Set or Candidate Set, it shall be deleted from the Neighbor Set.
- If a pilot is deleted from the Active Set, but not added to the Candidate Set, then it shall be added to the Neighbor Set with the AGE of 0.
- If a pilot is deleted from the Candidate Set, but not added to the Active Set, then it shall be added to the Neighbor Set with the AGE of 0.
- If the size of the Neighbor Set is greater than the maximum Neighbor Set supported by the access terminal, the access terminal shall delete enough pilots from the Neighbor Set such that the size of the Neighbor Set is the maximum size supported by the access terminal. The pilots with higher AGE are deleted first<sup>31</sup>.
- If the access terminal receives an *OverheadMessages.SectorParametersUpdated* indication, then:
  - The access terminal shall increment the AGE for every pilot in the Neighbor Set.
  - For each pilot in the neighbor list given as public data by the Overhead Messages Protocol that is a member of the Neighbor Set, the access terminal shall perform the following:
    - The access terminal shall set the AGE of this neighbor list pilot to the minimum of its current AGE and NeighborMaxAge.
  - For each pilot in the neighbor list given as public data by the Overhead Messages Protocol (in the order specified in the neighbor list) that is a member of the Remaining Set, the access terminal shall perform the following:
    - If the addition of this neighbor list pilot to the Neighbor Set would not cause the size of the Neighbor Set size to increase beyond the maximum Neighbor Set size supported by the access terminal, then the access terminal shall add this neighbor list pilot to the Neighbor Set with its AGE set to NeighborMaxAge.
    - If the addition of this neighbor list pilot would cause the size of the Neighbor Set to increase beyond the maximum Neighbor Set size supported by the access terminal and the Neighbor Set contains at least one pilot with AGE greater than NeighborMaxAge associated with the pilot's channel, then the access terminal shall delete the pilot in the Neighbor Set for which the difference between its AGE and the NeighborMaxAge associated with that pilot's channel (i.e., AGE - NeighborMaxAge) is the greatest and shall add this neighbor list pilot to the Neighbor Set with its AGE set to NeighborMaxAge associated with the pilot's channel.

---

<sup>31</sup> The order in which pilots of the same AGE are deleted does not matter in this case.

- 1                   – If the addition of this neighbor list pilot would cause the size of the Neighbor Set  
2 to increase beyond the maximum Neighbor Set size supported by the access  
3 terminal and the Neighbor Set does not contain a pilot with AGE greater than  
4 NeighborMaxAge associated with the pilot's channel, the access terminal shall  
5 not add this neighbor list pilot to the Neighbor Set.

#### 6 **6.6.5.3.7 Remaining set management**

7 The access terminal shall initialize the Remaining Set to contain all of the pilots whose PN offset  
8 index is an integer multiple of PilotIncrement and are not already members of any other set. In  
9 Connected State, the remaining set shall consist of pilots on at least all channels assigned in the  
10 ActiveSetAssignment message.

11 The access terminal shall add a pilot to the Remaining Set if it deletes the pilot from the Neighbor Set  
12 and if the pilot was not added to the Active Set or Candidate Set.

13 The access terminal shall delete the pilot from the Remaining Set if it adds it to another set.

#### 14 **6.6.5.4 Message sequence numbers**

15 The access network shall validate all received PilotReport messages as specified in 6.6.5.4.1.

16 The access terminal shall validate all received ActiveSetAssignment messages as specified in  
17 6.6.5.4.2.

18 The PilotReport message and the ActiveSetAssignment message carry a MessageSequence field that  
19 serves to flag duplicate or stale messages.

20 The MessageSequence field of the PilotReport message is independent of the MessageSequence field  
21 of the ActiveSetAssignment message.

#### 22 **6.6.5.4.1 PilotReport message validation**

23 When the access terminal first sends a PilotReport message, it shall set the MessageSequence field of  
24 the message to zero. Subsequently, the access terminal shall increment this field each time it sends a  
25 PilotReport message.

26 The access network shall consider all PilotReport messages it receives in the Idle State as valid.

27 The access network shall initialize the receive pointer,  $V(R)$ , to the MessageSequence field of the first  
28 PilotReport message it received in the Idle State, and the access network shall subsequently set it to  
29 the MessageSequence field of each received PilotReport message.

30 When the access network receives a PilotReport message in the Connected State, it shall validate the  
31 message using the procedure defined in 10.7. The access network shall discard the message if it is  
32 invalid.

#### 1 **6.6.5.4.2 ActiveSetAssignment message validation**

2 The access network shall set the MessageSequence field of the ActiveSetAssignment message it  
3 sends in the Idle State to zero. Subsequently, each time the access network sends a new  
4 ActiveSetAssignment message in the Connected State, it shall increment this field. If the access  
5 network is sending the same message multiple times, it shall not change the value of this field  
6 between transmissions.<sup>32</sup>

7 The access terminal shall initialize the receive pointer,  $V(R)$ , to the MessageSequence field of the first  
8 ActiveSetAssignment message that it receives in the Connected State.

9 When the access terminal receives an ActiveSetAssignment message in the Connected State, it shall  
10 validate the message using the procedure defined in 10.7. The access terminal shall discard the  
11 message if it is invalid.

#### 12 **6.6.5.5 Inactive state**

13 Upon entering this state, the access terminal shall perform the following:

- 14 ■ The access terminal shall set the Active Set, the Candidate Set, and the Neighbor Set to  
15 NULL.
- 16 ■ The access terminal shall initialize the Remaining Set to contain all of the pilots whose  
17 PN offset index is an integer multiple of PilotIncrement .
- 18 ■ The access terminal shall set  $(xL,yL)$ , the longitude and latitude of the sector in whose  
19 coverage area the access terminal last sent a PilotReport message, to (NULL, NULL).

#### 20 **6.6.5.6 Idle state**

21 In this state, PilotReport messages from the access terminal are based on the distance between the  
22 sector where the access terminal last sent a PilotReport message and the sector currently in its active  
23 set, or on the time elapsed since the last PilotReport message was sent.

##### 24 **6.6.5.6.1 Active set maintenance**

25 The access network shall not initially maintain an Active Set for the access terminal in this state.

26 The access terminal shall initially keep an Active Set of size one when it is in the Idle State. The  
27 Active Set pilot shall be the pilot associated with the Control Channel that the access terminal is  
28 currently monitoring. The access terminal shall return an *IdleHO* indication when the Active Set  
29 changes in the Idle State.

30 The access terminal shall not change its Active Set pilot at a time that causes it to miss a QuickPage  
31 packet. Other rules governing when to replace this Active Set pilot are beyond the scope of this  
32 specification.

33 If the access terminal receives an ActiveSetAssignment message, it shall set its Active Set to the list  
34 of pilots specified in the message.

---

<sup>32</sup> The access network may send a message multiple times to increase its delivery probability.

### 6.6.5.6.2 Pilot channel supervision in the idle state

The access terminal shall perform pilot channel supervision in the Idle State as follows:

- The access terminal shall monitor the pilot strength of the pilot in its active set, all of the pilots in the candidate set, and all of the pilots in the neighbor set that are on the same frequency.
- If the strength of all of the pilots that the access terminal is monitoring goes below the value specified by *PilotDrop*, the access terminal shall start a pilot supervision timer. The access terminal shall consider the timer to be expired after the time specified by *PilotDropTimer*.
- If the strength of at least one of the pilots goes above the value specified by *PilotDrop* while the pilot supervision timer is counting down, the access terminal shall reset and disable the timer.
- If the pilot supervision timer expires, the access terminal shall return a *NetworkLost* indication.

### 6.6.5.6.3 Processing access related indications in the Idle State

The following operation shall be performed at the access terminal if the *SharedSignalingMAC.AccessGrantReceived* Indication is received, and at the access network, if the *AccessChannelMAC.AccessProbeReceived* indication is received.

The Active Set Management Protocol at the access terminal shall construct the following *ActiveSetAssignment* block and place it in the public data. The fields in the block shall be decided based on the public data of the Overhead Messages Protocol, the received access grant, or the *InitialSetupConfigurationAttribute*.

The Active Set Management Protocol at the access network shall construct the following *ActiveSetAssignment* block and place it in the public data. The fields in the block shall be decided based on the public data of the Overhead Messages Protocol, the transmitted access grant, or the default values for the *InitialSetupConfigurationAttribute*.

Field	Length (bits)	Value
ChannelBandIncluded	1	1
ChannelBand	ChannelBandRecord Type Dependent	ChannelBand of the current Active Set Member
MaxPHYSubPacketSize	1	0
SingleServingSector	1	InitialSetupAttribute
RPICHEnabled	1	InitialSetupAttribute
NumRLControlSubbandsUser	4	Min(4,NumRLControlSubbands) <sup>33</sup>
RLControlSubbandUserOffset	4	0
EnhancedPilotReportEnabled	1	InitialSetupAttribute
EnhancedPilotReportRatio	4	InitialSetupAttribute
EnhancedPilotReportThreshold	4	InitialSetupAttribute
MinRequestInterval	2	InitialSetupAttribute
CQIReportInterval	2	InitialSetupAttribute
CQIReportPhase	3	InitialSetupAttribute
CQIPilotInterval	2	InitialSetupAttribute
CQIPilotPhase	3	InitialSetupAttribute
BFCHReportRate	4	InitialSetupAttribute
SFCHReportRate	4	InitialSetupAttribute
BFCHPowerOffset	3	InitialSetupAttribute
NumBFCHBits	2	InitialSetupAttribute
SFCHPowerOffset	3	InitialSetupAttribute
NumSFCHBits	2	InitialSetupAttribute
MandatoryCQICHCTRLReportingPeriod	3	InitialSetupAttribute
NumPilots	4	1
NumPilots occurrences of the following record:		
PilotPN	12	Pilot PN of sector that sent Access Grant
ActiveSetIndex	3	0

<sup>33</sup> NumRLControlSubbands is public data of the Overhead Messages Protocol for the sector and carrier the access grant was received from.

Field	Length (bits)	Value
SynchronousWithNextPilot	1	0
MAC ID	11	From AccessGrant in public data of SS MAC
AccessSequenceIDIncluded	1	0
AccessSequenceID	0	N/A
PowerControlStepUp	3	InitialSetupAttribute
PowerControlStepDown	3	InitialSetupAttribute
RDCHGainMin	6	InitialSetupAttribute
RDCHGainMax	2	InitialSetupAttribute

1

#### 2 6.6.5.6.4 Pilot report rules

3 The access terminal shall send PilotReport messages to update its location with the access network.

4 The access terminal shall not send a PilotReport message if the connection timer is active.

5 The access terminal shall comply with the following rules regarding PilotReport messages:

- 6 ■ The access terminal shall send a PilotReport message upon receiving a *SendPilotReport*  
7 command.
- 8 ■ The access terminal shall send a PilotReport message whenever it receives a  
9 PilotReportRequest message.
- 10 ■ The access terminal shall include in the PilotReport message the pilot PN, pilot strength,  
11 and drop timer status for every pilot in the Active Set and Candidate Set.
- 12 ■ The access terminal shall send a PilotReport message if the IdleStateRegistrationTimeOut  
13 attribute is nonzero, and the last PilotReport message was sent more than  
14 IdleStateRegistrationTimeOut time ago.
- 15 ■ If the RegistrationRadiusFlag is set to '0', the access terminal shall send a PilotReport  
16 message if the sector that is currently providing coverage to the access terminal has a  
17 latitude and longitude that is different from the latitude and longitude of the sector where  
18 the access terminal last sent a PilotReport message.
- 19 ■ If the RegistrationRadiusFlag is set to '1', the access terminal shall send a PilotReport  
20 message if the computed value  $r$  is greater than the value provided in the  
21 RegistrationRadius field of the SectorParameters message transmitted by the sector in  
22 which the access terminal last sent a PilotReport message.

1 If  $(x_L, y_L)$  are the longitude and latitude of the sector in whose coverage area the access terminal last  
 2 sent a PilotReport message, and  $(x_C, y_C)$  are the longitude and latitude of the sector currently providing  
 3 coverage to the access terminal, then  $r$  is given by<sup>34</sup>

$$r = \sqrt{\frac{\left[ (x_C - x_L) \times \cos\left(\frac{\pi}{180} \times \frac{y_L}{14400}\right) \right]^2 + [y_C - y_L]^2}{16}}$$

5 The access terminal shall compute  $r$  with an error of no more than  $\pm 5\%$  of its true value when  
 6  $|y_L/14400|$  is less than 60 and with an error of no more than  $\pm 7\%$  of its true value when  $|y_L/14400|$  is  
 7 between 60 and 70.<sup>35</sup>

### 8 **6.6.5.7 Connected state**

9 In this state, PilotReport messages from the access terminal are based on changes in the radio link  
 10 between the access terminal and the access network, obtained through pilot strength measurements at  
 11 the access terminal.

12 The access network determines the contents of the Active Set through ActiveSetAssignment  
 13 messages.

#### 14 **6.6.5.7.1 Active set maintenance**

##### 15 **6.6.5.7.1.1 Access network**

16 Whenever the access network sends an ActiveSetAssignment message to the access terminal, it shall  
 17 add to the Active Set any pilots listed in the message that are not currently in the Active Set. The  
 18 access network shall place the most recently transmitted ActiveSetAssignment message in the public  
 19 data of the Active Set Management Protocol.

20 The access network shall delete a pilot from the Active Set if the pilot was not listed in a  
 21 ActiveSetAssignment message and if the access network received the ActiveSetComplete message,  
 22 acknowledging that ActiveSetAssignment message.

23 The access network should send an ActiveSetAssignment message to the access terminal in response  
 24 to changing radio link conditions, as reported in the access terminal's PilotReport messages.

25 The access network should only specify a pilot in the ActiveSetAssignment message if it has  
 26 allocated the required resources in the associated sector. This means that the sector specified by the  
 27 pilot is ready to receive data from the access terminal and is ready to transmit queued data to the  
 28 access terminal should the access terminal directs a handoff request to that sector.

---

<sup>34</sup> The  $x$ 's denote longitude and the  $y$ 's denote latitude.

<sup>35</sup>  $x_L$  and  $y_L$  are given in units of 1/4 seconds.  $x_L/14400$  and  $y_L/14400$  are in units of degrees.

1 If the access network adds or deletes a pilot in the Active Set, it shall send an *ActiveSetUpdated*  
2 indication.

### 3 **6.6.5.7.1.2 Access terminal**

4 If the access terminal receives a valid ActiveSetAssignment message (see 6.6.5.4.2), it shall replace  
5 the contents of its current Active Set with the pilots specified in the message. The access terminal  
6 shall process the message as defined in 6.6.5.7.5.

### 7 **6.6.5.7.2 ResetReport message**

8 The access network may send a ResetReport message to reset the conditions under which PilotReport  
9 messages are sent from the access terminal. Access terminal usage of the ResetReport message is  
10 specified in the following section.

### 11 **6.6.5.7.3 Pilot strength report rules**

12 The access terminal sends a PilotReport message to the access network in this state to request  
13 addition or deletion of pilots from its Active Set. If DynamicThresholds is equal to '0', the access  
14 terminal shall include in the PilotReport message the pilot PN, pilot strength, and drop timer status for  
15 every pilot in the Active Set and Candidate Set. If DynamicThresholds is equal to '1', then the access  
16 terminal shall include in the PilotReport message the pilot PN, pilot strength, and drop timer status for  
17 every pilot in the Active Set, for each pilot in the Candidate Set whose strength is above the values  
18 specified by PilotAdd, and for each pilot in the Candidate set whose strength, PS, satisfies the  
19 following inequality:

$$20 \quad 10 \times \log_{10} PS > \frac{\text{SoftSlope}}{8} \times 10 \times \log_{10} PS_{N_A} + \frac{\text{AddIntercept}}{2}$$

21 The access terminal shall send a PilotReport message if any one of the following occurs:

- 22 ■ The access terminal receives a PilotReportRequest message. The access terminal shall set  
23 the DetailedInfoIncluded field of the PilotReport message to 1 if it receives a  
24 PilotRequest message with the ReportFormat set to 0x01.
- 25 ■ The Default Active Set Management Protocol receives a *SendPilotReport* command.
- 26 ■ If DynamicThresholds is equal to '0' and the strength of a Neighbor Set or Remaining  
27 Set pilot is greater than the value specified by PilotAdd.
- 28 ■ If DynamicThresholds is equal to '1' and the strength of a Neighbor Set or Remaining  
29 Set pilot, PS, satisfies the following inequality:

$$30 \quad 10 \times \log_{10} PS > \max\left(\frac{\text{SoftSlope}}{8} \times 10 \times \log_{10} PS_{N_A} + \frac{\text{AddIntercept}}{2}, \frac{\text{PilotAdd}}{2}\right)$$

- 31 ■ If DynamicThresholds is equal to '0' and the strength of a Candidate Set pilot is greater  
32 than the value specified by PilotCompare above an Active Set pilot, and a PilotReport  
33 message carrying this information has not been sent since the last ResetReport message  
34 was received.
- 35 ■ If DynamicThresholds is equal to '0' and the strength of a Candidate Set pilot is above  
36 PilotAdd, and a PilotReport message carrying this information has not been sent since the  
37 last ResetReport message was received.



- 1       ■ If DynamicThresholds is equal to '1' and
- 2       □ The strength of a Candidate Set pilot, PS, satisfies the following inequality:

$$3 \quad 10 \times \log_{10} PS > \frac{\text{SoftSlope}}{8} \times 10 \times \log_{10} PS_{N_A} + \frac{\text{AddIntercept}}{2}$$

- 4       □ A PilotReport message carrying this information has not been sent since the last
- 5       ResetReport message was received.

- 6       ■ If DynamicThresholds is equal to '1' and

- 7       □ The strength of a Candidate Set pilot is greater than the value specified by
- 8       PilotCompare above an Active Set pilot, and

- 9       □ The strength of a Candidate Set pilot, PS, satisfies the following inequality:

$$10 \quad 10 \times \log_{10} PS > \frac{\text{SoftSlope}}{8} \times 10 \times \log_{10} PS_{N_A} + \frac{\text{AddIntercept}}{2}$$

- 11      □ A PilotReport message carrying this information has not been sent since the last
- 12      ResetReport message was received.

- 13      ■ The pilot drop timer of an Active Set pilot has expired, and a PilotReport message
- 14      carrying this information has not been sent since the last ResetReport message was
- 15      received.

- 16      ■ If the Active Set Size is greater than one, and

- 17      □ EnhancedPilotReportEnabled=1, and

- 18      □ The strength of a pilot in the active set has changed (increased or decreased) by more
- 19      than EnhancedPilotReportThreshold since the last PilotReport was sent.

- 20      ■ If the Active Set Size is one, and

- 21      □ EnhancedPilotReportEnabled=1, and

- 22      □ The strongest current non-active pilot (for example, pilot j) has strength more than
- 23      the EnhancedPilotReportRatio fraction of the total interference, and

- 24      □ The strength of pilot j differs by more than EnhancedPilotReportThreshold from the
- 25      strength of the second strongest pilot at the time the last pilot report was sent
- 26      (regardless of whether the second strongest pilot at that time was pilot j or some other
- 27      pilot).

#### 28    **6.6.5.7.4 VCQI report rules**

29    The access terminal shall send a VCQI report every VCQIReportInterval superframes, where  
 30    VCQIReportInterval shall be part of the ActiveSetAssignment message. If the VCQIReportInterval is  
 31    set to zero, the access terminal shall not send any VCQI reports. The VCQI report shall be a  
 32    VCQIReportSISO if the CQIReportingMode of the ActiveSetAssignment message is set to CQISISO.  
 33    The VCQI report shall be a VCQIReportMIMO if the CQIReportingMode of the  
 34    ActiveSetAssignment message is set to CQISCW or CQIMCW.

### 6.6.5.7.5 Processing the ActiveSetAssignment message in the connected state

The access terminal shall process a valid ActiveSetAssignment message (see 6.6.5.4.2) as follows:

- The access terminal shall return an *ActiveSetUpdated* indication.
- The access terminal shall update its Active Set as defined in 6.6.5.7.1.2.
- The access terminal shall tune to the frequency defined by the ChannelBand record, if this record is included in the message. If more than one ChannelBand record is specified in the message, the access terminal may autonomously tune to one of the frequencies defined by the ChannelBand.
- The Active Set Management Protocol at the access terminal shall inform the physical layer about the synchronous status of sectors in the ActiveSetAssignment message according to the following rules. Two sectors shall be said to be synchronous both sectors are listed adjacent to each other in the ActiveSetAssignment message, and the sector listed first has the SynchronousWithNextSector bit set to '1'.

In all other cases, the Active Set Management Protocol shall inform the physical layer that the two sectors are not synchronous.

- The access terminal shall place the ActiveSetAssignment message in the public data. If the ActiveSetAssignment message has a non-zero value for the AdditionalFieldsStatus field for some pilot, the access terminal shall fill the additional fields (as defined in the ActiveSetAssignment message) according to rules followed by the access network for setting the AdditionalFieldsStatus field (as defined in the ActiveSetAssignment message).
- The access terminal shall send the access network an ActiveSetComplete message specifying the MessageSequence value received in the ActiveSetAssignment message.

### 6.6.5.7.6 Processing the ActiveSetComplete message

The access network should set a transaction timer when it sends an ActiveSetAssignment message. If the access network sets a transaction timer, it shall reset the timer when it receives an ActiveSetComplete message containing a MessageSequence field equal to the one sent in the ActiveSetAssignment message.

If the timer expires, the access network should return an *ActiveSetManagement.ConnectionLost* indication.

## 6.6.6 Message formats

The protocol uses the AttributeUpdateRequest, AttributeUpdateAccept, and AttributeUpdateReject messages of the Generic Attribute Update Protocol in 10.9 to update configurable attributes.

### 6.6.6.1 PilotReport

The access terminal sends the PilotReport message to notify the access network of its current location and provide it with an estimate of its surrounding radio link conditions.

Field	Length (bits)
MessageID	8
MessageSequence	8
ReferencePilotPN	12
ReferencePilotStrength	8
ReferenceKeep	1
NumPilots	5

NumPilots occurrences of the following record:

PilotPN	12
ChannelBandIncluded	1
ChannelBand	0 or ChannelBandRe cordType Dependent
PilotStrength	8
Keep	1
ExtendedChannelInfoAvailable	1
DetailedInfoIncluded	1
FineTimingOffset	0 or 16
SuperframeOffset	0 or 16
SectorID	0 or 128

Reserved	Variable
----------	----------

**MessageID** The access terminal shall set this field to 0x00.

**MessageSequence** The access terminal shall set this field to the sequence number of this message. The sequence number of this message is 1 more than the sequence number of the last PilotReport message (modulo 256) sent by this access terminal. If this is the first PilotReport message sent by the access terminal, it shall set this field to 0x00.

**ReferencePilotPN** The access terminal shall set this field to the access terminal's time reference (the reference pilot), relative to the zero offset pilot PN sequence in units of 64 PN chips.

1	ReferencePilotStrength	The access terminal shall set this field to $\lfloor -2 \times 10 \times \log_{10} PS \rfloor$ , where PS is
2		the strength of the reference pilot, measured as specified in 6.6.5.3.2. If this
3		value is less than 0, the access terminal shall set this field to '00000000'. If
4		this value is greater than '11111111', the access terminal shall set this field
5		to '11111111'.
6	ReferenceKeep	If the pilot drop timer corresponding to the reference pilot has expired, the
7		access terminal shall set this field to '0'; otherwise, the access terminal shall
8		set this field to '1'.
9	NumPilots	The access terminal shall set this field to the number of pilots that follow this
10		field in the message.
11	PilotPN	The PN offset in resolution of 1 chip of a pilot in the Active Set or Candidate
12		Set of the access terminal that is not the reference pilot.
13	ChannelBandIncluded	The access terminal shall set this field to '1' if the channel for this pilot offset
14		is not the same as the current channel. Otherwise, the access terminal shall
15		set this field to '0'.
16	ChannelBand	The access terminal shall include this field if the ChannelBandIncluded field
17		is set to '1'. The access terminal shall set this to the ChannelBand record
18		corresponding to this pilot (see 10.1). Otherwise, the access terminal shall
19		omit this field for this pilot offset.
20	PilotStrength	The access terminal shall set this field to $\lfloor -2 \times 10 \times \log_{10} PS \rfloor$ , where PS is
21		the strength of the pilot (PilotStrength) in the above field, measured as
22		specified in 6.6.5.3.2. If this value is less than 0, the access terminal shall set
23		this field to '00000000'. If this value is greater than '11111111', the access
24		terminal shall set this field to '11111111'.
25	Keep	If the pilot drop timer corresponding to the pilot in the above field has
26		expired, the access terminal shall set this field to '0'; otherwise, the access
27		terminal shall set this field to '1'.
28	ExtendedChannelInfoAvailable	
29		If the access terminal has a ExtendedChannelInfo block for the sector, and
30		the ExtendedChannelInfo block is valid, as defined by the Overhead
31		Messages Protocol, the access terminal shall set this field to '1'. Otherwise
32		the access terminal shall set this field to '0'.
33	DetailedInfoIncluded	The setting for this field is described in the rules for transmission of a
34		PilotReport message.
35	FineTimingOffset	This field shall be included if DetailedInfoIncluded is set to '1'. If the
36		beginning of a superframe m from the pilot being reported is during
37		superframe n of the reference pilot, this field shall be the time between the
38		beginning of superframes n and m, in units of microseconds. The all ones
39		value of this field is reserved, and indicates that timing information is not
40		available.

1	SuperframeOffset	This field shall be included if DetailedInfoIncluded is set to '1'. This field shall be set to the value $2^{15} + m - n$ , where m and n are described in the discussion of the FineTimingOffset. The all ones value of this field is reserved, and indicates that timing information is not available.
2		
3		
4		
5	SectorID	This field shall be set to the SectorID of the sector that transmits the pilot being reported.
6		
7	Reserved	The number of bits in this field is equal to the number needed to make the message length an integer number of octets. This field shall be set to all zeros.
8		
9		
10		

<b>Channels</b>	RTC	<b>SLP</b>	Reliable	Best Effort
<b>Addressing</b>	Unicast	<b>Security</b>	Required	

11

### 12 6.6.6.2 VCQIReportSISO

13 The access terminal shall send this message if the CQIReportingMode is set to CQISISO.

14

Field	Length (bits)
MessageID	8
NumPilots	3
NumPilots Occurrences of the following record{	
ActiveSetIndex	3
NumInterlaces	4
NumInterlaces instances of the following record{	
InterlaceID	4
NumSubbands	4
NumSubbands occurrences of the following record{	
SubbandID	4
VCQI	4
}}	
Reserved	Variable

15	MessageID	The access terminal shall set this field to 0x01.
16	NumPilots	The access terminal shall set this field to the number of sectors for which the report is being sent.
17		
18	ActiveSetIndex	The access terminal shall set this field to the ActiveSetIndex corresponding to the sector for which VCQI is being reported.
19		
20	NumInterlaces	The access terminal shall set this field to the number of interlaces in the system.
21		

1	InterlaceID	The access terminal shall set this field to the interlace number corresponding to the following record.
2		
3	NumSubbands	The number of subbands in the system.
4	SubbandID	This field shall be set to the subband corresponding to the following VCQI.
5	VCQI	This field shall be set to the VCQIValueSISO for this subband and interlace. VCQIValueSISO is defined in the CQICH Physical Layer Channel Procedures for the Reverse Control Channel MAC Protocol.
6		
7		
8	Reserved	The number of bits in this field is equal to the number needed to make the message length an integer number of octets. This field shall be set to all zeros.
9		
10		
11		

<b>Channels</b>	RTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Unicast	<b>Security</b>	Required

12

### 13 6.6.6.3 VCQIReportMIMO

14 The access terminal shall send this message if CQIReportingMode is set to CQISCW or CQIMCW.

15

Field	Length (bits)
MessageID	8
NumPilots	3
NumPilots Occurrences of the following record{	
ActiveSetIndex	3
NumInterlaces	4
NumInterlaces instances of the following record{	
InterlaceID	4
NumSubbands	4
NumSubbands occurrences of the following record{	
SubbandID	4
NumEffectiveAntennas	2
NumEffectiveAntennas occurrences of the following field{	
VCQI	4
}}}}	
Reserved	Variable

16 MessageID The access terminal shall set this field to 0x02.

17 NumPilots The access terminal shall set this field to the number of sectors for which the  
18 report is being sent.

1	ActiveSetIndex	The access terminal shall set this field to the ActiveSetIndex corresponding to the sector for which VCQI is being reported.
2		
3	NumInterlaces	The access terminal shall set this field to the number of interlaces in the system.
4		
5	InterlaceID	The access terminal shall set this field to the interlace number corresponding to the following record.
6		
7	NumSubbands	The number of subbands in the system.
8	SubbandID	This field shall be set to the subband corresponding to the following VCQI.
9	NumEffectiveAntennas	
10		This field shall be set as determined to the value in the QuickChannelInfo block for this sector, stored in the public data of the Overhead Parameters Protocol.
11		
12		
13	VCQI	If CQIReportingMode is set to CQISCW, this field shall be set to the VCQIValueSCW for this interlace, subband and rank.
14		
15		If CQIReportingMode is set to CQIMCW, this field shall be set to the VCQIValueMCW for this interlace, subband and layer. VCQIValueSCW and VCQIValueMCW are defined in the CQICH Physical Layer Channel Procedures for the Reverse Control Channel MAC Protocol.
16		
17		
18		
19	Reserved	The number of bits in this field is equal to the number needed to make the message length an integer number of octets. This field shall be set to all zeros.
20		
21		
22		

<b>Channels</b>	RTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Unicast	<b>Security</b>	Required

23

#### 24 6.6.6.4 ActiveSetAssignment

25 The access network sends the ActiveSetAssignment message to manage the access terminal's Active Set.

26

Field	Length (bits)
MessageID	8
MessageSequence	8
NumChannelBands	3
NumChannelBands instances of the following record{	
ChannelBandIncluded	1
ChannelBand	0 or ChannelBandRe cordType Dependent

Field	Length (bits)
MaxPHYSubPacketSize	1
SingleServingSector	1
RPICHEnabled	1
NumRLControlSubbandsUser	4
RLControlSubbandUserOffset	4
VCQIMeasureInterval	2
EnhancedPilotReportEnabled	1
EnhancedPilotReportRatio	4
EnhancedPilotReportThreshold	4
MinRequestInterval	2
CQIReportInterval	2
CQIReportPhase	2
CQIPilotInterval	2
CQIPilotPhase	2
BFCHReportRate	4
SFCHReportRate	4
NumBFCHBits	2
NumSFCHBits	2
MandatoryCQICHCTRLReportingPeriod	3
NumPilots	3
NumPilots occurrences of the following record{	
PilotPN	12
ActiveSetIndex	3
SynchronousWithNextPilot	1
AdditionalFieldsStatus	2
The following eight fields shall be included if AdditionalFieldsStatus is equal to '00' {	
MAC ID	11
AccessSequenceIDIncluded	1
AccessSequenceID	0 or 4
PowerControlStepUp	3
PowerControlStepDown	3
RDCHGainMin	6
RDCHGainMax	4
}}}	
Reserved	Variable

1 MessageID

The access network shall set this field to 0x03.



1	MessageSequence	The access network shall set this field to 1 higher than the MessageSequence field of the last ActiveSetAssignment message (modulo $2^S$ , $S=8$ ) sent to this access terminal.
2		
3		
4	NumChannelBands	The access network shall set this field to the number of ChannelBands included in the message. For each channel, the access network shall use the NumPilots field to determine the number of pilots included in the Active Set for that channel.
5		
6		
7		
8	ChannelBandIncluded	The access network shall set this field to '1' if the ChannelBand record is included for these pilots. Otherwise, the access network shall set this field to '0'.
9		
10		
11	ChannelBand	If the ChannelBandIncluded field is set to '0' the access network shall omit this field. The access network shall include this field if the ChannelBandIncluded field is set to '1'. The access network shall set this to the ChannelBand record corresponding to the following pilots (see 10.1). If the optional parameters of this ChannelBand record are different from the parameters for the sector transmitting this message, the access network shall include the optional parameters in the ChannelBand record. ..
12		
13		
14		
15		
16		
17		
18	MaxPHYSubPacketSize	This field shall determine the maximum subpacket size used by the Physical Layer. This parameter shall take the value 4096 if this field is set to '0' and shall take the value 8192 if this field is set to '1'. If the access network changes the value of this parameter at the access terminal, the access network should use a new set of MACIDs in this message.
19		
20		
21		
22		
23		
24	SingleServingSector	The access network shall set this field to '1' if the access terminal is required to maintain the same forward and reverse link serving sectors. The access network shall set this field to '0' otherwise.
25		
26		
27	RPICHEnabled	This field shall determine whether the R-PICH is transmitted.
28	NumRLControlSubbandsUser	This field shall determine the number of subbands over which the access terminal transmits the control segment waveform. This parameter shall take the value $n+1$ . The access network shall set this parameter to be the same for sectors in a synchronous subset.
29		
30		
31		
32		
33	RLControlSubbandUserOffset	This field shall determine the offset used to generate the set of subbands over which the access terminal transmits the control segment waveform. This is an integer between 0 and 15. The access network shall set this parameter to be the same for sectors in a synchronous subset.
34		
35		
36		
37		
38	CQIReportingMode	This field shall specify the configuration of MIMO CQI reports sent by the access terminal.
39		

**Table 51 Interpretation of CQIReportingMode**

CQIReportingMode	Interpretation
00	SISO CQI Report
01	CQISCW CQI Report
10	CQISCW CQI Report
11	Reserved

2	VCQIReportInterval	The access network shall set this field to the interval at which the access terminal shall send a VCQIReport message, in units of 0.25 seconds.
3		
4	VCQIMeasureInterval	The access network shall set this field to control VCQI reports. This parameter shall take the value $(2*n+1)$ superframes.
5		
6	VCQIMeasureThreshold	
7		The access network shall set this field to control VCQI reports.
8	MinVCQIReportInterval	
9		The access network shall set this field to control VCQI reports.
10	MaxVCQIReportInterval	
11		The access network shall set this field to control VCQI reports.
12	EnhancedPilotReportEnabled	
13		The access network shall use this field to determine if enhanced pilot reports shall be triggered.
14		
15	EnhancedPilotReportRatio	
16		The access network shall set this field to control pilot reports for the case when EnhancedPilotReportEnabled is set to '1'. This parameter shall take the value $n*0.5$ dB.
17		
18		
19	EnhancedPilotReportThreshold	
20		The access network shall set this field to control pilot reports for the case when EnhancedPilotReportEnabled is set to '1', in units of 1 dB.
21		
22	MinRequestInterval	The access network shall set this field to specify the minimum number of PHY Frames between two request transmissions by the access terminal. For example, if the field is set to zero, the access terminal may send a request transmission in PHY Frames that are one control period segment apart.
23		
24		
25		
26	CQIReportInterval	This field shall determine the periodicity at which an access terminal shall report a CQI value. For example, a value of 4 shall mean that every fourth control segment period shall contain a CQI report.
27		
28		

**Table 52 Interpretation of CQIReportInterval**

<b>CQIReportInterval</b>	<b>Periodicity of CQI report in number of control segment periods</b>
00	1
01	2
10	3
11	4

**CQIReportPhase** This field shall determine the phase of the CQI reports that are transmitted with periodicity CQIReportInterval. The access terminal shall transmit CQI report in control segment period number CQIReportPhase in superframe 0. For example, if CQIReportInterval is 2 and CQIReportPhase is 1, the access terminal shall transmit CQI in control segments number 1, 3, 5... starting at superframe 0. If CQIReportInterval is 2 and CQIReportPhase is 0, the access terminal shall transmit CQI in control segments number 0, 2, 4... starting at superframe 0.

**CQIPilotInterval** This field shall determine the periodicity at which an access terminal shall report a default CQICH Pilot. For example, a value of 3 shall mean that every third CQI report shall be set to a CQIPilot report.

**Table 53 Interpretation of CQIPilotInterval**

<b>CQIPilotInterval</b>	<b>Periodicity of default CQI report in number of CQIReportIntervals</b>
00	2
01	4
10	6
11	8

**CQIPilotPhase** This field shall determine the phase of the CQICH Pilot reports that are transmitted with periodicity CQIPilotInterval. The access terminal shall transmit CQICH Pilot report in CQI report number CQICH PilotPhase in superframe 0.

**BFCHReportRate** This field shall be set to the number of R-BFCH reports per super-frame.

**SFCHReportRate** This field shall be set to the number of R-SFCH reports per super-frame.

**NumBFCHBits** This field shall determine the number and type of bits sent on the R-BFCH. If this field is set to '11' the access terminal shall send all bits. If this field is set to '10', the access terminal sets the fields corresponding to the CQI value to 0. The values '00' and '01' are reserved.

1	NumSFCHBits	This field shall determine the number and type of bits sent on the R-SFCH. If
2		this field is set to '11' the access terminal shall send all bits. If this field is set
3		to '10', the access terminal sets the fields corresponding to the CQI value to
4		0. The values '00' and '01' are reserved.
5	MandatoryCQICHCTRLReportingPeriod	
6		This field shall determine the reporting period in multiples of
7		CQIReportInterval, where the AT is mandated to transmit R-CQICHCTRL
8		report in the R-CQICH channel. The value '000' is reserved.
9	NumPilots	The access network shall set this field to the number of pilots included in this
10		message.
11	PilotPN	The access network shall set this field to the PilotPN of a pilot included in
12		the message.
13	ActiveSetIndex	The access network shall set this field to enable CQI reporting to different
14		sectors in the active set.
15	SynchronousWithNextPilot	
16		The access network shall set this field to '1' if this sector is synchronous with
17		the next sector listed in the message. The access network shall set this field to
18		'0' if this sector is the last sector listed in the message, or if this sector is not
19		synchronous with the next sector listed in the message. Rules for determining
20		if two sectors are synchronous are given in the synchronization and timing
21		section of the physical layer (9.3.3).
22	AdditionalFieldsStatus	This field shall describe the status of the additional fields described in the
23		message definition. AdditionalFieldsStatus shall be set to '00' if the
24		additional fields are included in the message. AdditionalFieldsStatus shall be
25		set to '01' if the additional fields are not included in the message, and the
26		values of the additional fields for this sector are identical to the values in the
27		last transmitted ActiveSetAssignment message. AdditionalFieldsStatus shall
28		be set to '01' if the additional fields are not included in the message, and
29		values of the additional fields for this sector are identical to the values of the
30		additional fields for the first sector listed in the message. The value '11' shall
31		be reserved.
32	MAC ID	The access network shall set this field to the MAC ID assigned to the user in
33		this sector.
34	AccessSequenceIDIncluded	
35		The access network shall set this field to '1' if an AccessSequenceID is
36		included with the message.
37	AccessSequenceID	This field shall be set to the access sequence that the access terminal shall
38		use to acquire reverse link timing for this sector. This field shall take values
39		from zero to eight.

- 1 PowerControlStepUp This field shall determine the power increase at the access terminal when it  
2 receives a power up command from the access network. This parameter shall  
3 take the value  $(n+1)*0.25$  dB.
- 4 PowerControlStepDown  
5 This field shall be determine the power decrease at the access terminal when  
6 it receives a power down command from the access network. This parameter  
7 shall take the value  $(n+1)*0.25$  dB.
- 8 RDCHGainMin This field shall determine the lower limit of the delta value at the access  
9 terminal. This parameter shall take the value  $(0.25*n - 4)$  dB.
- 10 RDCHGainMax This field shall determine the upper limit of the delta value at the access  
11 terminal. This parameter shall take the value  $(RDCHGainMin + n)$  dB.
- 12 Reserved The number of bits in this field is equal to the number needed to make the  
13 message length an integer number of octets. This field shall be set to all  
14 zeros. The access terminal shall ignore this field.

<b>Channels</b>	FTC	<b>SLP</b>	Reliable
<b>Addressing</b>	Unicast	<b>Security</b>	Required

16

### 17 6.6.6.5 ActiveSetComplete

18 The access terminal sends the ActiveSetComplete message to provide an acknowledgment for the  
19 ActiveSetAssignment message.

20

Field	Length (bits)
MessageID	8
MessageSequence	8

21 MessageID The access terminal shall set this field to 0x04.

22 MessageSequence The access terminal shall set this field to the MessageSequence field of the  
23 ActiveSetAssignment message whose receipt this message is acknowledging.

24

<b>Channels</b>	RTC	<b>SLP</b>	Reliable
<b>Addressing</b>	Unicast	<b>Security</b>	Required

25

### 6.6.6.6 ResetReport

The access network sends the ResetReport message to reset the PilotReport transmission rules at the access terminal.

Field	Length (bits)
MessageID	8

MessageID The access network shall set this field to 0x05.

<b>Channels</b>	FTC	<b>SLP</b>	Reliable
<b>Addressing</b>	Unicast	<b>Security</b>	Required

### 6.6.6.7 PilotReportRequest

The access network sends a PilotReportRequest message to request the access terminal to send a PilotReport message.

Field	Length (bits)
MessageID	8
ReportFormat	8

MessageID The access network shall set this field to 0x06.

ReportFormat The access network shall set this field to indicate the format of the PilotReport it is requesting from the access terminal. The valid values for this field are 0x00 and 0x01, where 0x01 indicates that a detailed pilot report is requested.

<b>Channels</b>	FTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Unicast	<b>Security</b>	Required

## 6.6.7 Interface to other protocols

### 6.6.7.1 Commands

This protocol does not send any commands.

### 6.6.7.2 Indications

This protocol registers to receive the following indications:

- *OverheadMessages.SectorParametersUpdated*
- *SharedSignalingMAC.AccessGrantReceived*

- 1           ■ *ConnectedState.RegistrationRadiusUpdated*
- 2           ■ *IdleState.RegistrationRadiusUpdated*

### 3   **6.6.8 Configuration attributes**

4   The following complex attributes and default values are defined (see 10.3 for attribute record  
5   definition). The access terminal should not initiate modification of the following attributes.

6   Unless specified otherwise, the access terminal and the access network shall use the Generic Attribute  
7   Update Protocol in 10.9 to update configurable attributes belonging to the Default Active Set  
8   Management Protocol.

#### 9   **6.6.8.1 SearchParameters attribute**

Field	Length (bits)	Default Value
Length	8	N/A
AttributeID	8	N/A
PilotIncrement	4	4

- 11   **Length**                   Length of the complex attribute in octets. The access network shall set this  
12                               field to the length of the complex attribute excluding the Length field.
- 13   **AttributeID**             The access network shall set this field to 0x00.
- 14   **PilotIncrement**         The access network shall set this field to the pilot PN sequence increment, in  
15                               units of PN chips, that access terminals are to use for searching the  
16                               Remaining Set. The access network should set this field to the largest  
17                               increment such that the pilot PN sequence offsets of all its neighbor access  
18                               networks are integer multiples of that increment. The access terminal shall  
19                               support all of the valid values for this field.

### 6.6.8.2 SetManagementSameChannelBandParameters attribute

The access terminal shall use these attributes if the pilot being compared is on the same channel as the active set pilots' channel.

Field	Length (bits)	Default Value
Length	8	N/A
AttributeID	8	N/A
PilotAdd	6	0x0e
PilotCompare	6	0x05
PilotDrop	6	0x12
PilotDropTimer	4	3
DynamicThresholds	1	0
SoftSlope	0 or 6	N/A
AddIntercept	0 or 6	N/A
DropIntercept	0 or 6	N/A
NeighborMaxAge	4	0
Reserved	variable	N/A

Length	Length of the complex attribute in octets. The access network shall set this field to the length of the complex attribute excluding the Length field.
AttributeID	The access network shall set this field to 0x01.
PilotAdd	This value is used by the access terminal to trigger a PilotReport in the Connected State. The access network shall set this field to the pilot detection threshold, expressed as an unsigned binary number equal to $\lfloor -2 \times 10 \times \log_{10} Ec/I_0 \rfloor$ . The value used by the access terminal is $-0.5$ dB times the value of this field. The access terminal shall support all of the valid values specified by this field.
PilotDrop	This value is used by the access terminal to start a pilot drop timer for a pilot in the Active Set or the Candidate Set. The access network shall set this field to the pilot drop threshold, expressed as an unsigned binary number equal to $\lfloor -2 \times 10 \times \log_{10} Ec/I_0 \rfloor$ . The value used by the access terminal is $-0.5$ dB times the value of this field. The access terminal shall support all of the valid values specified by this field.
PilotCompare	Active Set versus Candidate Set comparison threshold, expressed as a 2's complement number. The access terminal transmits a PilotReport message when the strength of a pilot in the Candidate Set exceeds that of a pilot in the Active Set by this margin. The access network shall set this field to the threshold Candidate Set pilot to Active Set pilot ratio, in units of 0.5 dB. The access terminal shall support all of the valid values specified by this field.



1 PilotDropTimer Timer value after which an action is taken by the access terminal for a pilot  
 2 that is a member of the Active Set or Candidate Set, and whose strength has  
 3 not become greater than the value specified by PilotDrop. If the pilot is a  
 4 member of the Active Set, a PilotReport message is sent in the Connected  
 5 State. If the pilot is a member of the Candidate Set, it will be moved to the  
 6 Neighbor Set. The access network shall set this field to the drop timer value  
 7 shown in Table 54 corresponding to the pilot drop timer value to be used by  
 8 access terminals. The access terminal shall support all of the valid values  
 9 specified by this field.

10 **Table 54 Pilot drop timer values**

PilotDropTimer	Timer Expiration (seconds)	PilotDropTimer	Timer Expiration (seconds)
0	< 0.1	8	27
1	1	9	39
2	2	10	55
3	4	11	79
4	6	12	112
5	9	13	159
6	13	14	225
7	19	15	319

11 DynamicThresholds This field shall be set to '1' if the following three fields are included in this  
 12 record. Otherwise, this field shall be set to '0'.

13 SoftSlope This field shall be included only if DynamicThresholds is set to '1'. This  
 14 field shall be set to an unsigned binary number, which is used by the access  
 15 terminal in the inequality criterion for adding a pilot to the Active Set or  
 16 dropping a pilot from the Active Set. The access terminal shall support all of  
 17 the valid values specified by this field.

18 AddIntercept This field shall be included only if DynamicThresholds is set to '1'. This  
 19 field shall be set to a 2's complement signed binary number in units of dB.  
 20 The access terminal shall support all of the valid values specified by this  
 21 field.

22 DropIntercept This field shall be included only if DynamicThresholds is set to '1'. This  
 23 field shall be set to a 2's complement signed binary number in units of dB.  
 24 The access terminal shall support all of the valid values specified by this  
 25 field.

26 NeighborMaxAge The access network shall set this field to the maximum AGE value beyond  
 27 which the access terminal is to drop members from the Neighbor Set. The  
 28 access terminal shall support all of the valid values specified by this field.

1 Reserved The access network shall set this field to zero. The access terminal shall  
 2 ignore this field. The length of this field shall be such that the entire record is  
 3 octet-aligned.

#### 4 6.6.8.3 SetManagementDifferentChannelBandParameters attribute

5 The access terminal shall use these attributes if the pilot being compared is on a channel that is  
 6 different from the active set pilots' channel.

7

Field	Length (bits)	Default Value
Length	8	N/A
AttributeID	8	N/A
PilotAdd	6	0x0e
PilotCompare	6	0x05
PilotDrop	6	0x12
PilotDropTimer	4	3
DynamicThresholds	1	0
SoftSlope	0 or 6	N/A
AddIntercept	0 or 6	N/A
DropIntercept	0 or 6	N/A
NeighborMaxAge	4	0
Reserved	variable	N/A

8 Length Length of the complex attribute in octets. The access network shall set this  
 9 field to the length of the complex attribute excluding the Length field.

10 AttributeID The access network shall set this field to 0x02.

11 PilotAdd This value is used by the access terminal to trigger a PilotReport in the  
 12 Connected State. The access network shall set this field to the pilot detection  
 13 threshold, expressed as an unsigned binary number equal to  
 14  $\lfloor -2 \times 10 \times \log_{10} E_c/I_0 \rfloor$ . The value used by the access terminal is  $-0.5$  dB  
 15 times the value of this field. The access terminal shall support all of the valid  
 16 values specified by this field.

17 PilotDrop This value is used by the access terminal to start a pilot drop timer for a pilot  
 18 in the Active Set or the Candidate Set. The access network shall set this field  
 19 to the pilot drop threshold, expressed as an unsigned binary number equal to  
 20  $\lfloor -2 \times 10 \times \log_{10} E_c/I_0 \rfloor$ . The value used by the access terminal is  $-0.5$  dB  
 21 times the value of this field. The access terminal shall support all of the valid  
 22 values specified by this field.

1	PilotCompare	Active Set versus Candidate Set comparison threshold, expressed as a 2's complement number. The access terminal transmits a PilotReport message when the strength of a pilot in the Candidate Set exceeds that of a pilot in the Active Set by this margin. The access network shall set this field to the threshold Candidate Set pilot to Active Set pilot ratio, in units of 0.5 dB. The access terminal shall support all of the valid values specified by this field.
2		
3		
4		
5		
6		
7	PilotDropTimer	Timer value after which an action is taken by the access terminal for a pilot that is a member of the Active Set or Candidate Set, and whose strength has not become greater than the value specified by PilotDrop. If the pilot is a member of the Active Set, a PilotReport message is sent in the Connected State. If the pilot is a member of the Candidate Set, it will be moved to the Neighbor Set. The access network shall set this field to the drop timer value shown in Table 54 corresponding to the pilot drop timer value to be used by access terminals. The access terminal shall support all of the valid values specified by this field.
8		
9		
10		
11		
12		
13		
14		
15		
16	DynamicThresholds	This field shall be set to '1' if the following three fields are included in this record. Otherwise, this field shall be set to '0'.
17		
18	SoftSlope	This field shall be included only if DynamicThresholds is set to '1'. This field shall be set to an unsigned binary number, which is used by the access terminal in the inequality criterion for adding a pilot to the Active Set or dropping a pilot from the Active Set. The access terminal shall support all of the valid values specified by this field.
19		
20		
21		
22		
23	AddIntercept	This field shall be included only if DynamicThresholds is set to '1'. This field shall be set to a 2's complement signed binary number in units of dB. The access terminal shall support all of the valid values specified by this field.
24		
25		
26		
27	DropIntercept	This field shall be included only if DynamicThresholds is set to '1'. This field shall be set to a 2's complement signed binary number in units of dB. The access terminal shall support all of the valid values specified by this field.
28		
29		
30		
31	NeighborMaxAge	The access network shall set this field to the maximum AGE value beyond which the access terminal is to drop members from the Neighbor Set. The access terminal shall support all of the valid values specified by this field.
32		
33		
34	Reserved	The access network shall set this field to zero. The access terminal shall ignore this field. The length of this field shall be such that the entire record is octet-aligned.
35		
36		

#### 6.6.8.4 InitialSetupAttribute

This attribute shall be used to construct an ActiveSetAssignment locally at the access terminal. This ActiveSetAssignment shall be used between the times the protocol receives an *AccessChannelMAC.AccessGrantReceived* indication and an ActiveSetAssignment message. The access terminal should use default values for this attribute and the use of non-default values is not recommended.

Field	Length (bits)	Default
AttributeID	8	N/A
SingleServingSector	1	0
RPICHEEnabled	1	0
CQIReportingMode	2	00
VCQIReportInterval	4	0000
VCQIMeasureInterval	2	00
EnhancedPilotReportEnabled	1	0
EnhancedPilotReportRatio	4	0000
EnhancedPilotReportThreshold	4	1111
MinReportInterval	2	01
CQIReportInterval	2	00
CQIReportPhase	3	000
CQIPilotInterval	2	00
CQIPilotPhase	3	000
BFCHReportRate	4	0000
SFCHReportRate	4	0000
BFCHPowerOffset	3	000
NumBFCHBits	2	11
SFCHPowerOffset	3	000
NumSFCHBits	2	11
MandatoryCQICHCTRLReportingPeriod	3	111
PowerControlStepUp	3	111
PowerControlStepDown	3	111
RDCHGainMin	6	001000
RDCHGainMax	4	1000

AttributeID This field shall be set to 0x03

The remaining fields shall be interpreted as in the ActiveSetAssignment message.

### 6.6.8.5 IdleStateRegistrationTimeOut attribute

The default value of this attribute shall be 0x0000.

Attribute ID	Attribute	Values	Meaning
0x04	IdleStateRegistrationTimeOut	0x0000	Timer based registration disabled
		0x0001-0xffff	Timer based registration timeout in units of seconds

### 6.6.9 Protocol numeric constants

Constant	Meaning	Value
$N_{ASMPType}$	Type field for this protocol	Table 9
$N_{ASMPDefault}$	Subtype field for this protocol	0x0000
$N_{ASMPActive}$	Maximum size of the Active Set	8
$N_{ASMPCandidate}$	Maximum size of the Candidate Set	6
$N_{ASMPNeighbor}$	Minimum size of the Neighbor Set	20

### 6.6.10 Session state information

The Session State Information record (see 10.10) consists of parameter records.

This protocol defines the following parameter record in addition to the configuration attributes for this protocol.

### 6.6.10.1 ActiveSetManagement parameter

The following parameter shall be included in the Session State Information record only if the Session State Information is being transferred while the connection is open.

**Table 55 The Format of the parameter record for the ActiveSetManagement parameter**

Field	Length (bits)
ParameterType	8
Length	8
ASCMessagesSequence	8
ASAMessagesSequence	8
ASAMessageLength	16
ASAMessage	Variable
PRMessagesSequence	8
PRMessageLength	8
PRMessage	Variable
NumPilots	8
NumPilots occurrences of the following fields	
Reserved	4
PilotPN	12
SectorID	128

ParameterType	This field shall be set to 0x01 for this parameter record.
Length	This field shall be set to the length of this parameter record in units of octets excluding the Length field.
ASCMessagesSequence	This field shall be set to the MessageSequence field of the last ActiveSetComplete message received by the access network.
ASAMessagesSequence	This field shall be set to the MessageSequence field of the last ActiveSetAssignment message that was sent by the source access network.
ASAMessageLength	This field shall be set to the length of the last ActiveSetAssignment message that was sent by the source access network.
ASAMessage	Last ActiveSetAssignment message that was sent by the source access network.
PRMessagesSequence	This field shall be set to the MessageSequence field of the last PilotReport message that was received by the source access network.
PRMessageLength	This field shall be set to the length of the last PilotReport message that was received by the source access network.

1	PRMessage	This field shall be set to the last PilotReport message that was received by
2		the source access network.
3	NumPilots	This field shall be set to the NumPilots field in the last ActiveSetAssignment
4		message that was sent by the source access network.
5	Reserved	This field shall be set to '0000'.
6	PilotPN	This field shall be set to the corresponding PilotPN field in the last
7		ActiveSetAssignment message that was sent by the source access network.
8	SectorID	This field shall be set to the SectorID corresponding to the sector associated
9		with the PilotPN specified above.

## 10 6.7 Default Initialization State Protocol

### 11 6.7.1 Overview

12 The Default Initialization State Protocol provides the procedures and messages required for an access  
13 terminal to acquire a serving network. This protocol imposes the following two requirements on an  
14 access terminal

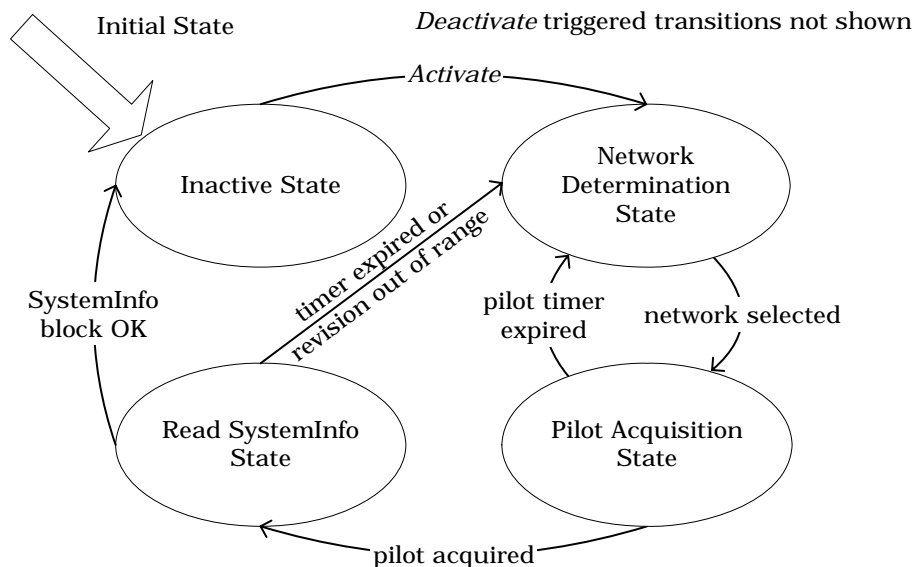
- 15 ■ The access terminal follows channel band information provided in a Redirect message of  
16 the Air Link Management Protocol.
- 17 ■ Prevents the access terminal from attempting to connect to access networks with out of  
18 range Revision number (as defined through MaximumRevision and MinimumRevision).

19 At the access network, this protocol does not define any states.

20 At the access terminal, this protocol operates in one of the following four states:

- 21 ■ *Inactive State*: In this state the protocol waits for an *Activate* command.
- 22 ■ *Network Determination State*: In this state the access terminal chooses an access network  
23 on which to operate.
- 24 ■ *Pilot Acquisition State*: In this state the access terminal acquires a Forward Pilot Channel.
- 25 ■ *Read SystemInfo State*: In this state the access terminal reads the SystemInfo block and  
26 determines if the MaximumRevision and MinimumRevision fields in the SystemInfo  
27 block can be supported by the access terminal.

1 Protocol states and events causing transition between states are shown in Figure 56.



2

3 **Figure 56 Default Initialization State Protocol state diagram (access terminal)**

#### 4 **6.7.2 Primitives and public data**

##### 5 **6.7.2.1 Commands**

6 This protocol defines the following commands:

- 7 ■ *Activate* (an optional ChannelBand record can be specified with the command)
- 8 ■ *Deactivate*

##### 9 **6.7.2.2 Return indications**

10 This protocol returns the following indications:

- 11 ■ *NetworkAcquired*

#### 12 **6.7.3 Public data**

##### 13 **6.7.3.1 Static public data**

- 14 ■ Selected ChannelBand
- 15 ■ System time
- 16 ■ PilotPN of the selected sector
- 17 ■ The following fields of the SystemInfo block:
  - 18 □ MaximumRevision
  - 19 □ MinimumRevision



### 1 **6.7.3.2 Dynamic public data**

- 2 ■ Subtype for this protocol

## 3 **6.7.4 Protocol initialization and swap procedures**

### 4 **6.7.4.1 Protocol initialization**

5 Upon initialization at the access terminal:

- 6 ■ The value for each attribute for this protocol instance shall be set to the default value for  
7 that attribute.
- 8 ■ The protocol shall enter the Inactive State.

9 Upon initialization at the access network:

- 10 ■ The value for each attribute for this protocol instance shall be set to the default value for  
11 that attribute.

### 12 **6.7.4.2 Protocol swap**

13 Upon swap at the access terminal, the protocol shall enter the Inactive State.

14 The access network shall define an empty swap procedure.

## 15 **6.7.5 Procedures**

16 The access network need not keep state for this protocol.

### 17 **6.7.5.1 Command processing**

18 The access network shall ignore all commands.

#### 19 **6.7.5.1.1 Activate**

20 If the protocol receives an *Activate* command in the Inactive State, the access terminal shall transition  
21 to the Network Determination State.

22 If the protocol receives this command in any other state, the access terminal shall ignore it.

#### 23 **6.7.5.1.2 Deactivate**

24 If the protocol receives a *Deactivate* command in the Inactive State, the access terminal shall  
25 ignore it.

26 If the protocol receives this command in any other state, the access terminal shall transition to the  
27 Inactive State.

### 28 **6.7.5.2 Inactive state**

29 In the Inactive State the access terminal waits for the protocol to receive an *Activate* command.

### 1 **6.7.5.3 Network determination state**

2 In the Network Determination State, the access terminal selects an OFDMA ChannelBand (see 10.1)  
3 on which to try and acquire the access network.

4 If a ChannelBand record was provided with the *Activate* command, the access terminal should select  
5 the system and channel specified by the record. Such a record may be provided by the access network  
6 when the Initialization State Protocol is activated due to a redirect message received by the Air Link  
7 Management Protocol.

8 The specific mechanisms to provision the access terminal with a list of preferred networks and with  
9 the actual algorithm used for network selection are beyond the scope of this specification.

10 Upon selecting a OFDMA Channel the access terminal shall enter the Pilot Acquisition State.

### 11 **6.7.5.4 Pilot acquisition state**

12 In the Pilot Acquisition State, the access terminal acquires the F-ACQCH of the selected OFDMA  
13 ChannelBand.

14 Upon entering the Pilot Acquisition State, the access terminal shall tune to the selected OFDMA  
15 ChannelBand and shall search for the pilot. If the access terminal acquires the pilot, it shall enter the  
16 Read SystemInfo State.<sup>36</sup> If the access terminal fails to acquire the pilot within  $T_{ISPPilotAcq}$  seconds of  
17 entering the Pilot Acquisition State, it shall enter the Network Determination State.

### 18 **6.7.5.5 Read SystemInfo state**

19 Upon entering this state, the access terminal shall issue the *ControlChannelMAC.Activate* and  
20 *OverheadMessages.Activate* commands.

21 If the access terminal fails to receive the SystemInfo block within  $T_{ISPSyncAcq}$  seconds of entering the  
22 Synchronization State, the access terminal shall issue *ControlChannelMAC.Deactivate* and  
23 *OverheadMessages.Deactivate* commands and shall enter the Network Determination State. While  
24 attempting to receive the SystemInfo block, the access terminal shall discard any other messages  
25 received on the Control Channel.

26 When the access terminal receives a SystemInfo block:

- 27 ■ If the access terminal's revision number is not in the range defined by the  
28 *MinimumRevision* and *MaximumRevision* fields (inclusive) specified in the message, the  
29 access terminal shall issue *ControlChannelMAC.Deactivate* and  
30 *OverheadMessages.Deactivate* commands and enter the Network Determination State.

---

<sup>36</sup> The Access Terminal Minimum Performance Requirements contains specifications regarding pilot acquisition performance.

- 1       ■ Otherwise, the access terminal shall:
  - 2           □ Set the access terminal time to the time specified in the SystemInfo block. The time
    - 3               specified in the block is the time at the beginning of the superframe where the
    - 4               message transmission started. Note that this time is accurate only to the twelve least
    - 5               significant digits, and is enough for the access terminal to read the
    - 6               ExtendedChannelInfo message (on the Forward Traffic Channel MAC) that contains
    - 7               the complete time.
  - 8           □ Return a *NetworkAcquired* indication.
  - 9           □ Enter the Inactive State.

## 10   6.7.6 Message formats

11   No messages are defined for this protocol.

## 12   6.7.7 Interface to other protocols

### 13   6.7.7.1 Commands

14   This protocol issues the following commands:

- 15       ■ *ControlChannelMAC.Activate*
- 16       ■ *ControlChannelMAC.Deactivate*
- 17       ■ *OverheadMessages.Activate*
- 18       ■ *OverheadMessages.Deactivate*

### 19   6.7.7.2 Indications

20   This protocol does not register to receive any indications.

## 21   6.7.8 Configuration attributes

22   No configuration attributes are defined for this protocol.

## 23   6.7.9 Protocol numeric constants

24   Constant	Meaning	Value	Comments
$N_{ISPT\text{Type}}$	Type field for this protocol	Table 9	
$N_{ISPD\text{Default}}$	Subtype field for this protocol	0x0000	
$T_{ISPP\text{PilotAcq}}$	Time to acquire pilot in access terminal	60 seconds	
$T_{ISPS\text{SyncAcq}}$	Time to acquire SystemInfo block in access terminal	5 seconds	

25

1 **6.7.10 Session state information**

2 The Session State Information record (see 10.10) consists of parameter records.

3 The parameter records for this protocol consist of the configuration attributes of this protocol.

4

## 7 Lower MAC Sublayer

### 7.1 Introduction

#### 7.1.1 General overview

The Lower MAC Sublayer contains rules for the formulation of Lower MAC Sublayer packets for transmission over physical channels and interpretation of Lower MAC Sublayer packets provided from Physical Layer channels. In particular, the Lower MAC Sublayer contains the rules governing the operation of the Forward Traffic Channel and the Reverse Traffic Channel. In addition to these data channels, the Lower MAC Sublayer specifies rules for controlling and processing physical layer signaling channels on both the forward and reverse links, as described below. This section presents the protocols for the Lower MAC Sublayer. Each of these protocols can be independently negotiated at the beginning of the session. A description of the various physical channels controlled by each protocol may be found in the Overview chapter of this specification.

The Lower MAC Sublayer contains the following protocols:

- *Control Channel MAC Protocol*: This protocol contains the rules governing the operation of transmissions over the following Physical Layer channels: the other-sector interference channel (F-OSICH), and the two primary broadcast channels (F-pBCH0 and F-pBCH1).
- *Access Channel MAC Protocol*: This protocol contains the rules governing access terminal (AT) transmission timing and power characteristics for the R-ACH physical layer channel.
- *Shared Signaling MAC Protocol*: This protocol contains the rules governing the operation of the F-SSCH physical layer channel. The protocol handles H-ARQ, power control, channel assignment, and access grant delivery.
- *Forward Traffic Channel MAC Protocol*: This protocol contains the rules governing the operation of the F-DCH physical layer channel.
- *Reverse Control Channel MAC Protocol*: This protocol contains the rules governing the operation of the R-REQCH, R-CQICH, R-BFCH, R-SFCH, R-ACKCH, and R-PICH physical layer channels.
- *Reverse Traffic Channel MAC Protocol*: This protocol contains the rules governing the operation of the R-DCH physical layer channel.

#### 7.1.2 Data encapsulation

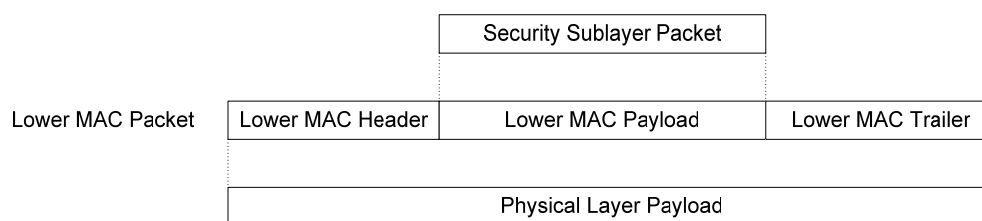
In the transmit direction at the access network, the Lower MAC Sublayer receives Security Sublayer packets, uses the Forward Traffic Channel MAC to add layer-related headers and trailers, and forwards the resulting packet to the Physical Layer for transmission on the F-DCH.

In the transmit direction at the access terminal, the Lower MAC Sublayer receives Security Sublayer packets, uses the Reverse Traffic Channel MAC to add layer-related headers and trailers, and forwards the resulting packet to the Physical Layer for transmission on the R-DCH.

1 In the receive direction at the access terminal, the Lower MAC Sublayer receives Lower MAC  
 2 packets from the Physical Layer F-DCH, uses the Forward Traffic Channel MAC to remove layer-  
 3 related headers and trailers, and forward the contained Security Sublayer packets to the Security  
 4 Sublayer.

5 In the receive direction at the access network, the Lower MAC Sublayer receives Lower MAC  
 6 packets from the Physical Layer R-DCH, uses the Reverse Traffic Channel MAC to remove layer-  
 7 related headers and trailers, and forwards the contained Security Sublayer packets to the Security  
 8 Sublayer.

9 Figure 57 illustrates the relationship between Security Sublayer packets, Lower MAC packets, and  
 10 Physical Layer packets. The content and formulation of Lower MAC headers and trailers is specified  
 11 for each Lower MAC Sublayer protocol in this chapter.



12  
 13 **Figure 57 Lower MAC Sublayer packet encapsulation**

### 14 7.1.3 Superframe timing

15 Forward and reverse link transmissions are divided into units of superframes. Superframes are further  
 16 divided into units of PHY Frames. Frequency Division Duplex (FDD) and Time Division Duplex  
 17 (TDD) superframe timing are described in the following sections.

18 Both forward and reverse link transmissions support H-ARQ. To provide time for the transmission  
 19 and reception of acknowledgements, PHY Frames on the forward and reverse links are grouped into  
 20 sets of PHY Frames called “interlaces,” and successive H-ARQ transmissions of MAC packets are  
 21 confined to a single interlace. The Lower MAC Sublayer chapter of this specification uses a PHY  
 22 Frame indexing scheme that is convenient for the descriptions herein, but may not be consistent with  
 23 indexing schemes used in other layers and sublayers in the specification. Refer to the FTC and RTC  
 24 MAC Protocols for the exact specification of H-ARQ operation.

#### 25 7.1.3.1 FDD

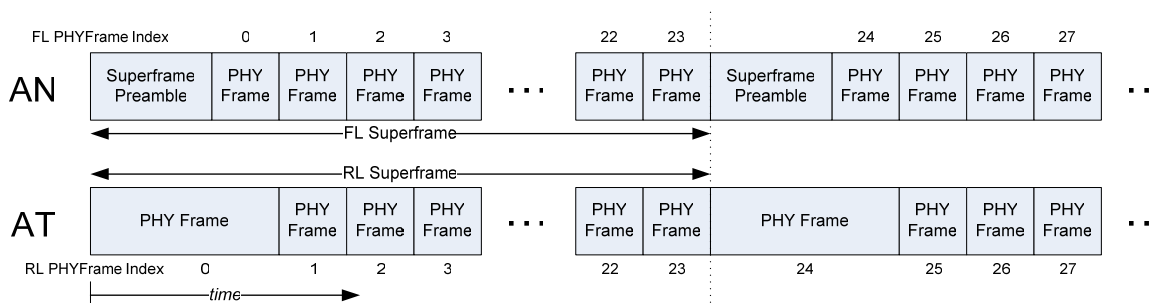
##### 26 7.1.3.1.1 Superframe timing

27 An FDD forward link superframe consists of a superframe preamble followed by  $N_{\text{FDD,FLPHYFrames}}$  FL  
 28 PHY Frames, and an FDD reverse link superframe consists of  $N_{\text{FDD,RLPHYFrames}}$  RL PHY Frames,  
 29 where  $N_{\text{FDD,FLPHYFrames}}$  represents the number of FDD FL PHY Frames per FL superframe and  
 30  $N_{\text{FDD,RLPHYFrames}}$  represents the number of FDD RL PHY Frames per RL superframe. These are  
 31 constants in the system and  $N_{\text{FDD,FLPHYFrames}} = N_{\text{FDD,RLPHYFrames}} = 24$ .

32 The first RL PHY Frame of each RL superframe is lengthened by the duration of the FL superframe  
 33 preamble to ensure superframe timing alignment between the forward link and reverse link.

- 1 Each superframe shall be uniquely identified by a superframe index that is incremented every  
 2 superframe.
- 3 ■ Each FL PHY Frame shall be uniquely identified by a FL PHY Frame Index. The FL  
 4 PHY Frame Index of the  $i^{\text{th}}$  FL PHY Frame,  $i = 0, 1, \dots, N_{\text{FDD,FLPHYFrames}}-1$ , in the  $j^{\text{th}}$   
 5 superframe,  $j = 0, 1, \dots$ , shall be given by  $j * N_{\text{FDD,FLPHYFrames}} + i$ .
  - 6 ■ Each RL PHY Frame shall be uniquely identified by a RL PHY Frame Index. The RL  
 7 PHY Frame Index of the  $i^{\text{th}}$  RL PHY Frame,  $i = 0, 1, \dots, N_{\text{FDD,RLPHYFrames}}-1$ , in the  $j^{\text{th}}$   
 8 superframe,  $j = 0, 1, \dots$ , shall be given by  $j * N_{\text{FDD,RLPHYFrames}} + i$ .

9 Henceforth, a PHY Frame (FL or RL) with PHY Frame index  $k$  is referred to as (FL or RL) PHY  
 10 Frame  $k$ . The exact timing details are provided by the Physical Layer Specification. The structure of  
 11 FDD forward link and reverse link superframes with  $N_{\text{FDD,FLPHYFrames}} = 24$  and  $N_{\text{FDD,RLPHYFrames}} = 24$  is  
 12 shown in Figure 58.

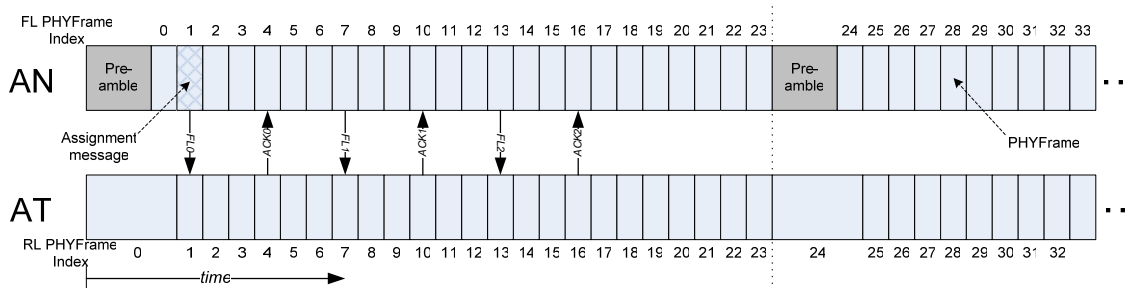


13  
 14 **Figure 58 FDD superframe timing**

15 **7.1.3.1.2 H-ARQ interlace structure**

16 There are six interlaces on both the forward and reverse links, and PHY Frame  $k$  is a member of  
 17 interlace  $k \bmod 6$ . Forward link assignments that arrive in FL PHY Frame  $k$  shall apply to the  
 18 interlace containing FL PHY Frame  $k$  and starting with FL PHY Frame  $k$ . A FL transmission of a  
 19 MAC packet on FL PHY Frame  $k$  is acknowledged on RL PHY Frame  $k+3$ . HARQ retransmissions  
 20 associated with the MAC packet that starts in PHY Frame  $k$  occur in PHY Frames  $k+6n$  where  $n$  is  
 21 the retransmission index,  $n=0,1, \dots$ .

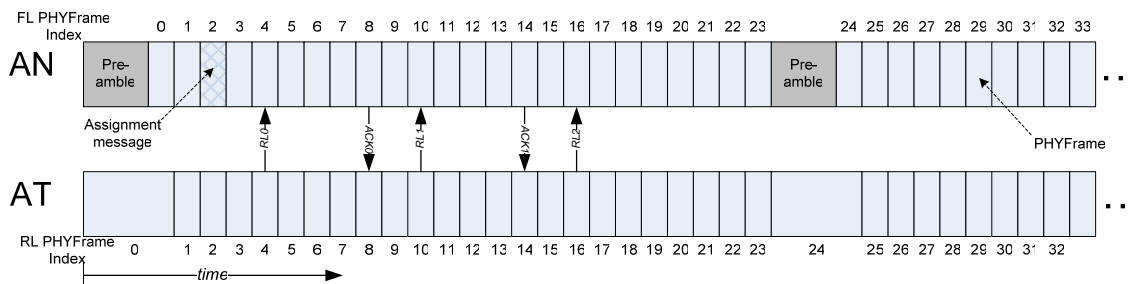
1 Figure 59 shows examples of the timing relationship between FL packet transmissions and the  
 2 associated acknowledgement transmissions for the FDD mode with  $N_{\text{FDD,FLPHYFrames}} = 24$  and  
 3  $N_{\text{FDD,RLPHYFrames}} = 24$ .



4  
5 **Figure 59 FDD FL H-ARQ interlace structure**

6 Reverse link assignments that arrive in FL PHY Frame  $k$  shall apply to the interlace containing RL  
 7 PHY Frame  $k+2$  and starting with RL PHY Frame  $k+2$ . A RL transmission of a MAC packet on RL  
 8 PHY Frame  $k$  is acknowledged on FL PHY Frame  $k+4$ . HARQ retransmissions associated with the  
 9 MAC packet that starts in PHY Frame  $k$  occur in PHY Frames  $k+6n$  where  $n$  is the retransmission  
 10 index,  $n=0,1, \dots$ .

11 Figure 60 shows the timing relationship between the RL packet transmissions and the associated  
 12 acknowledgment transmissions for the FDD mode with  $N_{\text{FDD,FLPHYFrames}} = 24$  and  $N_{\text{FDD,RLPHYFrames}} = 24$ .



13  
14 **Figure 60 FDD RL H-ARQ interlace structure**

### 15 7.1.3.1.3 H-ARQ interlace structure for Extended Transmission Duration Assignments

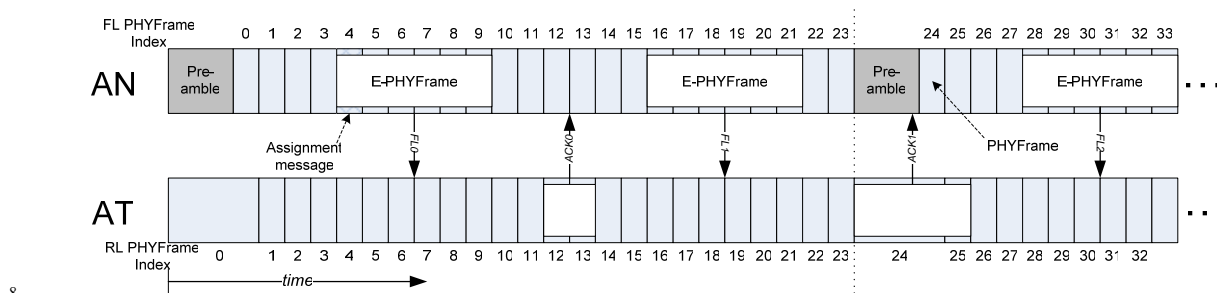
16 For FDD, in addition to the standard H-ARQ interlace structure, the system supports the use of  
 17 “Extended Transmission Duration” assignments. Such assignments utilize extended PHY Frames  
 18 (denoted E-PHY Frames, where E-PHY Frame  $k$  consists of PHY Frames  $k$  through  $k+5$ ) and alter  
 19 the timing of transmissions and corresponding ACK transmissions relative to the standard  
 20 assignments. Extended Transmission Duration assignments create a potential for resource assignment  
 21 collisions with standard assignments, and the AN should manage resource assignments to prevent  
 22 such collisions.

23 Forward link assignments that arrive in FL PHY Frame  $k$  shall apply to the E-PHY Frame and the  
 24 associated Extended Transmission Duration interlace starting with FL PHY Frame  $k$ . Note that with  
 25 Extended Transmission Duration assignments, the associated interlace can begin in any FL PHY  
 26 Frame. A FL transmission of a MAC packet using an Extended Transmission Duration assignment



1 that starts on FL PHY Frame  $k$  continues for six consecutive PHY Frames (an E-PHY Frame). The  
 2 E-PHY Frame is acknowledged using a combination of RL PHY Frames  $k+8$  and  $k+9$ . HARQ  
 3 retransmissions associated with the MAC packet that starts in FL PHY Frame  $k$  occur in FL PHY  
 4 Frames  $k+12n$  where  $n$  is the retransmission index.

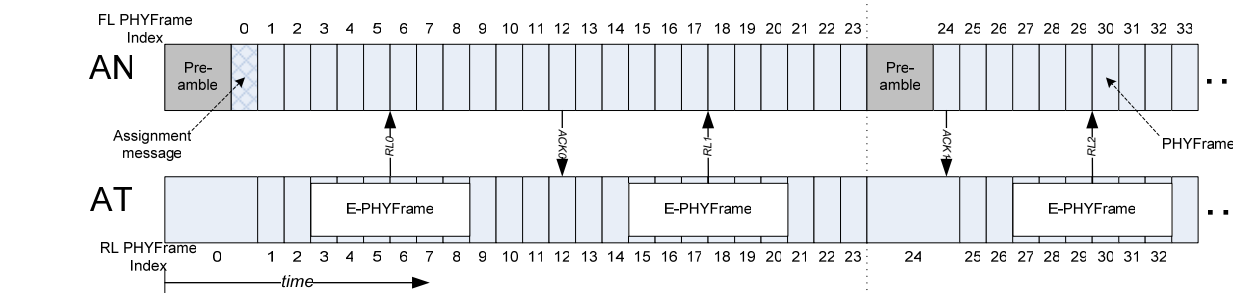
5 Figure 61 shows examples of the timing relationship between FL packet transmissions and the  
 6 associated acknowledgement transmissions for the FDD mode with  $N_{FDD,FLPHYFrames} = 24$  and  
 7  $N_{FDD,RLPHYFrames} = 24$ .



9 **Figure 61 FDD FL H-ARQ interlace structure for Extended Transmission Duration**  
 10 **Assignments**

11 Reverse link assignments that arrive in FL PHY Frame  $k$  shall apply to the E-PHY Frame and the  
 12 associated Extended Transmission Duration interlace starting with RL PHY Frame  $k+3$ . A RL  
 13 transmission of a MAC packet using an Extended Transmission Duration assignment that starts on  
 14 RL PHY Frame  $k$  continues for six consecutive PHY Frames (an E-PHY Frame). The E-PHY Frame  
 15 is acknowledged on FL PHY Frame  $k+9$ . HARQ retransmissions associated with the MAC packet  
 16 that starts in PHY Frame  $k$  occur in PHY Frames  $k+12n$  where  $n$  is the retransmission index.

17 Figure 62 shows examples of the timing relationship between RL packet transmissions and the  
 18 associated acknowledgement transmissions for the FDD mode with  $N_{FDD,FLPHYFrames} = 24$  and  
 19  $N_{FDD,RLPHYFrames} = 24$ .



21 **Figure 62 FDD RL H-ARQ interlace structure for Extended Transmission Duration**  
 22 **Assignments**

### 7.1.3.2 TDD

TDD mode has two associated variables called  $N_{FL\_BURST}$  and  $N_{RL\_BURST}$  (respectively given by the  $N\_FLBurst$  and  $N\_RLBurst$  parameters which are part of the public data of the Overhead Messages Protocol) that determine FL and RL partitioning. Specifically, the  $N_{FL\_BURST}:N_{RL\_BURST}$  partitioning refers to a partitioning wherein  $N_{FL\_BURST}$  FL PHY Frames are transmitted on the FL followed by  $N_{RL\_BURST}$  RL PHY Frames on the RL and so on in time. Both  $N_{FL\_BURST}$  and  $N_{RL\_BURST}$  are limited to the range 1-4 and are specified by 2 bit fields (see Overhead Messages Protocol).

A TDD forward link superframe consists of a superframe preamble and  $N_{TDD,FLPHYFrames}$  FL PHY Frames, and a TDD reverse link superframe consists of  $N_{TDD,RLPHYFrames}$  RL PHY Frames where  $N_{TDD,FLPHYFrames}$  is the number of FL PHY Frames per FL superframe, and  $N_{TDD,RLPHYFrames}$  is the number of RL PHY Frames per RL superframe.  $N_{TDD,FLPHYFrames}$  and  $N_{TDD,RLPHYFrames}$  are computed as follows:

$$N_{TDD,FLPHYFrames} = N_{FL\_BURST} \left\lceil \frac{24}{N_{FL\_BURST} + N_{RL\_BURST}} \right\rceil$$

$$N_{TDD,RLPHYFrames} = N_{RL\_BURST} \left\lceil \frac{24}{N_{FL\_BURST} + N_{RL\_BURST}} \right\rceil$$

The HARQ structure for the special case  $N_{FL\_BURST} = N_{RL\_BURST} = 1$ , or the 1:1 partitioning is described first. Then the 2:1 partitioning, followed by a general  $N_{FL\_BURST}:N_{RL\_BURST}$  partitioning is described. Note that while the general partition description does reduce to the 1:1 partitioning with  $N_{FL\_BURST} = N_{RL\_BURST} = 1$ , the 2:1 partitioning does not follow from the general partitioning because some optimizations have been used in the 2:1 partitioning to reduce H-ARQ retransmission latency on the FL.

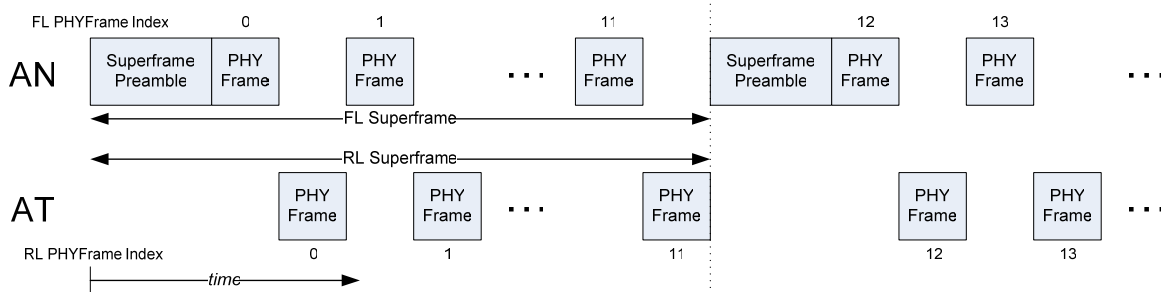
#### 7.1.3.2.1 Superframe timing for 1:1 partitioning

In the TDD mode, FL and RL transmissions alternate in time. Excluding the superframe preamble at the beginning of each FL superframe, the transmissions alternate between FL PHY Frames and RL PHY Frames throughout the superframe. An appropriate amount of guard interval is inserted between each FL-RL transition.

Each superframe shall be uniquely identified by a superframe index that is incremented every superframe.

- Each FL PHY Frame shall be uniquely identified by a FL PHY Frame Index. The FL PHY Frame Index of the  $i^{\text{th}}$  FL PHY Frame,  $i = 0, 1, \dots, N_{TDD,FLPHYFrames}-1$ , in the  $j^{\text{th}}$  superframe,  $j = 0, 1, \dots$ , shall be given by  $j * N_{TDD,FLPHYFrames} + i$ .
- Each RL PHY Frame shall be uniquely identified by a RL PHY Frame Index. The RL PHY Frame Index of the  $i^{\text{th}}$  RL PHY Frame,  $i = 0, 1, \dots, N_{TDD,RLPHYFrames}-1$ , in the  $j^{\text{th}}$  superframe,  $j = 0, 1, \dots$ , shall be given by  $j * N_{TDD,RLPHYFrames} + i$ .

Henceforth, a PHY Frame (FL or RL) with PHY Frame index  $k$  is referred to as (FL or RL) PHY Frame  $k$ . The exact timing details are provided by the Physical Layer specification. The structure of the TDD forward link and reverse link superframes with  $N_{TDD,FLPHYFrames} = 12$  and  $N_{TDD,RLPHYFrames} = 12$  is shown in Figure 63.

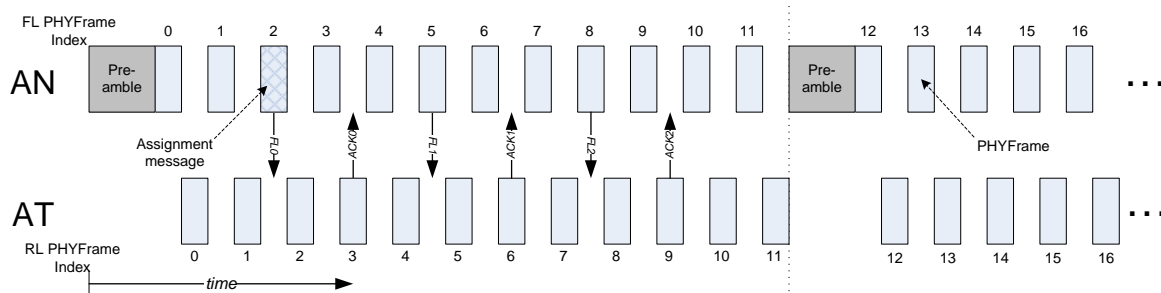


**Figure 63 TDD superframe timing for 1:1 partitioning**

**7.1.3.2.2 H-ARQ interlace structure for 1:1 partitioning**

There are three interlaces on both the forward and reverse links. Forward link assignments that arrive in FL PHY Frame  $k$  shall apply to the interlace containing FL PHY Frame  $k$  and starting with FL PHY Frame  $k$ . A FL transmission of a MAC packet on FL PHY Frame  $k$  is acknowledged on RL PHY Frame  $k+1$ . HARQ retransmissions associated with the MAC packet that starts in PHY Frame  $k$  occur in PHY Frames  $k+3n$  where  $n$  is the retransmission index,  $n=0,1, \dots$ .

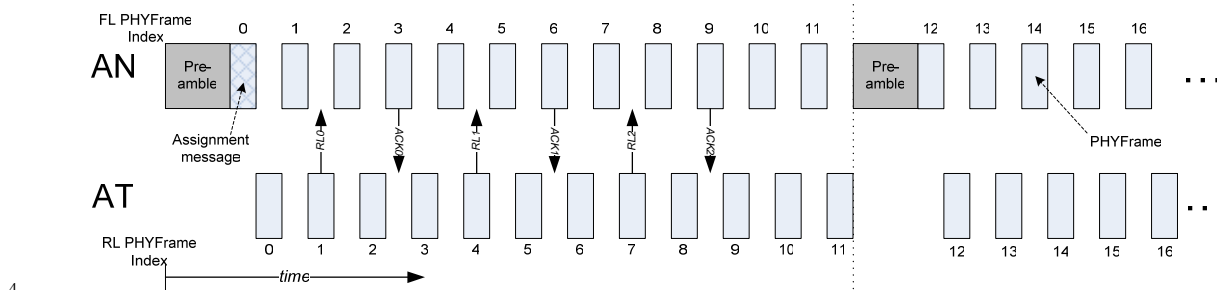
Figure 64 shows the timing relationship between the FL packet transmissions and the associated acknowledgment transmissions for the TDD mode with  $N_{TDD,FLPHYFrames} = 12$  and  $N_{TDD,RLPHYFrames} = 12$ .



**Figure 64 TDD FL H-ARQ interlace structure for 1:1 partitioning**

Reverse link assignments that arrive in FL PHY Frame  $k$  shall apply to the interlace containing RL PHY Frame  $k+1$  and starting with RL PHY Frame  $k+1$ . A RL transmission of a MAC packet on RL PHY Frame  $k$  is acknowledged on FL PHY Frame  $k+2$ . HARQ retransmissions associated with the MAC packet that starts in PHY Frame  $k$  occur in PHY Frames  $k+3n$  where  $n$  is the retransmission index.

1 Figure 65 shows the timing relationship between the RL packet transmissions and the associated  
 2 acknowledgment transmissions for the TDD mode with  $N_{TDD,FLPHYFrames} = 12$  and  $N_{TDD,RLPHYFrames} =$   
 3 12.



4  
5 **Figure 65 TDD RL H-ARQ interlace structure for 1:1 partitioning**

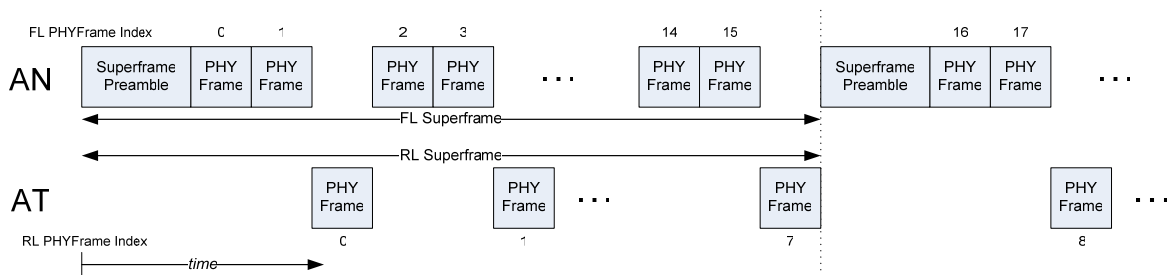
6 **7.1.3.2.3 Superframe timing for 2:1 partitioning**

7 In the TDD 2:1 mode, FL and RL transmissions alternate in time, with 2 FL PHY Frames followed by  
 8 1 RL PHY Frame in time. Excluding the superframe preamble at the beginning of each FL  
 9 superframe, the transmissions alternate between FL PHY Frames and RL PHY Frames throughout the  
 10 superframe. An appropriate amount of guard interval is inserted between each FL-RL transition.

11 Each superframe shall be uniquely identified by a superframe index that is incremented every  
 12 superframe.

- 13 ■ Each FL PHY Frame shall be uniquely identified by a FL PHY Frame Index. The FL  
 14 PHY Frame Index of the  $i^{th}$  FL PHY Frame,  $i = 0, 1, \dots, N_{TDD,FLPHYFrames}-1$ , in the  $j^{th}$   
 15 superframe,  $j = 0, 1, \dots$ , shall be given by  $j * N_{TDD,FLPHYFrames} + i$ .
- 16 ■ Each RL PHY Frame shall be uniquely identified by a RL PHY Frame Index. The RL  
 17 PHY Frame Index of the  $i^{th}$  RL PHY Frame,  $i = 0, 1, \dots, N_{TDD,RLPHYFrames}-1$ , in the  $j^{th}$   
 18 superframe,  $j = 0, 1, \dots$ , shall be given by  $j * N_{TDD,RLPHYFrames} + i$ .

19 Henceforth, a PHY Frame (FL or RL) with PHY Frame index  $k$  is referred to as (FL or RL) PHY  
 20 Frame  $k$ . The exact timing details are provided by the Physical Layer specification. The structure of  
 21 the TDD forward link and reverse link superframes with  $N_{TDD,FLPHYFrames} = 16$  and  $N_{TDD,RLPHYFrames} = 8$   
 22 is shown in Figure 66.

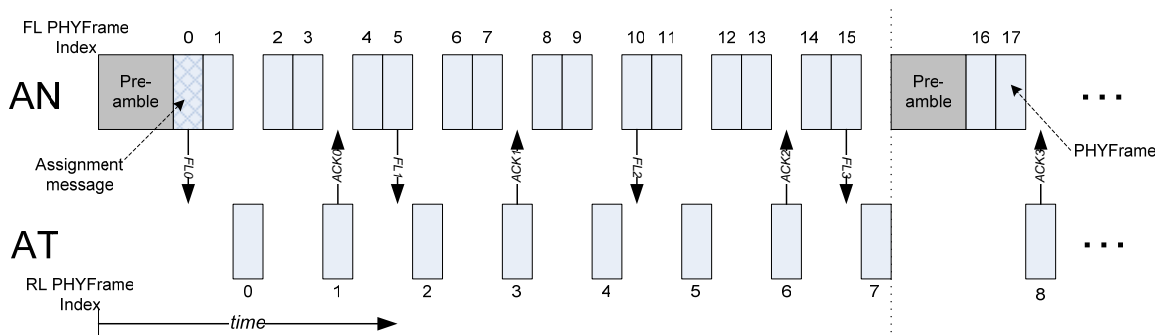


23  
24 **Figure 66 TDD superframe timing for 2:1 partitioning**

1 **7.1.3.2.4 H-ARQ interlace structure for 2:1 partitioning**

2 There are five interlaces on the forward and two interlaces on the reverse link. Forward link  
 3 assignments that arrive in FL PHY Frame  $k$  shall apply to the interlace containing FL PHY Frame  $k$   
 4 and starting with FL PHY Frame  $k$ . A FL transmission of a MAC packet FL PHY Frame  $k$  is  
 5 acknowledged in RL PHY Frame  $\lfloor k/2 \rfloor + 1$ . HARQ retransmissions associated with the MAC packet  
 6 that starts in PHY Frame  $k$  occur in PHY Frames  $k+5n$  where  $n$  is the retransmission index,  
 7  $n=0,1, \dots$

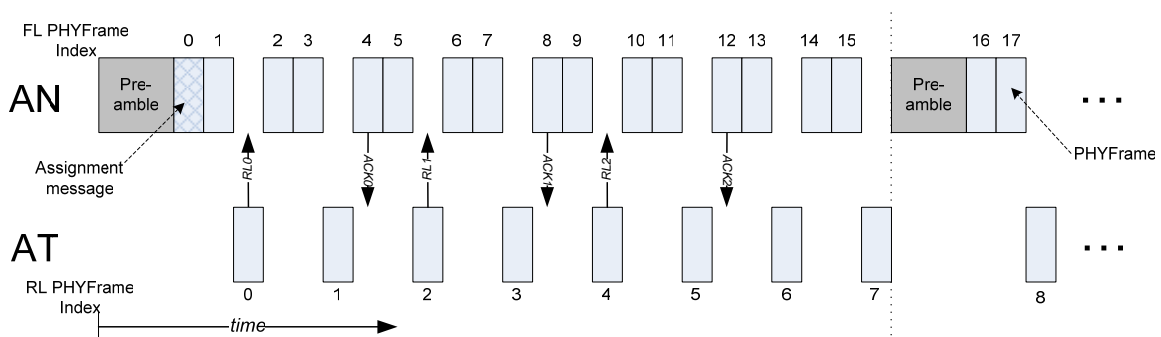
8 Figure 67 shows the timing relationship between the FL packet transmissions and the associated  
 9 acknowledgment transmissions for the TDD mode with  $N_{TDD,FLPHYFrames} = 16$  and  $N_{TDD,RLPHYFrames} = 8$ .



11 **Figure 67 TDD FL H-ARQ interlace structure for 2:1 partitioning**

12 Reverse link assignments that arrive in FL PHY Frame  $k$  shall apply to the interlace containing RL  
 13 PHY Frame  $\lfloor k/2 \rfloor$  and starting with the same RL PHY Frame. Note that due to assignment  
 14 demodulation requirements at the AT, Assignment messages are only sent in FL PHY Frames with an  
 15 even index. A RL transmission of a MAC packet on RL PHY Frame  $j$  is acknowledged on FL PHY  
 16 Frame  $2j + 4$ . HARQ retransmissions associated with the MAC packet that starts in PHY Frame  $k$   
 17 occur in PHY Frames  $k+2n$  where  $n$  is the retransmission index.

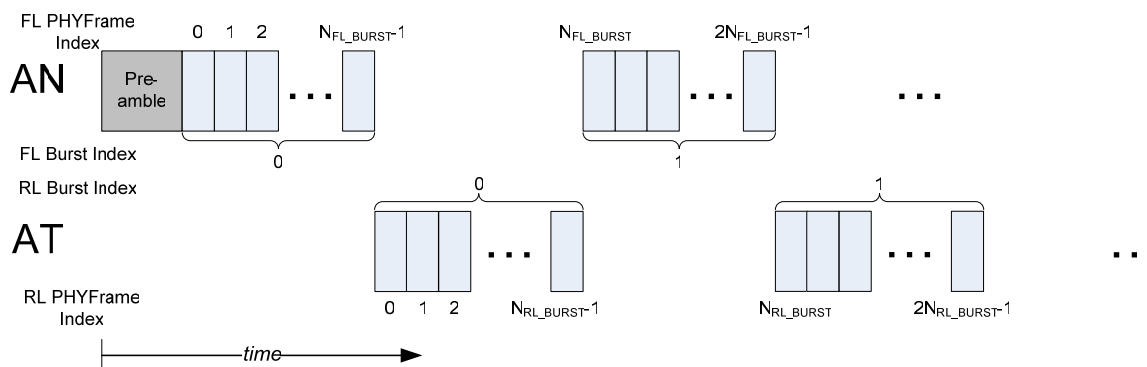
18 Figure 68 shows the timing relationship between the RL packet transmissions and the associated  
 19 acknowledgment transmissions for the TDD mode with  $N_{TDD,FLPHYFrames} = 16$  and  $N_{TDD,RLPHYFrames} = 8$ .



21 **Figure 68 TDD RL H-ARQ interlace structure for 2:1 partitioning**

### 7.1.3.2.5 Superframe timing and interlace structure for Generalized TDD partitioning

In addition to the 1:1 and 2:1 FL/RL TDD partitioning structures, the system supports a general TDD partitioning.<sup>37</sup> For convenience, let a FL burst be a set of  $N_{FL\_BURST}$  PHY Frames that are transmitted continuously in time on the FL, and a RL burst be a set of  $N_{RL\_BURST}$  PHY Frames that are transmitted continuously in time on the RL resulting in a  $N_{FL\_BURST}:N_{RL\_BURST}$  partitioning. FL and RL bursts then alternate in time, as shown in Figure 69. Let the “burst index” be an index of bursts, for example, FL Burst  $i$  consists of FL PHY Frames  $i*N_{FL\_BURST}$  through  $(i+1)N_{FL\_BURST} - 1$ .



**Figure 69 TDD Superframe Structure for other partitionings**

Assignment, acknowledgement, and retransmission timing can then be specified using the following formulas, where  $i$  is the FL Burst Index, and  $j$  is the RL Burst index.

Assignments:

FL PHY Frames in  $i$  are assigned in FL burst  $i$ .

RL PHY Frames in  $j$  are assigned in FL burst  $i=j$  if  $N_{FL\_BURST} > 1$  or  $i=j-1$  if  $N_{FL\_BURST} = 1$ .

Note that for RL assignments, if  $N_{FL\_BURST} \geq N_{RL\_BURST}$ , then an assignment in the  $k$ th PHY Frame of the FL burst applies to the interlace containing  $k$ th PHY Frame of the relevant RL burst. However, if  $N_{FL\_BURST} < N_{RL\_BURST}$ , the assignment will contain additional bits enabling the AT to determine the interlace of the assignment.

Acknowledgement:

FL PHY Frames in  $i$  are acknowledged in RL burst  $j=i+1$ .

RL PHY Frames in  $j$  are acknowledged in FL burst  $i=j+2$ .

See 7.6.6.3.11.2 for the details of the ACK channelization within the relevant burst.

Retransmissions:

FL PHY Frames in  $i$  are retransmitted in FL burst  $i+2$  if  $N_{RL\_BURST} > 1$  or  $i+3$  if  $N_{RL\_BURST} = 1$ .

RL PHY Frames in  $j$  are retransmitted in RL burst  $j+2$  if  $N_{FL\_BURST} > 1$  or  $j+3$  if  $N_{FL\_BURST} = 1$ .

Note that retransmissions occur in the same PHY Frame within a burst. Thus, if a transmission occurs in the second PHY Frame of the burst, then all retransmissions shall also occur in the second PHY Frame of the relevant subsequent bursts.

<sup>37</sup> Note that this generalized partitioning reduces to the 1:1 partitioning described above in section 7.1.3.2.1, but does not reduce to the 2:1 partitioning of section 7.1.3.2.3. The 2:1 partitioning has been optimized on the FL to minimize retransmission latency.

1 Resulting number of interlaces:

2  $2 * N_{FL\_BURST}$  FL interlaces if  $N_{RL\_BURST} > 1$  or  $3 * N_{FL\_BURST}$  FL interlaces if  $N_{RL\_BURST} = 1$ .

3  $2 * N_{RL\_BURST}$  RL interlaces if  $N_{FL\_BURST} > 1$  or  $3 * N_{RL\_BURST}$  RL interlaces if  $N_{FL\_BURST} = 1$ .

#### 4 7.1.4 Common definitions and terms

##### 5 7.1.4.1 Channel trees

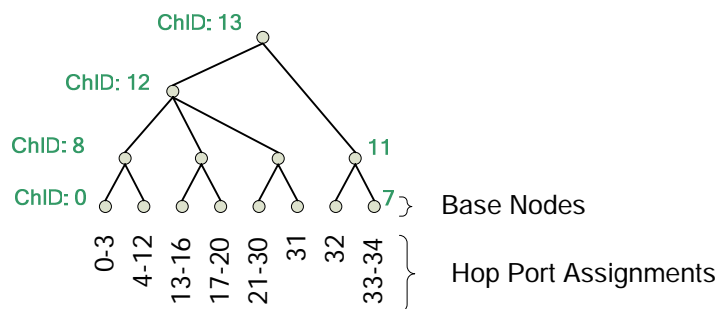
6 A channel tree is used to specify sets of hop-ports that are associated with each node identification  
7 number (NodeID). A set of hop-ports is said to be “mapped to a node” and a node “maps” a set of  
8 hop-ports. The following terms are used in association with channel trees in this specification:

- 9 ■ **Hop-port.** The fundamental unit of resource assignment. Each hop-port maps to one  
10 unique subcarrier. The mapping of hop-ports to subcarriers varies with time according to  
11 hopping rules specified in the Physical Layer specification.
- 12 ■ **Node.** A node corresponds to a single NodeID.
- 13 ■ **Children, Descendants.** Nodes that map a subset of the hop-ports mapped by a node.
- 14 ■ **Parents, Ancestors.** Nodes that map a superset of the hop-ports mapped by a node.
- 15 ■ **Base-nodes.** Nodes with no children. Base-nodes are assigned specific Physical Layer  
16 resources – in this case, hop-ports.
- 17 ■ **MinHopPortsPerNode.** The minimum number of hop-ports mapped to a node.

18 The indices of hop-ports follows a convention used throughout the Lower MAC Sublayer. Namely,  
19 hop-port indices are allocated sequentially starting from index 0. The base nodes are ordered in a left-  
20 to-right ordering as drawn, and hop-port indices are allocated sequentially starting from index 0 to the  
21 base nodes from left to right.

22 For example, the channel tree in Figure 70 has 14 nodes, numbered from 0 to 13. The base-nodes are  
23 NodeIDs 0 – 7. There are a total of 35 hop-ports available. Consider the node associated with NodeID  
24 10. This node has parent NodeID 12 and children NodeIDs 4 and 5. The node maps 11 hop-ports,  
25 namely hop-ports 21-31.

26 Because nodes define orthogonal assignments, the use of a node in the tree can restrict use of other  
27 nodes. Thus, if a node is in use, then all descendants and ancestors of the node are unavailable for use  
28 and are called “restricted” nodes.



30 **Figure 70 Example channel tree**

### 1 7.1.4.2 Power density

2 The Lower MAC Layer computes the power settings of Physical Layer channels and passes these  
3 settings to the Physical Layer for use. In some cases, “power density” is used to communicate  
4 transmission power levels, as defined below:

5 **Power density**            The expected energy per modulation symbol<sup>38</sup>

6 where “modulation symbol” refers to frequency domain (pre-IFFT) symbols, as defined in the  
7 Physical Layer specification. The “expected” power is used to cover the case of transmissions using  
8 non-constant modulus signal constellations. In all cases, the density shall be computed according to  
9 the parameters of the modulation symbols in the particular channel being specified.

## 10 7.2 Default Control Channel MAC Protocol

### 11 7.2.1 Overview

12 The Default Control Channel MAC Protocol provides the procedures and messages required for an  
13 access network (AN) to transmit and for an access terminal to receive the Control Channel. The  
14 protocol controls transmissions over the following Physical Layer channels: the other- sector  
15 interference channel (OSICH), and the primary broadcast channels (pBCH0 and pBCH1).

16 This specification assumes that the access network has one instance of this protocol for each sector in  
17 the network.

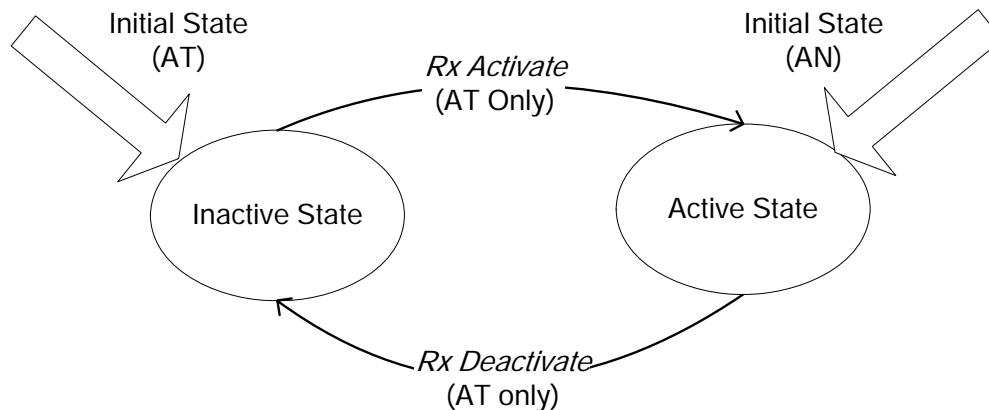
18 This protocol can be in one of two states:

- 19 ■ *Inactive State*: In this state, the protocol waits for an *Activate* command. This state  
20 applies only to the access terminal and occurs when the access terminal has not acquired  
21 an access network or is not monitoring the Control Channel.
- 22 ■ *Active State*: In this state, the access network transmits and the access terminal receives  
23 the Control Channel.

---

<sup>38</sup> Note that this definition of power density is not true in the strictest sense because of cyclic prefix and windowing overheads. However, it provides a convenient method of communicating power levels to the Physical Layer Protocol. The construction of transmitted waveforms as a function of power densities is specified by the Physical Layer Protocol.





1  
2 **Figure 71 Default Control Channel MAC Protocol state diagram**

3 **7.2.2 Primitives**

4 **7.2.2.1 Commands**

5 This protocol defines the following commands:

- 6     ■ *Activate*  
7     ■ *Deactivate*

8 **7.2.2.2 Return indications**

9 This protocol returns the following indications:

- 10     ■ *SupervisionFailed*  
11     ■ *PageReceived*  
12     ■ *QuickPageReceived*

13 **7.2.3 Public data**

14 **7.2.3.1 Static public data**

15 This protocol does not define any static public data.

16 **7.2.3.2 Dynamic public data**

- 17     ■ Subtype for this protocol.

18 **7.2.4 Protocol data unit**

19 The transmission unit of this protocol is defined separately for the three physical channels used by the  
20 protocol.(pBCH0, pBCH1, and OSICH), as defined in 7.2.6.1, 7.2.6.2, and 7.2.6.3.

## 7.2.5 Protocol initialization and swap

### 7.2.5.1 Protocol initialization

Upon initialization at the access terminal,

- The values of the attributes for this protocol instance shall be set to the default values specified for each attribute.
- The protocol shall enter the Inactive State.

Upon initialization at the access network,

- The values of the attributes for this protocol instance shall be set to the default values specified for each attribute.
- The protocol shall enter the Active State.

### 7.2.5.2 Protocol swap

Upon swap at the access terminal, the protocol instance shall enter the Inactive State.

Upon swap at the access network, the protocol instance shall enter the Active State.

## 7.2.6 Procedures

The following sections specify procedures for transmission and reception of the F-pBCH0, F-pBCH1, and F-OSICH Physical Layer channels.

### 7.2.6.1 Procedures for transmission and reception of the F-pBCH1 Physical Layer channel

If multi-carrier mode is MultiCarrierOn, this protocol shall deliver a separate payload for transmission on each carrier. An F-pBCH1 channel transmission consists of the following payload:

Field	Length (bits)
LoadControl	2
DataCtrlOffset	4
ACKCtrlOffset	4
BlockType0	2
QuickPage block	0 or $N_{QP\_BLK}$
QuickChannelInfo block	0 or as specified by Overhead Messages Protocol
BlockType1	0 or 2
QuickPage block	0 or $N_{QP\_BLK}$
QuickChannelInfo block	0 or as specified by Overhead Messages Protocol
BlockType2	0 or 2

Field	Length (bits)
QuickPage block	0 or $N_{QP\_BLK}$
QuickChannelInfo block	0 or as specified by Overhead Messages Protocol
BlockType3	0 or 2
QuickPage block	0 or $N_{QP\_BLK}$
QuickChannelInfo block	0 or as specified by Overhead Messages Protocol
Reserved	Variable

- 1 LoadControl This field shall determine the class of access terminals that are allowed to  
2 make an access attempt in the superframe.
- 3 DataCtrlOffset This field shall determine the offset between the power spectral densities of  
4 the Reverse CDMA Control Segment and the Reverse Traffic Channel in  
5 units of 0.5 dB
- 6 ACKCtrlOffset This field shall determine the offset between the power spectral densities of  
7 the Reverse CDMA Control Segment and the Reverse ACK Channel in units  
8 of 0.5 dB
- 9 BlockType<sub>j</sub> This field shall indicate the type of payload carried by the transmission. The  
10 block types '00' and '11' are reserved.
- 11 QuickPage block This field shall be included if BlockType preceding it is set to '01'. The  
12 transmitter shall construct the QuickPage block in accordance with the  
13 reception procedure 7.2.6.1.1.
- 14 QuickChannelInfo block This field shall be included if BlockType preceding it is set to '10'. The  
15 transmitter shall obtain the QuickChannelInfo block from the Overhead  
16 Messages Protocol.  
17
- 18 Reserved The transmitter shall set the number of reserved bits to zero so that the packet  
19 given to the physical layer is  $N_{CCMPpBCH1Bits}$  bits in length. The receiver shall  
20 ignore this field.

21 The fields above obey the following rules:

- 22 ■ BlockType<sub>j</sub> shall take the value '01' if it is followed by a QuickPage block.
- 23 ■ BlockType<sub>j</sub> shall take the value '10' if it is followed by a QuickChannelInfo block.
- 24 ■ The number of BlockType<sub>j</sub> fields shall be in accordance with  $N_{CCMPpBCH1Bits}$ .
- 25 ■ If a QuickChannelInfo block is included, it shall be the last block included.

1 The above rule makes it possible to carry, for example, two QuickPage and one QuickChannelInfo  
 2 block in the same pBCH1 transmission. Note that the actual number of blocks that can be transmitted  
 3 depends on the  $N_{CCMPpBCH1Bits}$ .

4 Also note that the above rules disallow more than one QuickChannelInfo blocks in the same pBCH1  
 5 transmission.

6 The transmitter shall set one of the BlockType fields to '10' in superframes with an even superframe  
 7 index. The transmitter shall set at least one of the BlockType to '01' in superframes with an odd  
 8 superframe index. At the transmit side, the protocol shall deliver the packet for transmission on F-  
 9 pBCH1.

10 The receiver shall use 7.2.6.1.1 to parse the received QuickPage block. The receiver shall forward the  
 11 received QuickChannelInfo block and reserved bits to the Overhead Messages Protocol.

### 12 7.2.6.1.1 Procedure for reception of the QuickPage block

13 The access terminal shall declare the QuickPage block to be in error if there is an error on the  
 14 F-pBCH1 channel that contains a QuickPage block (this error information may be used by the Idle  
 15 State Protocol). The access terminal shall process only those QuickPage blocks that occur in  
 16 superframes with superframe number contained in the PageTimes array that is public data of the Idle  
 17 State Protocol. The access terminal shall ignore QuickPage blocks in other superframes.

18 The QuickPage block may contain multiple QuickPages or a single Page (as decided by the access  
 19 network). The access terminal shall process each received QuickPage block as described below.

20 The format of the QuickPage block shall depend on the first three bits (NumPages field) in the block.  
 21 NumPages is the number of pages or quick pages in the QuickPage block. The access terminal  
 22 performs the following actions upon receiving a QuickPage block.

- 23 ■ If NumPages=0 the QuickPage block shall be ignored.
- 24 ■ If NumPages=1 and the ATI in the QuickPage block matches the ReceiveATIList of the  
 25 Address Management Protocol, the protocol shall issue a *PageReceived* indication.
- 26 ■ If  $1 < \text{NumPages} \leq 6$ , and the least significant bits of RQuickPage that is public data of the  
 27 Idle State Protocol match one of the RQuickPage $_i$  fields in the QuickPage block, the  
 28 protocol shall generate a *QuickPageReceived* indication.
- 29 ■ If NumPages=7, the protocol shall generate a *QuickPageReceived* indication if one of the  
 30 following conditions hold:
  - 31 □ The least significant bits of RQuickPage match one of the RQuickPage $_i$  fields in the  
 32 QuickPage block.
  - 33 □ RQuickPage7 is not equal to RQuickPage8, and the 4 least significant bits of  
 34 RQuickPage are larger than RQuickPage8.

35 **Table 56 QuickPage format with NumPages=0**

Field	Length (bits)	Value
NumPages	3	0x0
Reserved	$N_{QP\_BLK} - 3$	

36

**Table 57 QuickPage format with NumPages=1**

Field	Length (bits)	Value
NumPages	3	0x1
ATI	32	ATI of paged terminal
Reserved	$N_{QP\_BLK} - 35$	

**Table 58 QuickPage format with NumPages=2**

Field	Length (bits)	Value
NumPages	3	0x2
RQuickPage1	16	RQuickPageLSBs
RQuickPage2	16	RQuickPageLSBs
Reserved	$N_{QP\_BLK} - 35$	

**Table 59 QuickPage format with NumPages=3**

Field	Length (bits)	Value
NumPages	3	0x3
RQuickPage1	10	RQuickPageLSBs
RQuickPage2	10	RQuickPageLSBs
RQuickPage3	10	RQuickPageLSBs
Reserved	$N_{QP\_BLK} - 33$	

**Table 60 QuickPage format with NumPages=4**

Field	Length (bits)	Value
NumPages	3	0x4
RQuickPage1	8	RQuickPageLSBs
RQuickPage2	8	RQuickPageLSBs
RQuickPage3	8	RQuickPageLSBs
RQuickPage4	8	RQuickPageLSBs
Reserved	$N_{QP\_BLK} - 35$	

**Table 61 QuickPage format with NumPages=5**

Field	Length (bits)	Value
NumPages	3	0x5
RQuickPage1	6	RQuickPageLSBs
RQuickPage2	6	RQuickPageLSBs
RQuickPage3	6	RQuickPageLSBs
RQuickPage4	6	RQuickPageLSBs
RQuickPage5	6	RQuickPageLSBs
Reserved	$N_{QP\_BLK} - 33$	

**Table 62 QuickPage format with NumPages=6**

Field	Length (bits)	Value
NumPages	3	0x6
RQuickPage1	5	RQuickPageLSBs
RQuickPage2	5	RQuickPageLSBs
RQuickPage3	5	RQuickPageLSBs
RQuickPage4	5	RQuickPageLSBs
RQuickPage5	5	RQuickPageLSBs
RQuickPage6	5	RQuickPageLSBs
Reserved	$N_{QP\_BLK} - 33$	

**Table 63 QuickPage format with NumPages=7**

Field	Length (bits)	Value
NumPages	3	0x7
RQuickPage1	4	RQuickPageLSBs
RQuickPage2	4	RQuickPageLSBs
RQuickPage3	4	RQuickPageLSBs
RQuickPage4	4	RQuickPageLSBs
RQuickPage5	4	RQuickPageLSBs
RQuickPage6	4	RQuickPageLSBs
RQuickPage7	4	RQuickPageLSBs
RQuickPage8	4	RQuickPageLSBs
Reserved	$N_{QP\_BLK} - 35$	

### 7.2.6.2 Procedures for transmission and reception of the F-pBCH0 Physical Layer channel

An F-pBCH0 channel transmission has the following payload.

Field	Length (bits)
SystemInfo block	Specified by Overhead Messages Protocol
Reserved	Variable

**SystemInfo block** The transmitter shall obtain the SystemInfo block from the Overhead Messages Protocol.

**Reserved** The transmitter shall set the number of reserved bits to make the transmitted packet (including the CRC added by the Physical Layer) octet aligned. The receiver shall ignore this field.

At the transmit side, every  $N_{\text{pBCH0\_Period}}$  superframes, the protocol shall deliver the packet to the F-pBCH0 for transmission (where  $N_{\text{pBCH0\_Period}}$  is defined in the Physical Layer). If multi-carrier mode is MultiCarrierOn, then this protocol shall deliver a separate packet for each carrier.

At the receive side, the protocol shall forward the received payload to the Overhead Messages Protocol.

### 7.2.6.3 Procedures for transmission and reception of the F-OSICH Physical Layer channel

The Access Network shall forward OSI value to the Physical Layer for transmission over the F-OSICH Physical Layer channel. The computation of the OSI value is beyond the scope of this specification. If multi-carrier mode is MultiCarrierOn, then this protocol shall deliver a separate OSI value for each carrier. Note that in addition to the F-OSICH, the OSI values may be transmitted on the F-SSCH.

An F-OSICH channel transmission has the following payload.

Field	Length (bits)
OSIValue	2
Reserved	6

**OSIValue** The value shall indicate the level of interference at the sector. The value '11' is reserved.

**Reserved** The sender shall set this field to '000000'. The receiver shall ignore this field.

The Access Terminal shall monitor the OSICH channels of each sector in the OSIMonitorSet (see 7.7.6.4.1.6.1) and provide the received values to the Reverse Traffic Channel MAC Protocol.

## 1 7.2.6.4 Command processing

### 2 7.2.6.4.1 Activate

3 If this protocol receives an *Activate* command in the Inactive State, the access terminal shall transition  
4 to the Active State

5 If this protocol receives this command in the Active State, the command shall be ignored.

### 6 7.2.6.4.2 Deactivate

7 If this protocol receives a *Deactivate* command in the Inactive State, the command shall be ignored.

8 If this protocol receives this command in the Active State, the access terminal shall transition to the  
9 Inactive State

## 10 7.2.6.5 Inactive state

11 This state applies only to the access terminal.

12 When the protocol is in the Inactive State, the access terminal waits for an *Activate* command.

## 13 7.2.6.6 Active state

14 In this state, the access network transmits, and the access terminal monitors, the channels associated  
15 with the Control Channel MAC.

### 16 7.2.6.6.1 Access terminal requirements

#### 17 7.2.6.6.1.1 F-pBCH0 supervision

18 Upon entering the active state, the access terminal shall set the F-pBCH0 supervision timer for  
19  $T_{\text{CCMPSupervision0}}$ . If an F-pBCH0 MAC packet is received while the timer is active, the timer is  
20 restarted. If the timer expires, the protocol returns a *SupervisionFailed* indication and disables the  
21 timer.

#### 22 7.2.6.6.1.2 F-pBCH1 supervision

23 Upon entering the active state, the access terminal shall set the F-pBCH1 supervision timer for  
24  $T_{\text{CCMPSupervision1}}$ . If an F-pBCH1 MAC packet is received while the timer is active, the timer is  
25 restarted. If the timer expires, the protocol returns a *SupervisionFailed* indication and disables the  
26 timer.

## 27 7.2.7 Header and trailer formats

28 The headers and trailers for the F-pBCH0, F-pBCH1 and F-OSICH are described in the sections for  
29 procedures for transmission and reception for the respective channels (7.2.6.1, 7.2.6.2 and 7.2.6.3).



## 7.2.8 Interface to other protocols

### 7.2.8.1 Commands

This protocol does not issue any commands.

### 7.2.8.2 Indications

This protocol does not register to receive any indications.

## 7.2.9 Configuration attributes

No configuration attributes are defined for this protocol.

## 7.2.10 Protocol numeric constants

Constant	Meaning	Value
$N_{CCMPType}$	Type field for this protocol	Table 9
$N_{CCMPDefault}$	Subtype field for this protocol	0x0000
$T_{CCMPSupervision0}$	F-pBCH0 supervision timer value	$12 \times N_{pBCH0\_Period}$ superframes
$T_{CCMPSupervision1}$	F-pBCH1 supervision timer value	12 superframes
$N_{QP\_BLK}$	Number of bits in the QuickPagingBlock	35 bits
$N_{CCMPpBCH1Bits}$	Number of bits in the pBCH1 payload	52

## 7.2.11 Session state information

The Session State Information record (see 10.10) consists of parameter records.

The parameter records for this protocol consist of the configuration attributes of this protocol.

## 7.3 Default Access Channel MAC Protocol

### 7.3.1 Overview

The Default Access Channel MAC Protocol provides the procedures and messages required for an access terminal to transmit, and an access network to receive, the Access Probe. An access probe may be used for initial access or handoff within an Active Set. The access network responds to an access probe with an Access Grant over the Shared Signaling MAC Protocol.

This specification assumes that the access network has one instance of this protocol for each sector in the network.

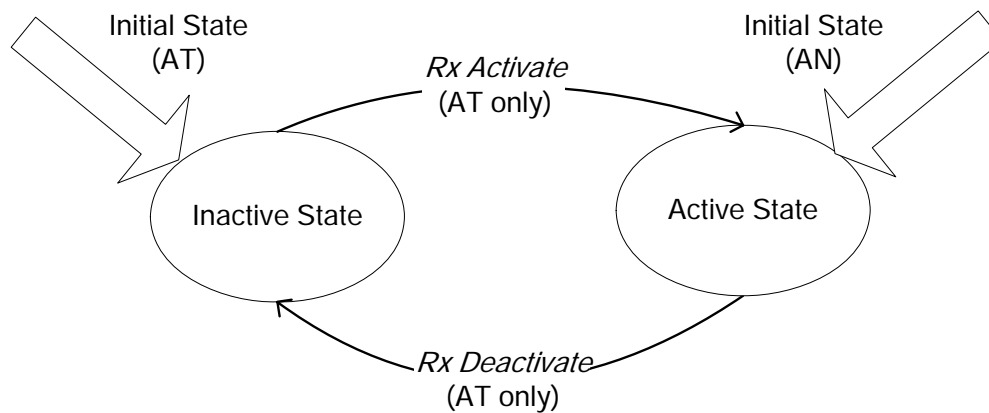
This protocol can be in one of two states at the access terminal:

- Inactive State:** In this state, the protocol waits for an *Activate* command. This state occurs when the access terminal has not acquired an access network.

- 1       ■ *Active State*: In this state, the access terminal may transmit on the Access Channel.

2 This protocol can be in only one state at the access network.

- 3       ■ *Active State*: In this state, the access network monitors the Access Channel.



4

5       **Figure 72 Default Access Channel MAC Protocol state diagram**

6 This protocol uses the following parameters and attributes.

7

Parameter Name	Where Defined	Comments
AccessCycleDuration	OMP	ExtendedChannelInfo
AccessSequencePartition	OMP	ExtendedChannelInfo
MaxProbesPerSequence	OMP	ExtendedChannelInfo
ProbeRampUpStepSize	OMP	ExtendedChannelInfo
PilotThreshold1	OMP	ExtendedChannelInfo
PilotThreshold2	OMP	ExtendedChannelInfo
OpenLoopAdjust	OMP	ExtendedChannelInfo
AccessRetryPersistence ( $N_{ACMPClass}$ values)	OMP	ExtendedChannelInfo
ReverseLinkSilencePeriod	OMP	ExtendedChannelInfo
ReverseLinkSilenceDuration	OMP	ExtendedChannelInfo
LoadControl	CC MAC	pBCH1
MaxProbeSequences	Configuration Attribute	
PageResponseBackoff	Configuration Attribute	
MaxProbeSequenceBackoff	Configuration Attribute	
RertyPersistenceOverride	Configuration Attribute	
TerminalAccessClass	Configuration Attribute	
RequestThreshold1	Configuration Attribute	
RequestThreshold2	Configuration Attribute	

8

## 7.3.2 Primitives

### 7.3.2.1 Commands

This protocol defines the following commands:

- *Activate*
- *Deactivate*
- *AttemptAccess*

### 7.3.2.2 Return indications

This protocol returns the following indications:

- *AccessGrantReceived*
- *TransmissionAborted*
- *AccessProbeReceived*
- *AccessFailed*

## 7.3.3 Public data

### 7.3.3.1 Static public data

This protocol does not define any static public data.

### 7.3.3.2 Dynamic public data

- Subtype for this protocol
- AccessSequenceID
- PilotPN
- AccessCarrier
- ProbePower

## 7.3.4 Protocol data unit

This protocol does not carry any payload on behalf of other protocols.

## 7.3.5 Protocol initialization and swap

### 7.3.5.1 Protocol initialization

Upon initialization at the access terminal,

- The values of the attributes for this protocol instance shall be set to the default values specified for each attribute.
- The protocol shall enter the Inactive State.

1 Upon initialization at the access network,

- 2     ■ The values of the attributes for this protocol instance shall be set to the default values  
3         specified for each attribute.
- 4     ■ The protocol shall enter the Active State.

### 5 **7.3.5.2 Protocol swap**

6 Upon swap at the access terminal, the protocol instance shall enter the Inactive State.

7 Upon swap at the access network, the protocol instance shall enter the Active State.

## 8 **7.3.6 Procedures**

### 9 **7.3.6.1 Command processing**

10 The access network shall ignore all commands.

#### 11 **7.3.6.1.1 Activate**

12 If this protocol receives an *Activate* command in the Inactive State,

- 13     ■ The access terminal shall transition to the Active State.
- 14     ■ The access network shall ignore this command.

15 If this protocol receives the command in the Active State, the command shall be ignored.

#### 16 **7.3.6.1.2 Deactivate**

17 If this protocol receives a *Deactivate* command in the Inactive State, the command shall be ignored.

18 If this protocol at the access terminal receives the command in the Active State, the access terminal  
19 shall

- 20     ■ Immediately cease transmitting on the Access Channel if it is in the process of sending a  
21         probe.
- 22     ■ Return a *TransmissionAborted* indication if it was in the process of sending an access  
23         probe.
- 24     ■ Clear all public data.
- 25     ■ Transition to the Inactive State.

26 The access network shall ignore this command.

#### 27 **7.3.6.1.3 AttemptAccess**

28 If the access terminal receives an *AttemptAccess* command, it shall invoke the procedure for  
29 processing an *AttemptAccess* command.

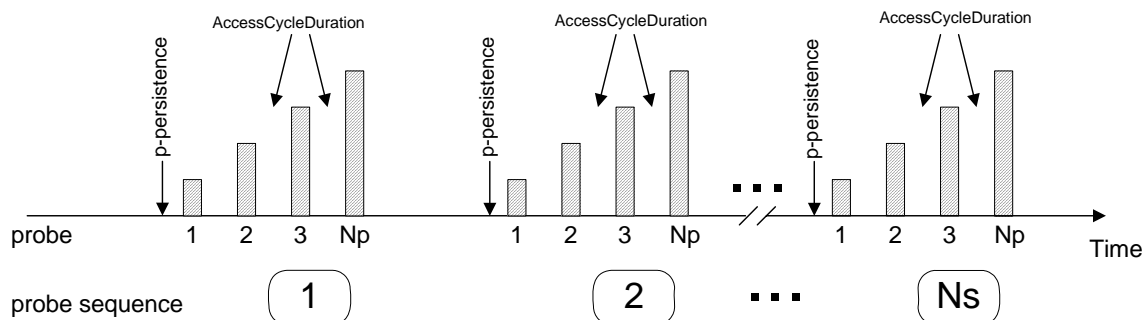
### 7.3.6.2 Access channel structure

Figure 73 illustrates the access probe structure and the access probe sequences. In the figure,  $N_s$  probe sequences are shown, where each probe sequence has  $N_p$  probes. The Access Channel MAC Protocol transmits access probes by instructing the Physical Layer to transmit a probe. With the instruction, the Access Channel MAC Protocol provides the Physical Layer with a power level, AccessSequenceID, PilotPN of the sector to which the access probe is to be transmitted and an AccessCarrier field. The Physical Layer allows the transmission of access probes only once every ControlSegmentPeriod frames (see 7.6.6.3 for definition).

Each probe in a sequence is transmitted at increased power until any of the following conditions are met:

- The access terminal receives an access grant,
- Transmission is aborted because the protocol received a *Deactivate* command, or
- Maximum number of probes per sequence (MaxProbesPerSequence) has been transmitted. After a maximum number of probes has been transmitted, a new probe sequence may resume from a lower power level.

Prior to the transmission of the first probe of all probe sequences, the access terminal performs a persistence test (see section 7.3.6.4.1.3) that is used to control congestion on the Access Channel. This persistence test may return a zero value in some cases, depending on the cause for access.



**Figure 73 Access probe sequences.  $N_s$  sequences with  $N_p$  probes per sequence**

### 7.3.6.3 Inactive state

This state applies only to the access terminal.

In this state, the access terminal waits for an *Activate* command.

## 1 7.3.6.4 Active state

2 In this state, the access terminal is allowed to transmit on the Access Channel and the access network  
3 is monitoring the Access Channel.

### 4 7.3.6.4.1 Access terminal requirements

#### 5 7.3.6.4.1.1 Procedure for processing the AttemptAccess command

6 The access terminal shall process only one *AttemptAccess* command at a time. *AttemptAccess*  
7 commands shall cause an existing probe transmission to end, and a new probe transmission to begin.

8 If any of the following events occurs while the *AttemptAccess* command is being processed, the  
9 access terminal shall invoke the procedure for ending probe transmission 7.3.6.4.1.2.

- 10 ■ The access terminal receives an access grant on the carrier on which the last access probe  
11 was transmitted (*SSMAC.AccessGrantReceived* indication).
- 12 ■ The access grant timer expires.
- 13 ■ The protocol receives an *IdleState.IdleHO* indication.
- 14 ■ The protocol receives a new *AttemptAccess* command (this command shall be processed  
15 after the current probe transmission ends).

16 When the procedure for processing the *AttemptAccess* command is invoked, the access terminal shall  
17 perform the following initial steps:

- 18 ■ Set ProbeSequenceNumber to 1.
- 19 ■ Set ProbeNumber to 1.
- 20 ■ Set TerminalAccessRetryPersistence to the TerminalAccessClass number value of the  
21 AccessRetryPersistence values. For example, if the TerminalAccessClass is 0, the field  
22 shall be set to the first AccessRetryPersistence value in the AccessParametersGroup.
- 23 ■ Set PilotPN to the Active Set public data field of the Idle State Protocol, or if a PilotPN  
24 was specified with the *AttemptAccess* command, set PilotPN to the specified value.
- 25 ■ Wait till the beginning of the next ControlSegmentPeriod (as defined in 7.6.6.3).

26 After performing the initial steps, the access terminal shall perform the following steps to transmit  
27 probes:

- 28 1. If ProbeSequenceNumber is greater than MaxProbeSequences, it shall set an access grant  
29 timer for  $T_{ACMPANProbeTimeout}$  duration and not perform additional steps.
- 30 2. Determine the AccessSequenceID by invoking procedure 7.3.6.4.1.4.
- 31 3. Add the AccessSequenceID to the public data.
- 32 4. If ProbeNumber is greater than MaxProbesPerSequence it shall:
  - 33 a. Set ProbeNumber to 1.
  - 34 b. Increment ProbeSequenceNumber by 1.

- 1           c. Determine the AccessCarrier by monitoring the LoadControl bits on the different  
2 carriers. For the remainder of the procedures, the access terminal shall use overhead  
3 parameters corresponding to the selected AccessCarrier.
- 4           d. Add the AccessCarrier to the public data.
- 5       5. If ProbeNumber is 1, it shall determine DelayToNextProbe by invoking the procedure for  
6 determining probe sequence backoff time 7.3.6.4.1.3. Otherwise, it shall set  
7 DelayToNextProbe to the AccessCycleDuration.
- 8       6. Start a timer for DelayToNextProbe frames. After the timer is expired, the access  
9 terminal shall proceed to the next step. This timer shall be decremented only in frames  
10 that satisfy the following requirements:
- 11           a. The ReverseLinkSilenceDuration and ReverseLinkSilencePeriod for the current  
12 sector is not active in the frame.
- 13           b. The superframe containing that frame has the LoadControl bits transmitted on the  
14 Control Channel MAC set to a value less than or equal to the TerminalAccessClass  
15 configuration attribute.
- 16           c. The QuickChannelInfoUpToDate and ExtendedChannelInfoUpToDate public data  
17 parameters of the Overhead Messages Protocol are both set to one.
- 18       7. Calculate the InitialProbePower using the procedures given in 7.3.6.4.1.5.
- 19       8. Transmit a probe using AccessSequenceID, PilotPN, AccessCarrier, and power  
20 calculated as:
- 21           
$$\text{ProbePower} = \text{InitialAccessPower} + \text{ProbeRampUpStepSize} * (\text{ProbeNumber} - 1)$$
- 22       9. Increment ProbeNumber and return to step 1.

#### 23 **7.3.6.4.1.2 Procedure for ending probe transmissions**

24 If the access terminal receives an access grant (*SSMAC.AccessGrantReceived* indication), it shall:

- 25       ■ Terminate the probe transmission procedure.
- 26       ■ Return an *AccessGrantReceived* indication.
- 27       ■ Place the ProbePower (the power at which the last probe was transmitted) in the public  
28 data
- 29       ■ Clear the AccessSequenceID from the public data.
- 30       ■ Terminate this procedure.

31 If the access grant timer expires, it shall:

- 32       ■ Return an *AccessFailed* indication.
- 33       ■ Clear the public data.
- 34       ■ Terminate this procedure.

1 If the protocol receives an *ActiveSetManagement.IdleHO* indication, it shall:

- 2 ■ Return an *AccessFailed* indication.
- 3 ■ Terminate the probe transmission procedure.
- 4 ■ Clear the public data.
- 5 ■ Terminate this procedure.

#### 6 **7.3.6.4.1.3 Procedure for determining probe sequence backoff time**

7 This procedure shall compute the persistence interval in units of *ControlSegmentPeriods*. If the  
8 following returns an integer value  $n$ , this procedure shall return a probe sequence backoff time of  $n$   
9 *ControlSegmentPeriods* (see 7.6.6.3 for definition of *ControlSegmentPeriod*).

10 If the access attempt was made in response to a page and *ProbeSequenceNumber* is 1, the persistence  
11 interval shall be set as follows:

- 12 ■ If the access terminal detected only one *QuickPage* bit set in the superframe where the  
13 access terminal received the page, the persistence interval shall be set to zero.
- 14 ■ If the access terminal detected more than one *QuickPage* bit set in the superframe where  
15 the access terminal received the page, the persistence interval shall be set to a random  
16 integer drawn uniformly between 0 and  $\text{PageResponseBackoff} * 3$ .

17 Otherwise, if the access attempt was made in response to an *OpenConnection* command from the Air  
18 Link Management Protocol at the access terminal and the *ProbeSequenceNumber* is 1, then the  
19 persistence interval shall be set to zero.

20 Otherwise, the persistence interval shall be set as follows:

- 21 ■ Generate a geometric random variable  $n$  with parameter  
22  $p = \text{TerminalAccessRetryPersistence}$ , i.e., the random variable takes value  $n$  with  
23 probability  $p(1-p)^{n-1}$ .
- 24 ■ Set the persistence interval to the minimum of  $n$  and  $\text{MaxProbeSequenceBackoff}$ .

#### 25 **7.3.6.4.1.4 Procedure for determining AccessSequenceID**

26 The sequences available for initial access shall be divided into  $N_{\text{ACMPNumPartitions}}$  partitions. The access  
27 terminal shall determine the partition to be used for the access attempt based on the observed pilot  
28 power and buffer level (number of bytes in the transmit buffer). Once the partition is determined, the  
29 access terminal shall select the *AccessSequenceID* using a uniform probability distribution over the  
30 partition. Note that of the  $N_{\text{ACMPNumSequences}}$  available for access, sequences 0, 1, 2 ...,  
31  $N_{\text{ACMPSpecialSequences}} - 1$  are reserved for active set operations, and sequences  $N_{\text{ACMPSpecialSequences}}$  through  
32  $N_{\text{ACMPNumSequences}} - 1$  are available for initial access.



### 7.3.6.4.1.4.1 Partitioning the access sequence space

The size of each partition shall be determined by the AccessSequencePartition field in the ExtendedChannelInfo block. Partition number N shall consist of AccessSequenceIDs ranging from PartitionNLower to PartitionNUpper. PartitionNLower and PartitionNUpper are determined using PartitionNSize (see Table 64 for a description of PartitionNSize) and the following algorithm.

- Partition1Lower =  $N_{ACMP\text{SpecialSequences}}$
- For N between 1 and 8
  - Set PartitionNUpper =  $\min(N_{ACMP\text{NumSequences}} - 1, \text{PartitionNLower} + \text{PartitionNSize})$
  - Set Partition(N+1)Lower =  $\min(N_{ACMP\text{NumSequences}} - 1, \text{PartitionNUpper} + 1)$
- Partition9Upper =  $N_{ACMP\text{NumSequences}} - 1$

The mapping of AccessSequencePartition to the values of PartitionNSize is contained in Table 64.

**Table 64 Access sequence partitions**

AccessSequence Partition	PartitionNSize (N from 1 to 9)									
	N	1	2	3	4	5	6	7	8	9
00000	0	0	0	0	0	0	0	0	0	remaining
00001	S2	S2	S2	S2	S2	S2	S2	S2	S2	remaining
00010	S3	S3	S3	S1	S1	S1	S1	S1	S1	remaining
00011	S1	S1	S1	S3	S3	S3	S1	S1	S1	remaining
00100	S1	S1	S1	S1	S1	S1	S3	S3	S3	remaining
00101	S3	S1	S1	S3	S1	S1	S3	S1	S1	remaining
00110	S1	S3	S1	S1	S3	S1	S1	S3	S3	remaining
00111	S1	S1	S3	S1	S1	S3	S1	S1	S1	remaining
01000	S3	S3	S1	S3	S1	S1	S1	S1	S1	remaining
01001	S1	S1	S1	S3	S3	S1	S3	S1	S1	remaining

Where the constants S1 through S3 are interpreted as,

**Table 65 Constants for interpreting the access sequence partition table**

Constant	Value
S1	$\text{floor}(N_{ACMP\text{NumSequences}}/18)$
S2	$S1 * 2$
S3	$S1 * 4$

#### 7.3.6.4.1.4.2 Determining AccessSequenceID in Access State of Idle State Protocol

If the access probe transmission procedure is invoked when the Idle State Protocol is in the Access State, the Access Channel MAC Protocol shall select a partition space based on PilotLevel and RequestLevel as defined in Table 66. The access terminal shall then select an AccessSequenceID using a uniform probability distribution over sequences in the selected partition.

**Table 66 Mapping the BufferLevel and PilotLevel to access sequence segment**

PilotLevel	RequestLevel		
	1	2	3
1	1	2	3
2	4	5	6
3	7	8	9

The access terminal shall select its RequestLevel based on the number of bits it needs to transmit on the ReverseTrafficChannelMAC. The Request Level shall be set as follows:

- RequestLevel shall be set to 1 for RequestThreshold1 or fewer bytes.
- RequestLevel shall be set to 2 for RequestThreshold1+1 to RequestThreshold2 bytes.
- RequestLevel shall be set to 3 for more than RequestThreshold2 bytes.

To determine the PilotLevel, the access terminal shall use PilotThreshold1 and PilotThreshold2 as described below.

The access terminal shall select its PilotLevel based on the ratio (measured in dB) of the acquisition pilot power from the sector where the access attempt is being made to the total power received in the acquisition channel time slot (acquisition pilot power plus interference power).

- If this ratio is below PilotThreshold1, the PilotLevel is set to 1.
- If the ratio is not below PilotThreshold1 and is below PilotThreshold2, the PilotLevel is set to 2.
- Otherwise the PilotLevel is set to 3.
- In case the partition of the sequence space is not available (because, for example, the AccessSequencePartition field of the Access Parameters has not been read), the AccessSequenceID shall be selected uniformly from between ( $N_{ACMPNumSequences}-S1-1$ ) and ( $N_{ACMPNumSequences}-1$ ). From Table 64 it may be observed that the minimum size of Partition 9 is S1, and the procedure in this paragraph always selects a sequence in Partition 9.

The acquisition pilot power referred to above is the received power of the second OFDM symbol of the F-ACQCH, as described in the Physical Layer specification (OFDM symbol with index 6 from the superframe preamble). See 9.3.2.4.4. The acquisition channel time slot refers to the same symbol, i.e., the OFDM symbol with index 6 from the superframe preamble.

### 7.3.6.4.1.4.3 Determining AccessSequenceID outside Access State of Idle State Protocol

The following describes the access probe transmission procedure when the Idle State Protocol is not in Access State. These procedures are used for

- “hard” handoff between different sectors of a synchronous subset
- handoff between different synchronous subsets
- handoff between sectors having different frequencies
- timing and power correction for a sector

The AccessSequenceID shall depend on the identity of the handoff target sector, as follows.

The AccessSequenceID for the purpose of handoff shall be set as follows

- If the ActiveSetAssignment message has AccessSequenceIDIncluded set to 0 for the target sector, the AccessSequenceID shall be set using the PilotLevel and RequestLevel fields (described in 7.3.6.4.1.4.2) as follows:

$$\text{AccessSequenceID} = \left\lceil N_{\text{ACMPSpecialSequences}} / 2 \right\rceil + 3 * (\text{PilotLevel} - 1) + \text{RequestLevel} - 1.$$

- If the ActiveSetAssignment message has AccessSequenceIDIncluded set to 1 for the target sector, the AccessSequenceID shall be set to the AccessSequenceID field for the target sector in the ActiveSetAssignment message.

The AccessSequenceID for the purpose of timing or power correction shall be set as follows

- The AccessSequenceID shall be set using the PilotLevel field described in 7.3.6.4.1.4.2.

$$\text{AccessSequenceID} = 3 * (\text{PilotLevel} - 1).$$

Note that the Physical Layer will scramble any transmission of a special access sequence (one with AccessSequenceID from 0..N<sub>ACMPSpecialSequences</sub>-1) with the MACID of the transmitting AT in the target sector.

### 7.3.6.4.1.5 Procedure for determining InitialAccessPower

The parameter InitialAccessPower shall be determined based on the OpenLoopAdjust parameter and the received power of the pilot from the sector where the access attempt is being made.

$$\text{InitialAccessPower} = - \text{MeanRxPower (dBm)} + \text{OpenLoopAdjust}$$

where the MeanRxPower shall be updated throughout the access procedure based on the received traffic channel acquisition pilot power, as measured at the receive antenna of the access terminal.

#### 1 **7.3.6.4.2 Access network requirements**

2 If the access network receives an access probe, it shall generate an *AccessProbeReceived* indication,  
3 and place the *AccessSequenceID* in its public data.

4 If the protocol receives a *SharedSignalingMAC.AccessGrantSent* indication, it shall issue the  
5 following commands:

- 6 ■ *ReverseTrafficChannelMAC.Activate*
- 7 ■ *ReverseControlChannelMAC.Activate*

8 The access network should monitor and control the load on the R-ACH. The access network may  
9 control the load by adjusting the *LoadControl* bits.

#### 10 **7.3.7 Header formats**

11 This protocol does not define any headers.

#### 12 **7.3.8 Message formats**

13 The protocol uses the *AttributeUpdateRequest*, *AttributeUpdateAccept*, and *AttributeUpdateReject*  
14 messages of the Generic Attribute Update Protocol in 10.9 to update configurable attributes.

#### 15 **7.3.9 Interface to other protocols**

##### 16 **7.3.9.1.1 Commands**

17 This protocol issues the following commands (access network only):

- 18 ■ *ReverseTrafficChannelMAC.Activate*
- 19 ■ *ReverseControlChannelMAC.Activate*

##### 20 **7.3.9.1.2 Indications**

21 This protocol registers to receive the following indications:

- 22 ■ *SharedSignalingMAC.AccessGrantSent*
- 23 ■ *SharedSignalingMAC.AccessGrantReceived*

#### 24 **7.3.10 Configuration attributes**

25 The negotiable simple attributes for this protocol are listed in Table 67. The access terminal and the  
26 access network shall use as default the values in Table 67 listed in ***bold italics***.

27 Unless specified otherwise, the access terminal and the access network shall use the Generic Attribute  
28 Update Protocol in 10.9 to update configurable attributes belonging to the Default Access Channel  
29 MAC Protocol.

1

**Table 67 Configurable values**

<b>Attribute ID</b>	<b>Attribute</b>	<b>Values</b>	<b>Meaning</b>
0xff	RetryPersistenceOverride	<i>0xff</i>	The access terminal shall use the persistence probability value as specified by the appropriate AccessRetryPersistence field of the AccessParameters message.
		0x3f	The access terminal shall use zero as the persistence probability.
		0x00 to 0x3e	The access terminal shall use $2^{-n/4}$ as the persistence probability.
		All other values	Reserved
0xfe	TerminalAccessClass	<i>0x00-0x03</i>	The access class of the user. This parameter controls the AccessRetryPersistence used by the terminal.
0xfd	MaxProbeSequences	<i>0x03</i>	The maximum number of probe sequences that can be transmitted as part of one access attempt.
		<i>0x01, 0x02, 0x04 to 0x0f</i>	Other allowed values for this parameter.
		All other values	Reserved
0xfc	PageResponseBackoff	<i>0x01</i>	Expedited response to pages not enabled
		<i>0x00,</i>	Expedited response to pages enabled
		All other values	Reserved
0xfb	MaxProbeSequenceBackoff	<i>0x08</i>	Maximum backoff between probe sequences
		<i>0x09 to 0x20</i>	Other allowed values
		All other values	Reserved
0xfa	RequestThreshold1	<i>0x30</i>	Used to determine the RequestLevel when making an access attempt in idle state.
		Other values	Other allowed values
0xfa	RequestThreshold2	<i>0x80</i>	Used to determine the RequestLevel when making an access attempt in idle state.
		Other values	Other allowed values

2

### 7.3.11 Protocol numeric constants

Constant	Meaning	Value
$N_{ACMPType}$	Type field for this protocol	Table 9
$N_{SIACMP}$	Subtype field for this protocol	0x0000
$N_{ACMPClass}$	Number of different persistence values	4
$N_{ACMPNumSequences}$	Number of available access sequences	1024
$N_{ACMPNumPartitions}$	Number of partitions of the access sequence space	9
$N_{ACMPSpecialSequences}$	Number of access sequences reserved for handoff or power and timing correction.	18
$T_{ACMPANProbeTimeout}$	Maximum time to send an acknowledgment for a probe at the access network	5 PHYFrames

### 7.3.12 Session state information

The Session State Information record (see 10.10) consists of parameter records.

The parameter records for this protocol consist of the configuration attributes of this protocol.

## 7.4 Default Shared Signaling Channel MAC Protocol

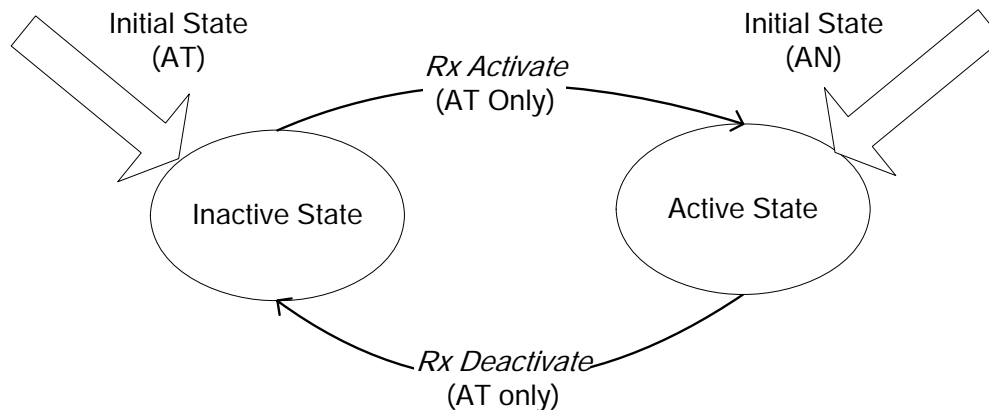
### 7.4.1 Overview

The Default Shared Signaling MAC Protocol provides the procedures and messages required for Lower MAC Sublayer signaling. This includes access network transmissions on the F-SSCH Physical Layer channels.

This specification assumes that the access network has one instance of this protocol for each sector in the network. However, any implementation that behaves identically is compliant.

This protocol can be in one of two states:

- *Inactive State*: In this state, the protocol waits for an *Activate* command. This state applies only to the access terminal and occurs when the access terminal has not acquired an access network or is not monitoring the F-SSCH.
- *Active State*: In this state, the access network transmits and the access terminal receives the F-SSCH.



**Figure 74 Default Shared Signaling Channel MAC Protocol state diagram**

This protocol shall use the following parameters and attributes.

Parameter Name	Where Defined	Comments
SSCHNumHopports	OMP	QuickChannelInfo block
EffectiveNumAntennas	OMP	QuickChannelInfo block
FLFirstRestrictedSetSubband	OMP	QuickChannelInfo block
FLNumRestrictedSetSubbands	OMP	QuickChannelInfo block
FLChannelTreeIndex	OMP	QuickChannelInfo block
RLChannelTreeIndex	OMP	ExtendedChannelInfo
FLPCReportInterval	OMP	ExtendedChannelInfo
MACIDRange	OMP	ExtendedChannelInfo
RLCtrlPCMode	OMP	ExtendedChannelInfo
FastOSIEnabled	OMP	ExtendedChannelInfo
AccessCarrier	Access Channel MAC Protocol	Public data
AccessSequenceID	Access Channel MAC Protocol	Public data
FLSS	RCC MAC Protocol	Public data
DFLSS	RCC MAC Protocol	Public data
RLSS	RCC MAC Protocol	Public data
DRLSS	RCC MAC Protocol	Public data
TuneAwayStatus	Connected State Protocol	Public data
SelectedInterlaceMode	Connected State Protocol	Public data
SelectedInterlaceAssignment	Connected State Protocol	Public data
MultiCarrierOn	Physical Layer Protocol	Public data

## 7.4.2 Primitives

### 7.4.2.1 Commands

This protocol defines the following commands:

- *Activate*
- *Deactivate*

### 7.4.2.2 Return indications

This protocol returns the following indications:

- *AccessGrantSent*
- *AccessGrantReceived*
- *SupervisionFailed*

## 7.4.3 Public data

### 7.4.3.1 Static public data

This protocol does not define any static public data.

### 7.4.3.2 Dynamic public data

- Subtype for this protocol
- AccessGrant message most recently transmitted or received by the protocol
- ActiveCarriers: For each MACID, the public data shall store ActiveCarriers, four bits which specify the carriers in which the access terminal can operate. If the  $i^{\text{th}}$  bit for a particular access terminal is equal to '1', that indicates the access terminal can operate in the  $i^{\text{th}}$  carrier. (Note that ActiveCarriers is only relevant when multi-carrier mode is equal to MultiCarrierOn.)
- REQCarrier: For each MACID, the public data shall store REQCarrier, two bits which specify on which carrier the access terminal shall send requests. The two bits represent an integer from 0 to 3, which directly specifies the carrier index. (Note that this is only relevant when multi-carrier mode is equal to MultiCarrierOn.)

## 7.4.4 Protocol data unit

This protocol does not carry payload on the behalf of other protocols.

## 7.4.5 Protocol initialization and swap

### 7.4.5.1 Protocol initialization

Upon initialization at the access terminal,

- The values of the attributes for this protocol instance shall be set to the default values specified for each attribute.
- The protocol shall enter the Inactive State.



1 Upon initialization at the access network,

- 2     ■ The values of the attributes for this protocol instance shall be set to the default values  
3         specified for each attribute.
- 4     ■ The protocol shall enter the Active State.

#### 5 **7.4.5.2 Protocol swap**

6 Upon swap at the access terminal, the protocol instance shall enter the Inactive State.

7 Upon swap at the access network, the protocol instance shall enter the Active State.

### 8 **7.4.6 Procedures**

9 The following sections specify procedures for transmission and reception of signals on the F-SSCH  
10 Physical Layer channel.

#### 11 **7.4.6.1 Command processing**

12 The access network shall ignore all commands.

##### 13 **7.4.6.1.1 Activate**

14 If this protocol receives an *Activate* command in the Inactive State, the access terminal shall transition  
15 to the Active State.

16 If this protocol receives this command in the Active State, the command shall be ignored.

##### 17 **7.4.6.1.2 Deactivate**

18 If this protocol receives a *Deactivate* command in the Inactive State, the command shall be ignored.

19 If this protocol receives this command in the Active State, the access terminal shall transition to the  
20 Inactive State.

#### 21 **7.4.6.2 Inactive state**

22 This state applies only to the access terminal.

23 When the protocol is in the Inactive State, the access terminal waits for an *Activate* command.

#### 24 **7.4.6.3 Active state**

25 In this state, the access network transmits, and the access terminal monitors, the Physical Layer  
26 channels managed by this MAC protocol.

### 1 7.4.6.3.1 Access network requirements

#### 2 7.4.6.3.1.1 Hop-port assignment for the F-SSCH

3 The F-SSCH is a physical layer channel, and is present in all FL PHY Frames.

4 In the following, restricted hop-ports are those belonging to restricted subbands, as specified by  
5 FLFirstRestrictedSetSubband and FLNumRestrictedSetSubbands, and those that are not in the usable  
6 hop-ports, as specified by the PHY Layer Protocol.

7 If multi-carrier mode is equal to MultiCarrierOff, then the F-SSCH operates over a set of hop-ports  
8 that map to a set of base nodes in the FL channel tree with index FLChannelTreeIndex. The exact set  
9 of hop-ports shall be determined based on SSCHNumHopports, as follows.

10 Consider the hop-ports indexed from 0 to  $N_{\text{CARRIER\_SIZE}}-1$ , where  $N_{\text{CARRIER\_SIZE}}$  is defined in the  
11 Physical Layer Protocol. Let  $q$  be a hop-port index in this range. Let  $y$  be the least integer that is  
12 greater than or equal to  $N_{\text{CARRIER\_SIZE}}/(\text{MinHopPortsPerNode} * \text{SSCHNumHopports})$ , where  
13 MinHopPortsPerNode is a characteristic of the channel tree in use and is defined in 7.1.4.1, and let  $K$   
14 be the number of hop-ports already allocated to F-SSCH by the following algorithm.

- 15 1. Set  $K=0$ , and  $q=0$ .
- 16 2. Let the base node mapped by hop-port  $q$  be specified by base node  $Q$ . The access  
17 network shall check if the following condition holds: Base node  $Q$  maps to no hop-  
18 ports that are restricted, and maps to no hop-ports that have been already allocated to  
19 the F-SSCH.

20 If this condition is satisfied, the access network shall allocate all the hop-ports mapped by  
21 base node  $Q$  to F-SSCH, and shall increment  $K$  by the number of hop-ports mapped by  
22 base node  $Q$ . Proceed to step 4.

23 If this condition is not satisfied, proceed to step 3:

- 24 3. Increment  $q$  by 1. If  $q > N_{\text{CARRIER\_SIZE}}-1$ , let  $q=0$  and return to step 2.
- 25 4. If  $K \geq \text{SSCHNumHopports}$ , all necessary hop-ports for F-SSCH have been allocated,  
26 and this algorithm shall end.

27 If  $K < \text{SSCHNumHopports}$ , let  $q=q+y$ . If  $q > N_{\text{CARRIER\_SIZE}}-1$ , let  $q=0$ . Return to step 2.

28 If multi-carrier mode is equal to MultiCarrierOn, then the F-SSCH shall operate over a set of hop-  
29 ports in each carrier. The set of hop-ports in the carrier  $j$  allocated to the SSCH shall be determined  
30 using the procedure described above setting the values of FLChannelTreeIndex and  
31 SSCHNumHopports to those sent in the overhead parameters message of carrier  $j$ . Note that these  
32 values may be different for each carrier.

33 The Physical Layer Protocol shall be passed the set of hop-ports to be used for F-SSCH on each  
34 carrier.

### 7.4.6.3.1.2 Blocks for the F-SSCH

The F-SSCH can send multiple blocks. The fields of each block are defined in the descriptions of the blocks that follow the table. Exceptions are the following commonly used fields:

4	MACID	Sector-specific access terminal identifier.
5	NodeID	Field used to identify a set of hop-ports. Mapping of NodeID to hop-ports is defined in 7.5.6.6 for the forward link and in 7.7.6.7 for the reverse link.
6	FrameResID	Field used to resolve the interlace of applicability for AccessGrants and RLABs when duplexing mode is TDD and $N_{FL\_BURST} < N_{RL\_BURST}$ .
7	PF	Field used to identify packet formats, and is defined in 7.5.6.7 and 7.7.6.8.

The length of the FrameResID is zero (0) bits if duplexing mode is FDD. If duplexing mode is TDD, the length of the FrameResID,  $N_{FRID}$  is

$$N_{FRID} = \begin{cases} \left\lceil \log_2 \left( \frac{N_{RL\_BURST}}{N_{FL\_BURST}} \right) \right\rceil & , N_{RL\_BURST} > N_{FL\_BURST} \\ 0 & , N_{RL\_BURST} \leq N_{FL\_BURST} \end{cases}$$

To resolve the RL interlace of applicability of an RLAB/AccessGrant, let  $i \in [0, N_{FL\_BURST} - 1]$  indicate the FL PHY frame of arrival of the RLAB/AccessGrant within an FL burst, and let  $j \in [0, N_{RL\_BURST} - 1]$  indicate the RL PHY frame within the following RL burst that is in the interlace of applicability. Then,

$$j = \left\lfloor i * \frac{N_{RL\_BURST}}{N_{FL\_BURST}} \right\rfloor + \text{FrameResID}$$

The ‘Extended Transmission’ field, although it may be present in the blocks as described below, is absent if the duplexing mode is TDD.

The parameter  $N_{EFF\_TX\_ANT}$  is used in the tables, where

$$N_{EFF\_TX\_ANT} = \min(\text{EffectiveNumAntennas}, 4)$$

The length of the NodeID field used to identify forward link hop-ports is given by  $N_{FL\_NODEID} = \lceil \log_2(\text{NumFLNodeIDs}) \rceil$ , where  $\text{NumFLNodeIDs}$  is the number of nodes in the forward link channel tree with index  $\text{FLChannelTreeIndex}$  in the carrier of interest. The length of the NodeID field used to identify reverse link hop-ports is given by  $N_{RL\_NODEID} = \lceil \log_2(\text{NumRLNodeIDs}) \rceil$ , where  $\text{NumRLNodeIDs}$  is the number of nodes in the reverse link channel tree with index  $\text{RLChannelTreeIndex}$  in the carrier of interest.

The length of the MACID field is given by  $N_{MACID} = \text{ceil}(\log_2(N_{CARRIER\_SIZE} * N_{CARRIERS}))$ , where  $N_{CARRIERS}$  is given in the Physical Layer Protocol.

1

Table 68 F-SSCH blocks

BLOCK Name	Header (binary)	Length (bits)	Fields	[#bits]
AccessGrant	0000	$6 + N_{\text{MACID}} + N_{\text{RL\_NODEID}} + N_{\text{FRID}}$	MACID NodeID FrameResID TimingAdjust	$[N_{\text{MACID}}]$ $[N_{\text{RL\_NODEID}}]$ $[N_{\text{FRID}}]$ [6]
NS-FLAB	0001	$9 + N_{\text{MACID}} + N_{\text{FL\_NODEID}}$	MACID NodeID PF Duration ExtendedTransmission	$[N_{\text{MACID}}]$ $[N_{\text{FL\_NODEID}}]$ [6] [2] [1]
NS-MCWFLAB1	0010	$8 + N_{\text{MACID}} + N_{\text{FL\_NODEID}}$	MACID NodeID PFLayer1 Duration Extended Transmission	$[N_{\text{MACID}}]$ $[N_{\text{FL\_NODEID}}]$ [5] [2] [1]
NS-MCWFLAB2	0011	$N_{\text{MACID}} + 4 * (N_{\text{EFF\_TX\_ANT}} - 1)$	MACID PFLayer2 ... PFLayer $N_{\text{EFF\_TX\_ANT}}$	$[N_{\text{MACID}}]$ [4] ... [4]
NS-SCWFLAB	0100	$10 + N_{\text{MACID}} + N_{\text{FL\_NODEID}}$	MACID NodeID PF NumLayers Duration Extended Transmission	$[N_{\text{MACID}}]$ $[N_{\text{FL\_NODEID}}]$ [5] [2] [2] [1]
FLAB	0101	$8 + N_{\text{MACID}} + N_{\text{FL\_NODEID}}$	MACID NodeID PF Extended Transmission Supplemental	$[N_{\text{MACID}}]$ $[N_{\text{FL\_NODEID}}]$ [6] [1] [1]
MCWFLAB1	0110	$6 + N_{\text{MACID}} + N_{\text{FL\_NODEID}}$	MACID NodeID PFLayer1 Extended Transmission Supplemental	$[N_{\text{MACID}}]$ $[N_{\text{FL\_NODEID}}]$ [5] [1] [1]
MCWFLAB2	0111	$N_{\text{MACID}} + 4 * (N_{\text{EFF\_TX\_ANT}} - 1)$	MACID PFLayer2 ... PFLayer $N_{\text{EFF\_TX\_ANT}}$	$[N_{\text{MACID}}]$ [4] ... [4]

BLOCK Name	Header (binary)	Length (bits)	Fields	[#bits]
SCWFLAB	1000	$9 + N_{MACID} + N_{FL\_NODEID}$	MACID NodeID PF NumLayers Extended Transmission Supplemental	$[N_{MACID}]$ $[N_{FL\_NODEID}]$ [5] [2] [1] [1]
RLAB	1001	$10 + N_{MACID} + N_{RL\_NODEID} + N_{FRID}$	MACID NodeID FrameResID PF Extended Transmission Supplemental Delta	$[N_{MACID}]$ $[N_{RL\_NODEID}]$ $[N_{FRID}]$ [5] [1] [1] [3]
NS-RLAB	1010	$10 + N_{MACID} + N_{RL\_NODEID} + N_{FRID}$	MACID NodeID FrameResID PF Duration Extended Transmission Delta	$[N_{MACID}]$ $[N_{RL\_NODEID}]$ $[N_{FRID}]$ [5] [1] [1] [3]
CCB	1011	$6 + N_{MACID}$	MACID ActiveCarriersChange REQCarrierChange	$[N_{MACID}]$ [4] [2]
FLAB-HO	1100	$10 + N_{MACID} + N_{FL\_NODEID}$	MACID NodeID PF Extended Transmission DesiredSector	$[N_{MACID}]$ $[N_{FL\_NODEID}]$ [6] [1] [3]
RLAB-HO	1101	$9 + N_{MACID} + N_{RL\_NODEID} + N_{FRID}$	MACID NodeID FrameResID PF Extended Transmission DesiredSector	$[N_{MACID}]$ $[N_{RL\_NODEID}]$ $[N_{FRID}]$ [5] [1] [3]

1

2 **7.4.6.3.1.2.1 AccessGrant**

3 A block that is sent in response to a detected access sequence transmission that allocates a MACID to  
4 the access terminal and an initial NodeID for use by the access terminal. The Access Grant block is  
5 scrambled with a hash of the AccessSequenceID used by the access terminal during transmission of  
6 the associated access sequence. The hashing procedure is defined in the Physical Layer. A  
7 TimingAdjust field is provided to inform the access terminal of the timing offset to use for  
8 subsequent RL transmissions. The access terminal shall advance its transmission timing by the

1 amount:  $\text{offset} = \text{TimingAdjust} * N_{\text{FFT}} / 128$  chips, where TimingAdjust is interpreted as an unsigned  
2 integer.

### 3 7.4.6.3.1.2.2 NS-FLAB

4 Non-sticky Forward Link Assignment Block. If the MACID in this block is not the broadcast  
5 MACID, this block informs the access terminal that holds the specific MACID that hop-ports  
6 specified by the NodeID field have been assigned to the access terminal, and informs that access  
7 terminal of the PF that should be used on its assigned hop-ports. The PF field is described in  
8 7.5.6.7.1. The duration of the assignment is specified in the duration field, resulting in a duration of  
9 3,6,9, or 12 PHY frames. If the MACID in this message is the broadcast MACID, then all access  
10 terminals are assigned the hop-ports specified by the NodeID field, with the given packet format.

11 If the extended transmission field is equal to '0', an NS-FLAB assigns hop-ports for a particular  
12 interlace consisting of standard PHY Frames, and the duration field specifies the number of standard  
13 PHY Frames to be used for transmission for this assignment. If the Extended Transmission field is  
14 equal to '1', an NS-FLAB assigns hop-ports for an interlace consisting of extended PHY Frames (6  
15 contiguous standard PHY Frames), and the duration field specifies the number of extended PHY  
16 Frames to be used for transmission for this assignment.

### 17 7.4.6.3.1.2.3 NS-MCWFLAB

18 NS-MCWFLAB1 Non-sticky Multiple Code Word MIMO Forward Link Assignment Block,  
19 part one. This block informs the access terminal that holds a specific MACID  
20 that hop-ports specified by the NodeID field have been assigned to the access  
21 terminal. It also informs the access terminal, via the PFLayer1 field, which  
22 packet format should be used on the first MIMO layer. See the Physical  
23 Layer specification for the definition of MIMO layers. The packet format  
24 field (PFLayer1) is described in 7.5.6.7.2. The packet formats that should be  
25 used on the remaining layers are specified in MCWFLAB2.

26  
27 The duration of the assignment is specified in the duration field, resulting in  
28 a duration of 3,6,9, or 12 PHY frames (standard or extended).

29  
30 If the extended transmission field is equal to '0', an NS-MCWFLAB1 assigns  
31 hop-ports for a particular interlace consisting of standard PHY frames, and  
32 the duration field specifies the number of standard PHY frames to be used for  
33 transmission on this assignment. If the Extended Transmission field is equal  
34 to '1', an NS-MCWFLAB1 assigns an hop-ports for an interlace consisting of  
35 extended PHY frames, and the duration field specifies the number of  
36 extended PHY frames to be used for transmission on this assignment.

37 NS-MCWFLAB2 Non-sticky Multiple Code Word MIMO Forward Link Assignment Block,  
38 part two. This block informs the access terminal that holds a specific MACID  
39 of the packet formats to be used for MIMO layer 2, up through  $N_{\text{EFF\_TX\_ANT}}$ ,  
40 via the fields PFLayer2 ... PFLayer $N_{\text{EFF\_TX\_ANT}}$ . The packet format field  
41 (PFLayer $_i$ ) is described in 7.5.6.7.2.

#### 1 **7.4.6.3.1.2.4 NS-SCWFLAB**

2 Non-sticky Single Code Word MIMO Forward Link Assignment Block. This block informs the  
3 access terminal that holds a specific MACID that hop-ports specified by the NodeID field have been  
4 assigned to the access terminal. It also informs the access terminal, via the PF field, which packet  
5 format should be used. The PF field is described in 7.5.6.7.3. The NumLayers field indicates the  
6 number of MIMO layers that shall be transmitted on the assignment. See the Physical Layer  
7 specification for the definition of MIMO layers.

8 The duration of the assignment is specified in the duration field, resulting in a duration of 3, 6, 9, or  
9 12 PHY frames (standard or extended).

10 If the extended transmission field is equal to '0', an NS-SCWFLAB assigns hop-ports for a particular  
11 interlace consisting of standard PHY frames, and the duration field specifies the number of standard  
12 PHY frames to be used for transmission for this assignment. If the Extended Transmission field is  
13 equal to '1', an NS-SCWFLAB assigns hop-ports for an interlace consisting of extended PHY frames,  
14 and the duration field specifies the number of extended PHY frames to be used for transmission for  
15 this assignment.

#### 16 **7.4.6.3.1.2.5 FLAB**

17 Forward Link Assignment Block. This block informs the access terminal that holds a specific  
18 MACID that hop-ports specified by the NodeID field have been assigned to the access terminal, and  
19 informs that access terminal of the PF that should be used on its assigned hop-ports. The PF field is  
20 described in 7.5.6.7.1.

21 If the extended transmission field is equal to '0', an FLAB assigns hop-ports for a particular interlace  
22 consisting of standard PHY Frames. If the Extended Transmission field is equal to '1', an FLAB  
23 assigns hop-ports for an interlace consisting of extended PHY Frames.

24 The access network shall set the Supplemental field in the message to '1', if the assigned hop-ports  
25 should be added to the existing access terminal assignment on the interlace to be occupied by the  
26 given assignment, and to '0', if the assignment should replace any existing assignment on the  
27 interlace to be occupied by the given assignment.

28 Note that an existing extended transmission duration assignment should only be supplemented with  
29 another extended transmission duration assignment (i.e., an FLAB with the Extended Transmission  
30 field set to '1'), and an existing standard (i.e., non-extended) assignment should only be  
31 supplemented with another standard assignment (i.e., an FLAB with the Extended Transmission field  
32 set to '0'.)

### 7.4.6.3.1.2.6 MCWFLAB

#### MCWFLAB1

Multiple Code Word MIMO Forward Link Assignment Block , part one. This block informs the access terminal that holds a specific MACID that hop-ports specified by the NodeID field have been assigned to the access terminal. It also informs the access terminal, via the PFLayer1 field, which packet format should be used on the first MIMO layer. See the Physical Layer specification for the definition of MIMO layers. The packet format field (PFLayer1) is described in 7.5.6.7.2. The packet formats that should be used on the remaining layers are specified in MCWFLAB2.

If the extended transmission field is equal to '0', a MCWFLAB1 assigns hop-ports for a particular interlace consisting of standard PHY Frames. If the Extended Transmission field is equal to '1', a MCWFLAB assigns hop-ports for an interlace consisting of extended PHY Frames.

The access network shall set the Supplemental field in the message to '1', if the assigned hop-ports should be added to the existing access terminal assignment on the interlace to be occupied by the given assignment, and to '0', if the assignment should replace any existing assignment on the interlace to be occupied by the given assignment.

Note that an existing extended transmission duration assignment should only be supplemented with another extended transmission duration assignment (i.e., an MCWFLAB1 with the Extended Transmission field set to '1'), and an existing standard (i.e., non-extended) assignment should only be supplemented with another standard assignment (i.e., an MCWFLAB1 with the Extended Transmission field set to '0'.)

#### MCWFLAB2

Multiple CodeWord MIMO Forward Link Assignment Block, part two. This block informs the access terminal that holds a specific MACID of the packet formats to be used for MIMO layer 2, up through  $N_{EFF\_TX\_ANT}$ , via the fields PFLayer2 ... PFLayer $N_{EFF\_TX\_ANT}$ . The packet format field (PFLayer $i$ ) is described in 7.5.6.7.2.

### 7.4.6.3.1.2.7 SCWFLAB

Single Code Word MIMO Forward Link Assignment Block. This block informs the access terminal that holds a specific MACID that hop-ports specified by the NodeID field have been assigned to the access terminal. It also informs the access terminal, via the PF field, which packet format should be used. The PF field is described in 7.5.6.7.3. The NumLayers field indicates the number of MIMO layers that shall be transmitted on the assignment. See the Physical Layer specification for the definition of MIMO layers.

If the extended transmission field is equal to '0', a SCWFLAB assigns hop-ports for a particular interlace consisting of standard PHY Frames. If the Extended Transmission field is equal to '1', a SCWFLAB assigns hop-ports for an interlace consisting of extended PHY Frames.



1 The access network shall set the Supplemental field in the message to '1', if the assignment should be  
 2 added to the existing access terminal assignment on the interlace to be occupied by the given  
 3 assignment, and to '0', if the assignment should replace any existing assignment on the interlace to be  
 4 occupied by the given assignment.

5 Note that an existing extended transmission duration assignment should only be supplemented with  
 6 another extended transmission duration assignment (i.e., an SCWFLAB with the Extended  
 7 Transmission field set to '1'), and an existing standard (i.e., non-extended) assignment should only be  
 8 supplemented with another standard assignment (i.e., an SCWFLAB with the Extended Transmission  
 9 field set to '0'.)

#### 10 **7.4.6.3.1.2.8 RLAB**

11 Reverse Link Assignment Block. This block informs the access terminal that holds a specific MACID  
 12 that hop-ports specified by the NodeID field have been assigned to the access terminal, and informs  
 13 the access terminal, via the PF field, of the packet format to be used for transmission on its  
 14 assignment. The PF field is described in 7.7.6.8.

15 If the extended transmission field is equal to '0', an RLAB assigns hop-ports for a particular interlace  
 16 consisting of standard PHY frames. If the Extended Transmission field is equal to '1', an RLAB  
 17 assigns hop-ports for an interlace consisting of extended PHY frames.

18 The access network shall set the Supplemental field in the message to '1', if the assignment should be  
 19 added to the existing access terminal assignment on the interlace to be occupied by the given  
 20 assignment, and to '0', if the assignment should replace any existing assignment on the interlace to be  
 21 occupied by the given assignment.

22 Note that an existing extended transmission duration assignment should only be supplemented with  
 23 another extended transmission duration assignment (i.e., an RLAB with the Extended Transmission  
 24 field set to '1'), and an existing standard (i.e., non-extended) assignment should only be  
 25 supplemented with another standard assignment (i.e., an RLAB with the Extended Transmission field  
 26 set to '0'.)

27 The Delta field is used by the access terminal to determine the power at which to send reverse link  
 28 data, as specified by the RTC MAC.

#### 29 **7.4.6.3.1.2.9 NS-RLAB**

30 Non-sticky Reverse Link Assignment Block.. This block informs the access terminal that holds a  
 31 specific MACID that hop-ports specified by the NodeID field have been assigned to the access  
 32 terminal, and informs the access terminal, via the PF field, of the packet format to be used for  
 33 transmission on its assignment. The PF field is described in 7.7.6.8.

34 The duration of the assignment is specified in the duration field, resulting in a duration of 3 or 6 PHY  
 35 frames (standard or extended).

36 If the extended transmission field is equal to '0', an NS-RLAB assigns hop-ports for a particular  
 37 interlace consisting of standard PHY frames, and the duration field specifies the number of standard  
 38 PHY frames to be used for transmission for this assignment. If the Extended Transmission field is  
 39 equal to '1', an NS-RLAB assigns hop-ports for an interlace consisting of extended PHY frames, and

1 the duration field specifies the number of extended PHY frames to be used for transmission for this  
2 assignment.

3 The Delta field is used by the access terminal to determine the power at which to send reverse link  
4 data, as specified by the RTC MAC.

#### 5 **7.4.6.3.1.2.10 CCB**

6 ChangeCarrierBlock. This block only applies when multi-carrier mode is MultiCarrierOn. It notifies  
7 the access terminal that it should change the carriers on which it is operating. The  
8 ActiveCarriersChange field indicates the carriers on which the AT should now operate. This field is a  
9 four bit field. When the  $i^{\text{th}}$  bit is equal to '1', this indicates that the access terminal should operate on  
10 the  $i^{\text{th}}$  carrier. When this block is sent, ActiveCarriers, for the MACID specified in the MACID field,  
11 shall be updated to the value in the ActiveCarriersChange field.

12 The REQCarrierChange field notifies the access terminal that it should change the carrier on which it  
13 is sending requests to the carrier specified by the index value in this field. When this block is sent,  
14 REQCarrier, for the MACID specified in the MACID field, shall be updated to the value in the  
15 REQCarrierChange field. In addition, when this block is sent, other sectors in the access terminal's  
16 Active Set shall be notified of the updated values of the ActiveCarriers and REQCarrier for the access  
17 terminal.

#### 18 **7.4.6.3.1.2.11 FLAB-HO**

19 Forward Link Assignment Block Handoff. This block informs the access terminal that holds a  
20 specific MACID that hop-ports specified by the NodeID field have been assigned to the access  
21 terminal, for the sector indicated in the DesiredSector field. (If the NodeID provided is equal to  
22 NodeID<sub>DEASSIGN</sub>, then no resources are allocated). The DesiredSector field shall be interpreted as an  
23 active set index, and indicates that the FLSS will switch to the sector specified in this field. The PF  
24 field informs the access terminal of the packet format that should be used on its assigned hop-ports.  
25 The PF field is described in 7.5.6.7.1.

26 If the extended transmission field is equal to '0', an FLAB-HO assigns hop-ports for a particular  
27 interlace consisting of standard PHY Frames. If the Extended Transmission field is equal to '1', an  
28 FLAB-HO assigns hop-ports for an interlace consisting of extended PHY Frames.

#### 29 **7.4.6.3.1.2.12 RLAB-HO**

30 Reverse Link Assignment Block Handoff. This block informs the access terminal that holds a specific  
31 MACID that hop-ports specified by the NodeID field have been assigned to the access terminal, for  
32 the sector indicated in the DesiredSector field. (If the NodeID provided is equal to NodeID<sub>DEASSIGN</sub>,  
33 then no resources are allocated). The DesiredSector field shall be interpreted as an active set index,  
34 and indicates that the RLSS will switch to the sector specified in this field. The PF field informs the  
35 access terminal of the packet format to be used for transmission on its assignment. The PF field is  
36 described in 7.7.6.8.

37 If the extended transmission field is equal to '0', an RLAB-HO assigns hop-ports for a particular  
38 interlace consisting of standard PHY frames. If the Extended Transmission field is equal to '1', an  
39 RLAB-HO assigns hop-ports for an interlace consisting of extended PHY frames.

### 1 7.4.6.3.1.3 General rules for F-SSCH

2 In addition to the above blocks, the F-SSCH also transmits ACK bits for RL traffic, Power Control  
3 (PC) and CQI Erasure Indication (CEI) bits, and Fast OSI value.

4 Hop-ports allocated from different F-SSCH blocks constitute assignments of different types. Note  
5 however, that MCWFLAB1 and MCWFLAB2 allocate hop-ports for the same assignment type.  
6 Similarly, NS-MCWFLAB1 and NS-MCWFLAB2 allocate hop-ports for the same assignment type.

7 The access network should not send an assignment block to an access terminal with the supplemental  
8 bit set to '1' for an interlace where the access terminal has no assignment of the same type on that  
9 interlace, as such blocks will be ignored.

10 The access network should not send an assignment to an access terminal for an interlace, with the  
11 supplemental bit set to '1' which does not change the access terminal's assignment for that interlace;  
12 such blocks will be ignored. (See the FTC-MAC protocol and RTC-MAC protocol for assignment  
13 management rules.)

14 If the access network wants to assign more hop-ports (for an assignment of a single type) to an access  
15 terminal via F-SSCH on a single PHY Frame than can be specified by a single NodeID, the access  
16 network should send multiple Link Assignment Blocks of the same type to the access terminal's  
17 MACID. Except for the NodeID field, all of the fields in these Link Assignment Blocks should be  
18 identical. In the case of MCWFLABs, only 1 MCWFLAB2 need be sent, since this information is  
19 common to all the MCWFLAB1s that are sent to a particular user in the same PHY Frame. Similarly,  
20 for the case of NS-MCWFLABs, only 1 NS-MCWFLAB2 need be sent, since this information is  
21 common to all the NS-MCWFLAB1s that are sent to a particular user in the same PHY frame.

22 The access network should not transmit any F-SSCH blocks, ACK bits, or power control bits  
23 intended for an access terminal while TuneAwayStatus is equal to '1'. NS-FLABs with broadcast  
24 MACIDs may still be sent since these blocks are not directed to a particular access terminal.

25 If the access terminal has SelectedInterlaceMode set to '1', and SelectedInterlaceAssignment contains  
26 the PilotPN of the current sector, then the access network should only send blocks on F-SSCH to this  
27 access terminal over the interlaces specified in SelectedInterlaceAssignment. Additionally, the access  
28 network should only send assignment blocks to this access terminal if the Extended Transmission  
29 field is equal to '0'. NS-FLABs with broadcast MACIDs may still be sent on the restricted interlaces,  
30 and may have the Extended Transmission field set to '1', since these blocks are not directed to a  
31 particular access terminal.

32 If multi-carrier mode is MultiCarrierOn, the access network should not send any F-SSCH blocks,  
33 ACKs, or power control bits over F-SSCH intended for a particular MACID on carriers that are not  
34 marked as active carriers in the ActiveCarriers parameter of the public data for that MACID.

35 The AN should not send an RLAB-HO with the DesiredSector field set to a sector which is outside  
36 the synchronous subset to which the current RLSS belongs, unless the NodeID field of the RLAB-HO  
37 is NodeID<sub>DEASSIGN</sub>. Similarly, the AN should not send an FLAB-HO with the DesiredSector field set  
38 to a sector which is outside the synchronous subset to which the current FLSS belongs, unless the  
39 NodeID field of the FLAB-HO is NodeID<sub>DEASSIGN</sub>.

1 If the AN sends an access grant over SSCH, it should also send assignment block/blocks to  
 2 deassign/decrement the assignments of any access terminal/terminals which would otherwise be using  
 3 the hop-ports being assigned by the access grant.

4 If the AN sends an RLAB-HO over SSCH, it should ensure that assignment block/blocks are sent  
 5 from the sector specified in the DesiredSector field of the RLAB-HO, which would  
 6 deassign/decrement the assignments of any access terminal/terminals which would otherwise be using  
 7 the hop-ports being assigned by the RLAB-HO. Similarly, If the AN sends an FLAB-HO over SSCH,  
 8 it should ensure that assignment block/blocks are sent from the sector specified in the DesiredSector  
 9 field of the FLAB-HO, which would deassign/decrement the assignments of any access  
 10 terminal/terminals which would otherwise be using the hop-ports being assigned by the FLAB-HO.

#### 11 7.4.6.3.1.4 Framing F-SSCH blocks

12 The framing of blocks in SSCH packets follows a specific format as defined in 7.4.6.3.1.2. An SSCH  
 13 packet consists of a 4-bit header followed by the block, followed by reserved bits which are used to  
 14 make all the SSCH packets the same size. The PHY shall add the CRC to each packet.

15 **Table 69 F-SSCH Block Structure**

Field	Length (bits)
Header	4
Block	variable
Reserved	variable

16 Header A field that identifies the subsequent Block fields. See table in 7.4.6.3.1.2 for  
 17 Header values and the respective blocks and block lengths.

18 Block See table in 7.4.6.3.1.2.

19 Reserved The access network shall set this field to zero. The access terminal shall  
 20 ignore this field. The length of this field shall be such that the total number of  
 21 bits in the SSCH packets is  $10 + N_{\text{MACID}} + N_{\text{FL\_NODEID}} + N_{\text{FRID}} + 4$  header bits.

22 For each PHY Frame, the access network shall decide the specific blocks to be sent on SSCH and  
 23 shall compute the power density at which each SSCH block is transmitted, for use by the Physical  
 24 Layer. The framed blocks shall be passed to the Physical Layer in order of decreasing transmit power  
 25 density (blocks with higher transmit power density passed first). Algorithms for choosing which  
 26 blocks to be sent, and computing the transmit power density for each block are beyond the scope of  
 27 this specification.

#### 28 7.4.6.3.1.5 Procedures for sending an access grant

29 The access network should send an access grant message when it receives an  
 30 *AccessChannelMAC.AccessProbeReceived* indication.

31 If the AccessSequenceID placed in the public data of the AccessChannelMAC is the  
 32  $0, \dots, N_{\text{ACMPSpecialSequences}} - 1$  AccessSequenceID, then the AccessGrant shall contain the MACID that was  
 33 used to generate the Physical Layer sequence corresponding to the received access probe, and may

1 contain  $\text{NodeID}_{\text{DEASSIGN}}$ . Refer to the Access Channel MAC Protocol for the procedures for  
 2 determining  $\text{AccessSequenceID}$ .

3 If the  $\text{AccessSequenceID}$  placed in the public data of the  $\text{AccessChannelMAC}$  is higher than  
 4  $N_{\text{ACMPSpecialSequences}}-1$ , then the  $\text{AccessGrant}$  shall contain a MAC ID that is not in use in the system.  
 5 Based on the space to which the received  $\text{AccessSequenceID}$  corresponds, the access network may  
 6 decide the  $\text{NodeID}$  in the access grant. When the access grant is sent, the protocol shall generate a  
 7 *SharedSignalingMAC.AccessGrantSent* indication.

#### 8 **7.4.6.3.1.6 Procedures for the power control bits**

9 For each access terminal that contains the sector in the active set, and whose  $\text{MACID}$  is within  
 10  $\text{MACIDRange}$ , the access network shall transmit one CQI Erasure Indication (CEI) bit on F-SSCH  
 11 every  $\text{FLPCReportInterval}$  frames. The access network shall determine the value of the CEI bits for  
 12 all  $\text{MACIDs}$  within  $\text{MACIDRange}$ , and the power at which each is to be transmitted, for use by the  
 13 Physical Layer. The MAC Layer Protocol shall pass the CEI bits in order of increasing  $\text{MACID}$  to the  
 14 Physical Layer Protocol.

15 In addition to the CEI bits, if  $\text{RLCtrlPCMode}$  is set to ‘UpDown’, the access network shall transmit  
 16 one Power Control (PC) bit on F-SSCH every  $\text{FLPCReportInterval}$  frames, for each access terminal  
 17 whose  $\text{RLSS}$  is this sector, and whose  $\text{MACID}$  is within the  $\text{MACIDRange}$ . The access network shall  
 18 determine the value of the PC bits for all  $\text{MACIDs}$  within  $\text{MACIDRange}$ , and the power at which  
 19 each is to be transmitted, for use by the Physical Layer. The MAC Layer Protocol shall pass the PC  
 20 bits in order of increasing  $\text{MACID}$  to the Physical Layer Protocol.

21 The algorithms for determining the values of the CEI and PC bits and the power at which they are to  
 22 be transmitted are outside the scope of this specification.

#### 23 **7.4.6.3.1.7 Procedures for the Fast OSI value**

24 If  $\text{FastOSIEnabled}$  is set to ‘1’, then, in addition to the F-OSICH, the access network shall transmit  
 25 the OSI value on the F-SSCH, every PHY frame. The computation of the OSI value is beyond the  
 26 scope of this specification. If multi-carrier mode is  $\text{MultiCarrierOn}$ , then this protocol shall transmit a  
 27 separate value for each carrier.

#### 28 **7.4.6.3.1.8 Procedures for sending ACKs**

29 Acknowledgements for RL traffic are sent on F-SSCH. The timing for sending ACKs is given  
 30 in 7.1.3.

31 The access terminal’s  $\text{ACKNode}$  for a particular assignment on a particular interlace shall be the base  
 32 node mapped by the hop-port with the lowest numerical index in the access terminal’s assignment on  
 33 the carrier which has the most hop-ports for that assignment. If more than one carrier meets this  
 34 criteria, then the carrier in this subset that has the smallest carrier index shall be the carrier on which  
 35 the  $\text{ACKNode}$  is chosen.

36 Each PHY frame, the access network shall transmit ACKs intended for certain access terminals. In  
 37 order to positively acknowledge a particular access terminal, the Access network shall set the bit  
 38 corresponding to access terminal’s  $\text{ACKNode}$  to ‘1’, and shall set the bits corresponding to the rest of  
 39 the base nodes in the access terminal’s assignment to ‘0’. After the bits corresponding to the base  
 40 nodes for the users receiving acknowledgements have been set in this way, the bits corresponding to

1 all other base nodes shall be set to '0'. The MAC Layer Protocol shall pass to the Physical layer  
 2 protocol the sequence of bits corresponding to all base nodes: the base nodes for each carrier shall be  
 3 ordered from left to right, and the carriers shall be ordered via increasing carrier index. The access  
 4 network shall also determine the power used to transmit each of these bits, for use by the Physical  
 5 Layer. Algorithms for determining this power are outside the scope of this specification. In addition,  
 6 for every positive acknowledgement, the MAC shall pass to the Physical layer a valid MACID  
 7 corresponding to the acknowledged AT which is needed for MACID scrambling.

#### 8 **7.4.6.3.1.9 Procedures for maintaining ActiveCarriers and REQCarriers at the AN**

9 The access network shall initialize ActiveCarriers for a particular MACID to be the carrier specified  
 10 by AccessCarrier.

11 When the access network sends a CCB block in the F-SSCH to a particular access terminal,  
 12 ActiveCarriers (for the MACID specified in the MACID field of the CCB) shall be updated to the  
 13 value in the ActiveCarriersChange field of the CCB. The value of ActiveCarriers for a particular  
 14 access terminal may also change if another sector indicates the necessary change.

15 The access network shall initialize REQCarrier for a particular MACID to be the carrier specified by  
 16 AccessCarrier.

17 When the access network sends a CCB block in the F-SSCH to a particular access terminal,  
 18 REQCarrier (for the MACID specified in the MACID field of the CCB) shall be updated to the value  
 19 in the REQCarrierChange field of the CCB. The value of REQCarrier for a particular access terminal  
 20 may also change if another sector indicates the necessary change.

#### 21 **7.4.6.3.2 Access terminal requirements**

22 When this protocol is in active state, the following apply:

23 If the access terminal has an outstanding access probe and the AccessSequenceID is  
 24  $N_{ACMPSpecialSequences}/2$  through  $N_{ACMPSpecialSequences}-1$ , then the access terminal shall, at least, demodulate  
 25 all F-SSCH packets, as well as the relevant ACK and power control bits, in the F-SSCH of the access  
 26 terminal's DFLSS each PHY Frame. If the access terminal has no outstanding access probe with  
 27 AccessSequenceID  $N_{ACMPSpecialSequences}/2$  through  $N_{ACMPSpecialSequences}-1$ , then the access terminal shall  
 28 demodulate all F-SSCH packets, as well as the relevant ACK and power control bits, in the F-SSCHs  
 29 of its FLSS, DFLSS, RLSS, and DRLSS (see 7.6.6.3 for definitions of FLSS, DFLSS, RLSS, DFLSS)  
 30 every PHY Frame. An access terminal that has a DFLSS in a different synchronous subset than its  
 31 FLSS, and is not capable of demodulating the F-SSCH in both its DFLSS and its FLSS, should  
 32 immediately issue an access probe with AccessSequenceID  $N_{ACMPSpecialSequences}/2$  through  
 33  $N_{ACMPSpecialSequences}-1$ .

34 If multi-carrier mode equals MultiCarrierOn, the access terminal shall demodulate F-SSCH packets,  
 35 ACKs and power control bits, on those carriers specified by its ActiveCarriers parameter in the public  
 36 data.

37 If the access terminal has its parameter TuneAwayStatus in the public data of the Connected State  
 38 Protocol set to '1', the access terminal need not monitor any F-SSCH.

39 The following are potential error events and specified actions:

- 40 ■ Each block that does not correctly pass CRC shall be discarded.

1 The access terminal shall perform the following actions on the remaining blocks/bits:

- 2 ■ If the access terminal's ACK bit has an ACK value of 1, then the access terminal has  
3 received a positive ACK signal for its assignment on the corresponding interlace.  
4 Otherwise, the access terminal has received an implicit negative ACK signal for the  
5 assignment on the corresponding interlace. Actions contingent upon the reception of a  
6 positive or negative acknowledgement shall be performed as specified in the RTC MAC  
7 Protocol.
- 8 ■ The access terminal shall determine its transmit power based on its power control bits  
9 received on F-SSCH as specified in the RCC MAC Protocol.
- 10 ■ For each sector in the active set, if FastOSIEnabled is set to '1', the access terminal shall  
11 monitor the Fast OSI value and provide the received values to the Reverse Traffic  
12 Channel MAC Protocol.
- 13 ■ For each sector in the active set, if an AccessGrant block is received, the access terminal  
14 shall:
  - 15 □ If all the following are true
    - 16 – the access terminal has no assigned MACID on the sector,
    - 17 – and the AccessSequenceID used to scramble the AccessGrant matches the  
18 AccessSequenceID in the public data of the Access Channel MAC Protocol,
    - 19 – and the PilotPN of the sector that transmitted the AccessGrant matches the  
20 PilotPN in the public data of the Access Channel MAC Protocol,
    - 21 – and if multi-carrier mode is set to MultiCarrierOn, the carrier on which the  
22 AccessGrant is received matches the AccessCarrier in the public data of the  
23 Access Channel MAC Protocol,
    - 24 the access terminal shall:
      - 25 – Return an *AccessGrantReceived* indication specifying the applicable sector.
      - 26 – Place the received AccessGrant in the public data.
      - 27 – In the Physical Layer, adjust the timing for the sector that transmitted the access  
28 grant.
      - 29 – Instruct the RCC MAC to initialize the transmit power as described in 7.6.6.3.5.
  - 30 □ If all of the following are true:
    - 31 – the MAC ID in the AccessGrant matches a MACID assigned to the access  
32 terminal on the sector,
    - 33 – the AccessSequenceID used to scramble AccessGrant matches the  
34 AccessSequenceID in the public data of the Access Channel MAC Protocol,
    - 35 – the PilotPN of the sector that transmitted the AccessGrant matches the PilotPN in  
36 the public data of the Access Channel MAC Protocol
    - 37 – and if multi-carrier mode is set to MultiCarrierOn, the carrier on which the  
38 AccessGrant is received matches the AccessCarrier in the public data of the  
39 Access Channel MAC Protocol,

1 the access terminal shall:

- 2 – Return an *AccessGrantReceived* indication specifying the applicable sector.
- 3 – Place the received AccessGrant in the public data.
- 4 – In the Physical Layer, adjust the timing for the sector that transmitted the access
- 5 grant.
- 6 – Instruct the RCC MAC to initialize the transmit power as described in 7.6.6.3.5.

7 □ The access terminal shall invoke the procedures for processing an access grant in the  
8 access terminal Assignment Management section of the RTC MAC Protocol, unless  
9 the NodeID field is equal to NodeID<sub>DEASSIGN</sub>, in which case the access grant will not  
10 be passed to the RTC MAC Protocol for further processing.

- 11 ■ If an FLAB is received, the access terminal shall invoke the procedures for processing an  
12 FLAB in the access terminal Assignment Management section of the FTC MAC  
13 Protocol.
- 14 ■ If a NS-FLAB is received, the access terminal shall invoke the procedures for processing  
15 a NS-FLAB in the access terminal Assignment Management section of the FTC MAC  
16 Protocol.
- 17 ■ If an RLAB is received, the access terminal shall invoke the procedures for processing an  
18 RLAB in the access terminal Assignment Management sections of the RTC MAC  
19 Protocol.
- 20 ■ If an NS-RLAB is received, the access terminal shall invoke the procedures for  
21 processing an NS-RLAB in the access terminal Assignment Management sections of the  
22 RTC MAC Protocol.
- 23 ■ If an MCWFLAB is received (part 1 or part 2), the access terminal shall invoke the  
24 procedures for processing a MCWFLAB in the access terminal Assignment Management  
25 section of the FTC MAC Protocol.
- 26 ■ If an NS-MCWFLAB is received (part 1 or part 2), the access terminal shall invoke the  
27 procedures for processing a NS-MCWFLAB in the access terminal Assignment  
28 Management section of the FTC MAC Protocol.
- 29 ■ If a SCWFLAB is received, the access terminal shall invoke the procedures for  
30 processing an SCWFLAB in the access terminal Assignment Management section of the  
31 FTC MAC Protocol.
- 32 ■ If a NS-SCWFLAB is received, the access terminal shall invoke the procedures for  
33 processing an NS-SCWFLAB in the access terminal Assignment Management section of  
34 the FTC MAC Protocol.
- 35 ■ If an FLAB-HO is received, the access terminal shall invoke the procedures for  
36 processing an FLAB-HO in the access terminal Assignment Management section of the  
37 FTC MAC Protocol.
- 38 ■ If an RLAB-HO is received, the access terminal shall invoke the procedures for  
39 processing an RLAB-HO in the access terminal Assignment Management section of the  
40 RTC MAC Protocol.



### 7.4.6.3.2.1 Procedures for maintaining ActiveCarriers and REQCarrier at the AT

The access terminal shall initialize ActiveCarriers to be the carrier specified by AccessCarrier.

The access terminal shall initialize REQCarrier to be the carrier specified by AccessCarrier I.

If multi-carrier mode is equal to MultiCarrierOff, then the access terminal shall ignore all CCB blocks sent on F-SSCH.

If multi-carrier mode is equal to MultiCarrierOn, then the following apply:

When the access terminal receives a CCB block in the F-SSCH, the access terminal shall update ActiveCarriers to the value in the ActiveCarriersChange field of the CCB.

When the access terminal receives a CCB block in the F-SSCH, REQCarrier shall be updated to the value in the REQCarrierChange field of the CCB.

### 7.4.6.3.3 Supervision Procedures

The access terminal shall return a *SupervisionFailed* indication if the F-SSCH channel is not received for a time period of length  $T_{SSCMSupervision}$ .

### 7.4.7 Header and trailer formats

This protocol does not carry higher-layer payload, and thus header and trailer formats are not defined.

### 7.4.8 Interface to other protocols

#### 7.4.8.1.1 Commands

This protocol does not issue any commands.

#### 7.4.8.1.2 Indications

This protocol registers to receive the following indication:

- *AccessChannelMAC.AccessProbeReceived*

### 7.4.9 Configuration attributes

No configuration attributes are defined for this protocol.

### 7.4.10 Protocol numeric constants

Constant	Meaning	Value
$N_{SSCMPType}$	Type field for this protocol	Table 9
$N_{SSCMPDefault}$	Subtype field for this protocol	0x0000
$MACID_{BROADCAST}$	Broadcast MACID	0x0
$T_{SSCMSupervision}$	Supervision timer	1 s

### 7.4.11 Session state information

The Session State Information record (see 10.10) consists of parameter records.

The parameter records for this protocol consist of the configuration attributes of this protocol.

## 7.5 Default Forward Traffic Channel MAC Protocol

### 7.5.1 Overview

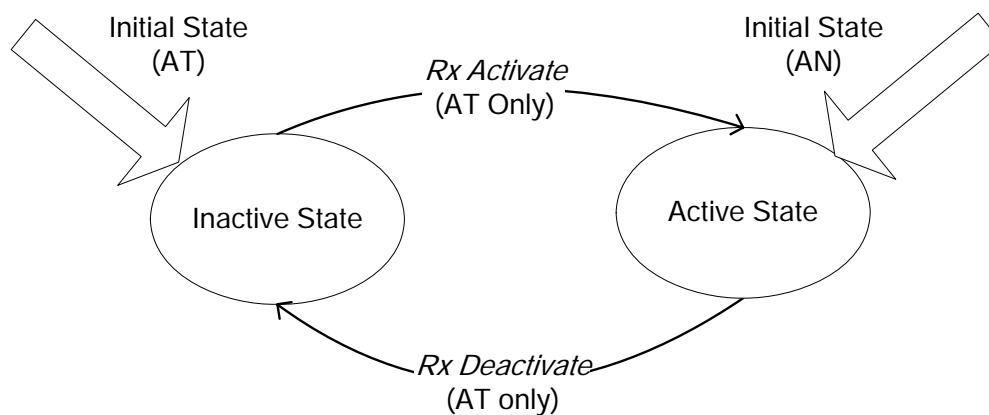
The Default Forward Traffic Channel MAC Protocol provides the procedures and messages required for an access network to transmit, and an access terminal to receive, the Forward Traffic Channel.

The access network maintains an instance of this protocol for every assigned MAC ID.

This protocol at the access terminal operates in one of two states:

- *Inactive State*: In this state, the access terminal cannot receive any packets on the Forward Traffic Channel. When the protocol is in this state, it waits for an *Activate* command.
- *Active State*: In this state, the access terminal receives the forward traffic channel based on link assignment blocks received from SS MAC.

The protocol states and allowed transitions between the states are shown in Figure 75. The rules governing these transitions are provided in 7.5.6.3 and 7.5.6.4.



**Figure 75 Default Forward Traffic Channel MAC Protocol state diagram**

1 This protocol shall use the following parameters and attributes:  
2

Parameter Name	Where Defined	Comments
FLChannelTreeIndex	OMP	QuickChannelInfo block
FLNumSDMADimensions	OMP	QuickChannelInfo block
FLImplicitDeassignEnabled	Connected State Protocol	Public data
SelectedInterlaceMode	Connected State Protocol	Public data
TuneAwayStatus	Connected State Protocol	Public data
FLSS	RCC MAC Protocol	Public data
DFLSS	RCC MAC Protocol	Public data
MultiCarrierOn	Physical Layer Protocol	Public data

## 3 7.5.2 Primitives

### 4 7.5.2.1 Commands

5 This protocol defines the following commands:

- 6 ■ *Activate*
- 7 ■ *Deactivate*

### 8 7.5.2.2 Return indications

9 This protocol returns the following indication:

- 10 ■ *ForwardTrafficPacketsMissed*
- 11 ■ *SupervisionFailed*
- 12 ■ *PageReceived*
- 13 ■ *UATIReceived*
- 14 ■ *SessionLost*

## 15 7.5.3 Public data

### 16 7.5.3.1 Static public data

17 This protocol does not define any static public data.

### 18 7.5.3.2 Dynamic public data

- 19 ■ Subtype for this protocol.

## 1 **7.5.4 Protocol data unit**

2 The transmission unit of this protocol is a Forward Traffic Channel Lower MAC Sublayer packet.  
3 Each packet consists of one Security Sublayer packet.

## 4 **7.5.5 Protocol initialization and swap**

### 5 **7.5.5.1 Protocol initialization**

6 Upon initialization at the access terminal,

- 7 ■ The values of the attributes for this protocol instance shall be set to the default values  
8 specified for each attribute.
- 9 ■ The protocol shall enter the Inactive State.

10 Upon initialization at the access network.

- 11 ■ The values of the attributes for this protocol instance shall be set to the default values  
12 specified for each attribute.
- 13 ■ The protocol shall enter the Active State.

### 14 **7.5.5.2 Protocol Swap**

15 Upon swap, the protocol instance shall enter the Inactive State.

## 16 **7.5.6 Procedures**

17 This protocol constructs a Forward Traffic Channel Lower MAC Sublayer packet out of the Security  
18 Sublayer packet by adding the Lower MAC Sublayer header and trailer as defined in 7.5.7. The  
19 protocol then sends the packet for transmission to the Physical Layer.

### 20 **7.5.6.1 Command processing**

#### 21 **7.5.6.1.1 Activate**

22 If this protocol receives an *Activate* command in the Inactive State, the access terminal and the access  
23 network shall transition to the Active State.

24 If this protocol receives the command in any other state, the command shall be ignored.

#### 25 **7.5.6.1.2 Deactivate**

26 If the protocol receives a *Deactivate* command in the Active State:

- 27 ■ The access terminal shall cease monitoring the Forward Traffic Channel and shall  
28 transition to the Inactive State.
- 29 ■ The access network shall ignore the command.

30 If this command is received in the Inactive State, the command shall be ignored.

## 1 7.5.6.2 Forward traffic channel addressing

2 Transmission on the Forward Traffic Channel is multiplexed in time and frequency. An assignment  
3 on the Forward Traffic Channel shall be specified by a set of hop-ports and an interlace. Each hop-  
4 port is specified by a hop-port index as well as a carrier index. If the duplex mode is FDD, then the  
5 interlace may be composed of standard PHY Frames or extended PHY Frames, (as specified by the  
6 assignment blocks received from the SS MAC protocol). Extended PHY Frames are defined  
7 in 7.1.3.1.3.

8 The duration of an assignment of hop-ports may or may not be pre-specified. Assignments whose  
9 durations are pre-specified are known as non-sticky assignments, and assignments whose durations  
10 are not pre-specified are known as sticky assignments.

11 The set of hop-ports assigned for a particular interlace for a particular access terminal via sticky  
12 assignment blocks (received from SS MAC) is referred to as a “Forward Link Access Terminal  
13 Assignment” or FL-ATA. Sticky assignment blocks include the following three types: FLABs (and  
14 FLAB-HOs), SCWFLABs, and MCWFLABs. (See 7.4.6.3.1 for complete definitions of these block  
15 types.) A FL-ATA for a particular interlace may only be assigned via a single type of sticky  
16 assignment block. A FL-ATA that has been assigned via a particular type of assignment block is also  
17 associated with that type. More precisely, a FL-ATA may be of type FLAB, SCWFLAB, or  
18 MCWFLAB. The access terminal may have multiple FL-ATAs, one for each nonoverlapping  
19 interlace (note that overlapping interlaces can be created only by the use of extended PHY Frames).

20 A non-sticky assignment block with a unicast MACID field (referred to as a non-sticky unicast  
21 assignment block) assigns hop-ports for a particular access terminal for a particular interlace. The set  
22 of hop-ports assigned for a particular access terminal for a particular interlace via non-sticky unicast  
23 assignment blocks (received from SS MAC) is referred to as a “Forward Link Non-Sticky Access  
24 Terminal Unicast Assignment” or FL-NS-ATA<sub>UC</sub>. Non-sticky unicast assignment blocks include the  
25 following three types: NS-FLABs, NS-SCWFLABs, and NS-MCWFLABs (all with unicast MACID  
26 fields). See 7.4.6.3.1 for complete definitions of these block types. A FL-NS-ATA<sub>UC</sub> for a particular  
27 interlace may only be assigned via a single type of non-sticky unicast assignment block. A  
28 FL-NS-ATA<sub>UC</sub> that has been assigned via a particular type of assignment block is also associated with  
29 that type. More precisely, a FL-NS-ATA<sub>UC</sub> may be of type NS-FLAB, NS-SCWFLAB, or  
30 NS-MCWFLAB. An access terminal may have multiple FL-NS-ATA<sub>UC</sub>s, one for each  
31 nonoverlapping interlace.

32 A non-sticky assignment block with a broadcast MACID field (referred to as a non-sticky broadcast  
33 assignment block) assigns hop-ports for multiple access terminals. The set of hop-ports used by a  
34 particular access terminal on a particular interlace, assigned via non-sticky broadcast assignment  
35 blocks is referred to as a “Forward Link Non Sticky Access Terminal Broadcast Assignment” or  
36 FL-NS-ATA<sub>BC</sub>. Non-sticky broadcast assignment blocks include only NS-FLABs. An access terminal  
37 may have multiple FL-NS-ATA<sub>BC</sub>s, one for each nonoverlapping interlace on each active carrier.

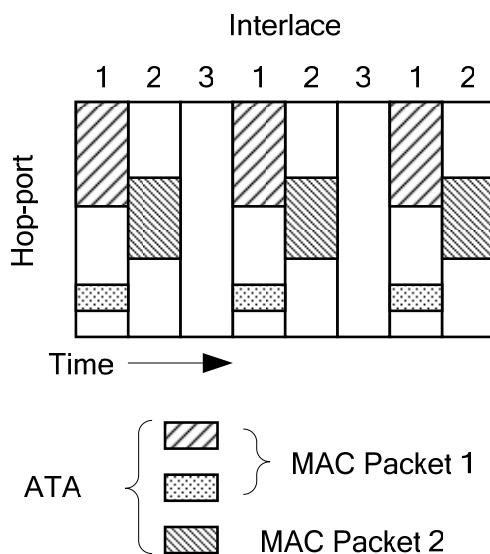
38 Sets of hop-ports assigned in assignment blocks received from SS MAC are specified using the  
39 channel tree that is currently in use for the FL. The channel tree that is in use is specified by  
40 FLChannelTreeIndex. Channel trees are specified in 7.5.6.6.

41 Packets transmitted over the Forward Traffic Channel are transmitted over the F-DCH physical layer  
42 channel. F-DCH is the primary channel for user data transmission, and access terminals are assigned  
43 F-DCH resources (FL-ATAs/-FL-NS-ATAs) via assignment blocks that are sent over the F-SSCH.

1 The following rules apply regarding the coexistence of FL-ATAs/FL-NS-ATA<sub>BC</sub>s/FL-NS-ATA<sub>UC</sub>s:

- 2 ■ An access terminal shall have no more than one FL-ATA per interlace. If duplex mode is  
3 FDD, then additionally, an access terminal shall not have any FL-ATAs that overlap in  
4 time.
- 5 ■ An access terminal shall have no more than one FL-NS-ATA<sub>UC</sub> per interlace. If duplex  
6 mode is FDD, then additionally, an access terminal shall not have any FL-NS-ATA<sub>UC</sub>s  
7 that overlap in time.
- 8 ■ An access terminal shall have no more than one FL-NS-ATA<sub>BC</sub> per interlace per carrier.  
9 If duplex mode is FDD, then additionally, an access terminal shall not have any  
10 FL-NS-ATA<sub>BC</sub>s that overlap in time on the same carrier.
- 11 ■ An access terminal shall not have an a non-empty FL-ATA and a non-empty  
12 FL-NS-ATA<sub>UC</sub> in the same interlace. If duplex mode is FDD, then additionally, an access  
13 terminal shall not have a non-empty FL-ATA and a non-empty FL-NS-ATA<sub>UC</sub> that  
14 overlap in time.

15 The FL-ATA for an interlace can be accumulated via multiple assignment blocks of the same type,  
16 sent over multiple PHY Frames, as specified in 7.5.6.4.1.1. The FL-NS-ATA<sub>UC</sub> for an interlace may  
17 be accumulated over multiple assignment blocks of the same type, but these blocks must be sent in  
18 the same PHY Frame, as specified in 7.5.6.4.1.1.1. Similarly, The FL-NS-ATA<sub>BC</sub> for an interlace may  
19 be accumulated over multiple assignment blocks of the same type, but these blocks must be sent in  
20 the same PHY frame, as specified in 7.5.6.4.1.1.1. All hop-ports in an FL-ATA/FL-NS-ATA<sub>UC</sub>/  
21 FL-NS-ATA<sub>BC</sub> for a particular interlace shall be combined for transmission over the Physical Layer  
22 channel (F-DCH). Different interlaces shall always carry separate MAC packets with independent  
23 H-ARQ termination. An example is illustrated in Figure 76.



24  
25 **Figure 76 F-DCH addressing example**

### 1 **7.5.6.3 Inactive state**

2 When the protocol is in the Inactive State, the access terminal waits for an *Activate* command.

3 The Inactive State is not defined for the Access Network.

### 4 **7.5.6.4 Active state**

5 In the Active State, the access network shall transmit over the F-DCH using the FL-ATAs or  
6 FL-NS-ATAs, and PFs selected by the access network and signaled to the access terminal over the  
7 F-SSCH Physical Layer channel. The access terminal processes assignment blocks from the SS MAC  
8 protocol to maintain its FL-ATAs/FL-NS-ATAs, configures the Physical Layer for reception of  
9 packets according to the ATAs, and controls transmission of ACK/NACK information via the RCC  
10 MAC Protocol based on pass or fail of the MAC packet, as indicated by the PHY. The MACID  
11 assigned to the access terminal for each sector in its active set shall be given in the  
12 ActiveSetAssignment message that is public data of the Active Set Management Protocol.  
13 Assignment and H-ARQ logic for the access terminal and access network are specified in 7.5.6.4.1  
14 and 7.5.6.4.2, respectively.

15 Note that an FL-NS-ATA shall be used for transmission of a single Lower MAC packet (see 7.1.2).

#### 16 **7.5.6.4.1 Access terminal requirements**

17 An access terminal shall keep a variable NumLayers for each interlace, which shall be initialized to  
18 zero when this protocol enters the Active state. On reception of a SCWFLAB/NS-SCWFLAB for a  
19 particular interlace from the SS MAC protocol that is not discarded due to assignment management  
20 rules given in 7.5.6.4.1.2, NumLayers for that interlace shall be set to the NumLayers field of the  
21 SCWFLAB/NS-SCWFLAB. On reception of a MCWFLAB/NS-MCWFLAB for a particular  
22 interlace from the SS MAC protocol that is not discarded due to assignment management rules given  
23 in 7.5.6.4.1.2, NumLayers for that interlace shall be set to the number of non-zero packet formats  
24 given in the MCWFLAB/NS-MCWFLAB. NumLayers for a particular interlace shall also be  
25 modified to reflect rank adjustments sent in the FL MAC header.

##### 26 **7.5.6.4.1.1 Access terminal assignment management for sticky assignments**

27 In this section, the term FLAB will be used to indicate all types of sticky FL assignment blocks,  
28 including FLAB, MCWFLAB, and SCWFLAB, unless otherwise specified. (FLAB-HOs are  
29 discussed in 7.5.6.4.1.3) Similarly, the term NS-FLAB will be used to indicate all types of non-sticky  
30 FL assignment blocks, including NS-FLAB, NS-MCWFLAB, and NS-SCWFLAB, unless otherwise  
31 specified.

32 The access terminal shall maintain and manage its FL-ATAs by monitoring FLABs and NS-FLABs  
33 delivered from the SS MAC protocol.

34 The logic in this section assumes that all of the FLABs/NS-FLABs are being sent from the same  
35 serving sector, the FLSS (see 7.6.6.3 for the definition of FLSS). The logic for access terminal  
36 assignment management during handoff is found in 7.5.6.4.1.3.

37 If FLImplicitDeassignEnabled is equal to '1', then upon receiving a *TunedAway* indication from the  
38 Connected State Protocol, the access terminal shall expire all its FL-ATAs.

1 If SelectedInterlaceMode is equal to '1', then the access terminal shall ignore all FLABs that have the  
2 Extended Transmission field set to '1'.

3 If the extended transmission field is equal to '0', an FLAB assigns an interlace consisting of standard  
4 PHY Frames, as shown in 7.1.3.1.2. If the Extended Transmission field is equal to '1', an FLAB  
5 assigns an interlace consisting of extended PHY Frames, as shown in 7.1.3.1.3.

#### 6 **7.5.6.4.1.1 Simultaneous assignments**

7 If duplex mode is FDD, FLABs received in the same PHY frame could be for different interlaces  
8 because some of the FLABs could be for the interlace consisting of standard PHY frames, and some  
9 of the FLABs could be for the interlace consisting of extended PHY frames. If multiple FLABs for  
10 the access terminal's MACID are received in the same PHY frame from SS MAC, and they are not all  
11 for the same interlace, then the access terminal shall assume an error has occurred, and shall ignore all  
12 of these FLABs, unless there is an FLAB with NodeID set to NodeID<sub>DEASSIGN</sub>, in which case this  
13 FLAB shall not be ignored.

14 If multiple FLABs for the access terminal's MACID are received from the SS MAC Protocol in the  
15 same PHY frame, for the same interlace, and one of the FLABs has a NodeIDs set to NodeID<sub>DEASSIGN</sub>,  
16 then all FLABs except for the latter shall be discarded. This rule trumps all those which follow in this  
17 section.

18 If multiple FLABs of the same type, which have the access terminal's MACID, are received from the  
19 SS MAC protocol in the same PHY Frame, for the same interlace, and all the FLABs have the same  
20 values in all the fields except the NodeID field, then the access terminal shall treat these FLABs as a  
21 single FLAB assigning the union of the hop-ports mapped by the constituent NodeIDs.

22 If multiple FLABs of the same type, which have the access terminal's MACID, are received from the  
23 SS MAC protocol in the same PHY Frame, for the same interlace, and if the values in at least one of  
24 the fields (excluding the NodeID field) are not the same, then the access terminal shall assume an  
25 error has occurred, and shall ignore all of these FLABs.

26 If multiple FLABs, which have the access terminal's MACID, are received from the SS MAC  
27 protocol in the same PHY frame, for the same interlace, and all of the FLABs are not of the same  
28 type, the access terminal shall assume an error has occurred, and shall ignore all of these FLABs.

#### 29 **7.5.6.4.1.1.2 Supplemental and non-supplemental assignments**

30 The following assumes that an FLAB/FLABs have been received from the SS MAC with the MACID  
31 matching that of the access terminal.

32 If the Supplemental field of an FLAB for a particular interlace is equal to '1', then the new FL-ATA  
33 on that interlace shall be the union of hop-ports included in the old FL-ATA on that interlace and the  
34 hop-ports specified by the new NodeID, provided the old FL-ATA and the new FLAB are of the same  
35 type, and the old FL-ATA is non-empty. The PF specified in the received FLAB shall be used in  
36 place of any PFs that may have been specified in any previous assignment of NodeIDs (hop-ports) on  
37 the interlace.

38 If the Supplemental field of an FLAB for a particular interlace is equal to '0', then the FL-ATA for  
39 the relevant interlace shall be cleared before adding the hop-ports specified by the NodeID in the  
40 FLAB to the FL-ATA for the interlace.



1 If the duplex mode is FDD, and the supplemental field of an FLAB for a particular interlace is equal  
 2 to '0', then the following apply. If the extended transmission field of the FLAB is equal to '1', all  
 3 FL-ATAs shall be expired except for the FL-ATA, should there be one, which is an extended  
 4 transmission duration assignment and does not overlap in time with the new assignment. The hop-  
 5 ports specified by the NodeID in the FLAB shall then be given to the FL-ATA in the corresponding  
 6 interlace.

7 If the duplex mode is FDD, and the supplemental field of an FLAB, for a particular interlace, is equal  
 8 to '0', and if the extended transmission field of the FLAB is equal to '0', all FL-ATAs that are  
 9 extended transmission duration assignments shall be expired. The hop-ports specified by the NodeID  
 10 in the FLAB shall be given to the FL-ATA in the corresponding interlace.

11 If an FLAB is received for a particular interlace with the supplemental field set to '1' when the  
 12 FL-ATA is empty on that interlace, then the access terminal shall ignore this FLAB.

13 If an FLAB is received for a particular interlace with the supplemental field set to '1', and the current  
 14 FL-ATA on that interlace has a different type then the FLAB, the access terminal shall ignore this  
 15 FLAB.

16 If an FLAB is received for a particular interlace with the supplemental field set to '1' that does not  
 17 change the access terminal's FL-ATA for that interlace, then the access terminal shall ignore this  
 18 FLAB.

#### 19 **7.5.6.4.1.1.3 Decrementing Assignments**

20 If duplex mode is TDD, then the access terminal shall decrement its FL-ATAs as follows. If an FLAB  
 21 or NS-FLAB<sub>UC</sub> is received that contains a MACID other than the access terminal's MACID with an  
 22 assignment for a particular interlace, then all of the hop-ports in the FL-ATA for that interlace that  
 23 intersect with hop-ports specified by the NodeID included in the FLAB/NS-FLAB<sub>UC</sub> shall be expired  
 24 (removed from the FL-ATA) for that interlace.

25 If duplex mode is FDD, then the access terminal shall decrement its FL-ATAs as follows: If an FLAB  
 26 or NS-FLAB<sub>UC</sub> is received that contains a MACID other than the access terminal's MACID with an  
 27 assignment for a particular interlace, any FL-ATA whose interlace overlaps in time with this  
 28 interlace, and whose hop ports intersect with the hop ports assigned by the FLAB/NS-FLAB<sub>UC</sub>, shall  
 29 have its overlapping hop-ports removed from the FL-ATA.

30 If the access terminal receives, in the same PHY frame, an FLAB with its MACID, and an  
 31 NS-FLAB<sub>UC</sub> with a MACID other than its MACID, for the same interlace, and the hop ports assigned  
 32 by the FLAB intersect with the hop ports assigned by the NS-FLAB<sub>UC</sub>, then the access terminal shall  
 33 expire the intersecting hop ports from its FL-ATA.

34 If duplex mode is FDD, the following applies. If the access terminal receives, in the same PHY frame,  
 35 an FLAB with its MACID, and an NS-FLAB<sub>UC</sub> with a MACID other than its MACID, for time  
 36 overlapping interlaces, and the hop ports assigned by the FLAB intersect with the hop ports assigned  
 37 by the NS-FLAB<sub>UC</sub>, then the access terminal shall expire the intersecting hop ports from its FL-ATA.

38 If the access terminal receives, in the same PHY frame, an FLAB with its MACID, and an FLAB  
 39 with a MACID other than its MACID, for the same interlace, and the hop ports assigned by the  
 40 FLABs intersect, then the access terminal shall ignore the FLAB with its MACID.

1 If duplex mode is FDD, the following applies. If the access terminal receives, in the same PHY frame,  
 2 an FLAB with its MACID, and an FLAB with a MACID other than its MACID, for time overlapping  
 3 interlaces, and the hop ports assigned by the FLABs intersect, then the access terminal shall ignore  
 4 the FLAB with its MACID.

#### 5 **7.5.6.4.1.1.4 Deassigning Assignments**

6 If an FLAB is received for a particular interlace, for the access terminal's MACID, that assigns the  
 7 reserved NodeID<sub>DEASSIGN</sub>, then the FL-ATA on that interlace shall be expired.

#### 8 **7.5.6.4.1.1.5 Overlapping NS broadcast assignments and sticky assignments**

9 If an access terminal receives (from SS MAC) a NS-FLAB<sub>BC</sub> for a particular interlace such that its  
 10 FL-NS-ATA<sub>BC</sub> on that interlace overlaps with its (sticky) FL-ATA on that interlace, then the access  
 11 terminal shall update its FL-NS-ATA<sub>BC</sub> according to the NS-FLAB<sub>BC</sub>, and shall remove any  
 12 overlapping hop-ports (between the FL-NS-ATA<sub>BC</sub> and FL-ATA) from the FL-ATA on that interlace  
 13 for the duration of that particular FL-NS-ATA<sub>BC</sub>.

14 If duplex mode is FDD, the following apply. If an access terminal receives (from SS MAC) a NS-  
 15 FLAB<sub>BC</sub> for a particular interlace such that its FL-NS-ATA<sub>BC</sub> on that interlace overlaps in frequency  
 16 with a (sticky) FL-ATA on a time overlapping interlace, then the access terminal shall update its FL-  
 17 NS-ATA<sub>BC</sub> according to the NS-FLAB<sub>BC</sub>, and shall remove any overlapping hop-ports (between the  
 18 FL-NS-ATA<sub>BC</sub> and FL-ATA) from the FL-ATA for the duration of that particular FL-NS-ATA<sub>BC</sub>.

19 If an access terminal receives (from SS MAC) a NS-FLAB<sub>BC</sub>, and an FLAB in the same PHY Frame,  
 20 such that its FL-NS-ATA<sub>BC</sub> overlaps with its FL-ATA on that interlace, then the access terminal shall  
 21 update its FL-NS-ATA<sub>BC</sub> according to the NS-FLAB<sub>BC</sub>, and shall remove any overlapping hop-ports  
 22 (between the FL-NS-ATA<sub>BC</sub> and FL-ATA) from the FL-ATA on that interlace for the duration of that  
 23 particular FL-NS-ATA<sub>BC</sub>.

24 If duplex mode is FDD, the following apply. If an access terminal receives (from SS MAC) a  
 25 NS-FLAB<sub>BC</sub> for a particular interlace, and an FLAB in the same PHY Frame, such that its  
 26 FL-NS-ATA<sub>BC</sub> overlaps in frequency with the FL-ATA on a time overlapping interlace, then the  
 27 access terminal shall update its FL-NS-ATA<sub>BC</sub> according to the NS-FLAB<sub>BC</sub>, and shall remove any  
 28 overlapping hop-ports (between the FL-NS-ATA<sub>BC</sub> and FL-ATA) from the FL-ATA for the duration  
 29 of that particular FL-NS-ATA<sub>BC</sub>.

#### 30 **7.5.6.4.1.1.6 Time overlapping NS unicast and sticky assignments**

31 If the access terminal receives a unicast NS-FLAB of any type with its MACID for a particular  
 32 interlace, while it already has a non-empty FL-ATA on that interlace, the access terminal shall keep  
 33 the resulting FL-NS-ATA<sub>UC</sub> and clear the FL-ATA on that interlace.

34 Furthermore, if duplex mode is FDD, and if the access terminal receives a unicast NS-FLAB of any  
 35 type with its MACID, that has the Extended Transmission field set to '1', the access terminal shall  
 36 expire all FL-ATAs that it has, unless the FL-ATA is also an extended transmission duration  
 37 assignment, and occupies an interlace whose extended PHY Frame does not overlap with the  
 38 extended PHY Frame of the new non-sticky unicast assignment. If the access terminal gets an  
 39 NS-FLAB<sub>UC</sub> with the Extended Transmission field set to '0', then all FL-ATAs that are extended  
 40 duration assignments shall be expired.

1 If the access terminal receives, in the same PHY Frame, a NS-FLAB with its MACID, giving it a  
 2 non-empty FL-NS-ATA<sub>UC</sub>, and an FLAB giving it a non-empty FL-ATA, then it shall ignore the  
 3 FLAB.

#### 4 **7.5.6.4.1.1.7 Multi-code word assignments**

5 If the access terminal receives a MCW-FLAB1 with its MACID from the SS MAC Protocol with the  
 6 supplemental field set to '0', then there should be a MCW-FLAB2 with its MACID received in the  
 7 same PHY Frame: otherwise, the access terminal shall ignore the MCW-FLAB1. If the access  
 8 terminal receives a MCW-FLAB1 with its MACID from the SS MAC protocol with the supplemental  
 9 bit set to '1', then if no MCW-FLAB2 with its MACID is received in the same PHY Frame, the  
 10 access terminal shall use the current packet formats for layers 2 and above (provided there is an  
 11 existing FL-ATA of type MCWFLAB). Otherwise, if a MCWFLAB2 with its MACID is received in  
 12 the same PHY frame, the access terminal shall update the packet formats for layers 2 and above  
 13 according to the MCWFLAB2.

#### 14 **7.5.6.4.1.1.8 Inband packet format switch**

15 If the access terminal receives an InBandPacketFormatSwitch message, it shall update its packet  
 16 format for the specified interlace, and use this updated packet format to try to decode all subsequent  
 17 packets on that interlace, provided the InBandPacketFormatSwitch message is consistent with the  
 18 assignment type on that interlace. If the InBandPacketFormatSwitch message is not consistent with  
 19 the assignment type on that interlace, the access terminal shall ignore the InBandPacketFormatSwitch  
 20 message.

#### 21 **7.5.6.4.1.2 Access terminal assignment management for non-sticky assignments**

22 In this section the term NS-FLAB will be used to indicate all types of non-sticky FL assignment  
 23 blocks, including NS-FLAB, NS-MCWFLAB, and NS-SCWFLAB, unless otherwise specified. The  
 24 term NS-FLAB shall also be used for both broadcast and unicast NS-FLABs, unless otherwise  
 25 specified. Note that unicast NS-FLABs may be any type, but broadcast NS-FLABs may not be NS-  
 26 SCWFLABs or NS-MCWFLABs. Additionally, the term FLAB will be used to indicate all types of  
 27 sticky FL assignment blocks, including FLAB, MCWFLAB, and SCWFLAB, unless otherwise  
 28 specified.

29 The access terminal shall maintain and manage its FL-NS-ATAs by monitoring FLABs and NS-  
 30 FLABs delivered from the SS MAC protocol. (Unless otherwise specified, FL-NS-ATA includes both  
 31 FL-NS-ATA<sub>BCS</sub> and FL-NS-ATA<sub>UCS</sub>).

32 The logic in this section assumes that all of the FLABs/NS-FLABs are being sent from the FLSS. The  
 33 logic for access terminal assignment management during handoff is found in 7.5.6.4.1.3.

34 If FLImplicitDeassignEnabled is equal to '1', then upon receiving a *TunedAway* indication from the  
 35 Connected State Protocol, the access terminal shall expire all its FL-NS-ATAs.

36 If SelectedInterlaceMode is equal to '1', then the access terminal shall ignore all NS-FLABs that have  
 37 the Extended Transmission field set to '1'.

38 When the access terminal receives a non-sticky forward link assignment block with broadcast  
 39 MACID (NS-FLAB<sub>BC</sub>) from the SS MAC Protocol for a particular interlace on a particular carrier,  
 40 the access terminal shall then temporarily maintain a FL-NS-ATA<sub>BC</sub> (non-sticky broadcast

1 assignment) for that interlace and carrier. When the access terminal receives, from the SS MAC  
 2 Protocol, a non-sticky forward link assignment block with unicast MACID (NS-FLAB<sub>UC</sub>) for a  
 3 particular interlace, where the MACID matches its MACID, the access terminal shall temporarily  
 4 maintain a FL-NS-ATA<sub>UC</sub> (non-sticky access terminal unicast assignment) for that interlace.

5 If the extended transmission field is equal to '0', an NS-FLAB assigns hop-ports for a particular  
 6 interlace consisting of standard PHY Frames as shown in 7.1.3.1.2. The duration field specifies the  
 7 number of standard PHY Frames to be used for transmission of this assignment. If the Extended  
 8 Transmission field is equal to '1', an NS-FLAB assigns hop-ports for an interlace consisting of  
 9 extended PHY Frames, as shown in 7.1.3.1.3. The duration field specifies the number of extended  
 10 PHY Frames to be used for transmission of this assignment.

11 If multi-carrier mode is equal to MultiCarrierOn, the access terminal may maintain a FL-NS-ATA<sub>BC</sub>  
 12 for each non-overlapping interlace for each active carrier. Active Carriers are specified by the  
 13 ActiveCarriers parameter in the public data of the SS MAC Protocol.

14 The access terminal shall give up its FL-NS-ATA<sub>BC</sub> for a particular interlace after it has kept it for the  
 15 assignment duration specified in the corresponding NS-FLAB<sub>BC</sub>. Similarly, the access terminal shall  
 16 give up its FL-NS-ATA<sub>UC</sub> for a particular interlace after it has kept it for the assignment duration  
 17 specified in the corresponding NS-FLAB<sub>UC</sub>.

#### 18 **7.5.6.4.1.2.1 Simultaneous assignments**

19 If duplex mode is FDD, NS-FLABs received in the same PHY frame could be for different interlaces  
 20 because some of the NS-FLABs could be for the interlace consisting of standard PHY frames, and  
 21 some of the NS-FLABs could be for the interlace consisting of extended PHY frames. If multiple  
 22 NS-FLAB<sub>UCS</sub> for the access terminal's MACID are received in the same PHY frame from SS MAC,  
 23 and they are not all for the same interlace, then the access terminal shall assume an error has  
 24 occurred, and shall ignore all of these NS-FLAB<sub>UCS</sub>, unless there is an NS-FLAB<sub>UC</sub> with NodeID set  
 25 to NodeID<sub>DEASSIGN</sub>, in which case this NS-FLAB<sub>UC</sub> shall not be ignored. If multiple NS-FLAB<sub>BCS</sub> for  
 26 the access terminal's MACID are received from SS MAC in the same PHY frame, for the same  
 27 carrier, and the NS-FLAB<sub>BCS</sub> are not all for the same interlace, then the access terminal shall assume  
 28 an error has occurred, and shall ignore all of these NS-FLAB<sub>BCS</sub>.

29 If multiple NS-FLAB<sub>UCS</sub> for the access terminal's MACID are received from the SS MAC Protocol in  
 30 the same PHY frame, and one of the NS-FLAB<sub>UCS</sub> has NodeID set to NodeID<sub>DEASSIGN</sub>, then all the  
 31 NS-FLAB<sub>UCS</sub> except for the latter shall be discarded. This rule trumps all those which follow in this  
 32 section.

33 If multiple NS-FLAB<sub>BCS</sub> are received from the SS MAC Protocol in the same PHY Frame and carrier,  
 34 and the values of all the fields, except for the NodeID field, are the same, the access terminal shall  
 35 treat these NS-FLAB<sub>BCS</sub> as a single NS-FLAB<sub>BC</sub>, assigning the union of the hop-ports mapped by the  
 36 constituent NodeIDs. If multiple NS-FLAB<sub>BCS</sub> are received from the SS MAC protocol in the same  
 37 PHY Frame and carrier, and if all the fields (other than the NodeID field) are not the same, the access  
 38 terminal shall consider an error has occurred and ignore these blocks.

39 If multiple NS-FLAB<sub>UCS</sub> of the same type are received from the SS MAC Protocol in the same PHY-  
 40 frame, for the access terminal's MACID, and the values of all the fields, except for the NodeID field,  
 41 are the same, the access terminal shall treat these NS-FLAB<sub>UCS</sub> as a single NS-FLAB<sub>UC</sub> assigning the  
 42 union of the hop-ports mapped by the constituent NodeIDs. If multiple NS-FLAB<sub>UCS</sub> of the same type  
 43 are received from the SS MAC protocol in the same PHY Frame, for the access terminal's MACID,

1 and if all the fields (other than the NodeID field) are not the same, the access terminal shall consider  
2 an error has occurred and shall ignore these blocks.

3 If multiple NS-FLAB<sub>UCS</sub> are received from the SS MAC Protocol in the same PHY frame for the  
4 access terminal's MACID, and these NS-FLAB<sub>UCS</sub> are not of the same type, then the access terminal  
5 shall consider an error has occurred and ignore these blocks.

#### 6 **7.5.6.4.1.2.2 Deassigning Assignments**

7 If an NS-FLAB for a particular interlace is received with the access terminal's MACID, and assigns  
8 the reserved NodeID<sub>DEASSIGN</sub>, then the FL-NS-ATA on that interlace shall be expired.

#### 9 **7.5.6.4.1.2.3 Time Overlapping broadcast assignments**

10 If a NS-FLAB<sub>BC</sub> is received from the SS MAC Protocol for a particular interlace and carrier, and the  
11 access terminal already has a FL-NS-ATA<sub>BC</sub> for that interlace and carrier, then the new assignment  
12 block takes precedence: the access terminal shall stop trying to decode on the old FL-NS-ATA<sub>BC</sub> for  
13 that interlace (shall clear this FL-NS-ATA<sub>BC</sub>), and shall update its FL-NS-ATA<sub>BC</sub> for that interlace  
14 according to the new NS-FLAB<sub>BC</sub>.

15 If duplex mode is FDD, the following apply. If an NS-FLAB<sub>BC</sub> is received from the SS MAC  
16 Protocol for a particular carrier that has the Extended Transmission field set to '1', then all  
17 FL-NS-ATA<sub>BCS</sub> on that carrier shall be expired, unless there is an FL-NS-ATA<sub>BC</sub> that is an extended  
18 transmission duration assignment whose extended PHY Frames do not overlap with those of the new  
19 assignment. If an NS-FLAB<sub>BC</sub> is received from the SS MAC protocol for a particular carrier that has  
20 the Extended Transmission field set to '0', then all FL-NS-ATA<sub>BCS</sub> on that carrier that are extended  
21 transmission duration assignments shall be expired. The access terminal shall update its  
22 FL-NS-ATA<sub>BC</sub> according to the new NS-FLAB<sub>BC</sub>.

#### 23 **7.5.6.4.1.2.4 Time Overlapping unicast assignments**

24 If a NS-FLAB<sub>UC</sub> is received from the SS MAC protocol for a particular interlace with the access  
25 terminal's MACID, and the access terminal already has a FL-NS-ATA<sub>UC</sub> for that interlace, then the  
26 new assignment block takes precedence: the access terminal shall stop trying to decode on the old  
27 FL-NS-ATA<sub>UC</sub> for that interlace (shall clear this FL-NS-ATA<sub>UC</sub>), and shall update its FL-NS-ATA<sub>UC</sub>  
28 for that interlace according to the new FL-NS-FLAB<sub>UC</sub>.

29 If duplex mode is FDD, the following apply. If an NS-FLAB<sub>UC</sub> is received from the SS MAC  
30 Protocol, that has the access terminal's MACID and has the Extended Transmission field set to '1',  
31 then all FL-NS-ATA<sub>UCS</sub> shall be expired, unless there is an FL-NS-ATA<sub>UC</sub> that is an extended  
32 transmission duration assignment whose extended PHY Frames do not overlap with those of the new  
33 assignment. If an NS-FLAB<sub>UC</sub> is received that has the access terminal's MACID, and has the  
34 Extended Transmission field set to '0', then all FL-NS-ATA<sub>UCS</sub> that are extended transmission  
35 duration assignments shall be expired. The access terminal shall update its FL-NS-ATA<sub>UC</sub> according  
36 to the new NS-FLAB<sub>UC</sub>.

#### 1 **7.5.6.4.1.2.5 Overlapping broadcast and unicast assignments**

2 If the access terminal receives a NS-FLAB<sub>BC</sub> from the SS MAC protocol for a particular interlace  
3 such that its FL-NS-ATA<sub>BC</sub> on that interlace overlaps with its FL-NS-ATA<sub>UC</sub> on that interlace, then  
4 the most recent assignment block takes precedence; that is, the access terminal shall expire the old  
5 FL-NS-ATA<sub>UC</sub> assignment, and update its FL-NS-ATA<sub>BC</sub> according to the new NS-FLAB<sub>BC</sub> .

6 Furthermore, if duplex mode is FDD, and if the access terminal receives an NS-FLAB<sub>BC</sub> from the SS  
7 MAC protocol with the Extended Transmission field set to '1', the access terminal shall expire all  
8 FL-NS-ATA<sub>UCS</sub> that overlap in frequency with the FL-NS-ATA<sub>BC</sub>, unless the FL-NS-ATA<sub>UC</sub> is also  
9 an extended transmission duration assignment, and occupies an interlace whose extended PHY Frame  
10 does not overlap with the extended PHY Frame of the new non-sticky broadcast assignment. If the  
11 access terminal receives an NS-FLAB<sub>BC</sub> from the SS MAC protocol with the Extended Transmission  
12 field set to '0', then all FL-NS-ATA<sub>UCS</sub> that are extended duration assignments and overlap in  
13 frequency with the FL-NS-ATA<sub>BC</sub> shall be expired.

14 If the access terminal receives a NS-FLAB<sub>UC</sub> with its MACID, from the SS MAC protocol, for a  
15 particular interlace such that its FL-NS-ATA<sub>UC</sub> on that interlace overlaps with its FL-NS-ATA<sub>BC</sub> on  
16 that interlace, then the most recent assignment takes precedence, that is, the access terminal shall  
17 expire the old assignment, the FL-NS-ATA<sub>BC</sub>, and update its FL-NS-ATA<sub>UC</sub> according to the new  
18 NS-FLAB<sub>UC</sub>

19 Furthermore, if duplex mode is FDD, and if the access terminal receives an NS-FLAB<sub>UC</sub> with its  
20 MACID from the SS MAC protocol with the Extended Transmission field set to '1', the access  
21 terminal shall expire all FL-NS-ATA<sub>BCS</sub> that overlap in frequency with the FL-NS-ATA<sub>UC</sub>, unless the  
22 FL-NS-ATA<sub>BC</sub> is also an extended transmission duration assignment, and occupies an interlace whose  
23 extended PHY Frame does not overlap with the extended PHY Frame of the new non-sticky unicast  
24 assignment. If the access terminal receives an NS-FLAB<sub>UC</sub> with its MACID from the SS MAC  
25 Protocol with the Extended Transmission field set to '0', then all FL-NS-ATA<sub>BCS</sub> that are extended  
26 transmission duration assignments and overlap in frequency with the FL-NS-ATA<sub>UC</sub> shall be expired.

27 If the access terminal receives a NS-FLAB<sub>BC</sub> and a NS-FLAB<sub>UC</sub> in the same PHY frame for the same  
28 interlace, and the corresponding FL-NS-ATA<sub>UC</sub> and FL-NS-ATA<sub>BC</sub> overlap in frequency, then the  
29 access terminal shall assume an error has occurred, and shall ignore the NS-FLAB<sub>UC</sub>.

30 If duplex mode is FDD, then if the access terminal receives a NS-FLAB<sub>BC</sub> and a NS-FLAB<sub>UC</sub> in the  
31 same PHY frame for time overlapping interlaces, and the corresponding FL-NS-ATA<sub>UC</sub> and  
32 FL-NS-ATA<sub>BC</sub> overlap in frequency, then the access terminal shall assume an error has occurred, and  
33 shall ignore the NS-FLAB<sub>UC</sub>.

#### 34 **7.5.6.4.1.2.6 Time Overlapping NS unicast and sticky assignments**

35 If an access terminal receives an FLAB with its MACID from the SS MAC protocol for a particular  
36 interlace while it already has a FL-NS-ATA<sub>UC</sub> on that interlace, then it shall give up the  
37 FL-NS-ATA<sub>UC</sub> on that interlace and only try to decode on the FL-ATA for that interlace.

38 Furthermore, if duplex mode is FDD, and if the access terminal receives an FLAB with its MACID  
39 from the SS MAC protocol with the Extended Transmission field set to '1', the access terminal shall  
40 expire all FL-NS-ATA<sub>UCS</sub> that it has, unless the FL-NS-ATA<sub>UC</sub> is also an extended transmission  
41 duration assignment, and occupies an interlace whose extended PHY Frame does not overlap with the  
42 extended PHY Frame of the new assignment. If the access terminal receives an FLAB with its

1 MACID from the SS MAC protocol with the Extended Transmission field set to '0', then all  
2 FL-NS-ATA<sub>UCS</sub> that are extended duration assignments shall be expired.

### 3 **7.5.6.4.1.2.7 Overlapping NS broadcast and sticky assignments**

4 If duplex mode is TDD, the following apply. If the Access terminal gets an FLAB such that its  
5 updated FL-ATA on an interlace overlaps with its FL-NS-ATA<sub>BC</sub> for that interlace, the access  
6 terminal shall expire the FL-NS-ATA<sub>BC</sub>.

7 If duplex mode is FDD, the following apply. If the Access terminal gets an FLAB such that its  
8 updated FL-ATA on an interlace overlaps (in time and frequency) with a FL-NS-ATA<sub>BC</sub>, the access  
9 terminal shall expire the FL-NS-ATA<sub>BC</sub>.

### 10 **7.5.6.4.1.2.8 Overlapping assignments from other ATs**

11 If an FLAB or NS-FLAB<sub>UC</sub> is received from the SS MAC protocol that contains a MACID other than  
12 the access terminal's MACID with an assignment for a particular interlace, and if the hop-ports  
13 specified by the NodeID in the FLAB/NS-FLAB<sub>UC</sub> intersect with the access terminal's FL-NS-ATA  
14 <sub>UC</sub> on that interlace, then the access terminal shall expire its FL-NS-ATA<sub>UC</sub> for that interlace.

15 Furthermore, if duplex mode is FDD, and if an FLAB/NS-FLAB<sub>UC</sub> is received from the SS MAC  
16 protocol that contains a MACID other than the access terminal's MACID and has the Extended  
17 Transmission field set to '1', the access terminal shall expire all FL-NS-ATA<sub>UCS</sub> whose hop-ports  
18 overlap with the new assignment, unless the FL-NS-ATA<sub>UC</sub> is also an extended transmission duration  
19 assignment, and occupies an interlace whose extended PHY Frame does not overlap with the  
20 extended PHY Frame of the new assignment. If an FLAB/NS-FLAB<sub>UC</sub> is received from the SS MAC  
21 protocol that contains a MACID other than the access terminal's MACID and has the Extended  
22 Transmission field set to '0', then all FL-NS-ATA<sub>UCS</sub> that are extended transmission duration  
23 assignments, and have hop-ports that overlap with the new assignment, shall be expired.

24 If an FLAB or NS-FLAB<sub>UC</sub> is received from the SS MAC protocol that contains a MACID other than  
25 the access terminal's MACID with an assignment for a particular interlace, and if the hop-ports  
26 specified by the NodeID in the FLAB/NS-FLAB<sub>UC</sub> intersect with the access terminal's FL-NS-ATA  
27 <sub>BC</sub> on that interlace for a particular carrier, then the access terminal shall expire its FL-NS-ATA<sub>BC</sub> for  
28 that interlace and carrier.

29 Furthermore, if duplex mode is FDD, and if an FLAB/NS-FLAB<sub>UC</sub> is received from the SS MAC  
30 protocol that contains a MACID other than the access terminal's MACID and has the Extended  
31 Transmission field set to '1', the access terminal shall expire all FL-NS-ATA<sub>BCS</sub> whose hop-ports  
32 overlap with the new assignment, unless the FL-NS-ATA<sub>BC</sub> is also an extended transmission duration  
33 assignment, and occupies an interlace whose extended PHY Frame does not overlap with the  
34 extended PHY Frame of the new assignment. If an FLAB/NS-FLAB<sub>UC</sub> is received from the SS MAC  
35 protocol that contains a MACID other than the access terminal's MACID and has the Extended  
36 Transmission field set to '0', then all FL-NS-ATA<sub>BCS</sub> that are extended transmission duration  
37 assignments, and have hop-ports that overlap with the new assignment, shall be expired.

38 If the access terminal receives, in the same PHY frame, an NS-FLAB<sub>BC</sub> with its MACID and an NS-  
39 FLAB<sub>UC</sub> with a MACID other than its MACID, for the same interlace, and the hop ports assigned by  
40 the NS-FLAB<sub>UC</sub> overlap with the hop ports assigned by the NS-FLAB<sub>BC</sub>, the access terminal assumes  
41 as error has occurred, and shall ignore the NS-FLAB<sub>UC</sub>.

1 If duplex mode is FDD, the following holds. If the access terminal receives, in the same PHY frame,  
 2 an NS-FLAB<sub>BC</sub> with its MACID and an NS-FLAB<sub>UC</sub> with a MACID other than its MACID, for time  
 3 overlapping interlaces, and the hop ports assigned by the NS-FLAB<sub>UC</sub> overlap with the hop ports  
 4 assigned by the NS-FLAB<sub>BC</sub>, the access terminal assumes as error has occurred, and shall ignore the  
 5 NS-FLAB<sub>UC</sub>.

6 If the access terminal receives, in the same PHY frame, an NS-FLAB<sub>UC</sub> with its MACID and an  
 7 NS-FLAB<sub>UC</sub> with a MACID other than its MACID, for the same interlace, and the hop ports assigned  
 8 by the NS-FLAB<sub>UC</sub>s overlap, the access terminal assumes as error has occurred, and shall ignore the  
 9 NS-FLAB<sub>UC</sub> with its MACID.

10 If duplex mode is FDD, the following holds. If the access terminal receives, in the same PHY frame,  
 11 an NS-FLAB<sub>UC</sub> with its MACID and an NS-FLAB<sub>UC</sub> with a MACID other than its MACID, for time  
 12 overlapping interlaces, and the hop ports assigned by the NS-FLAB<sub>UC</sub>s overlap, the access terminal  
 13 assumes as error has occurred, and shall ignore the NS-FLAB<sub>UC</sub> with its MACID.

14 If the access terminal receives, in the same PHY frame, an NS-FLAB<sub>BC</sub>, and an FLAB with a MACID  
 15 other than the access terminal's MACID, for the same interlace, and the hop ports assigned by the  
 16 NS-FLAB<sub>BC</sub> intersect with the hop ports assigned by the FLAB, the access terminal shall ignore the  
 17 FLAB (in the sense that the NS-ATA<sub>BC</sub> assigned by the NS-FLAB<sub>BC</sub> will not be modified or expired  
 18 because of the FLAB.)

19 If duplex mode is FDD, the following applies. If the access terminal receives, in the same PHY frame,  
 20 an NS-FLAB<sub>BC</sub>, and an FLAB with a MACID other than the access terminal's MACID, for time  
 21 overlapping interlaces, and the hop ports assigned by the NS-FLAB<sub>BC</sub> intersect with the hop ports  
 22 assigned by the FLAB, the access terminal shall ignore the FLAB (in the sense that the NS-ATA<sub>BC</sub>  
 23 assigned by the NS-FLAB<sub>BC</sub> will not be modified or expired because of the FLAB).

24 If the access terminal receives, in the same PHY frame, an NS-FLAB<sub>UC</sub> with its MACID, and an  
 25 FLAB with a MACID other than the access terminal's MACID, for the same interlace, and the hop  
 26 ports assigned by the NS-FLAB<sub>UC</sub> intersect with the hop ports assigned by the FLAB, the access  
 27 terminal shall ignore the FLAB (in the sense that the NS-ATA<sub>UC</sub> assigned by the NS-FLAB<sub>UC</sub> will not  
 28 be modified or expired because of the FLAB).

29 If duplex mode is FDD, the following applies. If the access terminal receives, in the same PHY frame,  
 30 an NS-FLAB<sub>UC</sub> with its MACID, and an FLAB with a MACID other than the access terminal's  
 31 MACID, for time overlapping interlaces, and the hop ports assigned by the NS-FLAB<sub>UC</sub> intersect  
 32 with the hop ports assigned by the FLAB, the access terminal shall ignore the FLAB (in the sense that  
 33 the NS-ATA<sub>UC</sub> assigned by the NS-FLAB<sub>UC</sub> will not be modified or expired because of the FLAB).

#### 34 **7.5.6.4.1.3 Access terminal assignment management during handoff**

35 If an *FLSSChanged* indication from the RCC MAC protocol is received, the access terminal shall  
 36 clear all FL-ATAs and FL-NS-ATAs associated with the old FLSS.

37 If the access terminal receives an FLAB-HO with its MACID, and the sector in the DesiredSector  
 38 field is different from the FLSS, but in the same synchronous subset as the FLSS, the access terminal  
 39 shall issue a *ReverseControlChannelMAC.ChangeFLSS* to change the FLSS to the sector in the  
 40 DesiredSector field of the FLAB-HO. If the access terminal then receives an *FLSSChanged* indication  
 41 from the RCC MAC Protocol, the access terminal shall process the *FLSSChanged* indication as  
 42 specified above, and in addition shall update its new FL-ATA to the hop ports specified by the



1 NodeID field in the FLAB-HO for the interlace specified by the ExtendedTransmission field in the  
 2 FLAB-HO, for the sector specified in the DesiredSector field. The packet format to be used is  
 3 specified in the PF field of the FLAB-HO.

4 If the access terminal receives an FLAB-HO with its MACID, and the sector in the DesiredSector  
 5 field is not different from the FLSS, or is in a different synchronous subset as the FLSS, the access  
 6 terminal shall ignore this FLAB-HO.

7 If the access terminal receives multiple FLAB-HOs with its MACID, and all the fields are the same  
 8 except for the NodeID fields, then the access terminal shall treat these FLAB-HOs as a single FLAB-  
 9 HO assigning the union of the hop ports specified by the NodeID fields.

10 If the access terminal receives multiple FLAB-HOs with its MACID, and all the fields other than the  
 11 NodeID field are not the same, then the access terminal shall ignore these FLAB-HOs.

12 If the access terminal receives an FLAB/NS-FLAB with the access terminal's MACID, that has the  
 13 supplemental field set to '0', from the DFLSS, while the DFLSS is different from the FLSS, the  
 14 access terminal shall issue a *ReverseControlChannelMAC.ChangeFLSS* command to change from the  
 15 FLSS to the DFLSS. If the access terminal then receives an *FLSSChanged* indication from the RCC  
 16 MAC Protocol, the access terminal shall process the *FLSSChanged* indication as specified above, and  
 17 in addition shall update the appropriate FL-ATA/FL-NS-ATA according to the new FLAB/NS-  
 18 FLAB.

19 The access terminal shall ignore all FLABs or NS-FLABs that come from sectors other than its  
 20 current FLSS or its DFLSS.

#### 21 **7.5.6.4.1.4 Access terminal assignment processing for sticky assignments**

22 The access terminal shall attempt to decode a MAC packet on each interlace with a non-empty  
 23 FL-ATA (contains one or more hop-ports). The access terminal may attempt to detect erasure  
 24 sequences that are transmitted by the access network whenever a MAC packet is not available for  
 25 transmission. Note that if duplex mode is FDD, erasures may be sent on standard PHY Frames or  
 26 extended PHY Frames. Exact algorithms for detecting erasure sequences and start-of-packet for MAC  
 27 packets that span multiple PHY Frames are beyond the scope of this specification.

28 When an access terminal's FL-ATA on an interlace changes, and the duration of the change is not  
 29 pre-specified (as would be the case if the FL-ATA was punctured due to a FL-NS-ATA<sub>BC</sub>), the access  
 30 terminal shall terminate any existing packet decoding on the interlace and restart packet decoding  
 31 attempts on the interlace (as long as the new ATA is non-empty).

32 When an access terminal's FL-ATA on an interlace changes, and the duration of the change is pre-  
 33 specified, as is the case if the FL-ATA is punctured due to a FL-NS-ATA<sub>BC</sub>, the access terminal shall  
 34 continue to decode as usual, except that the FL-ATA is reduced for the specified duration.

35 If a MAC packet on the FL-ATA is successfully decoded, as indicated by the PHY, the access  
 36 terminal shall reset the supervision timer and transmit a positive ACK via the RCC MAC to the  
 37 access network. Refer to 7.1.3 for detailed interlace structure and acknowledgment timing for both  
 38 FDD and TDD modes. The payload of the packet is then passed up to the Security Sublayer for  
 39 further processing.

1 If a MAC packet on the FL-ATA fails to decode successfully, as indicated by the PHY, and the  
 2 access terminal determines that the packet has been transmitted for the maximum number of PHY  
 3 Frames for the relevant PF (see 7.5.6.7), then the access terminal shall expire the FL-ATA for the  
 4 interlace in which the packet failed. For an access terminal that is in MIMO MCW mode, the  
 5 maximum number of PHY Frames is the maximum over all of the specified PFs. Exact algorithms to  
 6 estimate the number of H-ARQ packet transmissions that have been sent prior to successful decode  
 7 are beyond the scope of this specification.

#### 8 **7.5.6.4.1.5 Access terminal assignment processing for non-sticky assignments**

9 For each interlace with a non-empty FL-NS-ATA<sub>UC</sub>, the access terminal may attempt to decode a  
 10 MAC packet. The access terminal shall not attempt to detect erasure sequences for FL-NS-ATA<sub>UC</sub>. If  
 11 a MAC packet on the FL-NS-ATA<sub>UC</sub> is successfully decoded, as indicated by the PHY, the access  
 12 terminal shall reset the supervision timer, transmit a positive ACK via the RCC MAC to the access  
 13 network. The access terminal may choose to expire its FL-NS-ATA<sub>UC</sub> at this time if it is not  
 14 performing ACK to NACK detection. After sending an ACK, the access terminal may choose to  
 15 expire its FL-NS-ATA<sub>UC</sub> earlier than the time the assignment is set to expire (this time is derived  
 16 from the duration field in the assignment block). Refer to 7.1.3 for detailed interlace structure and  
 17 acknowledgment timing for both FDD and TDD modes. The payload of the packet is then passed up  
 18 to the Security Sublayer for further processing.

19 For each active carrier, for each interlace with a non-empty FL-NS-ATA<sub>BC</sub>, the access terminal may  
 20 attempt to decode a MAC packet. The access terminal shall not attempt to detect erasure sequences  
 21 for FL-NS-ATA<sub>BC</sub>. Even if a MAC packet on the FL-NS-ATA<sub>BC</sub> is successfully decoded, as indicated  
 22 by the PHY, the access terminal shall not transmit a positive ACK message via the RCC MAC to the  
 23 access network.

#### 24 **7.5.6.4.1.6 Header Processing**

25 If a MAC packet on the FL-NS-ATA<sub>UC</sub> is successfully decoded, as indicated by the PHY, the packet  
 26 shall be processed according to the following rules:

- 27 ■ The access terminal shall parse the packet according to the header and trailer specified for  
 28 unicast transmissions.
- 29 ■ The access terminal shall pass the payload of the packet to the Security Sublayer for  
 30 further processing
- 31 ■ If the UATIInfoIncluded header field of the packet is equal to '1',
  - 32 □ The access terminal shall generate a *UATIReceived* indication accompanied by the  
 33 contents of the MAC header.
  - 34 □ If the FailureCode header field of the packet is set to SessionLost, the access terminal  
 35 shall generate a *SessionLost* indication.

36 If a MAC packet on the FL-NS-ATA<sub>BC</sub> is successfully decoded, as indicated by the PHY, the packet  
 37 shall be processed according to the following rules:

- 38 ■ The access terminal shall parse the packet according to the header and trailer specified for  
 39 broadcast transmissions.

- 1           ■ The access terminal shall forward the Security Sublayer packet to the Security Sublayer if  
2 either of the following two conditions are met:
- 3           □ If the ATIType field and the ATI field of the first ATI Record in the Lower MAC  
4 Sublayer header is equal to the ATIType and ATI fields of any member of the  
5 Address Management Protocol's ReceiveATIList.
- 6           □ If the ATIType of any of the ATI Record in the MAC Layer header of a Security  
7 Sublayer packet is equal to '00' (i.e., BATI) and the ReceiveATIList includes a  
8 record with ATIType set to '00'.
- 9           ■ If the ATIType field and the ATI field of any ATI Record in the Lower MAC Sublayer  
10 header of a is equal to the ATIType and ATI fields of any member of the Address  
11 Management Protocol's ReceiveATIList, and the OpenConnectionRequired field  
12 associated with any of the matching ATI field is set to "1", then this protocol shall return  
13 an *PageReceived* indication. This indicates that the access terminal has been paged.
- 14          ■ Otherwise, the access terminal shall discard the received packet.

#### 15 **7.5.6.4.2 Access network requirements**

16 The access network shall keep a variable NumLayers for each interlace, which shall be initialized to  
17 zero when this protocol enters the Active state. On transmission of a SCWFLAB/NS-SCWFLAB for  
18 a particular interlace via the SS MAC protocol, NumLayers for that interlace shall be set to the  
19 NumLayers field of the SCWFLAB/NS-SCWFLAB. On transmission of a MCWFLAB/NS-  
20 MCWFLAB for a particular interlace via the SS MAC protocol, NumLayers for that interlace shall be  
21 set to the number of non-zero packet formats given in the MCWFLAB/NS-MCWFLAB. NumLayers  
22 for a particular interlace shall also be modified to reflect rank adjustments sent in the FL MAC  
23 header.

##### 24 **7.5.6.4.2.1 Access network assignment management**

25 If FLImplicitDeassignEnabled is equal to '1', then upon receiving a *TunedAway* indication from the  
26 Connected State Protocol, the access network shall expire all the FL-ATAs and FL-NS-ATAs for that  
27 access terminal.

28 If the FLNumSDMADimensions > 1, then the access network shall ensure that assignments to  
29 different ATs that contain hop-ports that map to the same subcarriers have the same F-DPICH format,  
30 as described in 7.5.6.7.

##### 31 **7.5.6.4.2.2 F-DCH transmissions associated with sticky assignments**

32 The access network may formulate and transmit a MAC packet on an interlace according to the  
33 FL-ATA on the interlace and the PF selected for transmission. The MAC shall pass the Physical  
34 Layer Protocol the set of hop ports specified by the FL-ATA, and shall specify whether or not the  
35 assignment is an Extended Transmission Duration Assignment (see 7.1.3.1.3). The access network  
36 informs the access terminal of its assignment using signaling blocks as described in 7.4.6.3.1.2.

37 If TuneAwayStatus is equal to '1', then the access network shall not send any MAC packets on any  
38 ATAs on any interlaces to this access terminal.

1 If a positive ACK addressed to the access terminal's MACID is received that corresponds to the  
 2 transmitted packet, transmission of the MAC packet shall terminate, and the interlace is immediately  
 3 available for the next packet. (Note that for multi-code word assignments assigned by MCW-FLABs,  
 4 the ACKs for all the layers need to be received before the transmission of the MAC packet  
 5 terminates, and the interlace is available for the next packet.) If the access network transmits a packet  
 6 for the maximum number of transmissions of the PF selected for this MAC packet without receiving  
 7 an ACK, the access network shall expire the FL-ATA for the interlace, and shall return a  
 8 *ForwardTrafficPacketsMissed* indication along with parameters that uniquely identify the lost packet.  
 9 The method of uniquely identifying the packet is out of the scope of this specification.

10 The access terminal should send no more than one MAC packet over a FL-ATA per interlace. If two  
 11 interlaces overlap in time, the access terminal should only send a MAC packet over a FL-ATA on one  
 12 of the interlaces.

13 If no packet is available for transmission in a given PHY Frame (standard or extended), then an  
 14 erasure sequence shall be transmitted on the FL-ATA, or a subset of the FL-ATA, for that PHY  
 15 Frame (as specified in the Physical Layer specification), or the access network shall expire the FL-  
 16 ATA for the interlace.

#### 17 **7.5.6.4.2.3 F-DCH transmissions associated with non-sticky assignments**

18 The access network may formulate and transmit a MAC packet on an interlace according to the  
 19 NS-ATA on the interlace and the PF selected for transmission. The MAC shall pass the Physical  
 20 Layer Protocol the set of hop ports specified by the FL-NS-ATA, and shall specify whether or not the  
 21 assignment is an Extended Transmission Duration Assignment (see 7.1.3.1.3). The access network  
 22 shall not start an F-DCH transmission containing a page in the first frame of a superframe. (A page  
 23 can be sent using an FTC MAC packet transmission to a broadcast MACID wherein the MAC header  
 24 includes the UATI field. See 7.5.7.2).

25 The number of H-ARQ retransmissions over a FL-NS-ATA<sub>UC</sub> is equal to the duration of the  
 26 FL-NS-ATA<sub>UC</sub>, as specified by the duration field in the non-sticky assignment block (see 7.4.6.3.1.2).  
 27 Note that the maximum number of transmissions may be more than 6, which is the number specified  
 28 in 7.5.6.7.1.

29 If TuneAwayStatus is equal to '1', then the access network shall not send any MAC packets on any  
 30 FL-NS-ATA<sub>UCS</sub> on any interlaces to this access terminal. MAC packets being sent on FL-NS-ATA<sub>BCS</sub>  
 31 may still be sent.

32 The access network should send no more than one MAC packet over a FL-NS-ATA<sub>UC</sub> per interlace. If  
 33 two interlaces overlap in time, the access network should only send a MAC packet over a FL-NS-  
 34 ATA<sub>UC</sub> on one of the interlaces. The access network should send no more than one MAC packet over  
 35 a FL-NS-ATA<sub>BC</sub> per carrier, per interlace. For a given carrier, if two interlaces overlap in time, the  
 36 access network should only send a MAC packet over a FL-NS-ATA<sub>BC</sub> on one of the interlaces. The  
 37 access network should not send MAC packets to the same MACID over both an FL-ATA and a FL-  
 38 NS-ATA<sub>UC</sub> on the same interlace, or on time overlapping interlaces.

39 If a positive ACK corresponding to the access terminal's MACID is received that corresponds to the  
 40 transmitted packet on a FL-NS-ATA<sub>UC</sub>, transmission of the MAC packet shall terminate, and the  
 41 access network shall expire the FL-NS-ATA<sub>UC</sub>. (Note that for multi-code word assignments assigned  
 42 by NS-MCW-FLABs, the ACKs for all the layers need to be received before the transmission of the  
 43 MAC packet terminates, and the interlace is available for the next packet.) If no ACK is received for

1 this packet, the access network shall return a *ForwardTrafficPacketsMissed* indication along with  
 2 parameters that uniquely identify the lost packet. The method of uniquely identifying the packet is out  
 3 of the scope of this specification.

4 FL-NS-ATAs shall be expired after the specified duration (see 7.4.6.3.1.2), if not already expired.

### 5 **7.5.6.5 Supervision procedures**

6 The access terminal shall maintain a supervision timer of duration *InactivityTimeout*. This timer shall  
 7 be reset as described in 7.5.6.4.1.4 and 7.5.6.4.1.5. The access terminal shall issue a  
 8 *SupervisionFailed* indication when the supervision timer expires.

### 9 **7.5.6.6 Channel trees**

10 A channel tree defines the mapping of each *NodeID* to a set of hop-ports. A channel tree on the FL is  
 11 indexed by the *FLChannelTreeIndex* and the number of subcarriers mapped by the channel tree,  
 12  $N_{\text{CARRIER\_SIZE}}$ , a parameter that is defined by the Physical Layer protocol. See 7.1.4.1 for common  
 13 terms used for describing channel trees in this specification. Hop-ports shall be numbered numerically  
 14 from 0.

15  $Q_{\text{SDMA}}$  equals *FLNumSDMADimensions*.

16 The set of hop-ports specified by a *NodeID* shall be the union of usable hop-ports mapped by all  
 17 base-nodes that are descendants of the node specified by *NodeID*, minus the hop-ports allocated to F-  
 18 SSCH (as defined in 7.4.6.3.1.1). “Usable hop-ports” are defined by the Physical Layer Protocol.

19 The number of hop-ports indexed by the tree shall equal  $Q_{\text{SDMA}} * N_{\text{CARRIER\_SIZE}}$ , and the total number of  
 20 nodes in the tree shall be a function of  $N_{\text{CARRIER\_SIZE}}$  and  $Q_{\text{SDMA}}$ .

21 Note that when multi-carrier mode is equal to *MultiCarrierOn*, there is an independent channel tree  
 22 per carrier, and the channel tree in use for the carrier is signaled on the overhead channels of that  
 23 carrier. Further, when a specific *NodeID* or set of hop-ports is communicated with other protocols in  
 24 this specification, the associated carrier must also be communicated.

#### 25 **7.5.6.6.1 FL channel tree index 0**

26 Channel trees associated with *FLChannelTreeIndex* 0 are illustrated in Figure 81 for  $N_{\text{CARRIER\_SIZE}} =$   
 27 512, 1024, and 2048. For  $N_{\text{CARRIER\_SIZE}} = 512$ , all nodes above the dashed line marked with  
 28  $N_{\text{CARRIER\_SIZE}} = 512$  are included in the channel tree; For  $N_{\text{CARRIER\_SIZE}} = 1024$ , all nodes above the  
 29 dashed line marked with  $N_{\text{CARRIER\_SIZE}} = 1024$  are included in the channel tree; For  $N_{\text{CARRIER\_SIZE}} =$   
 30 2048, all nodes above the dashed line marked with  $N_{\text{CARRIER\_SIZE}} = 2048$  are included in the channel  
 31 tree.

32 *MinHopPortsPerNode* equals 16 for *FLChannelTreeIndex* 0.

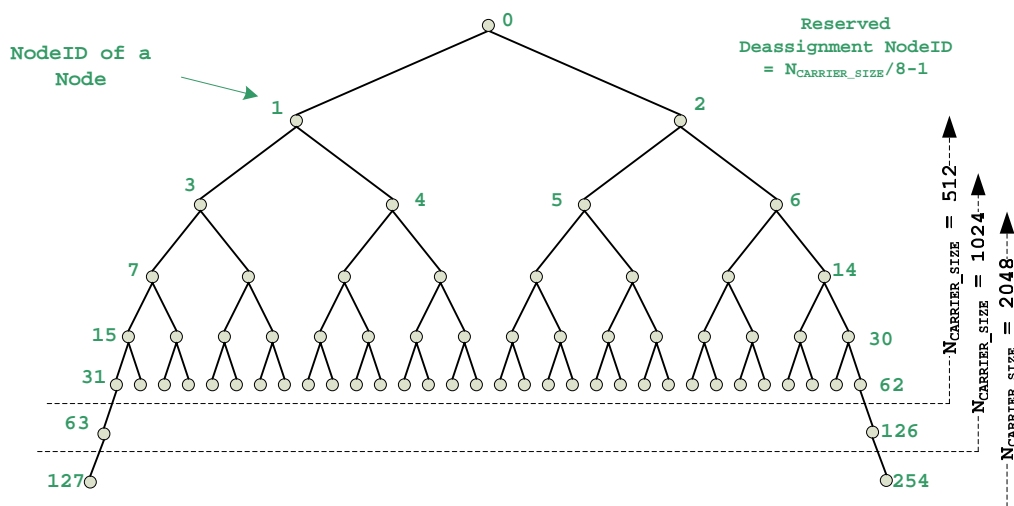
33 The figure shows only  $1/Q_{\text{SDMA}}$  of the total tree, and there are  $Q_{\text{SDMA}}$  identical versions of the  
 34 illustrated tree each with unique *NodeIDs* and mapping unique hop-ports. For example, if the total  
 35 tree is composed of  $Q_{\text{SDMA}}$  identical trees indexed by  $i=0, 1, \dots, Q_{\text{SDMA}}-1$ , then *NodeIDs* of nodes on  
 36 the  $i$ th tree can be obtained from the illustrated tree by adding  $i$   
 37  $*N_{\text{CARRIER\_SIZE}}/(\text{MinHopPortsPerNode}/2)$  to the *NodeID* of the illustrated tree. The number of hop-  
 38 ports indexed by the tree shall equal  $Q_{\text{SDMA}}N_{\text{CARRIER\_SIZE}}$ , and the total number of nodes in the tree

1 shall be a function of  $N_{\text{CARRIER\_SIZE}}$ . Namely, the base-nodes are defined by the intervals  $\text{NodeID} = i * N_{\text{CARRIER\_SIZE}} / (\text{MinHopPortsPerNode} / 2) + \{ N_{\text{CARRIER\_SIZE}} / \text{MinHopPortsPerNode} - 1, N_{\text{CARRIER\_SIZE}} / (\text{MinHopPortsPerNode} / 2) - 2 \}, i = 0, 1, \dots, Q_{\text{SDMA}} - 1$ . Thus, for  $N_{\text{CARRIER\_SIZE}} = 512$  and  $Q_{\text{SDMA}} = 4$ , there are 128 base-node NodeIDs, 31 through 62, 95 through 126, 159 through 190, and 223 through 254. For nodes on the same level of a channel tree, the NodeID associated with a node increases from left to right with step of 1. One deassignment NodeID,  $\text{NodeID}_{\text{DEASSIGN}}$ , is set to  $N_{\text{CARRIER\_SIZE}} / (\text{MinHopPortsPerNode} / 2) - 1$ .

8 The mapping of hop-ports to each base-node is described as follows. Each base-node maps to  $\text{MinHopPortsPerNode}$  hop-ports, the first  $\text{MinHopPortsPerNode}$  hop-ports (indices 0 to  $\text{MinHopPortsPerNode} - 1$ ) to the base-node with the lowest NodeID, the second  $\text{MinHopPortsPerNode}$  hop-ports to the next base-node, etc., until all hop-ports are mapped. See Table 70 for an example of this mapping for  $N_{\text{CARRIER\_SIZE}} = 512$ ,  $\text{MinHopPortsPerNode} = 16$ , and  $Q_{\text{SDMA}} = 4$ .

13 **Table 70 Base node NodeID to Hop-port Mapping Example**  
 14 **( $N_{\text{CARRIER\_SIZE}} = 512$ ,  $\text{MinHopPortsPerNode} = 16$ , and  $Q_{\text{SDMA}} = 4$ )**

Base node NodeID	Hop-ports mapped
31	0-15
32	16-31
...	...
62	496-511
95	512-527
96	528-543
...	...
126	1008-1023
...	...
254	2032-2047



16 **Figure 77 FL channel trees with index 0**

### 7.5.6.7 Packet formats

A packet format (PF) specifies the spectral efficiency, maximum number of transmissions, and the modulation format to be used for each transmission of a data packet. Each packet format is indexed by a packet format index. The modulation format is specified by the number of bits in each modulation symbol, which is denoted by modulation order. Modulation orders of 2, 3, 4, and 6 corresponding to QPSK, 8PSK, 16QAM and 64QAM modulations respectively. These modulations are described in detail in the Physical Layer specification. The size of the MAC packet that is provided to the Physical Layer is a function of the packet format as well as the set of hop-ports that are assigned to the data packet (to be transmitted on the F-DCH Physical Layer channel.) The computation of the packet size as a function of the set of hop-ports and the packet format is described in the Physical Layer.

#### 7.5.6.7.1 SISO mode packet formats (SSMAC)

The packet format consists of six bits. The first bit of the PF indicates the transmission mode. If the first bit is equal to '0' then the default mode is used for transmission. If the first bit is equal to '1' then the STTD mode is used for transmission. The second bit of the PF indicates the F-DPICH format as described in the F-DPICH section of the physical layer protocol. If the second bit is equal to '0', F-DPICH format 0 is used. If the second bit is equal to '1', F-DPICH format 1 is used. The remaining four bits index the spectral efficiency, maximum number of transmissions, and the modulation format to be used for each transmission of a data packet. The indexing is described in Table 71.

**Table 71 FL packet formats – SISO mode**

Packet Format Index	Spectral efficiency on 1 <sup>st</sup> transmission	Max number of transmissions	Modulation order for each transmission					
			1	2	3	4	5	6+
0	0.2	6	2	2	2	2	2	2
1	0.5	6	2	2	2	2	2	2
2	1.0	6	2	2	2	2	2	2
3	1.5	6	3	2	2	2	2	2
4	2.0	6	4	3	3	3	3	3
5	2.5	6	6	4	4	4	4	4
6	3.0	6	6	4	4	4	4	4
7	4.0	6	6	6	4	4	4	4
8	5.0	6	6	6	4	4	4	4
9	6.0	6	6	6	4	4	4	4
10	7.0	6	6	6	4	4	4	4
11	8.0	6	6	6	6	4	4	4
12	9.0	6	6	6	6	4	4	4
13	10.0	6	6	6	6	6	4	4
14	11.0	6	6	6	6	6	4	4
15	NULL							

1 PF index 15 is used to indicate NULL, or no PF, and is useful in MCW assignments that utilize fewer  
2 than the maximum number of layers for transmission (see 7.4.6.3.1).

### 3 **7.5.6.7.2 MIMO MCW mode packet formats**

4 The packet format consists of five bits. However, the first bit is sometimes omitted in the packet  
5 format field of SS MAC blocks when multiple blocks are used to communicate assignment  
6 information. In this case, the omitted first bit shall be assumed to be the same as the (non-omitted)  
7 first bit of the associated SS MAC block that contains all five packet format bits.

8 The first bit of the PF along with the parameter NumLayers (for the interlace being assigned)  
9 indicates the F-DPICH format as described in the F-DPICH section of the physical layer protocol. If  
10 NumLayers is equal to '1' or '2', then the F-DPICH format used is F-DPICH format 0 if the first bit  
11 is equal to '0' or F-DPICH format 1 if the first bit is equal to '1'. If NumLayers is equal to '3', the F-  
12 DPICH format used is F-DPICH format 0 (the first bit is ignored). If NumLayers is equal to '4', the  
13 F-DPICH format used is F-DPICH format 2 (the first bit is ignored). The remaining four bits index  
14 the spectral efficiency, maximum number of transmissions, and the modulation format to be used for  
15 each transmission of a data packet. The indexing is described in Table 71.

### 16 **7.5.6.7.3 MIMO SCW mode packet formats**

17 The packet format consists of five bits. The first bit of the PF along with the parameter NumLayers  
18 (for the interlace being assigned) indicates the F-DPICH format as described in the F-DPICH section  
19 of the physical layer protocol. If NumLayers is equal to '1' or '2', then the F-DPICH format used is  
20 F-DPICH format 0 if the first bit is equal to '0' or F-DPICH format 1 if the first bit is equal to '1'. If  
21 NumLayers is equal to '3', the F-DPICH format used is F-DPICH format 0 (the first bit is ignored). If  
22 NumLayers is equal to '4', the F-DPICH format used is F-DPICH format 2 (the first bit is ignored).  
23 The remaining four bits index the spectral efficiency, maximum number of transmissions, and the  
24 modulation format to be used for each transmission of a data packet. The indexing is described in  
25 Table 71.



## 7.5.7 Header and trailer formats

### 7.5.7.1 Header and trailer for unicast transmissions

The access network shall formulate a unicast packet for transmission over the Forward Traffic Channel using the following header and trailer:

#### 7.5.7.1.1 Header

Field	Length (bits)
UATIInfoIncluded	1
IsSecure	1
KeyChange	1
InBandControlIncluded	1
Reserved	4
ATIdentifier	0 or 128
UATIStatus	0 or 4
FailureCode	0 or 4
0 or more of the following fields	
InBandControl	8

**UATIInfoIncluded** The access network shall set this field to 1 if a UATI is included, and to zero otherwise. The access network shall set UATIInfoIncluded to '1' for the first packet sent on the F-DCH after the receipt of a packet with UATIInfoIncluded equal to '1' on the R-DCH.

**IsSecure** The access network shall set this field to '1' if the packet is secured by the Authentication and Encryption protocols. The access network shall set this field to '0' otherwise.

**KeyChange** This field shall be set by the Security Sublayer at the transmitter and communicated to the Security Sublayer along with the payload at the receiver.

**InBandControlIncluded** Used to signal the existence of in-band control bits. The access network shall set this field to '1' if the InBandControl field is present. Otherwise, the access network shall set this field to '0'. InBandControl messages are defined in 7.5.7.1.3.

**Reserved** The sender shall set this field to '0000'. The receiver shall ignore this field.

**ATIdentifier** If UATIInfoIncluded is 1, the access network shall set this field to the last ATIdentifier sent by the access terminal. Otherwise, the access network shall omit this field.

**UATIStatus** The access network shall include this field if UATIInfoIncluded is set to 1.

1

**Table 72 Encoding of the UATISStatus field**

Field value	Description
0x0	Reset timer
0x1	Expire timer
0x2	Ignore
All other values are reserved	

2 FailureCode

The access network shall set this field to a non-zero value if UATIInfoIncluded is set to 1 and UATISStatus is set to 1.

3

4

**Table 73 Encoding of the FailureCode field**

Field value	Description
0x00	No failure
0x01	Token not supported
0x02	Network Busy
0x03	Authentication or billing failure
0x04	Desired QoS unavailable
0x05	No route to host
0x06	Network maintenance
0x07	Connection closed due to terminal request
0x08	SessionLost
0x09	General failure
All other values are reserved	

5 InBandControl

Included only if InBandControl set to '1'. InBandControl messages are defined in 7.5.7.1.3.

6

7 **7.5.7.1.2 Trailer**

8 This protocol does not add a trailer.

### 7.5.7.1.3 FLInBandControl

#### 7.5.7.1.3.1 InBandPacketFormatSwitchSISO block

This block allows the access network to change the packet format being used for forward link transmissions. The block has the following format

Field	Length (bits)
ContinuationBit	1
BlockFormat	2
PacketFormat	6
Reserved	7

**ContinuationBit** The access network shall set the continuation bit to 0 if this is the last FLInBandControl block in the MAC header. Otherwise, the access terminal shall set the continuation bit to 1.

**BlockFormat** The access network shall indicate an InBandPacketFormat level block by setting the format indicator bits to 00.

**PacketFormat** The access network shall set this field to indicate the packet format that is to be used on subsequent transmissions on the same interlace. See 7.5.6.7.1 for the definition of this field.

**Reserved** The access network shall set this field to '0000000'. The access terminal shall ignore this field.

#### 7.5.7.1.3.2 InBandPacketFormatSwitchMIMOSCW block

This block allows the access network to change the packet format being used for forward link transmissions. The block has the following format

Field	Length (bits)
ContinuationBit	1
BlockFormat	2
PacketFormat	5
NumLayers	2
Reserved	6

**ContinuationBit** The access network shall set the continuation bit to 0 if this is the last FLInBandControl block in the MAC header. Otherwise, the access terminal shall set the continuation bit to 1.

**BlockFormat** The access network shall indicate an InBandPacketFormat level block by setting the format indicator bits to 01.

1	PacketFormat	The access network shall set this field to indicate the packet format that is to be used on subsequent transmissions on the same interlace. See 7.5.6.7.3 for the definition of this field.
2		
3		
4	NumLayers	The access network shall set this field to indicate the number of layers that is to be used on subsequent transmission on the same interlace.
5		
6	Reserved	The access network shall set this field to '000000'. The access terminal shall ignore this field.
7		

### 8 7.5.7.1.3.3 InBandPacketFormatSwitchMIMOMCW block

9 This block allows the access network to change the packet format being used for forward link  
10 transmissions. The block has the following format

11

Field	Length (bits)
ContinuationBit	1
BlockFormat	2
PacketFormat $i$	4
PacketFormat2	4
PacketFormat3	4
PacketFormat4	4
Reserved	6

12	ContinuationBit	The access network shall set the continuation bit to 0 if this is the last FLInBandControl block in the MAC header. Otherwise, the access terminal shall set the continuation bit to 1.
13		
14		
15	BlockFormat	The access network shall indicate an InBandPacketFormat level block by setting the format indicator bits to 01.
16		
17	PacketFormat $i$	The access network shall set this field to indicate the packet format that is to be used on layer $i$ of subsequent transmissions on the same interlace. See 7.5.6.7.2 for the definition of this field.
18		
19		
20	Reserved	The access network shall set this field to '000000'. The access terminal shall ignore this field.
21		

## 7.5.7.2 Header and trailer for broadcast transmissions

The access network shall formulate a broadcast packet for transmission using the following header and trailer:

### 7.5.7.2.1 Header

Field	Length (bits)
IsSecure	1
NumATIRecords	5
NumATIRecords number of the following 2 fields	
ATIRecord	2 or 34
OpenConnectionRequired	1
Reserved	0-7

IsSecure	The access network shall set this field to '1' if the packet is secured by the Authentication and Encryption protocols. The access network shall set this field to '0' otherwise.
NumATIRecords	This field shall be set to the number of ATIRecords included in the header.
ATI Record	Access Terminal Identifier Record. The access network shall set this field to the record specifying the access terminal's address. This record is defined in the Common Algorithms chapter (10) of this specification.
OpenConnectionRequired	If this field is set to '1', and the ATI record field matches as specified in 7.2.6.6.1.1, the terminal attempts to open a connection. This field may trigger a <i>PageReceived</i> indication at the access terminal.
Reserved	The length of this field shall be such that the entire message is octet-aligned. The sender shall set this field to zero. The receiver shall ignore this field.

### 7.5.7.2.2 Trailer

This protocol does not add a trailer.

## 7.5.8 Message formats

The protocol uses the AttributeUpdateRequest, AttributeUpdateAccept, and AttributeUpdateReject messages of the Generic Attribute Update Protocol in 10.9 to update configurable attributes.

## 7.5.9 Interface to other protocols

### 7.5.9.1 Commands sent

- ReverseControlChannelMAC.ChangeFLSS

## 7.5.9.2 Indications

This protocol registers to receive the following indications:

- *PhysicalLayer.ForwardDataCompleted*
- *ConnectedState.TunedAway*
- *RCCMAC.FLSSChanged*

## 7.5.10 Configuration attributes

The negotiable attributes for this protocol are listed in Table 74. The access terminal shall use as defaults the values in Table 74 that are listed in *bold italics*.

Unless specified otherwise, the access terminal and the access network shall use the Generic Attribute Update Protocol in 10.9 to update configurable attributes belonging to the Default Forward Traffic Channel MAC Protocol.

**Table 74 Configurable attributes**

Attribute ID	Attribute	Values	Meaning
0x00	InactivityTimeout	0x0000, 0x0100-0xffff	Reserved
		<i>0x0002</i>	InactivityTimeout=2 seconds
		0x0001-0x00ff	InactivityTimeout in seconds

## 7.5.11 Protocol numeric constants

Constant	Meaning	Value
N <sub>FTCMPType</sub>	Type field for this protocol	Table 9
N <sub>FTCMPDefault</sub>	Subtype field for this protocol	0x0000
MACID <sub>BROADCAST</sub>	Broadcast MACID	0x0

## 7.5.12 Session state information

The Session State Information record (see 10.10) consists of parameter records.

The parameter records for this protocol consist of the configuration attributes of this protocol.

## 7.6 Default Reverse Control Channel MAC Protocol

### 7.6.1 Overview

The Reverse Control Channel MAC defines the procedures for transmissions on the following Physical Layer channels: R-CQICH, R-BFCH, R-SFCH, R-PICH, R-REQCH, and R-ACKCH.

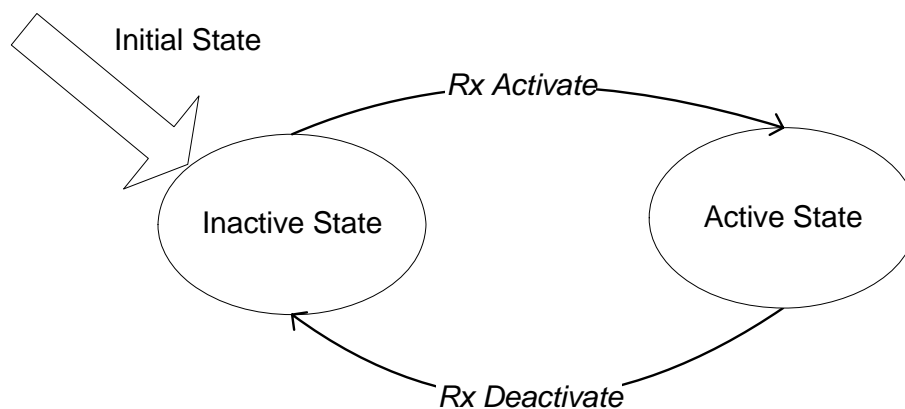
The R-CQICH is used by the access terminal to transmit the FL quantized channel quality from different sectors to the access network. The R-SFCH is a feedback channel, that is used by the access terminal to transmit the quantized FL channel quality measured for a subband in the FL serving sector (FLSS). The R-BFCH is a feedback channel, that is used by the access terminal to transmit the beam index defined in the Appendix, as well as supplemental channel quality information to enable SDMA transmission. SDMA transmission is defined in the Physical Layer Protocol. The R-PICH is a broadband pilot channel, and the R-REQCH is used by the access terminal to request resources. The R-ACKCH is used by the access terminal to acknowledge the FL MAC packets.

The access network maintains an instance of this protocol for every access terminal.

This protocol operates in one of two states:

- *Inactive State*: In this state, the access terminal is not assigned a MACID and shall not transmit on the Reverse Traffic Channel. When the protocol is in this state, it waits for an *Activate* command.
- *Active State*: In this state, the access terminal is assigned a MACID and may transmit data on the Reverse Traffic Channel.

The protocol states and the indications and events causing the transition between the states are shown in Figure 78.



**Figure 78 Default Reverse Control Channel MAC Protocol state diagram**

1 This protocol shall use the following parameters and attributes.  
2

Parameter Name	Where Defined	Comments
CQIReportingMode	ASMP	ActiveSetAssignmentMsg
CQIReportInterval	ASMP	ActiveSetAssignmentMsg
CQIReportPhase	ASMP	ActiveSetAssignmentMsg
CQIPilotReportInterval	ASMP	ActiveSetAssignmentMsg
CQIPilotReportPhase	ASMP	ActiveSetAssignmentMsg
BFCHReportRate	ASMP	ActiveSetAssignmentMsg
SFCHReportRate	ASMP	ActiveSetAssignmentMsg
BFCHPowerOffset	OMP	ExtendedChannelInfo
NumBFCHBits	ASMP	ActiveSetAssignmentMsg
SFCHPowerOffset	OMP	ExtendedChannelInfo
NumSFCHBits	ASMP	ActiveSetAssignmentMsg
BFCHBeamCodeBookIndex	OMP	ExtendedChannelInfo
MandatoryCQICHCTRLReportingPeriod	ASMP	ActiveSetAssignmentMsg
FLNumSDMADimensions	OMP	QuickChannelInfo block
RACKBandwidthFactor	OMP	ExtendedChannelInfo
ActiveSetIndex	ASMP	ActiveSetAssignmentMsg
MultiCarrierOn	Physical Layer Protocol	Public data
ActiveCarriers	SS MAC Protocol	Public data
REQCarrier	SS MAC Protocol	Public data
EffectiveNumAntennas	OMP	QuickChannelInfo block
MinRequestInterval	ASMP	ActiveSetAssignmentMsg
RPICHEnabled	ASMP	ActiveSet Assignment Msg
ProbePower	ACMP	public data
OpenLoopTransitionTime	Configuration Attribute	
RLCtrlPCMode	OMP	ExtendedChannelInfo
PowerControlStepUp	ASMP	ActiveSetAssignmentMsg
PowerControlStepDown	ASMP	ActiveSetAssignmentMsg
ACKCtrlOffset	pBCH1	Part of pBCH1 transmission
ACKChannelGain <sub>j</sub> , j = 0 to 9	Configuration Attribute	
ACKStepUpSize	Configuration Attribute	
ACKExtendedFrameGain	Configuration Attribute	
REQChannelGain <sub>j</sub> , j=0,1,2,3	OMP	ExtendedChannelInfo
CtrlAccessOffset	OMP	ExtendedChannelInfo
PICHPowerOffset	OMP	ExtendedChannelInfo
CQICHPowerBoostForHandoff	Configuration Attribute	
REQCHPowerBoostForHandoff	Configuration Attribute	
VCQIMeasureInterval	ASMP	ActiveSetAssignmentMsg
SingleServingSector	ASMP	ActiveSetAssignmentMsg

3



## 7.6.2 Primitives

### 7.6.2.1 Commands

This protocol defines the following commands:

- *ChangeFLSS*
- *ChangeRLSS*
- *Activate*
- *Deactivate*

### 7.6.2.2 Return indications

The protocol returns the following indications:

- *FLSSChanged*
- *RLSSChanged*
- *DFLSSChanged*
- *DRLSSChanged*

## 7.6.3 Public data

### 7.6.3.1 Static public data

This protocol does not define any static public data.

### 7.6.3.2 Dynamic public data

- Subtype for this protocol
- FLSS
- RLSS
- DFLSS
- DRLSS

## 7.6.4 Protocol data unit

This is a control protocol and does not carry any payload on the behalf of other protocols.

## 7.6.5 Protocol initialization and swap

### 7.6.5.1 Protocol initialization

Upon initialization at the access terminal,

- The values of the attributes for this protocol instance shall be set to the default values specified for each attribute.
- The protocol shall enter the Inactive State.

1 Upon initialization at the access network,

- 2       ■ The values of the attributes for this protocol instance shall be set to the default values
- 3       specified for each attribute.
- 4       ■ The protocol shall enter the Inactive State.

### 5 **7.6.5.2 Protocol swap**

6 Upon swap, the protocol instance shall enter the Inactive State.

## 7 **7.6.6 Procedures**

### 8 **7.6.6.1 Command processing**

#### 9 **7.6.6.1.1 Activate**

10 If the protocol receives an *Activate* command in the Inactive State, the access terminal and the access  
11 network shall perform the following:

- 12       ■ Transition to the Active State

13 If the protocol receives this command in any other state, the command shall be ignored.

#### 14 **7.6.6.1.2 Deactivate**

15 If the protocol receives a *Deactivate* command in the Active State:

- 16       ■ The access terminal shall cease transmitting the Reverse Control Channel and shall  
17       transition to the Inactive State.
- 18       ■ The access network shall cease monitoring the Reverse Control Channel from this access  
19       terminal and shall transition to the Inactive State.

20 If the protocol receives a *Deactivate* command in the Inactive State, the command shall be ignored.

#### 21 **7.6.6.2 Inactive State**

22 The access terminal shall not transmit on the physical channels governed by this protocol.

#### 23 **7.6.6.3 Active State**

24 The access terminal shall transmit on the physical channels governed by this protocol according to the  
25 procedures in this section. This section makes frequent use of the following terms defined for each  
26 active access terminal:

27	Active Set	The set of sectors for which the access terminal has an assigned MACID.
28	DFLSS	(Desired Forward Link Serving Sector)
29		The sector within the active set that the access terminal autonomously
30		determines is the best sector for forward link data transmissions.

1	DRLSS	(Desired Reverse Link Serving Sector)
2		The sector within the active set that the access terminal autonomously
3		determines is the best sector for reverse link data transmissions.
4	FLSS	(Forward Link Serving Sector) The last DFLSS from which the access
5		terminal successfully received a sticky FLAB with supplemental field '0', or
6		a non-sticky FLAB with a unicast MACID (see 7.5.6.4.1.2.1).
7	RLSS	(Reverse Link Serving Sector) The last DRLSS from which the access
8		terminal successfully received an RLAB with supplemental field '0' or a
9		non-sticky RLAB with a unicast MACID (see 7.7.6.4.1.1.1).
10	Desired Subband	The subband that the access terminal autonomously determines is the best
11		segment for forward link data transmission.
12	SSCH Subbands	Subbands where F-SSCH is transmitted.
13	Control Segment	Set of hop-ports in selected RL PHY Frames, used by the access terminal to
14		transmit R-CQICH, R-REQCH, R-PICH, R-SFCH, and R-BFCH.
15	ControlSegmentPeriod	Time duration between consecutive Control Segment transmissions to the
16		RLSS, in units of number of RL PHY Frames.
17	TDM2 Power Density	Power density of the second OFDM symbol in the F-ACQCH in the Physical
18		Layer Protocol.

19 In this section, the parameters CQIReportingMode, CQIReportInterval, CQIReportPhase,  
 20 CQIPilotReportInterval, CQIPilotReportPhase, BFCHReportRate, SFCHReportRate, NumSFCHBits,  
 21 NumBFCHBits, MandatoryCQICHCTRLReportingPeriod, RPICHEnabled, and  
 22 VCQIMeasureInterval correspond to the FLSS, and the parameter MinRequestInterval corresponds to  
 23 the RLSS or DRLSS, depending on the sector to which the R-REQCH is being transmitted.

#### 24 **7.6.6.3.1 Synchronous Subsets**

25 A synchronous subset contains a set of sectors that are determined to be synchronous according to the  
 26 most recent Active Set Assignment message received by the access terminal. Each ChannelBand, as  
 27 defined in Chapter 1, contains one or more synchronous subsets.

- 28 ■ The DRLSS, RLSS, and FLSS of the access terminal shall be members of the same  
 29 synchronous subset.
- 30 ■ A synchronous subset that contain the FLSS is referred to as a serving synchronous  
 31 subset. A synchronous subset that does not contain the FLSS is referred to as a non-  
 32 serving synchronous subset.
- 33 ■ The access terminal shall not transmit R-CQICH, R-BFCH, R-SFCH, R-PICH, and  
 34 R-REQCH on Control Segment of sectors that are not in the ChannelBand of the FLSS.
- 35 ■ The access terminal may transmit the R-CQICH on the Control Segment of sectors of  
 36 non-serving synchronous subsets. The access terminal shall not transmit R-BFCH, R-  
 37 SFCH, R-PICH, and R-REQCH on the Control Segment of sectors of non-serving  
 38 synchronous subsets.

### 1 7.6.6.3.2 Serving Sector Maintenance

2 If the multi-carrier mode is set to MultiCarrierOn,

- 3 ■ the RLSS shall be the same for all carriers indicated by ActiveCarriers, and
- 4 ■ the FLSS shall be the same for all carriers indicated by ActiveCarriers, and
- 5 ■ the DFLSS shall be the same for all carriers indicated by ActiveCarriers, and
- 6 ■ the DRLSS shall be the same for all carriers indicated by ActiveCarriers.

7 When this protocol receives a *ChangeFLSS* command, it shall perform the following:

- 8 ■ The protocol shall update the FLSS in the Public data to be equal to the FLSS in the
- 9 *ChangeFLSS* command,
- 10 ■ issue a *FLSSChanged* indication,
- 11 ■ the protocol shall update the DFLSS in the Public data of this protocol to be equal to the
- 12 FLSS in the *ChangeFLSS* command,
- 13 ■ issue a *DFLSSChanged* indication.
- 14 ■ In addition, if the FLSS in the *ChangeFLSS* command is a member of a non-serving
- 15 synchronous subset, or if the SingleServingSector bit is set to '1', the protocol shall
- 16 additionally perform the following
  - 17 □ the protocol shall update the RLSS in the Public data of this protocol, to be equal to
  - 18 the FLSS in the *ChangeFLSS* command,
  - 19 □ issue a *RLSSChanged* indication,
  - 20 □ the protocol shall update the DRLSS in the Public data of this protocol, to be equal to
  - 21 the FLSS in the *ChangeFLSS* command,
  - 22 □ issue a *DRLSSChanged* indication.

23 When this protocol receives a *ChangeRLSS* command, it shall perform the following:

- 24 ■ If the RLSS in the *ChangeRLSS* command is a member of a non-serving synchronous
- 25 subset, the protocol shall ignore the *ChangeRLSS* command.
- 26 ■ Otherwise,
  - 27 □ it shall update the RLSS in the Public data of this protocol,
  - 28 □ issue a *RLSSChanged* indication,
  - 29 □ it shall update the DRLSS in the Public data of this protocol,
  - 30 □ issue a *DRLSSChanged* indication.
  - 31 □ If the SingleServingSector bit is set to '1' for the RLSS in the *ChangeRLSS*
  - 32 command, the protocol shall additionally perform the following
    - 33 – the protocol shall update the FLSS in the Public data of this protocol, to be equal
    - 34 to the RLSS in the *ChangeRLSS* command,
    - 35 – issue a *FLSSChanged* indication,

- 1                   – the protocol shall update the DFLSS in the Public data of this protocol, to be
- 2                   equal to the RLSS in the *ChangeRLSS* command,
- 3                   – issue a *DFLSSChanged* indication.

#### 4   **7.6.6.3.3 Reverse Link CQI Reporting Modes**

5   The RCC MAC Protocol has multiple reporting modes. The access terminal's reporting mode shall be  
6   set based upon the CQIReportingMode. The access terminal's CQIReportingMode can be one of the  
7   following:

- 8                   ■ the Single Code Word CQI Reporting Mode(CQISCW),
- 9                   ■ the Multiple Code Word CQI Reporting Mode (CQIMCW),
- 10                  ■ the SISO CQI Reporting Mode (CQISISO).

#### 11   **7.6.6.3.4 Access Terminal Procedures for Determining RL Channel Quality**

12   For the purposes of determining the RL channel quality, the access terminal shall monitor the FL CQI  
13   Erasure Indication (CEI) bits from all sectors in each synchronous subset for which R-CQICH is  
14   transmitted on the corresponding Control Segment. If the multi-carrier mode is set to MultiCarrierOn,  
15   the access terminal shall independently monitor the FL CEI bit sent on carriers indicated by  
16   ActiveCarriers.

17   For each sector and carrier, the access terminal shall interpret the first FL CEI bit intended for the  
18   access terminal in the F-SSCH, following an R-CQICH message, according to the following  
19   interpretation rules:

- 20                  ■ If the FL CEI bit value is equal to 1, then the access terminal shall interpret the FL CEI  
21                   bit as indicating that the R-CQICH transmission was erased at the corresponding sector  
22                   and carrier; otherwise, the access terminal shall interpret the FL CEI bit as indicating that  
23                   the last R-CQICH transmission was not erased at the corresponding sector and carrier,
- 24                  ■ If an R-CQICH report has not occurred since the previous FL CEI bit from a given sector  
25                   and carrier, intended for the access terminal, then the current FL CEI bit from the  
26                   corresponding sector and carrier does not indicate an R-CQICH erasure.
- 27                  ■ If CQICHPowerBoostForHandoff is not zero, then the AT shall ignore the FL CEI bits  
28                   from all the sectors within the synchronous subset that contains the DFLSS following an  
29                   R-CQICH report that carries an FL handoff request (i.e., if the DFLSS is different from  
30                   the FLSS, the CQI is a CQICHCTRL report, the ActiveSetIndex field in the CQI is the  
31                   active set index of the DFLSS, and the DFLSS flag in the CQI is set to '1').

32   Following the above rules, the access terminal shall compute the estimated erasure rates for all sectors  
33   and carriers, for each synchronous subset for which R-CQICH is transmitted on the corresponding  
34   Control Segment.

### 7.6.6.3.5 Access Terminal Procedures for Power Control

If the multi-carrier mode is set to MultiCarrierOff, the access terminal shall initialize an independent parameter  $P_{CTRL}$  for each synchronous subset. If the multi-carrier mode is set to MultiCarrierOn, the access terminal shall initialize an independent parameter  $P_{CTRL}$  for each synchronous subset, for each carrier indicated by ActiveCarriers. The initial value of the  $P_{CTRL}$  is computed as

$$P_{CTRL} = ProbePower + CtrlAccessOffset$$

Independent ProbePower parameters are maintained for each synchronous subset. Each ProbePower parameter refers to the mean output power of the Access Channel preamble for the corresponding synchronous subset, at the end of the last successful Access Channel probe. If the multi-carrier mode is set to MultiCarrierOn, independent ProbePower parameters are maintained for each synchronous subset, for each carrier indicated by ActiveCarriers. In this case, each ProbePower parameter refers to the mean output power of the Access Channel preamble for the corresponding synchronous subset and carrier, at the end of the last successful Access Channel probe.

#### 7.6.6.3.5.1 Open Loop Power Adjustment

If OpenLoopTransitionTime is set to zero, the access terminal shall not perform the open loop power adjustment procedures described in this section. Otherwise, the access terminal shall measure the mean received power during each superframe preamble of the RLSS, and compare it to the mean received power measured during the previous superframe preamble of the RLSS. Following a step change of ReceivedPowerChange dB in the mean received superframe preamble power since the previous superframe preamble of the RLSS, each  $P_{CTRL}$  value shall transition to its final value of  $P_{CTRL} - ReceivedPowerChange$  according to the following rule:

$$P_{CTRL}(t) = P_{CTRL}(0) - ReceivedPowerChange \times \min\left\{\frac{t}{OpenLoopTransitionTime}, 1\right\},$$

where  $t$  denotes the time since the last superframe preamble of the RLSS and is measured in the same units as OpenLoopTransitionTime. The above expression defines the step response for this transition, i.e., if multiple changes are measured within a single transition time, their effects on each  $P_{CTRL}$  value shall be accumulated.

#### 7.6.6.3.5.2 Closed Loop Power Adjustment

Define PCSector(non-serving synchronous subset) as the sector with the lowest estimated erasure rate in the non-serving synchronous subset. PCSector(non-serving synchronous subset) is defined for each non-serving synchronous subset for which R-CQICH is transmitted and erasure rate is measured.

The access terminal shall treat the FL CEI bit intended for the access terminal that is sent over the F-SSCH of the PCSector(non-serving synchronous subset) for each non-serving synchronous subset as the power control command for that non-serving synchronous subset. If RLCtrlPCMode for the RLSS is set to 'UpDown', then the access terminal shall treat the FL PC bit intended for the access terminal that is sent over the F-SSCH of the RLSS as the power control command for the synchronous subset that contains the RLSS. If RLCtrlPCMode for the RLSS is set to 'ErasureBased', then the access terminal shall treat the FL CEI bit intended for the access terminal that is sent in the F-SSCH of the RLSS as the power control command for the synchronous subset that contains the RLSS.

1 The access terminal shall adjust the appropriate  $P_{CTRL}$  in response to FL power control command (PC  
2 or CEI bits, as explained below) intended for the access terminal that is sent over the F-SSCH of the  
3 RLSS and PCSector (non-serving synchronous subset) for each non-serving synchronous subset. If  
4 the multi-carrier mode is set to MultiCarrierOn, the access terminal shall independently adjust  
5 appropriate  $P_{CTRL}$  for each carrier indicated by ActiveCarriers, in response to the power control  
6 commands (PC or CEI bits, as explained below) for the corresponding carrier.

7 Each  $P_{CTRL}$  is adjusted independently using the PowerControlStepUp or PowerControlStepDown  
8 parameters corresponding to the RLSS (for the serving synchronous subset) or PCSector (non-serving  
9 synchronous subset) (for each non-serving synchronous subset), according to the following rules.

- 10 ■ For each non-serving synchronous subset, and for the synchronous subset that contains  
11 the RLSS, if RLCtrlPCMode for the RLSS is set to 'ErasureBased',
  - 12 □ When the CEI bit transmitted over the appropriate F-SSCH is '1', the access terminal  
13 shall increase the corresponding  $P_{CTRL}$  by PowerControlStepUp dB.
  - 14 □ When the CEI bit transmitted over the appropriate F-SSCH is '0', the access terminal  
15 shall decrease the corresponding  $P_{CTRL}$  by PowerControlStepDown dB.
- 16 ■ For the synchronous subset that contains the RLSS, if RLCtrlPCMode for the RLSS is set  
17 to 'UpDown'
  - 18 □ When the PC bit transmitted over the F-SSCH of the RLSS is '1', the access terminal  
19 shall increase the corresponding  $P_{CTRL}$  by PowerControlStepUp dB.
  - 20 □ When the PC bit transmitted over the F-SSCH of the RLSS is '0', the access terminal  
21 shall decrease the corresponding  $P_{CTRL}$  by PowerControlStepUp dB.
- 22 ■ If CQICHPowerBoostForHandoff is not zero, and the DFLSS does not belong to the  
23 same synchronous subset as the RLSS, then the AT shall ignore the FL CEI bits from that  
24 synchronous subset, following an R-CQICH report that carries an FL handoff request  
25 (i.e., if the DFLSS is different from the FLSS, the CQI report is a CQICHCTRL, the  
26 ActiveSetIndex field in the CQI is the active set index of the DFLSS, and the DFLSS flag  
27 in the CQI is set to '1').
- 28 ■ If CQICHPowerBoostForHandoff is not zero, the DFLSS belongs to the same  
29 synchronous subset as the RLSS, and RLCtrlPCMode for the RLSS is set to  
30 'ErasureBased', then the AT shall ignore the FL CEI bit from the RLSS, following an  
31 R-CQICH report that carries an FL handoff request (i.e., if the DFLSS is different from  
32 the FLSS, the CQI report is a CQICHCTRL, the ActiveSetIndex field in the CQI is the  
33 active set index of the DFLSS, and the DFLSS flag in the CQI is set to '1').

34 These changes in  $P_{CTRL}$  values shall be in addition to any changes dictated by the open-loop power  
35 control algorithm described in Section 7.6.6.3.5.1.

36 The mean output power of reverse control channels for the RLSS and PCSector(non-serving  
37 synchronous subset), and if the multi-carrier mode is set to MultiCarrierOn, for each carrier indicated  
38 by ActiveCarriers, shall be independently derived from the corresponding  $P_{CTRL}$ , as described below.

1 The R-CQICH transmit power shall be computed according to the following rules:

- 2 ■ If the DFLSS is different from the FLSS, the CQI report is a CQICHCTRL, the  
3 ActiveSetIndex field in the CQI is the active set index of the DFLSS, and the DFLSS flag  
4 in the CQI is set to '1', then

$$5 \quad P_{CQICH} = P_{CTRL} + CQICHPowerBoostForHandoff$$

- 6 ■ Otherwise,

$$7 \quad P_{CQICH} = P_{CTRL}$$

8 The R-PICH transmit power shall be computed for the RLSS as

$$9 \quad P_{PICH} = P_{CTRL} + PICHPowerOffset$$

10 The R-BFCH transmit power shall be computed for the RLSS as

$$11 \quad P_{BFCH} = P_{CTRL} + BFCHPowerOffset$$

12 The R-SFCH transmit power shall be computed only for the RLSS as

$$13 \quad P_{SFCH} = P_{CTRL} + SFCHPowerOffset$$

14 The R-REQCH transmit power shall be computed for the RLSS and DRLSS according to the  
15 following rules:

- 16 ■ If the DRLSS is different from the RLSS, and the ActiveSetIndex field in the REQ is the  
17 active set index of the DRLSS, then

$$18 \quad P_{REQCH} = P_{CTRL} + REQChannelGain_j + REQCHPowerBoostForHandoff$$

- 19 ■ Otherwise,

$$20 \quad P_{REQ} = P_{CTRL} + REQChannelGain_j$$

21 where j is the QoSFlow field of the request message being transmitted (Table 90).

22 The R-ACKCH transmit power shall be computed only for the FLSS according to the following rules:

$$23 \quad P_{ACK} = P_{CTRL} + ACKCtrlOffset + ACKChannelGain + \\ ACKChannelGainAdjustment + ACKExtendedFrameGain$$

24 where ACKCtrlOffset is a parameter transmitted over pBCH1 of the FLSS. The access terminal shall  
25 use the estimated erasure rate seen by the sector for which the Acknowledgment is intended, and shall  
26 set the ACKChannelGain to ACKChannelGain<sub>j</sub> when the estimated erasure rate is between j\*10%  
27 and (j+1)\*10%, where j can take the integer values 0,1,2,...,8,9. The value of the  
28 ACKChannelGainAdjustment (dB) shall always be greater than or equal to 0 dB. The access terminal  
29 shall only use a non-zero ACKChannelGainAdjustment value if the access terminal is able to detect



1 an ACK-to-NACK error event reliably with accuracy higher than 99 percent.  
 2 ACKChannelGainAdjustment shall be set to zero at the beginning of every MAC packet transmission  
 3 and shall increase by ACKStepUpSize dB every time an ACK-to-NACK error event is detected by  
 4 the access terminal. The access terminal shall use a non-zero ACKExtendedFrameGain value if the  
 5 access terminal is transmitting an acknowledgement for a FL MAC packet sent on an E-PHY Frame.

6 If the AT is transmitting packets on multiple interlaces, then it shall maintain independent values of  
 7 ACKChannelGainAdjustment and ACKExtendedFrameGain for each interlace. Then, the AT shall use  
 8 the interlace specific gain values when transmitting R-ACKCH associated with that interlace.

9 If the access terminal is unable to transmit the Physical Layer channels corresponding to the RCC  
 10 MAC protocol and the RTC MAC protocol at the specified power due to access terminal's transmit  
 11 power constraint, the access terminal shall transmit at the maximum transmit power. In this scenario,  
 12 the access terminal may independently decide the relative transmission priority of these channels. In  
 13 this scenario, the access terminal may independently decide to gate-off the transmission of one or  
 14 more physical layer channels corresponding to RCC MAC protocol and RTC MAC protocol.

15 The access terminal shall not modify the power control variable  $P_{CTRL}$  after  
 16 *ConnectedState.TunedAway* indication and until the *ConnectedState.TunedBack* indication. The  
 17 access terminal shall not transmit the Physical Layer channels corresponding to the RCC MAC  
 18 protocol and shall not monitor the FL PC, FL CEI, and Fast OSI value after  
 19 *ConnectedState.TunedAway* indication and until the *ConnectedState.TunedBack* indication.

#### 20 **7.6.6.3.6 Procedures for the R-CQICH Physical Layer channel**

21 The access terminal measures the FL channel quality on the SSCH subbands and transmits the  
 22 quantized channel quality to the access network using the R-CQICH. The computation and signaling  
 23 method are a function of the access terminal's CQIReportingMode, CQIReportInterval,  
 24 CQIReportPhase, CQIPilotReportInterval, CQIPilotReportPhase and  
 25 MandatoryCQICHCTRLReportingPeriod.

26 If the multi-carrier mode is set to MultiCarrierOn, the CQIReportingMode, CQIReportInterval,  
 27 CQIPilotReportInterval, and MandatoryCQICHCTRLReportingPeriod are identical across carriers  
 28 indicated by ActiveCarriers.

29 The RCC MAC Protocol shall provide the Physical Layer Protocol with the  $P_{CQICH}$ , RLCTRLCarrier,  
 30 PilotPN, and MACID for each R-CQICH transmission. The PilotPN and MACID may correspond to  
 31 different sectors depending on the CQI report and synchronous subset, as discussed in 7.6.6.3.6.1 The  
 32  $P_{CQICH}$  is determined according to the rules in 7.6.6.3.5. If the multi-carrier mode is set to  
 33 MultiCarrierOn, the RLCTRLCarrier indicates the carrier, where the given R-CQICH is transmitted.  
 34 If the multi-carrier mode is set to MultiCarrierOff, the RLCTRLCarrier is not defined.

#### 35 **7.6.6.3.6.1 Reporting Rules**

36 A detailed description of the different CQI reports referred to in this section, can be found in  
 37 7.6.6.3.6.2.

38 The access terminal shall transmit R-CQICH on the Control Segment of the RLSS. In addition, the  
 39 access terminal may transmit R-CQICH on the Control Segments of sectors in one or more non-  
 40 serving synchronous subsets. If the DFLSS is in a non-serving synchronous subset, the access

- 1 terminal shall transmit R-CQICH on the control segment corresponding to the DFLSS, or issue  
2 command `ACMAC.AttemptAccess(DFLSS)`.
- 3 If the multi-carrier mode is set to `MultiCarrierOn`, the access terminal shall transmit independent  
4 R-CQICH on the carriers indicated by `ActiveCarriers`. For the synchronous subset containing the  
5 FLSS, the `PilotPN`, `MACID`, and `RLCTRLCarrier` shall correspond to the current FLSS, unless  $(\text{RL PHY Frame Index} - \text{CQIPilotReportPhase}) \bmod \text{CQIPilotReportInterval} = 0$ . For non-serving  
6 synchronous subsets, the `PilotPN` and `MACID` shall correspond to the first sector of the synchronous  
7 subset (Indicated by the field `PilotPN`) listed in the last `ActiveSetAssignment` message received by the  
8 access terminal.
- 9
- 10 If  $(\text{RL PHY Frame Index} - \text{CQIPilotReportPhase}) \bmod \text{CQIPilotReportInterval} = 0$ , the `PilotPN` and  
11 `MACID`, shall correspond to the first sector listed for each synchronous subset in the last  
12 `ActiveSetAssignment` message received by the access terminal.
- 13 The access terminal shall not transmit the R-CQICH after `ConnectedState.TunedAway` indication and  
14 until the `ConnectedState.TuneBack` indication.
- 15 If  $(\text{RL PHY Frame Index} - \text{CQIPilotReportPhase}) \bmod \text{CQIPilotReportInterval} = 0$  then the  
16 `CQICHPIlot` shall be reported. If  $(\text{RL PHY Frame Index} - \text{CQIPilotReportPhase}) \bmod$   
17  $\text{CQIPilotReportInterval} \neq 0$  and  $(\text{RL PHY Frame Index} - \text{CQIReportPhase}) \bmod \text{CQIReportInterval}$   
18  $= 0$  then `CQICHCTRL`, `CQICHSCW`, or `CQICHMCW` shall be reported.
- 19 The `CQIReportIndex` shall be computed from the `CQIReportInterval` and `CQIReportPhase` as  
20  $\text{CQIReportIndex} = (\text{RL PHY Frame Index} - \text{CQIReportPhase}) / \text{CQIReportInterval}$ .
- 21 If  $\text{CQIReportIndex} \bmod \text{MandatoryCQICHCTRLReportingPeriod} = 0$  and  $(\text{RL PHY Frame Index} -$   
22  $\text{CQIPilotReportPhase}) \bmod \text{CQIPilotReportInterval} \neq 0$ , then `CQICHCTRL` shall be reported.
- 23 If  $(\text{RL PHY Frame Index} - \text{CQIPilotReportPhase}) \bmod \text{CQIPilotReportInterval} \neq 0$  and the `CQI` is  
24 reported for a sector other than the FLSS, then `CQICHCTRL` shall be reported.
- 25 If  $(\text{RL PHY Frame Index} - \text{CQIPilotReportPhase}) \bmod \text{CQIPilotReportInterval} \neq 0$  and `DFLSS`  $\neq$   
26 `FLSS` then `CQICHCTRL` shall be reported.
- 27 If  $2 < \text{EffectiveNumAntennas} \leq 4$ , the `CQICHMCW` report consists of 2 parts. The `CQICHMCW`  
28 report with the `MCWIndex` field set to 0, is the first part. However, if  $\text{EffectiveNumAntennas} \leq 2$ , the  
29 `CQICHMCW` report consists of only the first part. The `CQICHMCW` report is not defined if  
30  $\text{EffectiveNumAntennas} > 4$ .
- 31 Provided the above rules are not violated, then the AT may determine which `CQI` reports to use  
32 consistent with its `CQIReportingMode`.

### 7.6.6.3.6.2 CQI Report

The CQI reports for each CQIReportingMode and R-CQICH are shown in Table 75.

**Table 75 CQI Reports for each CQIReportingMode**

	CQIReportingMode		
	CQISISO	CQISCW	CQIMCW
CQI Report	CQICHPIlot	CQICHPIlot	CQICHPIlot
	CQICHCTRL	CQICHCTRL	CQICHCTRL
		CQICHSCW	CQICHMCW

The format for the CQICHPIlot report is shown in Table 76.

**Table 76 Format for CQICHPIlot report**

Field	Length (bits)
ReservedValue	10

ReservedValue      The ReservedValue is set to 0.

The format for the CQICHCTRL is shown in Table 77.

**Table 77 Format for CQICHCTRL report**

Field	Length (bits)
FormatType	1
CQIValueSISO	4
DFLSSFlag	1
ActiveSetIndex	3
Reserved	1

FormatType      This bit is set to the value 0.

CQIValueSISO      Indicates FL SISO CQI value. See 7.6.6.3.6.3 for details.

DFLSSFlag      If the ActiveSetIndex is the current DFLSS, the DFLSSFlag bit shall be set to 1; otherwise, the DFLSSFlag bit shall be set to 0. If the multi-carrier mode is set to MultiCarrierOn, the DFLSSFlag shall be the same for the CQICHCTRL report transmitted on all carriers indicated by ActiveCarriers.

ActiveSetIndex      Indicates the sector to which the CQIValueSISO corresponds. If the multi-carrier mode is set to MultiCarrierOn, the ActiveSetIndex shall be the same for the CQICHCTRL report transmitted on all carriers indicated by ActiveCarriers.

Reserved      The sender shall set this field to '0'. The receiver shall ignore this field.

1 The format for the CQISCW is in Table 78.

2 **Table 78 Format for CQICHSCW report**

Field	Length (bits)
FormatType	1
CQIValueSCW	5
Rank	2
Reserved	2

3 FormatType This bit is set to the value 1.

4 CQIValueSCW Indicates FL MIMO SCW CQI value for the reported Rank. See 7.6.6.3.6.3  
5 for details.

6 Rank Indicates the desired number of MIMO layers in the FL MIMO SCW  
7 transmission.

8 Reserved The sender shall set this field to '00'. The receiver shall ignore this field.

9 If  $2 < \text{EffectiveNumAntennas} \leq 4$ , the CQICHMCW report consists of 2 parts. The format for the first  
10 part is shown in Table 79, and the format for the 2<sup>nd</sup> part is shown in Table 80. However, if  
11  $\text{EffectiveNumAntennas} \leq 2$ , the CQICHMCW report consists of only the first part, shown in  
12 Table 79. The CQICHMCW report is not defined if  $\text{EffectiveNumAntennas} > 4$ .

13 **Table 79 Format of first part of CQICHMCW report**

Field	Length (bits)
FormatType	1
MCWIndex	1
CQIValueMCWLayer1	4
CQIValueMCWLayer2	4

14 FormatType This bit is set to the value 1.

15 MCWIndex This bit is set to the value 0 to indicate the CQI report is the first part of the  
16 CQICHMCW report.

17 CQIValueMCWLayer1  
18 Indicates the FL MIMO MCW layer 1 CQI value. See 7.6.6.3.6.3 for details.

19 CQIValueMCWLayer2  
20 Indicates the FL MIMO MCW layer 2 CQI value. See 7.6.6.3.6.3 for details.

**Table 80 Format of second part of CQICHMCW report**

Field	Length (bits)
FormatType	1
MCWIndex	1
CQIValueMCWLayer3	4
CQIValueMCWLayer4	4

FormatType This bit is set to the value 1.

MCWIndex This bit is set to the value 1 to indicate the CQI report is the second part of the CQICHMCW report.

CQIValueMCWLayer3 Indicates the FL MIMO MCW layer 3 CQI value. See 7.6.6.3.6.3 for details.

CQIValueMCWLayer4 Indicates the FL MIMO MCW layer 4 CQI value. See 7.6.6.3.6.3 for details.

### 7.6.6.3.6.3 CQIValue

Depending on the CQIReportingMode, the CQIValue can indicate the field CQIValueSISO in CQICHCTRL report, the field CQIValueSCW in CQICHSCW report, or the fields CQIValueMCWLayer1, CQIValueMCWLayer2, CQIValueMCWLayer3, and CQIValueMCWLayer4 in CQICHMCW report.

In the BlockHopping mode, the access terminal computes CQIValue assuming the FL PHY Frames are transmitted using beam index 0, in the code book specified by the BFCHBeamCodeBookIndex. The beam index 0 assumes no preferred transmit precoding matrix. See Appendix for details.

Based on an unrestricted observation interval, the access terminal shall report the highest tabulated CQIValue such that a packet transmitted by the access network to the access terminal over all of the hop-ports on the SSCH subbands (and on the corresponding RLCTRLCarrier if the multi-carrier mode is set to MultiCarrierOn) sent at TDM2 Power Density using the specified packet format and the specified number of FL-PHY Frames, terminating 1 PHY Frame before the start of the PHY Frame that the R-CQICH is transmitted, would result in a packet error probability  $\leq 0.01$ . The CQI mappings are shown in Table 81.

1

**Table 81 CQI Mapping to the FL Packet Format and Number of FL-PHY Frames**

4-bit CQI Value	5-bit CQI Value	FL Packet Format	Number of FL-PHY Frames
0	0	N/A	0
	1	0	2
1	2	1	4
	3	2	3
2	4	2	1
	5	2	2
3	6	2	3
4	7	3	5
5	8	4	4
	9	4	3
6	10	10	2
	11	5	1
7	12	6	2
	13	8	2
8	14	11	1
	15	12	3
9	16	13	5
	17	14	4
10	18	9	4
	19	13	3
11	20	10	2
12	21	11	2
13	22	12	1
14	23	13	2
15	24	14	2
	25	N/A	N/A
	26	N/A	N/A
	27	N/A	N/A
	28	N/A	N/A
	29	N/A	N/A
	30	N/A	N/A
	31	N/A	N/A

2

#### 1 **7.6.6.3.6.4 VCQI Support**

2 In this section, if the multi-carrier mode is set to MultiCarrierOn, the subband indices used for VCQI  
3 support, shall only include the indices corresponding to the carriers indicated by ActiveCarriers.

4 Define CQIValueSISOSubband[Interlace] to indicate the FL SISO CQI value for the given subband  
5 and FL interlace; CQIValueSCWSubband[Interlace][Rank] to indicate the FL SCW CQI value for the  
6 given subband, FL interlace and Rank; CQIValueMCWSubband[Interlace][Layer] to indicate the FL  
7 MCW CQI value for each MIMO layer.

8 Based on an unrestricted observation interval, the access terminal shall compute the  
9 CQIValueSISOSubband[Interlace], CQIValueSCWSubband[Interlace][Rank] or the  
10 CQIValueMCWSubband[Interlace][Layer] depending on the CQIReportingMode, such that a packet  
11 transmitted by the access network to the access terminal over all of the hop-ports on the subband, sent  
12 at TDM2 Power Density using the specified packet format and the specified number of FL-PHY  
13 Frames, terminating 1 PHY Frame before the start of the PHY Frame where  
14 CQIValueSISOSubband[Interlace], CQIValueSCWSubband[Interlace][Rank] or the  
15 CQIValueMCWSubband[Interlace][Layer] is computed, would result in a packet error probability  $\leq$   
16 0.01. The CQI mappings are shown in Table 81.

17 If the CQIReportingMode is SISO, the access terminal shall compute the report  
18 VCQIValueSISO[Interlace][subband] by averaging the CQIValueSISOSubband computed for that  
19 particular interlace and subband, over an averaging interval specified by VCQIMeasureInterval,  
20 terminating 1 PHY Frame before the VCQI message is reported.

21 If the CQIReportingMode is SCW, the access terminal shall compute the report  
22 VCQIValueSCW[Interlace][subband][Rank] for each Rank, by averaging the  
23 CQIValueSCWSubband[Interlace][Rank] computed for that particular subband and Rank, over an  
24 averaging interval specified by VCQIMeasureInterval, terminating 1 PHY Frame before the VCQI  
25 message is reported.

26 If the CQIReportingMode is MCW, the access terminal shall compute the report  
27 VCQIValueMCW[Interlace][subband][Layer] by averaging the CQIValueMCWSubband[Interlace]  
28 [Layer] computed for that particular subband and MIMO layer, over an averaging interval specified  
29 by VCQIMeasureInterval, terminating 1 PHY Frame before the VCQI message is reported.

#### 30 **7.6.6.3.7 Procedures for the R-SFCH Physical Layer channel**

31 The access terminal uses the R-SFCH to transmit the subband index and the quantized FL channel  
32 quality corresponding to the reported subband index and the FLSS.

33 The RCC MAC Protocol shall provide the Physical Layer Protocol with the  $P_{SFCH}$ , RLCTRLCarrier,  
34 PilotPNand MACID for each R-SFCH transmission. The PilotPNand MACID correspond to the  
35 FLSS. The  $P_{SFCH}$  is determined according to the rules in 7.6.6.3.5. If the multi-carrier mode is set to  
36 MultiCarrierOn, the RLCTRLCarrier indicates the carrier where the R-SFCH is transmitted. If the  
37 multi-carrier mode is set to MultiCarrierOff, the RLCTRLCarrier is not defined.

### 7.6.6.3.7.1 Reporting Rules

The access terminal shall transmit the R-SFCH only on the Control Segment of the RLSS.

The access terminal shall send the R-SFCH reports so that the SFCHReportRate requirement is satisfied. The access terminal shall not transmit R-SFCH report, if SFCHReportRate is equal to the value 0.

If the multi-carrier mode is set to MultiCarrierOn, R-SFCH shall be transmitted on the carriers indicated by ActiveCarriers.

The access terminal shall not transmit the R-SFCH after *ConnectedState.TunedAway* indication and until the *ConnectedState.TunedBack* indication.

### 7.6.6.3.7.2 SFCH Report

The SFCH reports for each CQIReportingMode are shown in Table 82.

**Table 82 SFCH Report for each CQIReportingMode**

	CQIReportingMode		
	CQISISO	CQISCW	CQIMCW
<b>SFCH Report</b>	SFCHSISO	SFCHSCW	SFCHMCW

The format for the SFCHSISO report is shown in Table 83.

**Table 83 Format for SFCHSISO report**

Field	Length (bits)
SubBandIndex	4
SubBandCQIValueSISO	4
Reserved	2

**SubBandIndex** Indicates the subband for which the SubBandCQIValueSISO is reported. If the multi-carrier mode is set to MultiCarrierOn, the two MSBs of this field is set to value 0.

**SubBandCQIValueSISO** If NumSFCHBits is equal to 11, this field indicates the CQI Value for the reported SubBandIndex. See 7.6.6.3.7.3 for details. If NumSFCHBits is set to “10”, this field is set to the value 0.

**Reserved** The sender shall set this field to ‘00’. The receiver shall ignore this field.



The format for the SFCHSCW report is shown in Table 84.

**Table 84 Format for SFCHSCW report**

Field	Length (bits)
SubBandIndex	4
SubBandCQIValueSCW	4
SubBandRank	2

**SubBandIndex** Indicates the Subband for which the SubBandCQIValueSCW is reported. If the multi-carrier mode is set to MultiCarrierOn, the two MSBs of this field is set to value 0.

**SubBandCQIValueSCW** If NumSFCHBits is equal to “11”, this field indicates the CQI Value for the FL MIMO SCW transmission for the reported Rank and the reported SubBandIndex. See 7.6.6.3.7.3 for details. If NumSFCHBits is equal to “10”, then this field is set to the value 0.

**SubBandRank** Indicates the desired number of MIMO layers in the FL MIMO SCW transmission for the reported SubBandIndex.

If EffectiveNumAntennas = 4, the SFCHMCW report consists of 3 parts. The format for the first part is shown in Table 85, the format for the second part is shown in Table 86, and the format for the third part is shown Table 87. If EffectiveNumAntennas = 2 or EffectiveNumAntennas = 3, the SFCHMCW report consists of only the first two parts, shown in Table 85 and Table 86. If EffectiveNumAntennas = 1, the SFCHMCW report consists of only the first part, shown in Table 85. The SFCHMCW report is not defined if EffectiveNumAntennas > 4.

**Table 85 Format of first part of SFCHMCW report**

Field	Length (bits)
MCWIndex	2
SubBandIndex	4
SubBandCQIValueMCWLayer1	4

**MCWIndex** This field is set to “00” to indicate the SFCH report is the first part of the SFCHMCW report.

**SubBandIndex** Indicates the Subband for which the SubBandCQIValueMCWLayer1 is reported. If the multi-carrier mode is set to MultiCarrierOn, the two MSBs of this field is set to value 0.

**SubBandCQIValueMCWLayer1** If NumSFCHBits is equal to “11”, this field indicates the CQI Value for the FL MIMO MCW layer 1 transmission and the reported SubBandIndex. See 7.6.6.3.7.3 for details. If NumSFCHBits is equal to “10”, then this field is set to the value 0.

**Table 86 Format of second part of SFCHMCW report**

Field	Length (bits)
MCWIndex	2
SubBandCQIValueMCWLayer2	4
SubBandCQIValueMCWLayer3	4

MCWIndex This field is set to “01” to indicate the SFCH report is the second part of the SFCHMCW report.

SubBandCQIValueMCWLayer2  
If NumSFCHBits is equal to “11”, this field indicates the CQI Value for the FL MIMO MCW layer 2 transmission and the reported SubBandIndex. See 7.6.6.3.7.3 for details. If NumSFCHBits is equal to “10”, then this field is set to the value 0.

SubBandCQIValueMCWLayer3  
If NumSFCHBits is equal to “11”, this field indicates the CQI Value for the FL MIMO MCW layer 3 transmission and the reported SubBandIndex. See 7.6.6.3.7.3 for details. If NumSFCHBits is equal to “10”, then this field is set to the value 0.

**Table 87 Format of third part of SFCHMCW report**

Field	Length (bits)
MCWIndex	2
SubBandCQIValueMCWLayer4	4
Reserved	4

MCWIndex This field is set to “10” to indicate the SFCH report is the third part of the SFCHMCW report.

SubBandCQIValueMCWLayer4  
If NumSFCHBits is equal to “11”, this field indicates the CQI Value for the FL MIMO MCW layer 4 transmission and the reported SubBandIndex. See 7.6.6.3.7.3 for details. If NumSFCHBits is equal to “10”, then this field is set to the value 0.

Reserved The sender shall set this field to ‘0000’. The receiver shall ignore this field.

### 7.6.6.3.7.3 SubBandCQIValue

Depending on the CQIReportingMode, the SubBandCQIValue can indicate the field SubBandCQIValueSISO in SFCHSISO report, the field SubBandCQIValueSCW in SFCHSCW report, or the fields SubBandCQIValueMCWLayer1, SubBandCQIValueMCWLayer2, SubBandCQIValueMCWLayer3 and SubBandCQIValueMCWLayer4 in SFCHSCW report.

1 The access terminal may autonomously choose to include single user transmit processing gains, such  
 2 as STTD or transmission on a preferred precoding matrix, in the SubBandCQIValue. STTD and  
 3 precoding are defined in the Physical Layer Protocol.

4 The access terminal shall not incorporate any form of multi-user transmit processing losses, such as  
 5 SDMA losses, in the SubBandCQIValue. SDMA is defined in the Physical Layer Protocol.

6 Based on an unrestricted observation interval, the access terminal shall report the highest tabulated  
 7 SubBandCQIValue such that a packet transmitted by the access network to the access terminal over  
 8 all of the hop-ports on the subband corresponding to the SubBandIndex, sent at TDM2 Power Density  
 9 using the specified packet format and the specified number of FL-PHY Frames, terminating 1 PHY  
 10 Frame before the start of the PHY Frame that the R-SFCH is transmitted, would result in a packet  
 11 error probability  $\leq 0.01$ . The CQI mappings are shown in Table 81.

### 12 **7.6.6.3.8 Procedures for the R-BFCH Physical Layer channel**

13 The access terminal uses the R-BFCH to transmit the beam index, the subband index, and the CQI  
 14 value offset necessary for SDMA transmission for the current FLSS. SDMA is defined in the Physical  
 15 Layer Protocol. A description of the beam code books, indexed by the BFCHBeamCodeBookIndex is  
 16 provided in 12.

17 The BFCH reports are sent using the BFCHBeamIndexType1 report or BFCHBeamIndexType2  
 18 report, the details of which are in section 7.6.6.3.8.1.

19 The RCC MAC Protocol shall provide the Physical Layer Protocol with the  $P_{BFCH}$ , RLCTRLCarrier,  
 20 PilotPN and MACID for each R-BFCH transmission. The BFCHReportType is equal to the value 1 if  
 21 BFCHBeamIndexType1 report is transmitted and equal to the value 2 if BFCHBeamIndexType2  
 22 report is transmitted. The PilotPN and MACID correspond to the FLSS. The  $P_{BFCH}$  is determined  
 23 according to the rules in 7.6.6.3.5. If the multi-carrier mode is set to MultiCarrierOn, the  
 24 RLCTRLCarrier indicates the carrier, where the R-BFCH is transmitted. If the multi-carrier mode is  
 25 set to MultiCarrierOff, the RLCTRLCarrier is not defined.

#### 26 **7.6.6.3.8.1 Reporting Rules**

27 The access terminal shall transmit the R-BFCH only on the Control Segment of the RLSS.

28 If the multi-carrier mode is set to MultiCarrierOn, R-BFCH shall be transmitted on the carriers,  
 29 indicated by ActiveCarriers.

30 The access terminal shall not transmit the R-BFCH after *ConnectedState.TunedAway* indication and  
 31 until the *ConnectedState.TunedBack* indication.

32 The access terminal shall send R-BFCH reports so that the BFCHReportRate requirement is satisfied.  
 33 The access terminal shall not transmit R-BFCH report if BFCHReportRate is equal to the value '0'.

34 The access terminal shall transmit at most one BFCHBeamIndexType1 report per Control Segment  
 35 and Carrier. In addition, the access terminal shall transmit the necessary number of  
 36 BFCHBeamIndexType2 reports per Control Segment and Carrier, to satisfy the BFCHReportRate  
 37 requirement.

### 7.6.6.3.8.2 BFCH Report

The following definition will be used in this section. The definition assumes that BFCHBeamIndexType1 report is transmitted on RL PHY Frame Index  $k$  and carrier  $j$ .

**MasterSFCHReportSet** The set of most recently transmitted SFCH reports on carrier  $j$  and RL PHY Frame Index  $k$  and RL PHY Frame Index  $k$  - ControlSegmentPeriod.

The format for BFCHBeamIndexType1 report is shown in Table 88, and the format for BFCHBeamIndexType2 report is shown in Table 89.

**Table 88 Format for BFCHBeamIndexType1**

Field	Length (bits)
BeamIndex	6
SDMADeltaCQI	3
Reserved	1

**BeamIndex** If the number of SFCH reports in the MasterSFCHReportSet is equal to '1', the BeamIndex indicates the desired beam index from the code book specified by the BFCHBeamCodeBookIndex, for the SubBandIndex specified in the SFCH report.

If the number of SFCH reports in the MasterSFCHReportSet is not equal to '1', the access terminal shall set the BeamIndex field to the value '0'. The access point shall ignore this field.

If the SFCHReportRate is 0, the BeamIndex field is set to the value '0'.

**SDMADeltaCQI** This field is defined only if NumBFCHBits is equal to "11", FLNumSDMADimensions > 1 and the BeamIndex field corresponds to an SDMA beam. Otherwise, the access terminal shall set this field to the value '0'. The access point shall ignore this field.

If the SFCHReportRate > 0, the SDMADeltaCQI indicates the integer offset that is to be subtracted from the SubBandCQIValue field of the SFCH reports, to obtain the SDMACQIValue. See 7.6.6.3.8.3 for details on SDMACQIValue.

If the SFCHReportRate = 0, the SDMADeltaCQI indicates the integer offset that is to be subtracted from the appropriate CQIValue field of the CQI reports, to obtain the SDMACQIValue. See 7.6.6.3.8.3 for details on SDMACQIValue.

**Reserved** The sender shall set this field to '0'. The receiver shall ignore this field.

**Table 89 Format for BFCHBeamIndexType2**

Field	Length (bits)
BeamIndex	6
SubBandIndex	4

BeamIndex The BeamIndex indicates the desired beam index from the code book specified by the BFCHBeamCodeBookIndex, for the reported SubBandIndex.

SubBandIndex Indicates the subband for which the BeamIndex is reported. If the multi-carrier mode is set to MultiCarrierOn, the two MSBs of this field is set to value 0.

### 7.6.6.3.8.3 SDMACQIValue

The SDMACQIValue is to be used by the Access Network for SDMA transmission.

In this section, the parameters used to compute SDMACQIValue are to be interpreted as integers. If SDMACQIValue is computed to be a number smaller than 0, the SDMACQIValue shall be made equal to 0. SDMACQIValue shall be interpreted as a 4-bit CQI value if the CQIReportingMode is CQISISO or CQIMCW, and as a 5-bit CQI value if the CQIReportingMode is CQISCW. The CQI mappings to be used for SDMACQIValue are shown in Table 81.

If the SFCHReportRate is 0, the SDMACQIValue is computed as follows.

- If the CQIReportingMode is CQISISO,  $SDMACQIValue = CQIValueSISO - SDMA\Delta CQI$ , where CQIValueSISO is a field in the CQICHCTRL report of the R-CQICH, described in 7.6.6.3.6.
- If the CQIReportingMode is CQISCW,  $SDMACQIValue = CQIValueSCW - SDMA\Delta CQI$ , where CQIValueSCW is a field in the CQICHSCW report of the R-CQICH, described in 7.6.6.3.6.
- If the CQIReportingMode is CQIMCW, the SDMACQIValue for each MIMO layer 'k' defined as  $SDMACQIValue = CQIValueMCWLayer'k' - SDMA\Delta CQI$ , where CQIValueMCWLayer'k' is a field in the CQICHMCW report of the R-CQICH, described in 7.6.6.3.6.

If the SFCHReportRate is not 0, the SDMACQIValue is computed as follows.

- If the CQIReportingMode is CQISISO,  $SDMACQIValue = SubBandCQIValueSISO - SDMA\Delta CQI$ , where SubBandCQIValueSISO is a field in the SFCHSISO report of the R-SFCH, described in 7.6.6.3.7.
- If the CQIReportingMode is CQISCW,  $SDMACQIValue = SubBandCQIValueSISO - SDMA\Delta CQI$ , where SubBandCQIValueSCW is a field in the SFCHSCW report of the R-SFCH, described in 7.6.6.3.7.
- If the CQIReportingMode is CQIMCW, the SDMACQIValue for each MIMO layer 'k' defined as  $SDMACQIValue = SubBandCQIValueMCWLayer'k' - SDMA\Delta CQI$ , where SubBandCQIValueMCWLayer'k' is a field in the SFCHMCW report of the R-SFCH, described in 7.6.6.3.6.

1 If the SFCHReportRate > 0, the access terminal shall report the lowest tabulated SDMADeltaCQI  
 2 based on an unrestricted observation interval, such that the packet transmitted by the access network  
 3 to the access terminal over all of the hop-ports referred by the SubBandIndex fields of the SFCH  
 4 reports, sent at TDM2 Power Density using the specified packet format and the specified number of  
 5 FL-PHY Frames corresponding to SDMACQIValue, terminating 1 PHY Frame before the start of the  
 6 PHY Frame that the R-BFCH is transmitted, would result in a packet error probability  $\leq 0.01$ .

7 If the SFCHReportRate = 0, the access terminal shall report the lowest tabulated SDMADeltaCQI  
 8 based on an unrestricted observation interval, such that the packet transmitted by the access network  
 9 to the access terminal over all of the hop-ports on the SSCH subbands used in the computation of R-  
 10 CQICH, sent at TDM2 Power Density using the specified packet format and the specified number of  
 11 FL-PHY Frames corresponding to SDMACQIValue, terminating 1 PHY Frame before the start of the  
 12 PHY Frame that the R-BFCH is transmitted, would result in a packet error probability  $\leq 0.01$ .

### 13 **7.6.6.3.9 Procedures for the R-PICH Physical Layer channel**

14 The access terminal uses the R-PICH to transmit the broadband pilot.

15 The access terminal shall transmit the R-PICH only on the Control Segment of the RLSS.

16 The access terminal shall transmit the R-PICH only if RPICHEnabled is equal to the value 1.

17 The RCC MAC Protocol shall provide the Physical Layer Protocol with  $P_{PICH}$ , RLCTRLCarrier,  
 18 PilotPNand MACID for each R-PICH transmission. The PilotPNand MACID correspond to the  
 19 FLSS. The  $P_{PICH}$  is determined according to the rules in 7.6.6.3.5. If the multi-carrier mode is set to  
 20 MultiCarrierOn, the RLCTRLCarrier indicates the carrier, where the R-PICH is transmitted. If the  
 21 multi-carrier mode is set to MultiCarrierOff, the RLCTRLCarrier is not defined. If the multi-carrier  
 22 mode is set to MultiCarrierOn, independent R-PICH shall be transmitted on independent carriers  
 23 indicated by ActiveCarriers.

24 The access terminal shall not transmit the R-PICH after *ConnectedState.TunedAway* indication and  
 25 until the *ConnectedState.TunedBack* indication.

### 26 **7.6.6.3.10 Procedures for the R-REQCH Physical Layer channel**

27 The access terminal shall transmit the R-REQCH only on the Control Segment of the RLSS.

28 The RCC MAC Protocol shall provide the Physical Layer Protocol with the  $P_{REQCH}$ , RLCTRLCarrier,  
 29 PilotPNand MACID. The PilotPNand MACID correspond to the FLSS. The  $P_{REQCH}$  is determined  
 30 according to the rules in 7.6.6.3.5. If the multi-carrier mode is set to MultiCarrierOn, the  
 31 RLCTRLCarrier is equal to the REQCarrier, that indicates the carrier on which the R-REQCH is  
 32 transmitted. If the multi-carrier mode is set to MultiCarrierOff, the RLCTRLCarrier is not defined.

33 The access terminal shall not transmit the R-REQCH after *ConnectedState.TunedAway* indication and  
 34 until the *ConnectedState.TunedBack* indication.

35 The access terminal shall wait a minimum duration of  $\text{MinRequestInterval} * \text{ControlSegmentPeriod}$   
 36 between consecutive R-REQCH transmissions to DRLSS. For instance, if the MinRequestInterval is  
 37 two, and the access terminal last transmits R-REQCH to the DRLSS on RL PHY Frame with RL  
 38 PHY Frame Index k, then the access terminal shall only be permitted to transmit R-REQCH to that

1 same sector again on an RL PHY Frame with RL PHY Frame Indices  $\geq$  ControlSegmentPeriod \* 2 +  
2 k.

3 The R-REQCH message format is shown in Table 90.

4 **Table 90 R-REQCH message format**

Field	Length (bits)
QoSFlow	2
MaxNumSubCarriers	2
DRLSS	3
Reserved	3

5 QoSFlow These bits specify the RLP QoS flow corresponding to the request. The  
6 access terminal shall indicate the QoS of the highest QoS flow that contains  
7 data available for transmission. The QoS priority order shall be 00 highest,  
8 01 second, 10 third, and 11 lowest.

9 MaxNumSubCarriers These bits specify the maximum number of subcarriers the access terminal  
10 can currently support. The access terminal shall specify the highest value  
11 from Table 91 such that both the buffer level of the QoS flow given by the  
12 QoSFlow bits and the number of subcarriers that the access terminal can  
13 support at RDCHGain using the available transmit power are satisfied.

14 **Table 91 MaxNumSubCarriers lookup table**

Num Subcarriers supportable at RDCHGain (Subcarriers)	Buffer size of QoSFlow (Bytes)	MaxNumSubCarriers Field
$X < 16$	$Y < 50$	00
$16 \leq X < 32$	$50 \leq Y < 100$	01
$32 \leq X < 64$	$100 \leq Y < 200$	10
Otherwise	Otherwise	11

15 DRLSS This field shall be set to the 3-bit ActiveSetIndex corresponding to the access  
16 terminal's DRLSS.

17 Reserved The sender shall set this field to '000'. The receiver shall ignore this field.

### 18 7.6.6.3.11 Procedures for the R-ACKCH Physical Layer channel

#### 19 7.6.6.3.11.1 Definitions

20 The following definitions will be used in this section, and apply to a specified RL PHY Frame.

21 RACKCTRLCarrier Indicates the carrier where the acknowledgement is transmitted.

22 NumACK Number of FL PHY Frames acknowledged.

1	NumACKIndex	Index for each of the NumACK acknowledgements. The acknowledgements are assumed to be ordered so that so that the acknowledgment corresponding to the FL PHY Frame index $k$ has a smaller NumACKIndex than the ACK corresponding to FL PHY Frame Index $k+m$ , where $m > 1$ . NumACKIndex satisfies $0 \leq \text{NumACKIndex} \leq \text{NumACK}-1$ .
2		
3		
4		
5		
6	NumRACKBaseNodes	For the RACKCTRLCarrier, $\text{NumRACKBaseNodes} = \lceil N_{\text{CARRIER\_SIZE}} / \text{MinHopPortsPerNode} \rceil * \lceil \text{FLNumSDMADimensions} / \text{RACKBandwidthFactor} \rceil * \text{NumACK}$ .
7		
8		
9	BaseNodeIndex	Index for a base-node in the carrier. The base-nodes are assumed to be ordered in increasing order so that base-node mapped by the lowest index hop-port in the carrier has BaseNodeIndex = 0, and the base-node mapped by the highest index hop-port in the carrier has BaseNodeIndex = $(\lceil N_{\text{CARRIER\_SIZE}} / \text{MinHopPortsPerNode} \rceil * \text{FLNumSDMADimensions}) - 1$ .
10		
11		
12		
13		
14	SpatialOrder	Number of MIMO-MCW layers.

### 15 7.6.6.3.11.2 ACK Reporting Rules

16 The access terminal shall not transmit the R-ACKCH after *ConnectedState.TunedAway* indication and  
17 until the *ConnectedState.TunedBack* indication.

18 In the FDD mode, for the FL transmission of a MAC packet on FL PHY Frame Index  $k$ , the access  
19 terminal shall send an ACK on the RL PHY Frame with RL PHY Frame Index  $k+3$ , resulting in  
20 NumACK = 1.

21 In the FDD mode, for the FL transmission of a MAC packet using an E-PHY Frame that starts on FL  
22 PHY Frame Index  $k$ , the access terminal shall send an ACK on the RL PHY Frames with RL PHY  
23 Frame Indices  $k+8$  and  $k+9$ , resulting in NumACK = 1.

24 Define:

$$25 \quad A = \lceil N_{\text{FL\_BURST}} / N_{\text{RL\_BURST}} \rceil \text{ and } B = \lfloor N_{\text{FL\_BURST}} / N_{\text{RL\_BURST}} \rfloor \text{ in TDD mode.}$$

26 The parameters  $N_{\text{FL\_BURST}}$  and  $N_{\text{RL\_BURST}}$  are respectively given by the  $N_{\text{FLBurst}}$  and  $N_{\text{RLBurst}}$   
27 parameters which are part of the public data of the Overhead Messages Protocol. The access terminal  
28 shall send ACKs according to the following rules:

- 29 ■ If  $N_{\text{FL\_BURST}} \leq N_{\text{RL\_BURST}}$ , for the FL transmission of MAC packet sent on FL PHY Frame  
30 Index  $k$ , the access terminal shall send an ACK on RL PHY Frame with RL PHY Frame  
31 Index  $(q+1)N_{\text{RL\_BURST}} + f$ , where  $q = \lfloor k / N_{\text{FL\_BURST}} \rfloor$  and  $f = k \bmod N_{\text{FL\_BURST}}$ . This results  
32 in NumACK = 1 for the first  $N_{\text{FL\_BURST}}$  RL PHY Frames, and NumACK = 0 for the  
33 remaining RL PHY Frames, in the appropriate RL burst.
- 34 ■ If  $N_{\text{FL\_BURST}} > N_{\text{RL\_BURST}}$ , for the FL transmission of MAC packet sent on FL PHY Frame  
35 Index  $k$ , the access terminal shall send an ACK on RL PHY Frame with RL PHY Frame  
36 Index  $(q+1)N_{\text{RL\_BURST}} + r$ , where  $q = \lfloor k / N_{\text{FL\_BURST}} \rfloor$  and  $r = \lfloor (k \bmod N_{\text{FL\_BURST}}) / A \rfloor$ . This  
37 results in NumACK = A, in the first  $\lfloor N_{\text{FL\_BURST}} / A \rfloor$  RL PHY Frames, and NumACK = B  
38 in the subsequent RL PHY Frames, in the appropriate RL burst.



1 The RACKVal, RACKBaseNodeIndex, and RACKCTRLCarrier computation are described in  
2 7.6.6.3.11.2.1. The  $P_{ACKCH}$  is determined according to the rules in 7.6.6.3.4.

3 If the multi-carrier mode is set to MultiCarrierOff, the MAC shall provide the PHY with  
4 NumRACKBaseNodes,  $P_{ACKCH}$  and RACKBaseNodeIndex and RACKVal, for each  
5 acknowledgement sent on the RL PHY Frame.

6 If the multi-carrier mode is set to MultiCarrierOn, the parameters NumRACKBaseNodes,  
7 RACKBaseNodeIndex,  $P_{ACKCH}$  and RACKVal are defined for the corresponding RACKCTRLCarrier.

8 If the multi-carrier mode is set to MultiCarrierOn, and the acknowledgement is for FL MIMO-MCW  
9 transmission restricted to one carrier or for FL MIMO-SCW or SISO transmission, the MAC shall  
10 provide the PHY with NumRACKBaseNodes,  $P_{ACKCH}$ , RACKBaseNodeIndex and RACKVal, for  
11 one RACKCTRLCarrier, sent on the RL PHY Frame.

12 If the multi-carrier mode is set to MultiCarrierOn, and the acknowledgement is for FL MIMO-MCW  
13 transmission across multiple carriers, the MAC may provide the PHY with NumRACKBaseNodes,  
14  $P_{ACKCH}$  and RACKBaseNodeIndex and RACKVal, for multiple RACKCTRLCarriers.

15 The access point shall ensure that the base-node mapped by the lowest-indexed hop-port in the ATA  
16 for each RACKCTRLCarrier, has the corresponding BaseNodeIndex that is a multiple of  
17  $RACKBandwidthFactor \times MinHopPortsPerNode$ .

#### 18 **7.6.6.3.11.2.1 RACKVal, RACKBaseNodeIndex and RACKCTRLCarrier Computation**

19 For each acknowledgement transmitted in the RL PHY Frame, the RACKVal, RACKBaseNodeIndex  
20 and RACKCTRLCarrier are determined as follows.

- 21 1. For the SISO and MIMO-SCW acknowledgement, the RACKVal,  
22 RACKBaseNodeIndex and RACKCTRLCarrier are determined as follows.
  - 23 a. If the FL transmission of MAC packet passes CRC, the RACKVal is set to the value  
24 1; otherwise RACKVal is equal to the value 0.
  - 25 b. The RACKCTRLCarrier shall be the carrier with the most hop-ports in the ATA. If  
26 more than one carrier has the most hop-ports in the ATA, the RACKCTRLCarrier  
27 shall be the carrier corresponding to the lowest carrier index.
  - 28 c. The  $RACKBaseNodeIndex = [(N_{CARRIER\_SIZE} / MinHopPortsPerNode) * (FLNumSDMADimensions) / RACKBandwidthFactor * NumACKIndex] + [BaseNodeIndex / RACKBandwidthFactor]$ , where the BaseNodeIndex corresponds  
29 to the base-node mapped by the lowest-indexed hop-port in the ATA in the  
30 RACKCTRLCarrier.  
31  
32
- 33 2. For MIMO-MCW acknowledgement, RACKBaseNodeIndex and RACKCTRLCarrier  
34 for MIMO layer 0 is chosen according to rules 1b and 1c. If the FL MIMO layer 0  
35 transmission of a MAC packet passes CRC, then the corresponding RACKVal shall be  
36 equal to the value 1, otherwise the corresponding RACKVal shall be equal to the value  
37 0.

- 1           3. In the MIMO-MCW mode, for the MIMO layer  $k$  ( $k > 0$ ), the RACKVal ,  
 2           RACKBaseNodeIndex and RACKCTRLCarrier are computed as follows. Initialize  $k =$   
 3           1. Initialize RACKBaseNodeIndex and RACKCTRLCarrier according to rules 1b and  
 4           1c.
- 5           a. If RACKBaseNodeIndex  $<$  NumRACKBaseNodes -1, increment  
 6           RACKBaseNodeIndex by 1 and go to step 3c; otherwise go to step 3b.
- 7           b. If RACKBaseNodeIndex  $\geq$  NumRACKBaseNodes -1, determine  
 8           RACKCTRLCarrier by incrementing the carrier index by 1, determine  
 9           RACKBaseNodeIndex for MIMO layer  $k$  according to rule 1c for the updated  
 10           RACKCTRLCarrier, and go to step 3c.
- 11           c. If the FL MIMO layer 'k' transmission of a MAC packet passes CRC, then the  
 12           corresponding RACKVal shall be equal to the value 1, otherwise the corresponding  
 13           RACKVal shall be equal to the value 0. Go to step 3d.
- 14           d. Increment  $k$  by 1. Go to step 3a if  $k <$  SpatialOrder; otherwise declare  
 15           RACKCTRLCarrier and RACKBaseNodeIndex computation to be complete.

16           The MIMO-MCW acknowledgement shall have SpatialOrder total number of RACKBaseNodeIndex  
 17           and RACKVal, counted across one or more RACKCTRLCarriers.

## 18           **7.6.7 Message formats**

19           The protocol uses the AttributeUpdateRequest, AttributeUpdateAccept, and AttributeUpdateReject  
 20           messages of the Generic Attribute Update Protocol in 10.9 to update configurable attributes.

## 21           **7.6.8 Interface to other protocols**

### 22           **7.6.8.1.1 Commands**

23           This protocol issues the following command:

- 24           ■ ACMAC.AttemptAccess(DFLSS)

### 25           **7.6.8.1.2 Indications**

26           This protocol registers to receive the following indications:

- 27           ■ *ConnectedState.TunedAway*
- 28           ■ *ConnectedState.TunedBack*

## 7.6.9 Configuration attributes

The following complex attributes and default values are defined (see 10.3 for attribute record definition).

Unless specified otherwise, the access terminal and the access network shall use the Generic Attribute Update Protocol in 10.9 to update configurable attributes belonging to the Default Reverse Traffic Channel MAC Protocol.

### 7.6.9.1 PowerControl attribute

Field	Length (bits)	Default Value
Length	8	N/A
AttributeID	8	N/A
OpenLoopTransitionTime	8	0
ACKChannelGain0	8	152
ACKChannelGain1	8	160
ACKChannelGain2	8	168
ACKChannelGain3	8	176
ACKChannelGain4	8	182
ACKChannelGain5	8	188
ACKChannelGain6	8	196
ACKChannelGain7	8	204
ACKChannelGain8	8	212
ACKChannelGain9	8	224
ACKStepUpSize	8	136
ACKExtendedFrameGain	8	128
CQICHPowerBoostForHandoff	8	0
REQCHPowerBoostForHandoff	8	0

**Length** Length of the complex attribute in octets. The access network shall set this field to the length of the complex attribute excluding the Length field.

**AttributeID** This field shall be set to 0x00.

**ValueID** This field identifies this particular set of values for the attribute. The access network shall increment this field for each complex attribute-value record for a particular attribute.

**OpenLoopTransitionTime** This field determines the transition time for open loop power adjustment, in msec. A zero value indicates that the open loop power adjustment procedures shall not be performed.

1	ACKChannelGain $j$	This field is used to determine the gain of the R-ACKCH over the R-CQICH. The value of this parameter is $(n-128)*0.125$ dB and shall be used when the erasure rate is between $j*10\%$ and $(j+1)*10\%$ .
2		
3		
4	ACKStepUpSize	This field is used to determine the gain of the R-ACKCH over the R-CQICH. The value of this parameter is $(n-128)*0.125$ dB.
5		
6	ACKExtendedFrameGain	This field determines the gain of the R-ACKCH over the R-CQICH. The value of this parameter is $(n-128)*0.125$ dB.
7		
8		
9	CQICHPowerBoostForHandoff	This field determines the amount of power boost on the R-CQICH when signaling an FL handoff. The value of this parameter is $n*0.125$ dB.
10		
11		
12	REQCHPowerBoostForHandoff	This field determines the amount of power boost on the R-REQCH when signaling an RL handoff. The value of this parameter is $n*0.125$ dB.
13		
14		

## 15 7.6.10 Protocol numeric constants

16

Constant	Meaning	Value
$N_{RCCMPType}$	Type field for this protocol	Table 9
$N_{RCCMPDefault}$	Subtype field for this protocol	0x0000

17

## 18 7.6.11 Session state information

19 The Session State Information record (see 10.10) consists of parameter records.

20 The parameter records for this protocol consist of the configuration attributes of this protocol.

## 21 7.7 Default Reverse Traffic Channel MAC Protocol

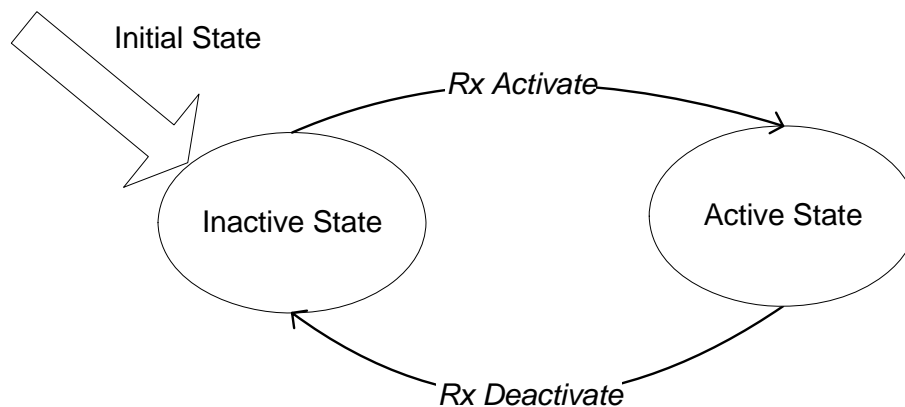
### 22 7.7.1 Overview

23 The Default Reverse Traffic Channel MAC Protocol provides the procedures and messages required  
 24 for an access terminal to transmit, and for an access network to receive, the Reverse Traffic Channel.  
 25 The access network maintains an instance of this protocol for every access terminal.

26 This protocol operates in one of two states:

- 27 ■ *Inactive State*: In this state, the access terminal is not assigned a MACID and cannot  
 28 transmit on the Reverse Traffic Channel. When the protocol is in this state, it waits for an  
 29 *Activate* command.
- 30 ■ *Active State*: In this state, the access terminal is assigned a MACID and may transmit data  
 31 on the Reverse Traffic Channel.

- 1 The protocol states and the indications and events causing the transition between the states are shown  
 2 in Figure 79.



3

4 **Figure 79 Default Reverse Traffic Channel MAC Protocol state diagram**

5 This protocol shall use the following parameters and attributes.

6

Parameter Name	Where Defined	Comments
RLChannelTreeIndex	OMP	ExtendedChannelInfo
RLControlSegmentDuration	OMP	ExtendedChannelInfo
DataCtrlOffset	pBCH1	Part of pBCH1 transmission
RDCHGainMin	ASMP	ActiveSetAssignmentMsg
RDCHGainMax	ASMP	ActiveSetAssignmentMsg
RDCHInitialPacketFormat	OMP	ExtendedChannelInfo
ErasureGain <sub>j</sub> , j=0, 1, 2, 3	OMP	ExtendedChannelInfo
RLNumSDMADimensions	OMP	ExtendedChannelInfo
ReverseLinkSilenceDuration	OMP	Public data
ReverseLinkSilencePeriod	OMP	Public data
RLImplicitDeassignEnabled	Connected State Protocol	Public data
SelectedInterlaceMode	Connected State Protocol	Public data
TuneAwayStatus	Connected State Protocol	Public data
MultiCarrierOn	Physical Layer Protocol	Public data
UpDecisionThresholdMin	Configuration Attribute	
DownDecisionThresholdMin	Configuration Attribute	
ChanDiffMax	Configuration Attribute	
ChanDiffMin	Configuration Attribute	
UpDecisionValue	Configuration Attribute	
DownDecisionValue	Configuration Attribute	
DataGainStepUp	Configuration Attribute	

Parameter Name	Where Defined	Comments
DataGainStepDown	Configuration Attribute	
RDCHGainAdjustmentThreshold	Configuration Attribute	
OSIMonitorSetSize	Configuration Attribute	
OSIMonitorThreshold	Configuration Attribute	
OSI2SequenceMax	Configuration Attribute	
FastOSIChanDiffThreshold	Configuration Attribute	

1

## 2 7.7.2 Primitives

### 3 7.7.2.1 Commands

4 This protocol defines the following commands:

- 5 ■ *Activate*
- 6 ■ *Deactivate*

### 7 7.7.2.2 Return indications

8 This protocol returns the following indications:

- 9 ■ *ReverseTrafficPacketsMissed*
- 10 ■ *SupervisionFailed*
- 11 ■ *UATIReceived*

## 12 7.7.3 Public data

### 13 7.7.3.1 Static public data

14 This protocol does not define any static public data.

### 15 7.7.3.2 Dynamic public data

- 16 ■ Subtype for this protocol

## 17 7.7.4 Protocol data unit

18 The transmission unit of this protocol is a Reverse Traffic Channel Lower MAC Sublayer packet.  
 19 Each packet contains one Security Sublayer packet.

## 7.7.5 Protocol initialization and swap

### 7.7.5.1 Protocol initialization

Upon initialization at the access terminal,

- The values of the attributes for this protocol instance shall be set to the default values specified for each attribute.
- The protocol shall enter the Active State.

Upon initialization at the access network

- The values of the attributes for this protocol instance shall be set to the default values specified for each attribute.
- The protocol shall enter the Active State.

### 7.7.5.2 Protocol Swap

Upon swap, the protocol instance shall enter the Inactive State.

## 7.7.6 Procedures

The protocol constructs a Reverse Traffic Channel Lower MAC Sublayer packet out of the Security Sublayer packet by adding the Lower MAC Sublayer header and trailer as defined in 7.7.7.

The protocol then sends the packet for transmission to the Physical Layer.

### 7.7.6.1 Command processing

#### 7.7.6.1.1 Activate

If the protocol receives an *Activate* command in the Inactive State, the access terminal and the access network shall perform the following:

- Transition to the Active State
- Internal variables of the protocol shall be initialized as mentioned in the definition of these variables.

If the protocol receives this command in any other state, the command shall be ignored.

#### 7.7.6.1.2 Deactivate

If the protocol receives a *Deactivate* command in the Active State:

- The access terminal shall cease transmitting the Reverse Traffic Channel and shall transition to the Inactive State.
- The access network shall cease monitoring the Reverse Traffic Channel from this access terminal and shall transition to the Inactive State.

If the protocol receives a *Deactivate* command in the Inactive State, the command shall be ignored.

## 1 7.7.6.2 Reverse traffic channel addressing

2 Transmission on the Reverse Traffic Channel is multiplexed in time and frequency. An assignment on  
3 the Reverse Traffic Channel shall be specified by a set of hop-ports and an interlace. Each hop-port is  
4 specified by a hop-port index as well as a carrier index. If the duplex mode is FDD, then the interlace  
5 may be composed of standard PHY Frames, or extended PHY Frames (as specified by the assignment  
6 blocks received from the SS MAC protocol). Extended PHY Frames are defined in 7.1.3.1.3.

7 The duration of an assignment of hop-ports may or may not be prespecified. Assignments whose  
8 durations are pre-specified are known as non-sticky assignments, and assignments whose durations  
9 are not pre-specified are known as sticky assignments.

10 The set of hop-ports assigned for a particular interlace for a particular access terminal via sticky  
11 assignment blocks (RLABs, RLAB\_HOs, or access grants) is referred to as the “Reverse Link Access  
12 Terminal Assignment” or RL-ATA for an interlace. An access terminal can have multiple RL-ATAs,  
13 one for each nonoverlapping interlace (note that overlapping interlaces can be created only by the use  
14 of extended PHY Frames).

15 The set of hop-ports assigned for a particular interlace for a particular access terminal via non-sticky  
16 assignment blocks (NS-RLABs) is referred to as the “Reverse Link Non-Sticky Access Terminal  
17 Assignment” or RL-NS-ATA for an interlace. An access terminal can have multiple RL-NS-ATAs,  
18 one for each nonoverlapping interlace

19 Sets of hop-ports in assignment blocks (from SS MAC) are specified using the channel tree specified  
20 by RLChannelTreeIndex. Channel tree tables are specified in 7.7.6.7.

21 Packets transmitted over the Reverse Traffic Channel are transmitted over the R-DCH Physical Layer  
22 channel. Access terminals are assigned R-DCH resources (RL-ATAs, RL-NS-ATAs) via assignment  
23 blocks (RLABs,RLAB-HOs, NS-RLABs) and AccessGrants that are sent over the F-SSCH.

24 The following rules apply regarding the coexistence of RL-ATAs and RL-NS-ATAs:

- 25 ■ An access terminal shall have no more than one RL-ATA per interlace. If duplex mode is  
26 FDD, then additionally, an access terminal shall not have any RL-ATA’s that overlap in  
27 time.
- 28 ■ An access terminal shall have no more than one RL-NS-ATA per interlace. If duplex  
29 mode is FDD, then additionally, an access terminal shall not have any RL-NS-ATA’s that  
30 overlap in time.
- 31 ■ An access terminal shall not have a non-empty RL-ATA and RL-NS-ATA on the same  
32 interlace. If duplex mode is FDD, then additionally, an access terminal shall not have a  
33 non-empty RL-ATA that overlaps in time with a non-empty RL-NS-ATA.

34 The RL-ATA for an interlace can be accumulated via multiple (sticky) assignment blocks as specified  
35 in 7.7.6.4.1.1. The RL-NS-ATA for an interlace can be accumulated via multiple (non-sticky)  
36 assignment blocks as specified in 7.7.6.4.1.4. All hop-ports in the RL-ATA/RL-NS-ATA for an  
37 access terminal in a single interlace shall be combined for transmission over the Physical Layer  
38 channel (R-DCH). Different interlaces shall always carry separate MAC packets with independent H-  
39 ARQ termination. An example is illustrated in Figure 80.



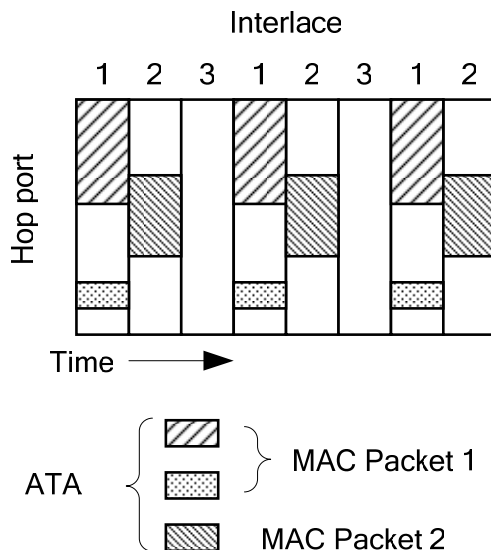


Figure 80 R-DCH addressing example

### 7.7.6.3 Inactive state

When the protocol is in the Inactive State, the access terminal and the access network wait for an *Activate* command.

### 7.7.6.4 Active state

In this section, the parameters *RDCHGainMin* and *RDCHGainMax* correspond to the RLSS.

In the Active State, the access terminal shall transmit over the R-DCH using the RL-ATAs or RL-NS-ATAs and PFs selected by the access network and signaled to the access terminal over the F-SSCH Physical Layer channel. The access terminal processes blocks from the SS MAC Protocol to maintain its RL-ATA and RL-NS-ATA, and configures the Physical Layer for transmission of packets according to the RL-ATA/RL-NS-ATA. The access network controls transmission of ACK/NACK information via the SS MAC Protocol (over the F-SSCH Physical Layer channel) based on pass or fail of the MAC packet, as determined by the PHY.

Note that an RL-NS-ATA shall be used for transmission of a single Lower MAC packet (see 7.1.2).

The MACID assigned to the access terminal for each sector in its active set shall be given in the *ActiveSetAssignment* message that is public data of the Active Set Management Protocol.

*OSI2SequenceNum* and *PilotPNStrongest* are an access terminal's locally maintained parameters. The initial value of *OSI2SequenceNum* shall be equal to 1, and the initial value of *PilotPNStrongest* shall be equal to -1.

*RDCHGain* is an access terminal's locally maintained parameter, whose initial value shall be equal to *RDCHGainMin*.

Assignment and H-ARQ logic for the access network and access terminal are specified in 7.7.6.4.1 and 7.7.6.4.2.

## 1 7.7.6.4.1 Access terminal requirements

### 2 7.7.6.4.1.1 Access terminal assignment management for sticky assignments

3 The access terminal shall maintain and manage its RL-ATA by monitoring RLABs, (and  
4 RLAB-HOs)and NS-RLABs, as well as AccessGrants received from the SS MAC protocol. For  
5 transmission on this ATA after an access grant, the access terminal shall use the packet format  
6 defined by RDCHInitialPacketFormat. After receiving an RLAB/NS-RLAB, the access terminal shall  
7 switch to the packet format specified in the RLAB/NS-RLAB.

8 If RLImplicitDeassignEnabled is equal to '1', then upon receiving a *TunedAway* indication from the  
9 Connected State Protocol, the access terminal shall expire all its RL-ATA's.

10 If SelectedInterlaceMode is equal to '1', then the access terminal shall ignore all RLABs that have the  
11 Extended Transmission field set to '1'.

12 In this section, it is assumed that all the RLABs/NS-RLABs are being sent from the same serving  
13 sector, the RLSS (see 7.6.6.3 for the definition of RLSS). The logic for access terminal assignment  
14 management during handoff is found in 7.7.6.4.1.3.

15 If the extended transmission field is set to '0', an RLAB assigns hop-ports for a particular interlace  
16 consisting of standard PHY frames as shown in 7.1.3.1.2. If the Extended Transmission field is equal  
17 to '1', an RLAB assigns hop-ports for an interlace consisting of extended PHY frames, as shown in  
18 7.1.3.1.3.

#### 19 7.7.6.4.1.1.1 Simultaneous assignments

20 If duplex mode is FDD, RLABs received in the same PHY frame could be for different interlaces  
21 because some of the RLABs could be for the interlace consisting of standard PHY frames, and some  
22 of the RLABs could be for the interlace consisting of extended PHY frames. If multiple RLABs for  
23 the access terminal's MACID are received in the same PHY frame from SS MAC, and they are not all  
24 for the same interlace, then the access terminal shall assume an error has occurred, and shall ignore all  
25 of these RLABs, unless there is an RLAB with the NodeID set to NodeID<sub>DEASSIGN</sub>, in which case this  
26 RLAB shall not be ignored.

27 If multiple RLABs for the access terminal's MACID are received from the SS MAC protocol in the  
28 same PHY frame, for the same interlace, and one of the RLABs has a NodeID set to NodeID<sub>DEASSIGN</sub>,  
29 then all RLABs except for the latter shall be discarded. This rule trumps all those which follow in this  
30 section.

31 If multiple RLABs for the access terminal's MACID are received from the SS MAC protocol in the  
32 same PHY frame, for the same interlace, and if all the values in all the fields except the NodeID field  
33 are the same, then the access terminal shall treat these RLABs as a single RLAB assigning the union  
34 of the hop-ports mapped by the constituent NodeIDs.

35 If multiple RLABs for the access terminal's MACID are received from the SS MAC protocol in the  
36 same PHY frame, for the same interlace, and if the values in at least one of the fields (excluding the  
37 NodeID field) are not the same, then the access terminal shall treat these RLABs as errors and shall  
38 ignore them.

#### 1 7.7.6.4.1.2 Supplemental and non-supplemental assignments

2 If an RLAB is received from the SS MAC protocol for the access terminal's MACID, then the hop-  
3 ports associated with the NodeID and interlace assigned by the RLAB shall be added to the RL-ATA  
4 (refer to 7.4.6.3.1.2 for the RLAB format and interpretation), according to the following rules.

5 If the Supplemental field of the RLAB for a particular interlace (see the SS MAC specification) is  
6 equal to '1', then the new RL-ATA on that interlace is the union of hop-ports included in the old  
7 RL-ATA on that interlace and hop-ports specified by the new NodeID, provided the old RL-ATA is  
8 non-empty. The PF specified in the received RLAB shall be used in place of any PFs that may have  
9 been specified in any previous assignment of NodeIDs (hop-ports) on the interlace.

10 If the Supplemental field of the RLAB is equal to '0', then the RL-ATA for the relevant interlace shall  
11 be cleared before adding the hop-ports specified by the NodeID in the RLAB to the RL-ATA for the  
12 interlace. If duplex mode is FDD, then all RL-ATAs that overlap in time with the new assignment  
13 shall be cleared before adding the hop-ports specified by the NodeID in the RLAB to the RL-ATA for  
14 the interlace.

15 The access terminal shall ignore this RLAB if any of the following conditions are satisfied:

- 16 ■ If the Supplemental field of the RLAB for a particular interlace is equal to '1', and the  
17 access terminal has an empty RL-ATA on that interlace, or
- 18 ■ If an RLAB is received from the SS MAC protocol for the access terminal's MACID and  
19 the resulting combined RL-ATA has hop-ports that belong to two sub-channel trees  
20 (see 7.7.6.7), or
- 21 ■ If an RLAB is received from the SS MAC for the access terminal's MACID for a  
22 particular interlace that does not change the access terminal's RL-ATA for that interlace.

#### 23 7.7.6.4.1.1.3 Decrementing assignments

24 If an RLAB or NS-RLAB is received from the SS MAC Protocol for a particular interlace that  
25 contains a MACID other than the access terminal's MACID, then all of the hop-ports in the RL-ATA  
26 on that interlace that intersect with hop-ports specified by the NodeID included in the RLAB/NS-  
27 RLAB shall be expired (removed from the RL-ATA) for that interlace. If duplex mode is FDD, and  
28 an RLAB/NS-RLAB is received from the SS MAC protocol for a particular interlace that contains a  
29 MACID other than the access terminal's MACID, then for each RL-ATA that overlaps in time and  
30 frequency with the hop-ports specified by this new assignment, the intersecting hop-ports shall be  
31 removed from the RL-ATA.

32 If the access terminal receives, in the same PHY frame, an RLAB with its MACID, and an NS-RLAB  
33 with a MACID other than its MACID, for the same interlace, and the hop ports assigned by the  
34 RLAB intersect with the hop ports assigned by the NS-RLAB, then the access terminal shall expire  
35 the intersecting hop ports from its RL-ATA.

36 If duplex mode is FDD, the following applies. If the access terminal receives, in the same PHY frame,  
37 an RLAB with its MACID, and an NS-RLAB with a MACID other than its MACID, for time  
38 overlapping interlaces, and the hop ports assigned by the RLAB intersect with the hop ports assigned  
39 by the NS-RLAB, then the access terminal shall expire the intersecting hop ports from its RL-ATA.

1 If the access terminal receives, in the same PHY frame, an RLAB with its MACID, and an RLAB  
 2 with a MACID other than its MACID, for the same interlace, and the hop ports assigned by the  
 3 RLABs intersect, then the access terminal shall ignore the RLAB with its MACID.

4 If duplex mode is FDD, the following applies. If the access terminal receives, in the same PHY frame,  
 5 an RLAB with its MACID, and an RLAB with a MACID other than its MACID, for time overlapping  
 6 interlaces, and the hop ports assigned by the RLABs intersect, then the access terminal shall ignore  
 7 the RLAB with its MACID.

#### 8 **7.7.6.4.1.1.4 Deassigning assignments**

9 If an RLAB is received from the SS MAC protocol for the access terminal's MACID that assigns the  
 10 reserved NodeID<sub>DEASSIGN</sub>, then the RL-ATA on that interlace shall be expired.

#### 11 **7.7.6.4.1.1.5 Time overlapping sticky and non-sticky assignments**

12 If the access terminal receives, from SS MAC, a NS-RLAB with its MACID for a particular interlace,  
 13 while it already has a non-empty RL-ATA on that interlace, the access terminal shall keep the  
 14 resulting RL-NS-ATA, and clear the RL-ATA on that interlace.

15 Furthermore, if duplex mode is FDD, and the access terminal receives, from the SS MAC, a NS-  
 16 RLAB with its MACID, then the access terminal shall clear any RL-ATA that overlaps in time with  
 17 the new RL-NS-ATA.

18 If the access terminal receives, in the same PHY frame, a NS-RLAB giving it a non-empty  
 19 RL-NS-ATA, and an RLAB giving it a non-empty RL-ATA, then the access terminal shall ignore the  
 20 RLAB.

#### 21 **7.7.6.4.1.2 Access terminal assignment management for non-sticky assignments**

22 The access terminal shall maintain and manage its RL-NS-ATAs by monitoring RLABs and NS-  
 23 RLABs delivered from the SS MAC Protocol.

24 If RLImplicitDeassignEnabled is equal to '1', then upon receiving a *TunedAway* indication from the  
 25 Connected State Protocol, the access terminal shall expire all its RL-NS-ATA's.

26 If SelectedInterlaceMode is equal to '1', then the access terminal shall ignore all NS-RLABs that  
 27 have the Extended Transmission field set to '1'.

28 In this section, it is assumed that all the RLABs/NS-RLABs are being sent from the same serving  
 29 sector, the RLSS (see 7.6.6.3 for the definition of RLSS). The logic for access terminal assignment  
 30 management during handoff is found in 7.7.6.4.1.3.

31 If the extended transmission field is equal to '0', an NS-RLAB assigns hop-ports for a particular  
 32 interlace consisting of standard PHY frames as shown in 7.1.3.1.2. The duration field specifies the  
 33 number of standard PHY frames to be used for transmission of this assignment. If the Extended  
 34 Transmission field is equal to '1', an NS-RLAB assigns hop-ports for an interlace consisting of  
 35 extended PHY frames, as shown in 7.1.3.1.3. The duration field specifies the number of extended  
 36 PHY frames to be used for transmission of this assignment.

#### 1 **7.7.6.4.1.2.1 Simultaneous assignments**

2 If duplex mode is FDD, NS-RLABs received in the same PHY frame could be for different interlaces  
 3 because some of the NS-RLABs could be for the interlace consisting of standard PHY frames, and  
 4 some of the NS-RLABs could be for the interlace consisting of extended PHY frames. If multiple  
 5 NS-RLABs for the access terminal's MACID are received in the same PHY frame from SS MAC,  
 6 and they are not all for the same interlace, then the access terminal shall assume an error has  
 7 occurred, and shall ignore all of these NS-RLABs, unless there is an RLAB with the NodeID set to  
 8 NodeID<sub>DEASSIGN</sub>, in which case this RLAB shall not be ignored.

9 If multiple NS-RLABs for the access terminal's MACID are received from the SS MAC Protocol in  
 10 the same PHY frame, and one of the NS-RLABs has a NodeIDs set to NodeID<sub>DEASSIGN</sub>, then all  
 11 NS-RLABs except for the latter shall be discarded. This rule trumps all those which follow in this  
 12 section.

13 If multiple NS-RLABs for the access terminal's MACID are received from the SS MAC protocol in  
 14 the same PHY frame, if all the values all the fields except the NodeID field are the same, then the  
 15 access terminal shall treat these NS-RLABs as a single NS-RLAB assigning the union of the hop-  
 16 ports mapped by the constituent NodeIDs.

17 If multiple NS-RLABs for the access terminal's MACID are received from the SS MAC Protocol in  
 18 the same PHY frame, and if the values in at least one of the fields (excluding the NodeID field) are  
 19 not the same, then the access terminal shall treat these NS-RLABs as errors and shall ignore them.

#### 20 **7.7.6.4.1.2.2 Deassigning assignments**

21 If an NS-RLAB is received from the SS MAC protocol for the access terminal's MACID that assigns  
 22 the reserved NodeID<sub>DEASSIGN</sub>, then the RL-NS-ATA on that interlace shall be expired.

#### 23 **7.7.6.4.1.2.3 Time overlapping non-sticky assignments**

24 If a NS-RLAB is received from the SS MAC Protocol for a particular interlace with the access  
 25 terminal's MACID, and the access terminal already has a RL-NS-ATA for that interlace, then the new  
 26 assignment block takes precedence: the access terminal shall stop trying to decode on the old  
 27 RL-NS-ATA for that interlace (shall clear this RL-NS-ATA), and shall update its RL-NS-ATA for  
 28 that interlace according to the new NS-RLAB.

29 If duplex mode is FDD, and a NS-RLAB is received from the SS MAC Protocol with the access  
 30 terminal's MACID that assigns a non-empty RL-NS-ATA, then all RL-NS-ATAs that overlap in time  
 31 with the new RL-NS-ATA shall be expired.

#### 32 **7.7.6.4.1.2.4 Time overlapping sticky and non-sticky assignments**

33 If the access terminal receives, from SS MAC, a RLAB with its MACID for a particular interlace,  
 34 while it already has a non-empty RL-NS-ATA on that interlace, the access terminal shall keep the  
 35 resulting RL-ATA, and clear the RL-NS-ATA on that interlace.

36 Furthermore, if duplex mode is FDD, and the access terminal receives, from the SS MAC, an RLAB  
 37 with its MACID, then the access terminal shall clear any RL-NS-ATA that overlaps in time with the  
 38 new RL-ATA.

#### 1 7.7.6.4.1.2.5 Overlapping assignments from other ATs

2 If an RLAB or NS-RLAB is received from the SS MAC Protocol for a particular interlace that  
3 contains a MACID other than the access terminal's MACID, then if the hop-ports specified by the  
4 NodeID included in the RLAB/NS-RLAB overlap with the RL-NS-ATA on that interlace, the  
5 RL-NS-ATA shall be expired. If duplex mode is FDD, then all RL-NS-ATAs that intersect in time  
6 and frequency with the new assignment shall be expired.

7 If the access terminal receives, in the same PHY frame, an NS-RLAB with its MACID, and an RLAB  
8 with a MACID other than the access terminal's MACID, for the same interlace, and the hop ports  
9 assigned by the NS-RLAB intersect with the hop ports assigned by the RLAB, the access terminal  
10 shall ignore the RLAB (in the sense that the NS-ATA resulting from the NS-RLAB will not be  
11 modified/expired because of the RLAB).

12 If duplex mode is FDD, the following applies. If the access terminal receives, in the same PHY frame,  
13 an NS-RLAB with its MACID, and an RLAB with a MACID other than the access terminal's  
14 MACID, for time overlapping interlaces, and the hop ports assigned by the NS-RLAB intersect with  
15 the hop ports assigned by the RLAB, the access terminal shall ignore the RLAB (in the sense that the  
16 NS-ATA resulting from the NS-RLAB will not be modified/expired because of the RLAB).

17 If the access terminal receives, in the same PHY frame, an NS-RLAB with its MACID and an  
18 NS-RLAB with a MACID other than its MACID, for the same interlace, and the hop ports assigned  
19 by the NS-RLABs overlap, the access terminal assumes as error has occurred, and shall ignore the  
20 NS-RLAB with its MACID.

21 If duplex mode is FDD, the following holds. If the access terminal receives, in the same PHY frame,  
22 an NS-RLAB with its MACID and an NS-RLAB with a MACID other than its MACID, for time  
23 overlapping interlaces, and the hop ports assigned by the NS-RLABs overlap, the access terminal  
24 assumes as error has occurred, and shall ignore the NS-RLAB with its MACID.

#### 25 7.7.6.4.1.3 Access terminal assignment management during handoff

26 If an *RLSSChanged* indication from the RCC MAC Protocol is received, the access terminal shall  
27 clear all RL-ATAs and RL-NS-ATAs associated with the old RLSS.

28 If the access terminal receives an RLAB-HO with its MACID, and the sector in the DesiredSector  
29 field is different from the RLSS, the access terminal shall issue a  
30 *ReverseControlChannelMAC.ChangeRLSS* to change the RLSS to the sector in the DesiredSector  
31 field of the RLAB-HO. If the access terminal then receives an *RLSSChanged* indication from the  
32 RCC MAC Protocol, the access terminal shall process the *RLSSChanged* indication as specified  
33 above, and in addition shall update its new RL-ATA to the hop ports specified by the NodeID field in  
34 the RLAB-HO for the interlace specified by the ExtendedTransmission field in the RLAB-HO, for the  
35 sector specified in the DesiredSector field. The packet format to be used is specified in the PF field of  
36 the RLAB-HO.

37 If the access terminal receives an RLAB-HO with its MACID, and the sector in the DesiredSector  
38 field is not different from the RLSS, the access terminal shall ignore this RLAB-HO.

39 If the access terminal receives multiple RLAB-HOs with its MACID, and all the fields are the same  
40 except for the NodeID fields, then the access terminal shall treat these RLAB-HOs as a single RLAB-  
41 HO assigning the union of the hop ports specified by the NodeID fields.

1 If the access terminal receives multiple RLAB-HOs with its MACID, and all the fields other than the  
2 NodeID field are not the same, then the access terminal shall ignore all these RLAB-HOs.

3 If the access terminal receives an RLAB/NS-RLAB with the access terminal's MACID, that has the  
4 supplemental field set to '0', from the DRLSS, while the DRLSS is different from the RLSS, the  
5 access terminal shall issue a *ReverseControlChannelMAC.ChangeRLSS* command to change from the  
6 RLSS to the DRLSS. If the access terminal then receives an *RLSSChanged* indication from the RCC  
7 MAC Protocol, the access terminal shall process the *RLSSChanged* indication as specified above, and  
8 in addition shall update the appropriate RL-ATA/RL-NS-ATA according to the new  
9 RLAB/NS-RLAB.

10 If the access terminal receives an access grant for handoff<sup>39</sup> with its MACID, the access terminal shall  
11 issue a *ReverseControlChannelMAC.ChangeFLSS* command to change the FLSS to the sector from  
12 which the access grant was sent, and shall also issue a *ReverseControlChannelMAC.ChangeRLSS*  
13 command to change the RLSS to the sector from which the access grant was sent. If the access  
14 terminal then receives an *RLSSChanged* indication from the RCC Protocol, the access terminal shall  
15 clear all RL-ATAs/RL-NS-ATAs it may have, and in addition shall update an RL-ATA to be equal to  
16 the hop ports specified by the NodeID field in the AccessGrant, for the sector from which the access  
17 grant was sent.

18 The access terminal shall ignore all RLABs or NS-RLABs that come from sectors other than its  
19 current RLSS or its DRLSS.

#### 20 **7.7.6.4.1.4 Access terminal transmission logic for sticky assignments**

21 The access terminal may formulate and transmit a MAC packet on a PHY Frame on an interlace  
22 according to the RL-ATA on the interlace and the PF selected for transmission. The MAC shall pass  
23 the Physical Layer Protocol the set of hop ports specified by the RL-ATA, and shall specify whether  
24 or not the assignment is an Extended Transmission Duration Assignment (see 7.1.3.1.3). The power  
25 for transmission of a particular MAC packet is computed as specified in 7.7.6.4.1.6. The access  
26 network informs the access terminal of its assignment using signaling messages as described in the  
27 SS MAC Protocol specification.

28 If TuneAwayStatus is equal to '1', then the access terminal shall not send any MAC packets on any  
29 interlaces.

30 If a positive ACK signal corresponding to the RL-ATA is received (via the SS MAC Protocol) that  
31 corresponds to the transmitted packet, transmission of the MAC packet shall terminate, and the  
32 interlace is immediately available for the next packet. If the access terminal transmits a packet for the  
33 maximum number of transmissions of the PF selected for this MAC packet the access terminal shall  
34 automatically terminate transmission of the packet. If no ACK is received for this packet, the access  
35 terminal shall expire its RL-ATA, and return a *ReverseTrafficPacketsMissed* indication along with  
36 parameters that uniquely identify the lost packet. The method of uniquely identifying the packet is out  
37 of the scope of this specification.

---

<sup>39</sup> Some AccessGrants are sent for the purpose of obtaining timing and/or power offset information from a sector and should not be used for handoff logic. Such AccessGrants are scrambled using an AccessSequenceID that belongs to the set of IDs reserved for timing/power correction, as described in 7.3.6.4.1.4.3.

1 If an RLAB/NS-RLAB for a particular interlace is received from the SS MAC protocol that leaves the  
 2 RL-ATA on that interlace non-empty, and if the access terminal is currently transmitting a packet on  
 3 that interlace, then the access terminal shall return a *ReverseTrafficPacketsMissed* indication along  
 4 with parameters that uniquely identify the lost packet and shall cease transmitting this packet. The  
 5 method of uniquely identifying the packet is out of the scope of this specification. The access terminal  
 6 shall then update its RL-ATA in accordance with the new RLAB/NS-RLAB.

7 If no packet is available for transmission in a given PHY Frame, then an erasure sequence shall be  
 8 transmitted in the hop-ports, or subset of hop-ports, assigned to the access terminal in that PHY  
 9 Frame as specified in section Physical Layer specification. The power for transmission of an erasure  
 10 sequence is computed as specified in 7.7.6.4.1.6. If duplex mode is FDD, the PHY Frame may be  
 11 either a standard PHY Frame or an extended PHY Frame.

#### 12 **7.7.6.4.1.5 Access terminal transmission logic for non-sticky assignments**

13 The access terminal may formulate and transmit a MAC packet on a PHY Frame on an interlace  
 14 according to the RL-NS-ATA on the interlace and the PF selected for transmission. The MAC shall  
 15 pass the Physical Layer Protocol the set of hop ports specified by the RL-NS-ATA, and shall specify  
 16 whether or not the assignment is an Extended Transmission Duration Assignment (see 7.1.3.1.3). The  
 17 power for transmission of a particular MAC packet is computed as specified in 7.7.6.4.1.6. The access  
 18 network informs the access terminal of its assignment using signaling messages as described in the  
 19 SS MAC Protocol specification.

20 The number of H-ARQ retransmissions over a RL-NS-ATA is equal to the duration of the  
 21 RL-NS-ATA, as specified by the duration field in the non-sticky assignment block (see 7.4.6.3.1.2).

22 If TuneAwayStatus is equal to '1', then the access terminal shall not send any MAC packets on any  
 23 interlaces.

24 If a positive ACK signal corresponding to the RL-NS-ATA is received (via the SS MAC Protocol)  
 25 that corresponds to the transmitted packet, transmission of the MAC packet shall terminate, and the  
 26 access terminal shall expire the RL-NS-ATA. If no ACK is received for this packet, the access  
 27 terminal shall return an indication along with parameters that uniquely identify the lost packet. The  
 28 method of uniquely identifying the packet is out of the scope of this specification.

#### 29 **7.7.6.4.1.6 R-DCH power control**

30 During the transmission of the reverse link data, the power of the R-DCH,  $P_{DCH}$ , shall be:

$$31 \quad P_{DCH} = P_{CTRL} - 10 \log_{10}(N_{CTRL-SUBCARRIERS}) + 10 \log_{10}(N_c) + RDCHGain + \\ DataCtrlOffset$$

32 where  $P_{CTRL}$  is the reference value used by the access terminal in adjusting the mean output power of  
 33 the reverse control channels and is given in 7.6.6.3.5,  $N_c$  is the number of hop-ports in the  
 34 RL-ATA/RL-NS-ATA for this transmission,  $N_{CTRL-SUBCARRIERS}$  is the number of subcarriers allocated  
 35 for reverse control channels, *DataCtrlOffset* is a parameter transmitted over pBCH1 of the RLSS, and  
 36 *RDCHGain* is as specified in 7.7.6.4.1.7. The R-DCH power shall further be subject to the access  
 37 terminal's transmit power limitation and shall remain constant for the entire transmission of each  
 38 PHY Frame.



1 If no packet is available for transmission in a given PHY Frame, then an erasure sequence shall be  
 2 transmitted with the following power:

$$3 \quad P_{ERASURE} = P_{CTRL} - 10 \log_{10}(N_{CTRL-SUBCARRIERS}) + 10 \log_{10}(N_{c,erasure}) + \text{ErasureGain}_i \\ + \text{DataCtrlOffset}$$

4 where  $N_{c,erasure}$  is the number of subcarriers over which the erasure sequence is transmitted, and is  
 5 always set to 16. ErasureGain<sub>i</sub>,  $i=0,1,2,3$  is given in the OverheadParameterList of the Overhead  
 6 Messages Protocol. ErasureGain<sub>0</sub> is used when  $N_c > 64$  and the erasure is being sent over a single PHY  
 7 frame. ErasureGain<sub>1</sub> is used when  $N_c > 64$  and the erasure is being sent over an extended PHY frame.  
 8 ErasureGain<sub>2</sub> is used when  $N_c \leq 64$  and the erasure is being sent over a single PHY frame.  
 9 ErasureGain<sub>3</sub> is used when  $N_c \leq 64$  and the erasure is being sent over an extended PHY frame.

10 The reverse link erasure power shall further be subject to the access terminal's transmit power  
 11 limitation and shall remain constant for the entire transmission of each PHY Frame. If the access  
 12 terminal does not have enough power to send at  $P_{ERASURE}$  due to power limitations, then it shall send  
 13 at its maximum power.

#### 14 **7.7.6.4.1.6.1 OSIMonitorSet update**

15 The access terminal shall update the OSIMonitorSet according to the following rules:

- 16 ■ If the access terminal is monitoring the Fast OSI value on the F-SSCH of any active set  
 17 member other than the RLSS, then every FL PHY frame, the access terminal shall update  
 18 the OSIMonitorSet with a list of PilotPNs of the sectors in the active set whose Fast OSI  
 19 value are being monitored by the access terminal, and whose ChanDiff values, as defined  
 20 in 7.7.6.4.1.7, are smaller than or equal to FastOSIChanDiffThreshold, where  
 21 FastOSIChanDiffThreshold is a configuration attribute of the protocol.
- 22 ■ At the beginning of every superframe of the RLSS, the access terminal shall update the  
 23 OSIMonitorSet with a list of PilotPN's of the sectors whose PilotStrength is larger than  
 24 or equal to the OSIMonitorThreshold, where PilotPN and PilotStrength are parameters in  
 25 the OverheadParameterList of the Overhead Messages Protocol and  
 26 OSIMonitorThreshold is a configuration attribute of the protocol.

27 The OSIMonitorSet shall exclude the PilotPN of the RLSS. In addition, if the size of the list is larger  
 28 than or equal to OSIMonitorSetSize (where OSIMonitorSetSize is a configuration attribute of the  
 29 protocol), only OSIMonitorSetSize PilotPN's with strongest PilotStrength shall be kept.

#### 30 **7.7.6.4.1.7 RDCHGain determination**

31 If an RLAB or NS-RLAB is received from SS MAC which results in a non empty RL-ATA or RL-  
 32 NS-ATA, and the RLAB/NS-RLAB was received later than the last time RDCHGain was calculated,  
 33 and additionally the Delta field in the RLAB/NS-RLAB is not the reserved Delta field, then  
 34 RDCHGain will be set equal to the Delta specified by the Delta field. The encoding of the Delta field  
 35 shall be as follows

- 36 ■ The value 7 shall be reserved
- 37 ■ Values between 0 and 6 shall be interpreted as a linear interpolation between  
 38 RDCHGainMin and RDCHGainMax with  $\text{RDCHGainIndex} = \text{floor}[(6.5 /$   
 39  $(\text{RDCHGainMax} - \text{RDCHGainMin}) * (\text{RDCHGain} - \text{RDCHGainMin})]$ .

1 Otherwise, RDCHGain will be calculated as follows.

2 After each OSIMonitorSet update, the access terminal shall create an OSI vector whose  $i^{\text{th}}$  element,  
 3 i.e.,  $OSI_i$ , corresponds to the most recent OSIValue from the sector whose PilotPN is indicated by the  
 4  $i^{\text{th}}$  entry of the OSIMonitorSet. The most recent OSIValue can be a value received over the F-OSICH  
 5 of the sector or over the Fast OSI Segment of the F-SSCH of the sector. In addition, the access  
 6 terminal shall create a ChanDiff vector whose  $i^{\text{th}}$  element, i.e.,  $ChanDiff_i$ , corresponds to:

$$7 \quad ChanDiff_i = \frac{RxPower_{RL,SS}}{TransmitPower_{RL,SS}} \times \frac{TransmitPower_i}{RxPower_i}$$

8 where  $RxPower_{RL,SS}$  and  $RxPower_i$ , contained in the public data of the Active Set Management  
 9 Protocol, correspond to the average received power (across antenna) of the F-ACQCH of the RLSS,  
 10 and the average received power (across antenna) of the F-ACQCH of the sector whose PilotPN is  
 11 indicated by the  $i^{\text{th}}$  entry of the OSIMonitorSet, respectively.  $TransmitPower_{RL,SS}$  and  $TransmitPower_i$ ,  
 12 specified in the OverheadParameterList of the Overhead Messages Protocol, correspond to the  
 13 average transmit power of the F-ACQCH of the RLSS, and the average transmit power of the F-  
 14 ACQCH of the sector whose PilotPN is indicated by the  $i^{\text{th}}$  entry of the OSIMonitorSet, respectively.  
 15 The above calculation shall be done in a linear unit. The access terminal shall determine RDCHGain  
 16 as follows:

17 If the OSIMonitorSet is empty, the access terminal shall set RDCHGain to RDCHGainMax,  
 18 OSI2SequenceNum to 1 and PilotPNStrongest to -1. RDCHGainMax is a parameter in the  
 19 OverheadParameterList of the Overhead Messages Protocol.

20 If the OSIMonitorSet is not empty, the access terminal shall compute the RDCHGain as follows:

- 21 ■ The access terminal shall first compute a Decision Threshold vector, whose  $i^{\text{th}}$  element,  
 22 i.e.,  $DecisionThreshold_i$ ,  $1 \leq i \leq OSIMonitorSetSize$ , is given by:

$$23 \quad DecisionThreshold_i = \begin{cases} \max\{UpDecisionThresholdMin, (1-a)b_i\} & \text{if } OSI_i = 0 \\ \max\{DownDecisionThresholdMin, a(1-b_i)\} & \text{if } OSI_i = 1 \\ 1 & \text{if } OSI_i = 2 \end{cases}$$

24 where UpDecisionThresholdMin and DownDecisionThresholdMin are configuration  
 25 attributes of the protocol. Variables a and  $b_i$  are determined as follows:

$$26 \quad a = \frac{\min\{RDCHGain, RDCHGainMax\} - RDCHGainMin}{RDCHGainMax - RDCHGainMin}, \text{ and}$$

$$27 \quad b_i = \frac{\min\{ChanDiff_i, ChanDiffMax\} - ChanDiffMin}{ChanDiffMax - ChanDiffMin},$$

28 where ChanDiffMax and ChanDiffMin are configuration attributes of the protocol, and all  
 29 values in the above computations are in logarithmic scale (in units of dB).

- 1        ■ The access terminal shall produce a Decision vector whose  $i^{\text{th}}$  element, i.e.,  $Decision_i$ ,  
 2         $1 \leq i \leq OSIMonitorSetSize$ , is given by:

$$3 \quad Decision_i = \begin{cases} UpDecisionValue & \text{if } x_i \leq DecisionThreshold_i \text{ and } OSI_i = 0 \\ -DownDecisionValue & \text{if } x_i \leq DecisionThreshold_i \text{ and } OSI_i = 1 \text{ or } 2 \\ 0 & \text{otherwise} \end{cases}$$

4        where  $0 \leq x_i \leq 1$  is a uniform random variable (generated using the procedure specified in  
 5        the Common Algorithm Section) and  $UpDecisionValue$  and  $DownDecisionValue$  are  
 6        configuration attributes of the protocol.

- 7        ■ The access terminal shall then compute a weighted decision,  $D_w$ , according to:

$$8 \quad D_w = \frac{\sum_{i=1}^{OSIMonitorSetSize} \frac{1}{ChanDiff_i} Decision_i}{\sum_{i=1}^{OSIMonitorSetSize} \frac{1}{ChanDiff_i}}$$

9        The access terminal shall find the sector with the lowest  $ChanDiff$  in the  $OSIMonitorSet$  and call that  
 10        sector as sector  $k$ . Then the access terminal shall set the variable  $OSIStrongest$  to the  $OSI$  value of  
 11        sector  $k$  and  $PilotPNCcurrent$  to the  $PilotPN$  of sector  $k$ . Then, the access terminal shall update  
 12         $OSI2SequenceNum$  and  $PilotPNStrongest$  as follows:

$$13 \quad OSI2SequenceNum = \begin{cases} OSI2SequenceNum + 1 & \text{if } PilotPNCcurrent = PilotPNStrongest \text{ and} \\ & OSI2SequenceNum \leq OSI2SequenceNumMax - 1 \text{ and} \\ & OSIStrongest = 2 \\ 2 & \text{if } PilotPNCcurrent \neq PilotPNStrongest \text{ and} \\ & OSIStrongest = 2 \\ 1 & \text{otherwise} \end{cases}$$

$$14 \quad PilotPNStrongest = \begin{cases} PilotPNCcurrent, & \text{if } OSIStrongest = 2 \\ -1 & \text{, otherwise} \end{cases}$$

1 where OSI2SequenceNumMax is a configuration attribute of the protocol.

- 2 ■ The access terminal shall increase RDCHGain by DataGainStepUp dB if  $D_w$  is greater  
3 than RDCHGainAdjustmentThreshold and shall decrease RDCHGain by  
4 DataGainStepDown\*OSI2SequenceNum dB if  $D_w$  is less than or equal to  
5 -RDCHGainAdjustmentThreshold, where DataGainStepUp, DataGainStepDown, and  
6 RDCHGainAdjustmentThreshold are configuration attributes of the protocol.  
7 Furthermore, the RDCHGain shall always lie between RDCHGainMin and  
8 RDCHGainMax. That is, the access terminal shall set RDCHGain to RDCHGainMin if  
9 the resulting RDCHGain is smaller than RDCHGainMin and to RDCHGainMax if the  
10 resulting RDCHGain is larger than RDCHGainMax.

#### 11 **7.7.6.4.1.8 OSIReport message procedures**

12 If the access terminal receives a OSIReportRequest message, the access terminal shall respond with a  
13 OSIReport message within 2 superframes from the time of receipt of the OSIReportRequest message.,  
14 The access terminal shall include a history of duration RequestedHistory superframes where  
15 RequestedHistory is a field of the OSIReportRequest message.

#### 16 **7.7.6.4.2 Access network requirements for sticky assignments**

17 For each interlace with a non-empty RL-ATA (contains one or more hop-ports), the access network  
18 may attempt to decode a MAC packet transmitted on the interlace. The access network may attempt  
19 to detect erasure sequences that are transmitted by the access terminal whenever a MAC packet is not  
20 available for transmission. Exact algorithms for detecting erasure sequences and the start-of-packet  
21 for MAC packets that span multiple PHY Frames are beyond the scope of this specification.

22 If TuneAwayStatus is equal to '1', then the access network should not attempt to decode any packets  
23 from that access terminal.

24 If RLImplicitDeassignEnabled is equal to '1', then upon receiving a *TunedAway* indication from the  
25 Connected State Protocol, the access network shall expire any RL-ATAs for that access terminal.

26 If a MAC packet is successfully decoded, as indicated by the PHY, the access network shall transmit  
27 a positive ACK on the F-SSCH channel via the SS MAC Protocol. If a MAC packet fails to decode  
28 and the access network determines that the packet has been transmitted for the maximum number of  
29 PHY Frames for the relevant PF, then the access network shall expire the RL-ATA for that interlace.  
30 Refer to 7.1.3 for detailed interlaced structure and acknowledgment timing for both FDD and TDD  
31 modes.

32 Exact algorithms to determine the number of H-ARQ attempts prior to successful decode are beyond  
33 the scope of this specification.

34 If a MAC packet is successfully decoded, the payload of the packet is then passed up to the Security  
35 Sublayer for further processing. If the MAC packet has a UATIInfoIncluded header field set to '1',  
36 this protocol shall generate a *UATIReceived* indication, accompanied by the MAC header of the  
37 received packet.

38 If the RLNumSDMADimensions > 1, then the access network shall ensure that assignments to  
39 different ATs that contain hop-ports that map to the same subcarriers have the same F-DPICH format,  
40 as described in 7.7.6.8.

### 1 7.7.6.4.3 Access network requirements for non- sticky assignments

2 For each interlace with a non-empty RL-NS-ATA (contains one or more hop-ports), the access  
3 network may attempt to decode a MAC packet transmitted on the interlace.

4 If TuneAwayStatus is equal to '1', then the access network should not attempt to decode any packets  
5 from that access terminal.

6 If RLImplicitDeassignEnabled is equal to '1', then upon receiving a *TunedAway* indication from the  
7 Connected State Protocol, the access network shall expire any RL-NS-ATAs for that access terminal.

8 If a MAC packet is successfully decoded, as indicated by the PHY, the access network shall transmit  
9 a positive ACK on the F-SSCH channel via the SS MAC Protocol, and shall expire the RL-NS-ATA.

10 Exact algorithms to determine the number of H-ARQ attempts prior to successful decode are beyond  
11 the scope of this specification.

12 If a MAC packet is successfully decoded, the payload of the packet is then passed up to the Security  
13 Sublayer for further processing. If the MAC packet has a UATIInfoIncluded header field set to '1',  
14 this protocol shall generate a *UATIReceived* indication, accompanied by the MAC header of the  
15 received packet.

16 If the RLNumSDMADimensions > 1, then the access network shall ensure that (non-sticky)  
17 assignments to different ATs that contain hop-ports that map to the same subcarriers have the same  
18 F-DPICH format, as described in section 7.7.6.8.

19 The RL-NS-ATA shall expire after the specified duration (see 7.4.6.3.1.2), if it is not already expired.  
20 (This duration is specified by the assignment block received from SS MAC.)

### 21 7.7.6.5 Reverse link silence interval

22 The access terminal shall not transmit on any RL channel if the transmission of that channel would  
23 overlap with the Reverse Link Silence Interval<sup>40</sup>. This rule shall override any requirement for  
24 transmission stated elsewhere in this section or in the Reverse Control Channel MAC.

25 The Reverse Link Silence Interval is defined by ReverseLinkSilenceDuration and  
26 ReverseLinkSilencePeriod in the Overhead Messages Protocol.

### 27 7.7.6.6 Supervision procedures

28 The access terminal shall generate a *SupervisionFailed* indication when it does not receive a reverse  
29 traffic channel assignment during a time period of length  $T_{\text{RTCSupervision}}$  while it has sent non-empty  
30 request bits during all R-REQCH transmissions during the time period.

---

<sup>40</sup> This implies that the access terminal must not even start transmission on the Reverse Traffic Data Channel if the transmission of the Reverse Traffic Channel packet would overlap with the Reverse Link Silence Interval.

### 7.7.6.7 Channel trees

A channel tree defines the mapping of each NodeID to a set of hop-ports and the grouping of hop-ports into port-sets. A channel tree on the RL is indexed by RLChannelTreeIndex and the number of subcarriers mapped by the channel tree,  $N_{\text{CARRIER\_SIZE}}$ , a parameter that is defined by the Physical Layer protocol. See 7.1.4.1 for common terms used for describing channel trees in this specification. Hop-ports shall be numbered numerically from 0.

$Q_{\text{SDMA}}$  equals RLNumSDMADimensions.

The set of hop-ports specified by a NodeID shall be the union of all hop-ports mapped by all base-nodes that are descendants of the node specified by NodeID, minus unusable hop-ports, which are defined by the Physical Layer Protocol.

Hop-ports may be grouped into disjoint port-sets for frequency reuse purposes.

The number of hop-ports indexed by the tree shall equal  $Q_{\text{SDMA}} * N_{\text{CARRIER\_SIZE}}$ , and the total number of nodes in the tree shall be a function of  $N_{\text{CARRIER\_SIZE}}$  and  $Q_{\text{SDMA}}$ , where  $Q_{\text{SDMA}}$  is the RL multiplexing factor.

Note that when multi-carrier mode is equal to MultiCarrierOn, there is an independent channel tree per carrier, and the channel tree in use for the carrier is signaled on the overhead channels of that carrier. Further, when a specific NodeID or set of hop-ports is communicated with other protocols in this specification, the associated carrier must also be communicated.

#### 7.7.6.7.1 RL channel tree index 0

Channel trees associated with RLChannelTreeIndex 0 are illustrated in Figure 81 for  $N_{\text{CARRIER\_SIZE}} = 512, 1024, \text{ and } 2048$ . For  $N_{\text{CARRIER\_SIZE}} = 512$ , all nodes above the dashed line marked with  $N_{\text{CARRIER\_SIZE}} = 512$  are included in the channel tree; For  $N_{\text{CARRIER\_SIZE}} = 1024$ , all nodes above the dashed line marked with  $N_{\text{CARRIER\_SIZE}} = 1024$  are included in the channel tree; For  $N_{\text{CARRIER\_SIZE}} = 2048$ , all nodes above the dashed line marked with  $N_{\text{CARRIER\_SIZE}} = 2048$  are included in the channel tree.

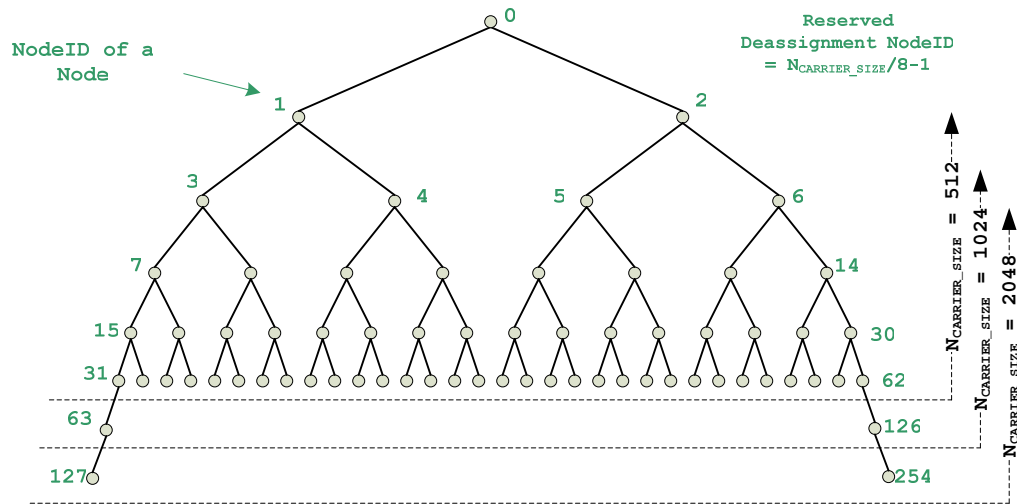
MinHopPortsPerNode equals 16 for RLChannelTreeIndex 0.

The figure shows only  $1/Q_{\text{SDMA}}$  of the total tree, and there are  $Q_{\text{SDMA}}$  identical versions of the illustrated tree each with unique NodeIDs and mapping unique hop-ports. For example, if the total tree is composed of  $Q_{\text{SDMA}}$  identical trees indexed by  $i=0, 1, \dots, Q_{\text{SDMA}}-1$  then NodeIDs of nodes on the  $i$ th tree can be obtained from the illustrated tree by adding  $i * N_{\text{CARRIER\_SIZE}} / (\text{MinHopPortsPerNode} / 2)$  to the NodeID of the illustrated tree. The number of hop-ports indexed by the tree shall equal  $Q_{\text{SDMA}} N_{\text{CARRIER\_SIZE}}$ , and the total number of nodes in the tree shall be a function of  $N_{\text{CARRIER\_SIZE}}$ . Namely, the base-nodes are defined by the intervals  $\text{NodeID} = i * N_{\text{CARRIER\_SIZE}} / (\text{MinHopPortsPerNode} / 2) + \{ N_{\text{CARRIER\_SIZE}} / \text{MinHopPortsPerNode} - 1, N_{\text{CARRIER\_SIZE}} / (\text{MinHopPortsPerNode} / 2) - 2 \}, i=0, 1, \dots, Q_{\text{SDMA}}-1$ . Thus, for  $N_{\text{CARRIER\_SIZE}}=512$  and  $Q_{\text{SDMA}}=4$ , there are 128 base-node NodeIDs, 31 through 62, 95 through 126, 159 through 190, and 223 through 254. For nodes on the same level of a channel tree, the NodeID associated with a node increases from left to right with step of 1. One deassignment NodeID, NodeID<sub>DEASSIGN</sub>, is set to  $N_{\text{CARRIER\_SIZE}} / (\text{MinHopPortsPerNode} / 2) - 1$ .

1 The mapping of hop-ports to each base-node is described as follows. Each base-node maps to  
 2 MinHopPortsPerNode hop-ports, the first MinHopPortsPerNode hop-ports (indices 0 to  
 3 MinHopPortsPerNode-1) to the base-node with the lowest NodeID, the second MinHopPortsPerNode  
 4 hop-ports to the next base-node, etc., until all hop-ports are mapped. See Table 92 for an example of  
 5 this mapping for  $N_{\text{CARRIER\_SIZE}}=512$ , MinHopPortsPerNode=16, and  $Q_{\text{SDMA}}=4$ .

6 **Table 92 Base node NodeID to Hop-port Mapping Example**  
 7 **( $N_{\text{CARRIER\_SIZE}}=512$ , MinHopPortsPerNode=16, and  $Q_{\text{SDMA}}=4$ )**

Base node NodeID	Hop-ports mapped
31	0-15
32	16-31
...	...
62	496-511
95	512-527
96	528-543
...	...
126	1008-1023
...	...
254	2032-2047



8 **Figure 81 RL channel trees with index 0**

### 7.7.6.8 Packet formats

A packet format (PF) specifies the pilot pattern, spectral efficiency, maximum number of transmissions, and the modulation format to be used for each transmission of a data packet. The packet format consists of five bits. The first bit of the PF indicates the F-DPICH format as described in the F-DPICH section of the physical layer protocol. If the first bit is equal to '0', F-DPICH format 0 is used. If the first bit is equal to '1', F-DPICH format 1 is used. The remaining four bits index the spectral efficiency, maximum number of transmissions, and the modulation format to be used for each transmission of a data packet. The indexing is described in Table 93.

The modulation format is specified by the number of bits in each modulation symbol, which is denoted by modulation order. Modulation orders of 2, 3, and 4 correspond to QPSK, 8PSK, and 16QAM modulations, respectively. The size of the MAC packet that is provided to the Physical Layer is a function of the packet format as well as the set of hop-ports that are assigned to the data packet (to be transmitted on the R-DCH Physical Layer channel).

**Table 93 RL packet formats**

Packet format index	Spectral efficiency on 1 <sup>st</sup> transmission	Max number of transmissions	Modulation order for each transmission					
			1	2	3	4	5	6+
0	0.25	6	2	2	2	2	2	2
1	0.50	6	2	2	2	2	2	2
2	1.0	6	2	2	2	2	2	2
3	1.5	6	3	2	2	2	2	2
4	2.0	6	3	3	2	2	2	2
5	2.67	6	4	4	3	3	3	3
6	4.0	6	4	4	3	3	3	3
7	6.0	6	4	4	4	3	3	3
8	8.0	6	4	4	4	4	4	3
9	4.0	6	6	6	4	4	4	4
10	5.0	6	6	6	4	4	4	4
11	6.0	6	6	6	4	4	4	4
12	7.0	6	6	6	4	4	4	4
13	8.0	6	6	6	6	4	4	4
14	9.0	6	6	6	6	4	4	4



## 7.7.7 Header and trailer and formats

The access terminal shall formulate a packet for transmission over the Reverse Traffic Channel using the following header and trailer:

### 7.7.7.1 Header

Field	Length (bits)
UATIInfoIncluded	1
IsSecure	1
KeyChange	1
InBandControlIncluded	1
Reserved	4

The following six fields shall be included if UATIInfoIncluded is '1'

SessionConfigurationToken	16
ATIdentifierType	1
ATIdentifier	128
ConnectCount	12
AccessReason	2
Reserved	1

The following field shall be included if InBandControlIncluded is '1'

RLInBandControl	0 or N x 8
-----------------	------------

6	UATIInfoIncluded	Used to signal the existence of access terminal fields in the header. These include SessionConfigurationToken, ATIdentifierType, and ATIdentifier. The access terminal shall set this field to '1' if these fields are present. Otherwise, the access terminal shall set this field to '0'. The access terminal shall set UATIInfoIncluded to '1' for the first packet sent on the R-DCH after the receipt of an AccessGrant.
7		
8		
9		
10		
11		
12	IsSecure	The access terminal shall set this field to '1' if the packet is secured by the Authentication and Encryption protocols. The access terminal shall set this field to '0' otherwise.
13		
14		
15	KeyChange	This field shall be set by the Security Sublayer at the transmitter and communicated to the Security Sublayer along with the payload at the receiver.
16		
17		
18	InBandControlIncluded	Used to signal the existence of in-band control bits. The access network shall set this field to '1' if the InBandControl field is present. Otherwise, the access network shall set this field to '0'.
19		
20		
21		
22	Reserved	This field shall be set to zero. The receiver shall ignore this field.

1	<b>SessionConfigurationToken</b>	
2		This field shall be set to the SessionConfiguration public data of the Session
3		Configuration Protocol.
4	<b>ATIdentifierType</b>	Used to signal the type of identifier to be signaled in the ATIdentifier field.
5		The access terminal shall set this field to ‘1’ if the ATIdentifier field contains
6		a 128-bit UATI. Otherwise, the access terminal shall set this field to ‘0’,
7		indicating that ATIdentifier field contains the SessionSeed.
8	<b>ATIdentifier</b>	Used to signal the access terminal identifier record. If the ATIdentifierType
9		field is 1 then this field shall be set to TransmitUATI public data of the
10		Address Management Protocol. Otherwise, the lower bits of this field shall
11		be set to the SessionSeed public data of the SessionManagementProtocol and
12		the upper bits shall be set to zero.
13	<b>ConnectCount</b>	This field shall be set to the ConnectCount field that is public data of the Idle
14		State Protocol.
15	<b>AccessReason</b>	This field shall be set to ‘00’ if the access attempt was made in response to a
16		page received by the access terminal. This field shall be set to ‘01’ otherwise.
17	<b>Reserved</b>	The sender shall set this field to ‘0000’. The receiver shall ignore this field.
18	<b>RLInBandControl</b>	The RL in-band bits are used to transmit power control, buffer level, and
19		packet latency information to the access network RL scheduler. The
20		RLInBandControl blocks are given in 7.7.7.2.1.

### 21 7.7.7.2 Trailer

22 This protocol does not specify a trailer.

#### 23 7.7.7.2.1 RLInBandControl

24 There are three possible RLInBandControl blocks. Multiple RLInBandControl blocks may be sent in  
25 one MAC Header.

##### 26 7.7.7.2.1.1 InBandPowerControl block

27 The first InBandPowerControl block, shown below, transmits RL power control information. The  
28 access terminal may include more than one InBandPowerControl block in a MAC packet. The  $j^{th}$   
29 InBandPowerControl block shall refer to the  $j^{th}$  carrier in the ActiveCarrier public data of the SSCH  
30 MAC Protocol (The ActiveCarrier public data shall be assumed to be sorted according to increasing  
31 carrier values).

32

Field	Length (bits)
ContinuationBit	1
BlockFormat	1
MaxSubCarriers	3
RDCHGainIndex	3

- 1 ContinuationBit The access terminal shall set the continuation bit to 0 if this is the last  
2 RLIInBandControl block in the MAC header. Otherwise, the access terminal  
3 shall set the continuation bit to 0.
- 4 BlockFormat The access terminal shall indicate an RL power control block by setting the  
5 format indicator bit to 0.
- 6 MaxSubCarriers The access terminal shall specify MaxSubCarriers, the maximum number of  
7 subcarriers that the access terminal can transmit at RDCHGain based on  
8 transmit power constraints with the following values:  
9

Maximum Number of Supportable Subcarriers at RDCHGain	MaxSubCarriers
16	000
32	001
64	010
128	011
256	100
512	101
1024	110
2048	111

- 10 RDCHGainIndex The access terminal shall set RDCHGainIndex to a 3-bit value between 0 and  
11 7 as a linear interpolation between RDCHGainMin and RDCHGainMax as  
12  $RDCHGainIndex = \text{floor}[(7.5 / (RDCHGainMax - RDCHGainMin)) * (RDCHGain - RDCHGainMin)]$ .  
13

#### 14 7.7.7.2.1.2 InBandBufferLevel block

- 15 The second RLIInBandControl block, shown below, transmits RL buffer level information. This block  
16 is used to provide the scheduler with a more accurate buffer level than the R-REQCH, as well as  
17 providing an in-band request channel. Multiple InBandBufferLevel blocks may be sent in one MAC  
18 packet for different QoS flows:  
19

Field	Length (bits)
ContinuationBit	1
BlockFormat	2
BufferLevel	3
QoS	2

- 20 ContinuationBit The access terminal shall set the continuation bit to 0 if this is the last  
21 RLIInBandControl block in the MAC header. Otherwise, the access terminal  
22 shall set the continuation bit to 1.
- 23 BlockFormat The access terminal shall indicate an RL buffer level block by setting the  
24 format indicator bits to 10.

1 BufferLevel The access terminal shall specify the RLP QoS flow to which the buffer level  
 2 corresponds. The buffer level is specified as follows:  
 3

RLP Buffer Level (Bytes)	BufferLevel (3bits)
$X = 0$	000
$0 < X < 50$	001
$50 \leq X < 400$	010
$400 \leq X < 1000$	011
$1000 \leq X < 2000$	100
$2000 \leq X < 3000$	101
$3000 \leq X < 9000$	110
$9000 \leq X$	111

4 QoS The access terminal shall set the QoS to indicate one of the four negotiated  
 5 QoS streams.

#### 6 7.7.7.2.1.3 InBandLatencyInfo block

7 The third RLPInBandControl block, shown in the following table, is used to transmit packet latency.  
 8 Multiple InBandLatencyInfo blocks may be sent in one MAC packet for different QoS flows.  
 9

Field	Length (bits)
ContinuationBit	1
BlockFormat	2
LatencyLevel	3
QoS	2

10 ContinuationBit The access terminal shall set the continuation bit to 0 if this is the last  
 11 RLPInBandControl block in the MAC header. Otherwise, the access terminal  
 12 shall set the continuation bit to 1.

13 BlockFormat The access terminal shall indicate an RL latency level block by setting the  
 14 format indicator bits to 11.

15 LatencyLevel The latency level is the largest latency of any packet in the specified QoS  
 16 flow not including the bits sent in the same MAC packet as the  
 17 RLPInBandBits. The latency is specified in the number of RL-PHY Frames  
 18 for which the packet has been waiting at the time the RLPInBandBit packet  
 19 starts transmission. The LatencyLevel value is given by the following lookup  
 20 table.  
 21

Latency of RLP head of line packet (PHY Frames)	LatencyLevel (3 bits)
$X < 4$	000
$4 \leq X < 10$	001
$10 \leq X < 20$	010
$20 \leq X < 40$	011
$40 \leq X < 80$	100
$80 \leq X < 150$	101
$150 \leq X < 300$	110
$300 \leq X$	111

1 QoS                      The access terminal shall specify the QoS flow to which the latency  
2 corresponds.

### 3 7.7.8 Message formats

4 The protocol uses the AttributeUpdateRequest, AttributeUpdateAccept, and AttributeUpdateReject  
5 messages of the Generic Attribute Update Protocol in 10.9 to update configurable attributes. In  
6 addition, this protocol defines the following messages.

#### 7 7.7.8.1 OSIReportRequest

8 The access network sends a OSIReportRequest message to request the access terminal to send a  
9 OSIReport message.

10

Field	Length (bits)
MessageID	8
RequestedHistory	8

11 MessageID              The access network shall set this field to 0x00.

12 RequestedHistory      The access network shall set this field to indicate the number of  
13 measurements the access terminal should include in the OSIReport message.

14

<b>Channels</b>	FTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Unicast	<b>Security</b>	Required

### 7.7.8.2 OSIReport

The access terminal sends the PilotReport message to notify the access network of OSI values received from other sectors in the past.

Field	Length (bits)
MessageID	8
SuperframeReferenceNumber	34
NumPilots	5
NumPilots occurrences of the following record{	
PilotPN	12
CarrierID	2
NumOSIValues	8
NumOSIValues instances of the following two fields(	
OSIValue	2
ChanDiff	8
NumFastOSIValues	8
NumFastOSIValues instances of the following two fields(	
FastOSIValue	2
ChanDiff	8
}}	
Reserved	Variable

**MessageID** The access terminal shall set this field to 0x01.

**SuperframeReferenceNumber**  
The access terminal shall set this field to the superframe number when the latest OSIValue contained in the message was received.

**NumPilots** The access terminal shall set this field to the number of pilots that follow this field in the message. The access terminal should include all pilots that have been in the Active Set or the OSIMonitorSet over the reported period.

**PilotPN** The access terminal shall set this field to the PilotPN to which the following OSIValues correspond.

**CarrierID** The access terminal shall set this field to the CarrierID of the carrier to which the following OSIValues correspond.

**NumOSIValues** The access terminal sets this field to the number of OSIValues included for this pilot.

**OSIValue** The  $i^{\text{th}}$  occurrence of this field for a given PilotPN refers to the OSI value received from this PilotPN in superframe number SuperframeReferenceNumber- $i+1$ . The OSIValue '11' shall correspond to an erased or unavailable OSIValue.

1	ChanDiff	The access terminal shall set this field to the ChanDiff for the given PilotPN in superframe number SuperframeReferenceNumber-i+1. This field shall be in a two's complement format in units of 1 dB (takes values between -128 and +127 dB). The ChanDiff field is defined in 7.7.6.4.1.7.
2		
3		
4		
5	NumFastOSIValues	The access terminal sets this field to the number of FastOSIValues included for this pilot.
6		
7	FastOSIValue	The $i^{\text{th}}$ occurrence of this field for a given PilotPN refers to the Fast OSI value received from this PilotPN in FL PHY Frame frame number FirstFrameNumber-i+1, where FirstFrameNumber is the frame number of the first FL PHY frame of superframe number SuperframeReferenceNumber. The OSIValue '11' shall correspond to an erased or unavailable OSIValue.
8		
9		
10		
11		
12	ChanDiff	The access terminal shall set this field to the ChanDiff for the given PilotPN in FL PHY Frame frame number FirstFrameNumber-i+1, where FirstFrameNumber is the frame number of the first FL PHY frame of superframe number SuperframeReferenceNumber. This field shall be in a two's complement format in units of 1 dB (takes values between -128 and +127 dB). The ChanDiff field is defined in 7.7.6.4.1.7.
13		
14		
15		
16		
17		
18	Reserved	The number of bits in this field is equal to the number needed to make the message length an integer number of octets. The sender shall set this field to zero. The receiver shall ignore this field.
19		
20		
21		

<b>Channels</b>	RTC	<b>SLP</b>	Best Effort
<b>Addressing</b>	Unicast	<b>Security</b>	Required

22

## 23 7.7.9 Interface to other protocols

### 24 7.7.9.1.1 Commands

25 This protocol issues the following commands:

- 26     ■ *ReverseControlChannelMAC.ChangeFLSS*
- 27     ■ *ReverseControlChannelMAC.ChangeRLSS*

### 28 7.7.9.1.2 Indications

29 This protocol registers to receive the following indications:

- 30     ■ *ConnectedState.TunedAway*
- 31     ■ *ReverseControlChannelMAC.RLSSChanged*

## 7.7.10 Configuration attributes

The following complex attributes and default values are defined (see 10.3 for attribute record definition).

Unless specified otherwise, the access terminal and the access network shall use the Generic Attribute Update Protocol in 10.9 to update configurable attributes belonging to the Default Reverse Control Channel MAC Protocol.

### 7.7.10.1 PowerParameters attribute

Field	Length (bits)	Default Value
Length	8	N/A
AttributeID	8	N/A
UpDecisionThresholdMin	8	3
DownDecisionThresholdMin	8	7
ChanDiffMax	8	50
ChanDiffMin	8	0
UpDecisionValue	8	160
DownDecisionValue	8	80
DataGainStepUp	8	8
DataGainStepDown	8	8
RDCHGainAdjustmentThreshold	8	136
OSIMonitorSetSize	8	2
OSIMonitorThreshold	8	17
OSI2SequenceMax	3	4
FastOSIChanDiffThreshold	8	40

**Length** Length of the complex attribute in octets. The access network shall set this field to the length of the complex attribute excluding the Length field.

**AttributeID** This field shall be set to 0x00.

**UpDecisionThresholdMin** This field is used to control parameters internal to the RTC MAC power control algorithm. The value of this field is  $\min(1, 2^{-7} * n)$ .

**DownDecisionThresholdMin** This field is used to control parameters internal to the RTC MAC power control algorithm. The value of this parameter is  $\min(1, 2^{-7} * n)$ .

**ChanDiffMax** This field determines maximum value of ChanDiff used in determining RDCHGain. The value of this parameter is  $2 * n$  dB.

**ChanDiffMin** This field is determines the minimum value of ChanDiff used in determining RDCHGain. The value of this parameter is  $n$  dB.



1	UpDecisionValue	This field is used to control parameters internal to the RTC MAC power control algorithm. The value of this parameter is $(n-128) * 2^{-5}$ .
2		
3	DownDecisionValue	This field is used to control parameters internal to the RTC MAC power control algorithm. The value of this parameter is $(n-128) * 2^{-5}$ .
4		
5	DataGainStepUp	This field is used to control parameters internal to the RTC MAC power control algorithm. The value of this parameter is $n * 2^{-5}$ dB.
6		
7	DataGainStepDown	This field is used to control parameters internal to the RTC MAC power control algorithm. The value of this parameter is $n * 2^{-5}$ dB.
8		
9	RDCHGainAdjustmentThreshold	
10		This field is used to control parameters internal to the RTC MAC power control algorithm. The value of this parameter is $(n-128) * 2^{-5}$ dB.
11		
12	OSIMonitorSetSize	This field is set to the size of the OSI Monitor Set.
13	OSIMonitorThreshold	This field determines a threshold such that only sectors with pilot signal to interference ratios above the threshold shall be added to the OSI monitor set. The value of this parameter is n dB.
14		
15		
16	OSI2SequenceMax	This field is set to control parameters internal to the RTC MAC power control algorithm.
17		
18	FastOSIChanDiffThreshold	
19		This field determines a threshold such that only active set members with a ChanDiff value smaller than this threshold shall be added to the OSI monitor set when the RDCHGain is being adjusted based on the Fast OSI values. The value of this parameter is $n*0.125$ dB.
20		
21		
22		

### 7.7.11 Protocol numeric constants

Constant	Meaning	Value
$N_{RTCMPType}$	Type field for this protocol	
$N_{RTCMPDefault}$	Subtype field for this protocol	0x0000
$T_{RTCSupervision}$	Supervision timer	2 s

### 7.7.12 Session state information

The Session State Information record (see 10.10) consists of parameter records.

The parameter records for this protocol consist of the configuration attributes of this protocol.

## 1 **8 Physical Layer**

### 2 **8.1 Default Physical Layer Protocol**

#### 3 **8.1.1 Overview**

4 This chapter contains the specification for the Default (Subtype 0) Physical Layer Protocol.

#### 5 **8.1.2 Primitives**

##### 6 **8.1.2.1 Commands**

7 This protocol does not define any commands.

##### 8 **8.1.2.2 Return indications**

9 This protocol does not return any indications.

#### 10 **8.1.3 Public data**

##### 11 **8.1.3.1 Static public data**

12 This protocol does not define any static public data.

##### 13 **8.1.3.2 Dynamic public data**

- 14 ■ Subtype for this protocol
- 15 ■ MultiCarrierOn mode

#### 16 **8.1.4 Protocol data unit**

17 The transmission unit of this protocol is the Physical Layer packet. Each Physical Layer packet  
18 contains one MAC Layer packet.

#### 19 **8.1.5 Protocol initialization and swap procedures**

##### 20 **8.1.5.1 Protocol initialization**

21 Upon creation, the instance of this protocol in the access terminal and access network shall perform  
22 the following:

- 23 ■ The value of the attributes for this protocol instance shall be set to the default values  
24 specified for each attribute.
- 25 ■ This protocol shall determine the values of the following parameters
  - 26 □ Duplexing mode (FDD or TDD)
  - 27 □ Synchronization mode (Semi-synchronous or Asynchronous)
  - 28 □ Multi-carrier mode (MultiCarrierOn or MultiCarrierOff)

## 8.1.6 Protocol swap

- This protocol defines an empty swap procedure.

## 8.1.7 Procedures

Procedures for the protocol are described in chapter 9.

## 8.1.8 Message formats

### 8.1.8.1 TimingCorrection

The access network shall send the timing correction message to correct the reverse link timing of the access terminal.

Field	Length (bits)
MessageID	8
NumSectors	2
NumSectors instances of the following fields	
PilotPN	12
TimingCorrection	16

MessageID	The access network shall set this field to 0x02.
NumSectors	The access network shall set this field to the number of sector records in the message.
PilotPN	The access network shall set this field to the PilotPN of the sector.
TimingCorrection	The access network shall set this field to the timing correction on the sector in twos complement format in units of 1/8 chips. A positive value shall advance the timing, and a negative value shall retard the timing.

## 8.1.9 Interface to other protocols

### 8.1.9.1 Commands

This protocol does not issue any commands.

### 8.1.9.2 Indications

This protocol does not register to receive any indications.

## 8.1.10 Configuration attributes

This protocol does not define any configuration attributes.

### 8.1.11 Protocol numeric constants and parameters

Constant	Meaning	Value
$N_{\text{PHYType}}$	Type field for this protocol	Table 9
$N_{\text{PHYDefault}}$	Subtype field for this protocol	0x0000
$N_{\text{FFT}}$	Number of subcarriers in an OFDM symbol	512, 1024, or 2048
$T_{\text{CHIP}}$	Basic unit of time for generating the OFDM waveform	Defined in Table 94 as a function of $N_{\text{FFT}}$
$N_{\text{CARRIER\_SIZE}}$	Number of subcarriers in one carrier	512 in MultiCarrierOn mode $N_{\text{FFT}}$ in MultiCarrierOff mode
$N_{\text{CARRIERS}}$	Number of carriers	$N_{\text{FFT}}/N_{\text{CARRIER\_SIZE}}$ (= 1 in MultiCarrierOff mode)
$T_{\text{CP,PR}}$	Cyclic prefix duration for the superframe preamble	$N_{\text{FFT}}T_{\text{CHIP}}/4$
$N_{\text{GUARD,PR}}$	Number of guard subcarriers in the superframe preamble	Any multiple of $N_{\text{CARRIER\_SIZE}}/8$ , ranging from $N_{\text{CARRIER\_SIZE}}/8$ through $7N_{\text{CARRIER\_SIZE}}/8$ . This field shall be set by the access network, and is determined by the access terminal.
$T_{\text{WGI}}$	Duration of windowing guard interval	$N_{\text{FFT}}T_{\text{CHIP}}/32$
$N_{\text{PREAMBLE}}$	Number of OFDM symbols in the superframe preamble	8
$N_{\text{FRAME,F}}$	Number of OFDM symbols in a forward link PHY Frame	8
$N_{\text{FRAME,R}}$	Number of OFDM symbols in a reverse link PHY Frame	8
$N_{\text{BLOCK}}$	Number of subcarriers in a tile.	16 in BlockHopping mode (FL) 1 in SymbolRateHopping mode (FL) 16 for the RL
$T_{\text{G,TDD,F}}$	Guard time between a forward link PHY Frame and the subsequent reverse link PHY frames	$3N_{\text{FFT}}T_{\text{CHIP}}/4$
$T_{\text{G,TDD,R}}$	Guard time between a reverse link PHY Frame and the subsequent forward link PHY frames	$5N_{\text{FFT}}T_{\text{CHIP}}/32$
$N_{\text{pBCH0\_Period}}$	Number of superframes over which F-pBCH0 is encoded	16
$N_{\text{MaxErasureHopPorts,F}}$	Maximum number of hop-ports to be used for transmitting a single erasure sequence on the forward link	16
$N_{\text{MaxErasureHopPorts,R}}$	Maximum number of hop-ports to be used for transmitting a single erasure sequence on the reverse link	16
$N_{\text{CRC,pBCH}}$	Number of CRC bits to be used for pBCH0 and pBCH1 packets.	12

Constant	Meaning	Value
$N_{\text{CRC,SSCH}}$	Number of CRC bits to be used for SSCH packets.	16
$N_{\text{CRC,Data}}$	Number of CRC bits to be used for F-DCH and R-DCH packets.	24
$N_{\text{FastOSI}}$	Number of modulation symbols in the Fast OSI segment.	8
$N_{\text{BLOCK, R-ACKCH}}$	Number of subcarriers in a R-ACKCH tile.	8
$N_{\text{R-ACKCH-SUBTILE-DURATION}}$	Number of OFDM symbols in a R-ACKCH subtile	2

1

2

**Table 94 Chip duration as a function of  $N_{\text{FFT}}$** 

$N_{\text{FFT}}$	$T_{\text{CHIP}}$ in $\mu\text{s}$
512	1/4.9152
1024	1/9.8304
2048	1/19.6608

3

**8.1.12 Session state information**

4

5 This protocol does not define any parameter record to be included in a Session State Information  
6 record.

## 9 Default Physical Layer

### 9.1 Physical layer modes

The physical layer specification consists of two different duplexing modes, two different forward link hopping modes, two different synchronization modes and two different multi-carrier modes. The possible duplexing modes are Time Division Duplexing (TDD) and Frequency Division Duplexing (FDD). The different forward link hopping modes are SymbolRateHopping and BlockHopping. The forward link hopping mode to be used is given by the BlockHoppingEnabled field, which is part of the public data of the Overhead Messages Protocol. The possible synchronization modes are SemiSynchronous and Asynchronous. The possible multi-carrier modes are MultiCarrierOn and MultiCarrierOff.

Parts of the physical layer specification are described separately for different duplexing modes, different forward-link hopping modes, synchronization modes and/or multi-carrier modes. Except in these cases, the specification shall apply to all values of the corresponding mode.

The TDD mode has two associated variables called  $N_{FL\_BURST}$  and  $N_{RL\_BURST}$  which determine the time partitioning between forward and reverse links. The values of these variables are given by the  $N\_FLBurst$  and  $N\_RLBurst$  parameters respectively, which are part of the public data of the Overhead Messages Protocol.

In the MultiCarrierOn mode, the total transmission bandwidth is divided into a multiplicity of carriers, as specified in 9.3.2.2. The number of carriers is given by  $N_{CARRIERS}$ . For convenience of exposition, the same terminology is sometimes used in the MultiCarrierOff mode as well. In this case, the total transmission bandwidth is divided into  $N_{CARRIERS} = 1$  carriers.

### 9.2 Encoding and modulation

This section describes the core encoding and modulation procedures, shown in Figure 82, that shall be used for constructing several of the physical layer channels. The procedures described in this section, namely packet-splitting, CRC insertion, encoding, channel interleaving, sequence repetition, scrambling, and modulation, together constitute a method for converting a k-bit packet (generated by an appropriate MAC protocol) into sequences of modulation symbols (one sequence per sub-packet), for any value of k that satisfies at least one of the following two conditions: (1) k is less than MaxPHYSubPacketSize (2) k is a multiple of 8. Here, MaxPHYSubPacketSize is public data of the Active Set Management Protocol.

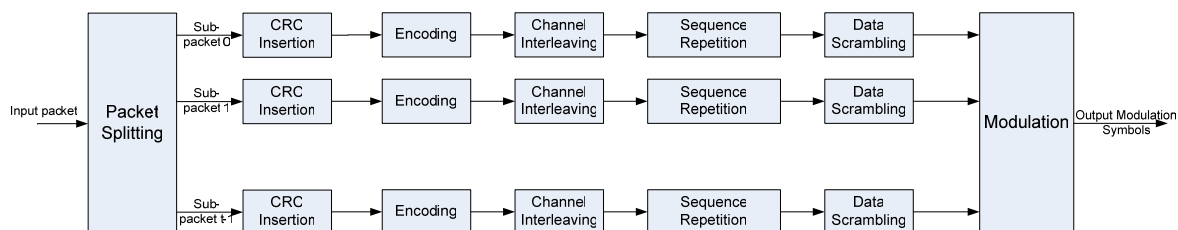


Figure 82 Encoding and modulation structure

## 9.2.1 Packet-splitting and CRC insertion

If the packet size  $k$  is larger than  $\text{MaxPHYSubPacketSize}$ , the packet shall be split into  $t$  sub-packets, indexed from 0 to  $t-1$ , where  $t = \lceil k / \text{MaxPHYSubPacketSize} \rceil$ . Here,  $\text{MaxPHYSubPacketSize}$  is public data of the Active Set Management Protocol. Let  $k_i$  denote the size in bits of sub-packet  $i$ . Define integers  $t_0 = (k/8) \bmod t$  and  $t_1 = t - t_0$ . Define two other integers  $b_0 = 8 \lceil k / (8t) \rceil$  and  $b_1 = 8 \lfloor k / (8t) \rfloor$ . The first  $t_0$  sub-packets shall consist of  $b_0$  bits, i.e.,  $k_0 = k_1 = \dots = k_{t_0-1} = b_0$ , while the last  $t_1$  sub-packets shall consist of  $b_1$  bits, i.e.,  $k_{t_0} = k_{t_0+1} = \dots = k_{t-1} = b_1$ . The bits are distributed to the different sub-packets in order, i.e., bits 0 through  $k_0-1$  form sub-packet 0, bits  $k_0$  through  $k_0+k_1-1$  form sub-packet 1, etc.

Each of the sub-packets so generated shall then be appended by a CRC as described in chapter 10. The number of CRC bits, denoted by  $N_{\text{CRC}}$ , to be appended is variable and is specified separately in the description of each physical layer channel using this procedure. The sizes of the resulting sub-packets at the end of this procedure are therefore given by  $k_i' = k_i + N_{\text{CRC}}$ , for  $i$  ranging from 0 through  $t-1$ .

At the receiver, a packet shall be declared to be in error if any of the constituent sub-packets are in error.

The operations described in Sections 9.2.2, 9.2.3, 9.2.4, and 9.2.5, namely encoding, channel interleaving, sequence repetition and scrambling operate independently on each of the sub-packets and are described only for the case  $t=1$ . The operation described in 9.2.6, namely modulation, operates jointly on all sub-packets and is described for all values of  $t$ .

## 9.2.2 Core encoders

The air-link shall support two basic encoding structures, namely a rate 1/5 parallel turbo code and a rate 1/3 convolutional code. The rate 1/5 turbo code shall be used for values of  $k$  larger than 128, while the rate 1/3 convolutional code shall be used for values of  $k$  less than or equal to 128.

### 9.2.2.1 Rate 1/3 convolutional encoding

The core rate-1/3 code is a non-systematic non-recursive convolutional code. The outputs of the convolutional code are punctured or repeated to achieve the desired number of convolutional encoder output bits.

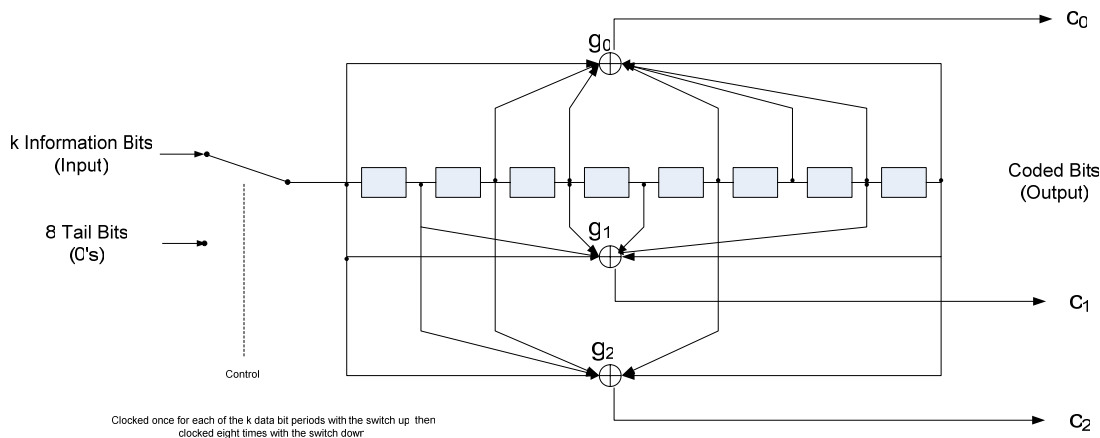
The transfer function for the convolutional code shall be:

$$G(D) = [g_0(D) \ g_1(D) \ g_2(D) ]$$

where  $g_0(D) = 1 + D^2 + D^3 + D^5 + D^6 + D^7 + D^8$ ,  $g_1(D) = 1 + D + D^3 + D^4 + D^7 + D^8$ , and  $g_2(D) = 1 + D + D^2 + D^5 + D^8$ , where  $D$  represents the delay operator.

The sequence of information bits shall be appended with a tail of eight 0s and input to the convolutional encoder. This code generates three code bits for each bit input to the encoder. Thus a total of  $3(k+8) = 3k+24$  coded bits are generated for a  $k$ -bit input packet. These code bits shall be output so that the code bit ( $c_0$ ) encoded with generator function  $g_0$  shall be output first, the code bit ( $c_1$ ) encoded with generator function  $g_1$  shall be output second, and the code bit ( $c_2$ ) encoded with

1 generator function  $g_2$  shall be output last. The state of the convolutional encoder, upon initialization,  
 2 shall be the all-zero state. The first code bit output after initialization shall be a code bit encoded with  
 3 generator function  $g_0$ . The encoder for this code is illustrated in Figure 83.



4  
 5 **Figure 83 Rate 1/3 convolutional encoder**

6 **9.2.2.2 Rate 1/5 turbo encoding**

7 The core turbo encoder is a rate 1/5 code that employs two systematic, recursive, convolutional  
 8 encoders connected in parallel, with an interleaver—the turbo interleaver—preceding the second  
 9 recursive convolutional encoder. The two recursive convolutional codes are called the constituent  
 10 codes of the turbo code. The outputs of the constituent encoders are punctured or repeated to achieve  
 11 the desired number of turbo encoder output bits.

12 The transfer function for the constituent code shall be:

$$G(D) = \left[ 1 \quad \frac{n_0(D)}{d(D)} \quad \frac{n_1(D)}{d(D)} \right]$$

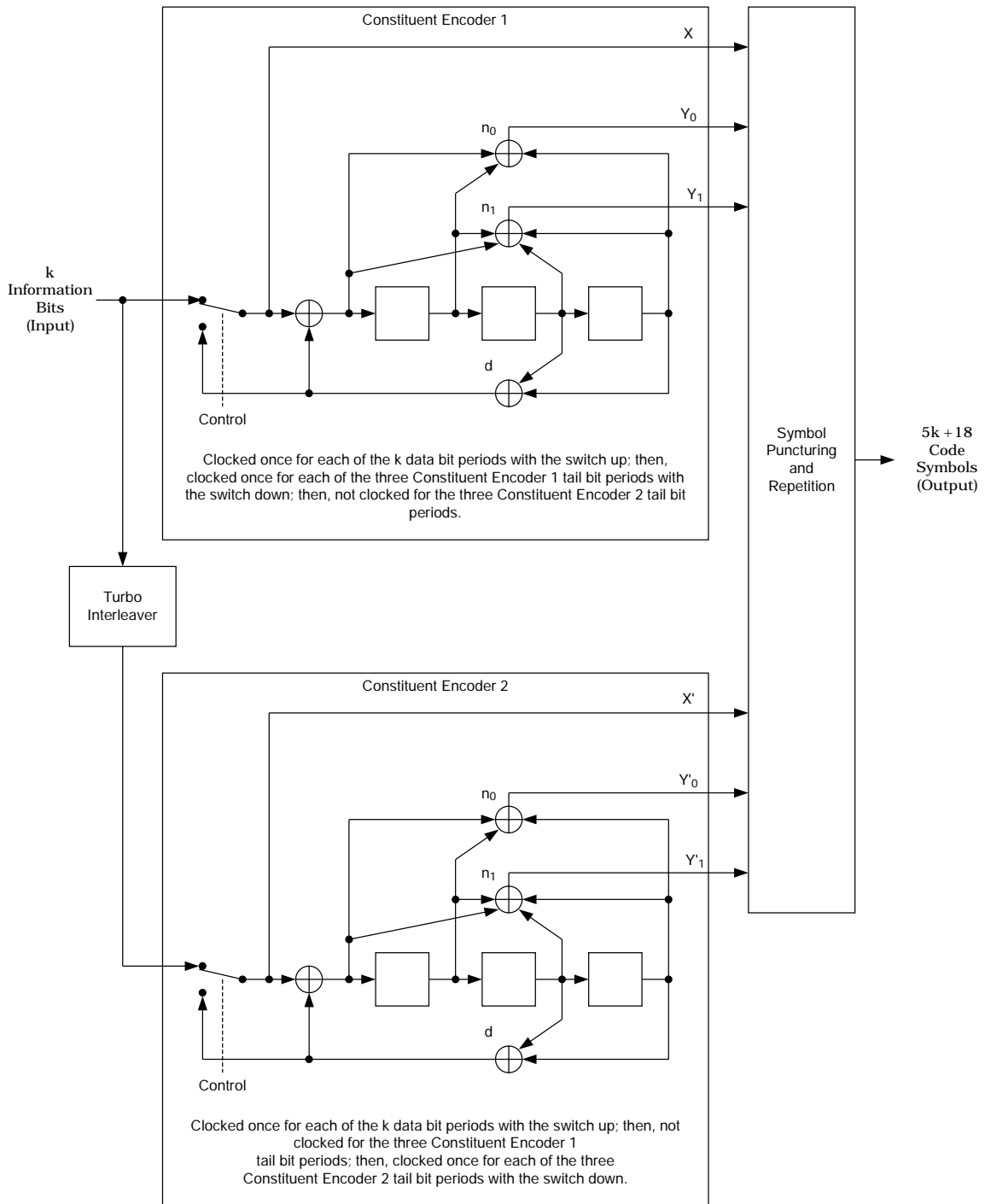
13  
 14 where  $d(D) = 1 + D^2 + D^3$ ,  $n_0(D) = 1 + D + D^3$ , and  $n_1(D) = 1 + D + D^2 + D^3$ , where  $D$  represents the  
 15 delay operator.

16 The turbo encoder shall generate an output bit sequence that is identical to the one generated by the  
 17 encoder shown in Figure 84. Initially, the states of the constituent encoder registers in this figure are  
 18 set to zero. Then, the constituent encoders are clocked with the switches in the positions noted.

19 The encoded data output bits are generated by clocking the constituent encoders  $k$  times with the  
 20 switches in the up positions, where  $k$  is the number of input bits into the turbo encoder. The  
 21 constituent encoder outputs for each bit period shall be output in the sequence  $X, Y_0, Y_1, Y'_0, Y'_1$   
 22 with the  $X$  output first. (The bit  $X$  shall not be part of the output sequence.)



1 The turbo encoder shall generate 18 tail output bits following the encoded data output bits. This tail  
2 output bit sequence shall be identical to the one generated by the encoder shown in Figure 84. The tail  
3 output bits are generated after the constituent encoders have been clocked  $k$  times with the switches in  
4 the up position. The first 9 tail output bits are generated by clocking Constituent Encoder 1 three  
5 times with its switch in the down position while Constituent Encoder 2 is not clocked. The constituent  
6 encoder outputs for each bit period shall be output in the sequence  $X, Y_0, Y_1$ , with the  $X$  output first.  
7 The last 9 tail output bits are generated by clocking Constituent Encoder 2 three times with its switch  
8 in the down position while Constituent Encoder 1 is not clocked. The constituent encoder outputs for  
9 each bit period shall be output in the sequence  $X', Y'_0, Y'_1$ , with the  $X'$  output first. The tail bit  
10 sequence ensures that both constituent encoders achieve the all-zeros state at the end of the encoding  
11 process.



1

2

**Figure 84 Turbo encoder**

### 9.2.2.2.1 Turbo Interleaving

The turbo interleaver, which is part of the turbo encoder, shall block interleave the turbo encoder input data that is fed to Constituent Encoder 2.

The turbo interleaver shall be functionally equivalent to an approach where the entire sequence of turbo interleaver input bits are written sequentially into an array at a sequence of addresses, and then the entire sequence is read out from a sequence of addresses that is defined by the procedure described in the following.

Let the sequence of input addresses be from 0 to  $k - 1$ . Then, the sequence of interleaver output addresses shall be equivalent to those generated by the procedure illustrated in Figure 85 and described in the following.<sup>41</sup>

1. Determine the turbo interleaver parameter,  $n$ , where  $n$  is the smallest integer such that  $k \leq 2^{n+5}$ .
2. Initialize an  $(n + 5)$ -bit counter  $x$  to 0.
3. Let  $x' = (\lfloor x/32 \rfloor + 1) \bmod 2^n$ .  $x'$  is generated by extracting the  $n$  most significant bits (MSBs) from the counter, adding one to form a new value, and then discarding all except the  $n$  least significant bits (LSBs) of this value.
4. Obtain the  $n$ -bit output of the table lookup defined in Table 95 with a read address equal to the five LSBs of the counter  $x$ , and call this output  $x''$ . Note that this table depends on the value of  $n$ .
5. Multiply the values  $x'$  and  $x''$  obtained in Steps 3 and 4, and discard all except the  $n$  LSBs to get a number  $y$ .  $y = x'x'' \bmod 2^n$ .
6. Bit-reverse the five LSBs of the counter  $x$  to get a five-bit number  $y'$ .
7. Form a tentative output address  $z$  that has its MSBs equal to the value  $y'$  obtained in Step 6 and its LSBs equal to the value  $y$  obtained in Step 5.
8. Accept the tentative output address  $z$  as an output address if it is less than  $k$ ; otherwise, discard it.
9. Increment the counter and repeat Steps 3 through 8 until all  $k$  interleaver output addresses are obtained.

---

<sup>41</sup> This procedure is equivalent to one where the counter values are written into a  $2^5$ -row by  $2^n$ -column array by rows, the rows are shuffled according to a bit-reversal rule, the elements within each row are permuted according to a row-specific linear congruential sequence, and tentative output addresses are read out by column. The linear congruential sequence rule is  $x(i + 1) = (x(i) + c) \bmod 2^n$ , where  $x(0) = c$  and  $c$  is a row-specific value from a table lookup.

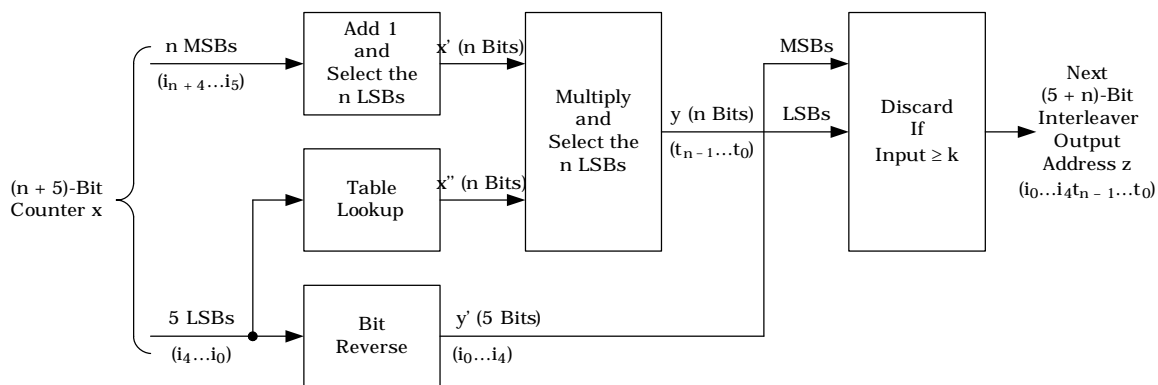


Figure 85 Turbo interleaver output address calculation procedure

Table 95 Turbo interleaver lookup table definition

Table Index	n = 2 Entries	n = 3 Entries	n = 4 Entries	n = 5 Entries	n = 6 Entries	n = 7 Entries	n = 8 Entries	n = 9 Entries
0	3	1	5	27	3	15	3	13
1	3	1	15	3	27	127	1	335
2	3	3	5	1	15	89	5	87
3	1	5	15	15	13	1	83	15
4	3	1	1	13	29	31	19	15
5	1	5	9	17	5	15	179	1
6	3	1	9	23	1	61	19	333
7	1	5	15	13	31	47	99	11
8	1	3	13	9	3	127	23	13
9	1	5	15	3	9	17	1	1
10	3	3	7	15	15	119	3	121
11	1	5	11	3	31	15	13	155
12	1	3	15	13	17	57	13	1
13	1	5	3	1	5	123	3	175
14	1	5	15	13	39	95	17	421
15	3	1	5	29	1	5	1	5
16	3	3	13	21	19	85	63	509
17	1	5	15	19	27	17	131	215
18	3	3	9	1	15	55	17	47
19	3	5	3	3	13	57	131	425
20	3	3	1	29	45	15	211	295
21	1	5	3	17	5	41	173	229
22	3	5	15	25	33	93	231	427
23	1	5	1	29	15	87	171	83
24	3	1	13	9	13	63	23	409

Table Index	n = 2 Entries	n = 3 Entries	n = 4 Entries	n = 5 Entries	n = 6 Entries	n = 7 Entries	n = 8 Entries	n = 9 Entries
25	1	5	1	13	9	15	147	387
26	3	1	9	23	15	13	243	193
27	1	5	15	13	31	15	213	57
28	3	3	11	13	17	81	189	501
29	1	5	3	1	5	57	51	313
30	1	5	15	13	15	31	15	489
31	3	3	5	13	33	69	67	391

1

## 2 9.2.3 Channel interleaving

3 The turbo or convolutional encoding shall be followed by channel interleaving, which consists of bit  
4 demultiplexing followed by bit permuting.

### 5 9.2.3.1 Bit demultiplexing

6 The output bits generated by the rate-1/3 convolutional encoder shall be reordered according to the  
7 following procedure:

- 8 1. All of the convolutional encoder output bits shall be demultiplexed into three sequences  
9 denoted  $V_0$ ,  $V_1$ ,  $V_2$ . The encoder output bits shall be sequentially distributed from the  $V_0$   
10 sequence to the  $V_2$  sequence with the first bit going to the  $V_0$  sequence, the second bit  
11 going to the  $V_1$  sequence, the third to the  $V_2$  sequence, the fourth to the  $V_0$  sequence, etc.
- 12 2. The  $V_0$ ,  $V_1$ , and  $V_2$  sequences shall be ordered according to  $V_0V_1V_2$ . That is, the  $V_0$   
13 sequence shall be first, the  $V_1$  sequence shall be second, and the  $V_2$  sequence shall be  
14 last.

15 The output bits generated by the rate-1/5 turbo encoder shall be reordered according to the following  
16 procedure:

- 17 1. All of the turbo encoder output data bits (i.e., the  $5k$  bits output in the first  $k$  clock  
18 periods) shall be demultiplexed into five sequences denoted  $U$ ,  $V_0$ ,  $V_1$ ,  $V'_0$ , and  $V'_1$ . The  
19 encoder output bits shall be sequentially distributed from the  $U$  sequence to the  $V'_1$   
20 sequence with the first encoder output bit going to the  $U$  sequence, the second to the  $V_0$   
21 sequence, the third to the  $V_1$  sequence, the fourth to the  $V'_0$  sequence, the fifth to the  $V'_1$   
22 sequence, the sixth to the  $U$  sequence, etc.
- 23 2. The 18 tail bits numbered 0 through 17 (i.e., the 18 bits generated during the last six  
24 clock periods) shall be distributed as follows: Tail bits numbered 0, 3, 6, 9, 12, and 15  
25 shall go to the  $U$  sequence, the tail bits numbered 1, 4, and 7 shall go to the  $V_0$  sequence,  
26 the tail bits numbered 2, 5, and 8 shall go to the  $V_1$  sequence, the tail bits numbered 10,  
27 13, and 16 shall go to the  $V'_0$  sequence, and the tail bits numbered 11, 14, and 17 shall go  
28 to the  $V'_1$  sequence. In other words, the tail bits of each non-systematic stream are  
29 allocated to the corresponding sequence.

- 1           3. The  $U$ ,  $V_0$ ,  $V_1$ ,  $V'_0$ , and  $V'_1$  sequences shall be ordered according to  $UV_0V'_0V_1V'_1$ . That  
2           is, the  $U$  sequence shall be first and the  $V'_1$  sequence shall be last.

### 3    **9.2.3.2 Bit permuting**

4    The demultiplexed bits shall be permuted in three separate interleaver blocks with rate-1/5 coding and  
5    in one block with rate-1/3 coding. For the rate 1/5 turbo code, the permuter input blocks shall consist  
6    of the  $U$  sequence, the  $V_0$  sequence followed by the  $V'_0$  sequence (denoted as  $V_0/V'_0$ ), and the  $V_1$   
7    sequence followed by the  $V'_1$  sequence (denoted as  $V_1/V'_1$ ). For the rate-1/3 convolutional code, the  
8    permuter input block shall consist of the  $V_0$  sequence followed by the  $V_1$  sequence followed by the  
9     $V_2$  sequence (denoted as  $V_0/V_1/V_2$ ). A Pruned Bit-Reversal Interleaver (PBRI) shall be used for  
10   permuted each of the blocks.

11   The PBRI shall be functionally equivalent to an approach where the entire sequence of input bits in  
12   the block are written sequentially into an array at a sequence of addresses, and then the entire  
13   sequence is read out from a sequence of addresses that is defined by the procedure described in the  
14   following.

15   Let the number of bits in the input block be  $k_b$ , and let the sequence of input addresses be from 0 to  $k_b$   
16   – 1. Then, the sequence of interleaver output addresses shall be equivalent to those generated by the  
17   procedure described in the following.

- 18           1. Determine the PBRI parameter,  $n$ , where  $n$  is the smallest integer such that  $k_b \leq 2^n$ .
- 19           2. Initialize a counter  $j$  to 0.
- 20           3. Form a tentative output address that is equal to the bit-reversed value of  $j$ , using an  $n$ -bit  
21           binary representation. For example, if  $n = 4$  and  $j = 3$ , then the bit reversed value of  $j$  is  
22           12.
- 23           4. Accept the tentative output address as an output address if it is less than  $k_b$ ; otherwise,  
24           discard it.
- 25           5. Increment the counter  $j$  and repeat Steps 3 through 5 until all  $k_b$  interleaver output  
26           addresses are obtained.

27   With rate-1/5 turbo coding, the interleaver output sequence shall be the interleaved  $U$  sequence of bits  
28   followed by the interleaved  $V_0/V'_0$  sequence of bits followed by the interleaved  $V_1/V'_1$  sequence of  
29   bits. With rate-1/3 convolutional coding, the interleaver output sequence shall be the interleaved  
30    $V_0/V_1/V_2$  sequence of bits.

### 31   **9.2.4 Sequence repetition**

32   Let  $x_0, x_1, \dots, x_{n-1}$  be the sequence of bits at the output of the channel interleaver. This sequence of  
33   bits shall be repeated to create a sequence of output bits  $y_0, y_1, \dots$ . The output buffer  $y_0, y_1, \dots$  is read  
34   sequentially by the modulator, described in 9.2.6, until the required number of modulation symbols  
35   has been generated. The number of repetitions of the interleaver output sequence  $x_i$  shall be such that  
36   the modulator does not reach the end of the output buffer while generating modulation symbols.

1 The output sequence  $y_0, y_1, \dots$  shall be equivalent to an infinite sequence described by the formula  $y_i$   
 2  $= x_{i \bmod n}$ , where  $n$  is the number of bits at the output of the channel interleaver.<sup>42</sup>

### 3 9.2.5 Data scrambling

4 The sequence  $y_0, y_1, \dots$  at the output of the sequence repetition stage shall be data-scrambled to  
 5 randomize the data prior to modulation. The scrambling sequence shall be equivalent to one generated  
 6 with a 17-tap linear feedback shift register with a generator sequence of  $h(D) = 1 + D^{14} + D^{17}$ , as  
 7 shown in Figure 86. The  $n$ 'th output  $s(n)$  of this shift register shall satisfy  $s(n) = s(n-14) \oplus s(n-17)$ . At  
 8 the start of the physical layer packet, the shift register shall be initialized to the state  
 9  $[b_{16}b_{15}b_{14}b_{13}b_{12}b_{11}b_{10}b_9b_8b_7b_6b_5b_4b_3b_2b_1b_0]$ , which is a bitwise XOR of the vectors  
 10  $[r_{10}r_9r_8r_7r_6r_5r_4r_3r_2r_1r_0 \ d_5d_4d_3d_2d_1d_0]$  and  $[t_{10}t_9t_8t_7t_6t_5t_4t_3t_2t_1t_0 \ f_5f_4f_3f_2f_1f_0]$ . Here, the  
 11  $r_{10}r_9r_8r_7r_6r_5r_4r_3r_2r_1r_0$  bits shall be the bits of a 11-bit MACID (with  $r_0$  being the LSB and  $r_{10}$  the  
 12 MSB) that will be specified in the description of each physical layer channel that uses this procedure.  
 13 If the length of the MACID is less than 11 bits, then the MACID shall be padded with 0's on the left  
 14 (i.e., in the MSBs) to generate a 11-bit MACID.

15 The  $d_5d_4d_3d_2d_1d_0$  bits shall be the 6 LSBs of a packet format index (with  $d_0$  being the LSB and  $d_5$   
 16 being the MSB), and will also be specified in the description of each physical layer channel that uses  
 17 this procedure.<sup>43</sup> If the specified packet format is less than 6 bits, then it shall be padded with 0's in  
 18 the beginning (i.e., the MSBs shall be set to 0) in order to achieve the desired length. The  
 19  $t_{10}t_9t_8t_7t_6t_5t_4t_3t_2t_1t_0$  bits shall be the 11 LSBs of the superframe index of the superframe in which the  
 20 first modulation symbol of this packet is transmitted. The  $f_5f_4f_3f_2f_1f_0$  bits shall be the PHY Frame  
 21 index (within the superframe) of the PHY Frame in which the first modulation symbol of this packet  
 22 is transmitted. An all-zeros PHY Frame index is used if packet transmission begins during the  
 23 superframe preamble.

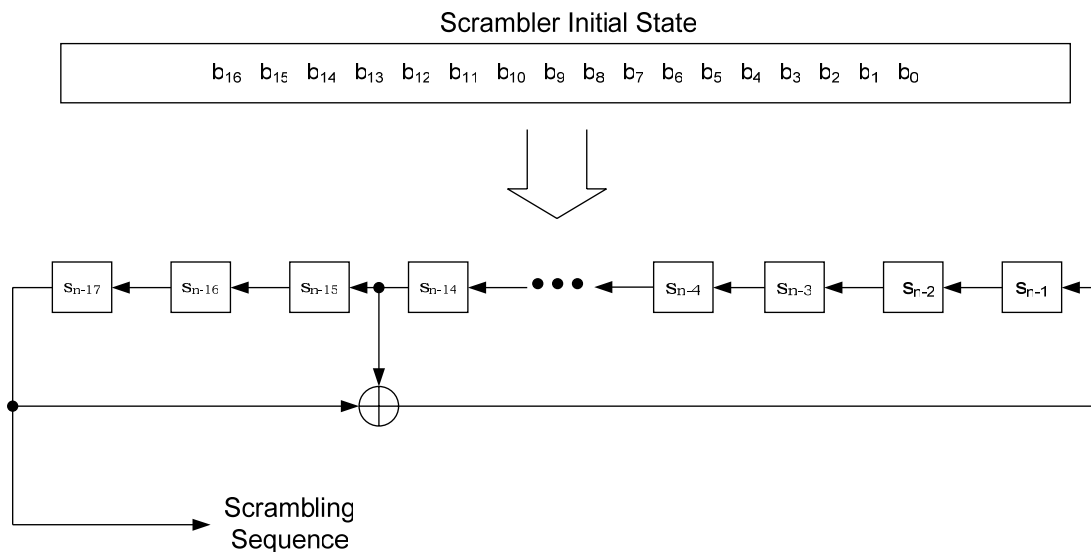
24 The first bit in the scrambling sequence shall be generated by the initial state of the shift register.  
 25 Each subsequent bit shall be generated by clocking the shift register once. Every bit at the output of  
 26 the sequence repetition stage shall be XORed with the corresponding bit of the scrambling sequence  
 27 to yield a scrambled bit.

28 The data-scrambling operation can be omitted for some physical layer channels. This shall be  
 29 specified in the description of the relevant channel.

---

<sup>42</sup> This procedure is equivalent to repeating the original bits as often as required. The modulator, described in Section 9.2.6, will read this sequence sequentially, and the number of repetitions should be such that the modulator does not reach the end of the repeated sequence before the required number of modulation symbols is generated.

<sup>43</sup> The MACID bits will normally correspond to the MACID of the target user where this makes sense, and will be set to all 0's otherwise. The packet format bits will correspond to the packet format for a data packet, and will be set to all 0's otherwise.



1

2

Figure 86 Data scrambler

3

### 9.2.6 Modulation

4

5

6

7

8

9

10

11

12

13

14

15

The outputs of the data-scrambler for all the different sub-packets, where a sub-packet is as defined in 9.2.1, shall be applied to a modulator that outputs complex numbers known as modulation symbols, which are modulated on to OFDM subcarriers. The modulator cycles between the different sub-packets while generating the modulation symbols, i.e., modulation symbol 0 is generated from sub-packet 0, modulation symbol 1 is generated from sub-packet 1, etc. The airlink supports QPSK, 8PSK, 16QAM and 64QAM modulation formats. The number of bits in one modulation symbol is called the modulation order. The modulation order is 2 for QPSK, 3 for 8PSK, 4 for 16QAM and 6 for 64QAM. The airlink also supports multiple modulation formats for a single encoded bit sequence. This section describes the procedure for generating a sequence of modulation symbols of varying modulation formats from an encoded bit sequence (after channel interleaving and scrambling). The number of modulation symbols and the modulation format of each symbol will be specified separately in the specification of each physical layer channel using this procedure.

16

17

The sequence of modulation symbols output from the modulator shall be equivalent to those generated by the following approach:

18

19

20

21

22

23

24

25

1. Let  $y(0,0), y(0,1), \dots$  be the infinite-length sequence of bits at the output of the scrambler corresponding to sub-packet 0,  $y(1,0), y(1,1), \dots$  the infinite-length sequence of bits at the output of the scrambler for sub-packet 1 and so on. Let  $t$  be the total number of sub-packets. Initialize  $t$  counters, denoted by  $i_0, i_1, \dots, i_{t-1}$ , to 0. Initialize another set of  $t$  counters  $j_0, j_1, \dots, j_{t-1}$ , to 0. Counter  $i_m$  counts the number of modulation symbols that have already been generated for the  $m^{\text{th}}$  sub-packet, while counter  $j_m$  is a pointer to the bits that were last modulated for the  $m^{\text{th}}$  sub-packet. Initialize another counter  $k = 0$ , which counts the total number of modulation symbols generated.

26

27

28

29

2. Let  $q$  be the desired modulation order of the next modulation symbol. Let  $m = k \bmod t$ . Collect the sequence of  $q$  bits  $y(m, j_m), y(m, j_m+1), \dots, y(m, j_m+q-1)$ . A sequence of bits  $z(0), \dots, z(q-1)$  is obtained by rotating this sequence by the value  $i_m \bmod q$ , i.e.,  $z(p) = y(m, j_m + ((i_m + p) \bmod q))$ .



3. The rotated sequence  $z(0), z(1), \dots, z(q-1)$  is then mapped to a modulation symbol using the mapping described in 9.2.6.1 for QPSK, in 9.2.6.2 for 8PSK, in 9.2.6.3 for 16QAM, and in 9.2.6.4 for 64QAM.
4. Increment counter  $i_m$  by 1. Increment counter  $j_m$  by  $q$ . Increment counter  $k$  by 1.
5. Repeat steps 2 through 4 until the desired number of modulation symbols have been generated.

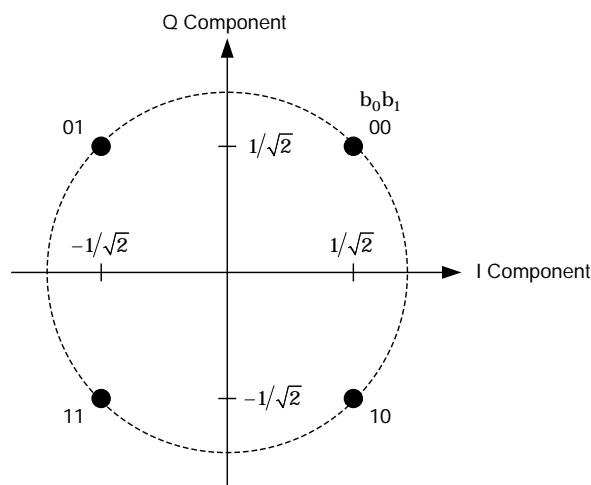
### 9.2.6.1 QPSK modulation

In the case of QPSK modulation, a group of 2 input bits ( $b_0, b_1$ ) is mapped into a complex modulation symbol ( $m_I(k), m_Q(k)$ ), as specified in Table 96. Figure 87 shows the signal constellation of the QPSK modulator.

**Table 96 QPSK modulation table**

Input Bits		Modulation Symbols	
$b_0$	$b_1$	$m_I(k)$	$m_Q(k)$
0	0	A	A
0	1	-A	A
1	0	A	-A
1	1	-A	-A

Note:  $A = 1/\sqrt{2}$



**Figure 87 Signal constellation for QPSK modulation**

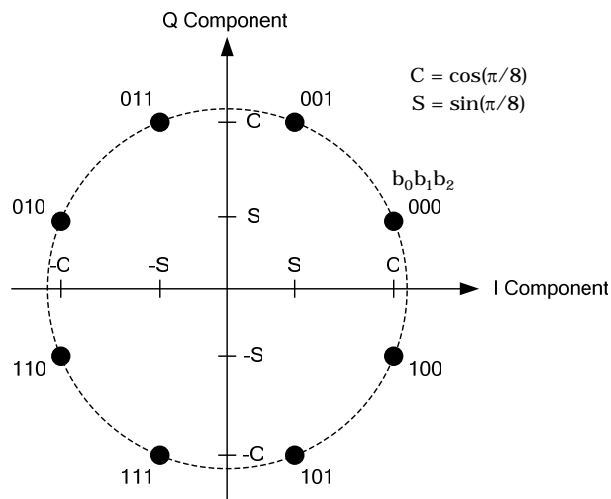
### 9.2.6.2 8-PSK modulation

In the case of 8-PSK modulation, a group of 3 input bits ( $b_0, b_1, b_2$ ) is mapped into a complex modulation symbol ( $m_I(k), m_Q(k)$ ) as specified in Table 97. Figure 88 shows the signal constellation of the 8-PSK modulator.

**Table 97 8-PSK modulation table**

Input Bits			Modulation Symbols	
$b_0$	$b_1$	$b_2$	$m_I(k)$	$m_Q(k)$
0	0	0	C	S
0	0	1	S	C
0	1	1	-S	C
0	1	0	-C	S
1	1	0	-C	-S
1	1	1	-S	-C
1	0	1	S	-C
1	0	0	C	-S

Note:  $C = \cos(\pi/8) \approx 0.9239$  and  $S = \sin(\pi/8) \approx 0.3827$



**Figure 88 Signal constellation for 8-PSK modulation**

### 9.2.6.3 16-QAM modulation

In the case of 16-QAM modulation, a group of 4 input bits ( $b_0, b_1, b_2, b_3$ ) is mapped into a complex modulation symbol ( $m_I(k), m_Q(k)$ ), as specified in Table 98. Figure 89 shows the signal constellation of the 16-QAM modulator.

**Table 98 16-QAM modulation table**

Input Bits				Modulation Symbols	
$b_0$	$b_1$	$b_2$	$b_3$	$m_Q(k)$	$m_I(k)$
0	0	0	0	3A	3A
0	0	0	1	3A	A
0	0	1	1	3A	-A
0	0	1	0	3A	-3A
0	1	0	0	A	3A
0	1	0	1	A	A
0	1	1	1	A	-A
0	1	1	0	A	-3A
1	1	0	0	-A	3A
1	1	0	1	-A	A
1	1	1	1	-A	-A
1	1	1	0	-A	-3A
1	0	0	0	-3A	3A
1	0	0	1	-3A	A
1	0	1	1	-3A	-A
1	0	1	0	-3A	-3A

Note:  $A = 1/\sqrt{10} \approx 0.3162$

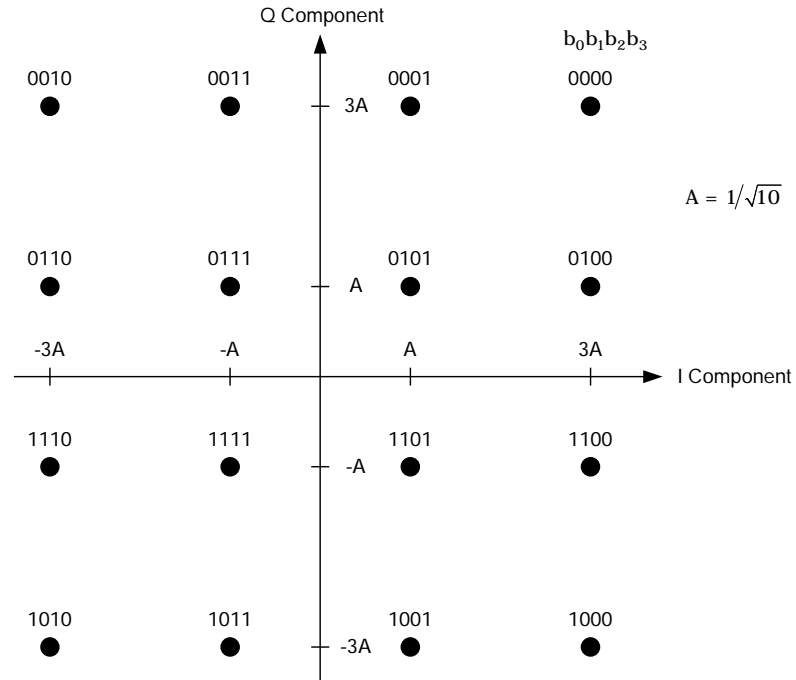


Figure 89 Signal constellation for 16-QAM modulation

#### 9.2.6.4 64-QAM modulation

In the case of 64-QAM modulation, a group of 6 input bits ( $b_0, b_1, b_2, b_3, b_4, b_5$ ) is mapped into a complex modulation symbol ( $m_I(k), m_Q(k)$ ) as specified in Table 99. Figure 90 shows the signal constellation of the 64-QAM modulator.

Table 99 64-QAM modulation table

Input Bits						Modulation Symbols	
$b_0$	$b_1$	$b_2$	$b_3$	$b_4$	$b_5$	$m_Q(k)$	$m_I(k)$
0	0	0	0	0	0	7A	7A
0	0	0	0	0	1	7A	5A
0	0	0	0	1	0	7A	A
0	0	0	0	1	1	7A	3A
0	0	0	1	0	0	7A	-7A
0	0	0	1	0	1	7A	-5A
0	0	0	1	1	0	7A	-A
0	0	0	1	1	1	7A	-3A
0	0	1	0	0	0	5A	7A
0	0	1	0	0	1	5A	5A
0	0	1	0	1	0	5A	A
0	0	1	0	1	1	5A	3A

Input Bits						Modulation Symbols	
$b_0$	$b_1$	$b_2$	$b_3$	$b_4$	$b_5$	$m_Q(k)$	$m_I(k)$
0	0	1	1	0	0	5A	-7A
0	0	1	1	0	1	5A	-5A
0	0	1	1	1	0	5A	-A
0	0	1	1	1	1	5A	-3A
0	1	0	0	0	0	A	7A
0	1	0	0	0	1	A	5A
0	1	0	0	1	0	A	A
0	1	0	0	1	1	A	3A
0	1	0	1	0	0	A	-7A
0	1	0	1	0	1	A	-5A
0	1	0	1	1	0	A	-A
0	1	0	1	1	1	A	-3A
0	1	1	0	0	0	3A	7A
0	1	1	0	0	1	3A	5A
0	1	1	0	1	0	3A	A
0	1	1	0	1	1	3A	3A
0	1	1	1	0	0	3A	-7A
0	1	1	1	0	1	3A	-5A
0	1	1	1	1	0	3A	-A
0	1	1	1	1	1	3A	-3A
1	0	0	0	0	0	-7A	7A
1	0	0	0	0	1	-7A	5A
1	0	0	0	1	0	-7A	A
1	0	0	0	1	1	-7A	3A
1	0	0	1	0	0	-7A	-7A
1	0	0	1	0	1	-7A	-5A
1	0	0	1	1	0	-7A	-A
1	0	0	1	1	1	-7A	-3A
1	0	1	0	0	0	-5A	7A
1	0	1	0	0	1	-5A	5A
1	0	1	0	1	0	-5A	A
1	0	1	0	1	1	-5A	3A
1	0	1	1	0	0	-5A	-7A
1	0	1	1	0	1	-5A	-5A
1	0	1	1	1	0	-5A	-A
1	0	1	1	1	1	-5A	-3A
1	1	0	0	0	0	-A	7A

Input Bits						Modulation Symbols	
$b_0$	$b_1$	$b_2$	$b_3$	$b_4$	$b_5$	$m_Q(k)$	$m_I(k)$
1	1	0	0	0	1	-A	5A
1	1	0	0	1	0	-A	A
1	1	0	0	1	1	-A	3A
1	1	0	1	0	0	-A	-7A
1	1	0	1	0	1	-A	-5A
1	1	0	1	1	0	-A	-A
1	1	0	1	1	1	-A	-3A
1	1	1	0	0	0	-3A	7A
1	1	1	0	0	1	-3A	5A
1	1	1	0	1	0	-3A	A
1	1	1	0	1	1	-3A	3A
1	1	1	1	0	0	-3A	-7A
1	1	1	1	0	1	-3A	-5A
1	1	1	1	1	0	-3A	-A
1	1	1	1	1	1	-3A	-3A

Note:  $A = \sqrt{1/42} \approx 0.1543$

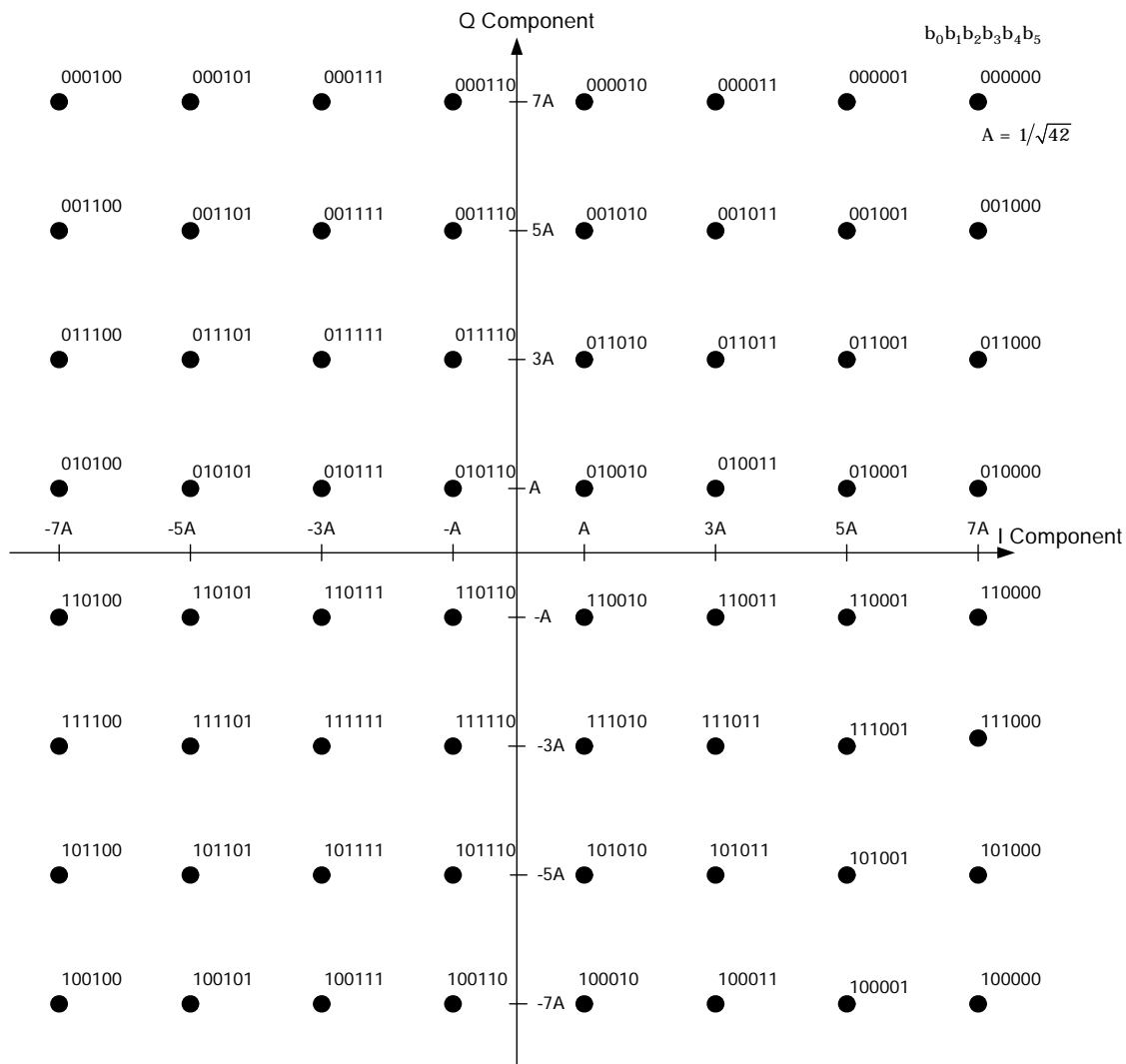


Figure 90 Signal constellation for 64-QAM modulation

### 9.3 Access network requirements

This section defines requirements specific to access network (AN) equipment and operation.

#### 9.3.1 Transmitter

The transmitter shall reside in each sector of the access network. These requirements apply to the transmitter in each sector.

Each sector is assigned an integer identifier in the range 0-4095 (including 0 and 4095) called the PilotPN. This identifier may also be referred to as a 12-bit binary number with the leftmost bit being the MSB and the rightmost bit being the LSB.

### 9.3.2 Modulation characteristics

#### 9.3.2.1 Superframe timing

The forward link transmission is divided into units of superframes. The structure of a forward link superframe shall be as shown in Figure 91 for FDD and as shown in Figure 92, Figure 93, Figure 94 for TDD for different values of the parameters  $N_{FL\_BURST}$  and  $N_{RL\_BURST}$ . A superframe shall consist of a superframe preamble followed by a series of  $N_{FDD,FLPHYFrames}$  FL PHY Frames in FDD, and by  $N_{TDD,FLPHYFrames}$  FL PHY Frames in TDD. In TDD mode with parameter  $N_{FL\_BURST}$  and  $N_{RL\_BURST}$ ,  $N_{FL\_BURST}$  FL PHY Frames alternate with the mute time reserved for RL PHY Frames. Here,  $N_{FDD,FLPHYFrames}$  and  $N_{TDD,FLPHYFrames}$ <sup>44</sup> are as defined by the Lower MAC sublayer. The superframe preamble carries the F-CPICH, the F-pBCH, the F-ACQCH, and the F-OSICH. The FL PHY Frames carry the F-CPICH, F-AuxPICH, the F-SSCH and the F-DCH physical channels for SymbolRateHopping mode and the F-DPICH, the F-CPICH, the F-SSCH, and the F-DCH physical channels in BlockHopping mode. The structure of the superframe preamble and each FL PHYFrame shall be as shown in Figure 95 for SymbolRateHopping mode and in Figure 96 for BlockHopping mode.

Each superframe shall be identified by a superframe index that is incremented every superframe. The superframe index is related to the System Time as described in Chapter 1. Each superframe also has an associated quantity called the PilotPhase, defined as  $(PilotPN + Superframe Index) \bmod 4096$ .

The PHY layer chapter of this specification uses a FL PHY Frame indexing scheme that is convenient for the descriptions herein, but is not necessarily consistent with indexing schemes used in other layers and sublayers in the specification. In this indexing scheme, the FL PHY Frames in a given superframe shall be indexed sequentially from 0 through  $N_{FDD,FLPHYFrames} - 1$  in FDD mode and from 0 through  $N_{TDD,FLPHYFrames} - 1$  in TDD mode. The FL PHY Frame index is sometimes also referred to using its 6-bit binary representation.

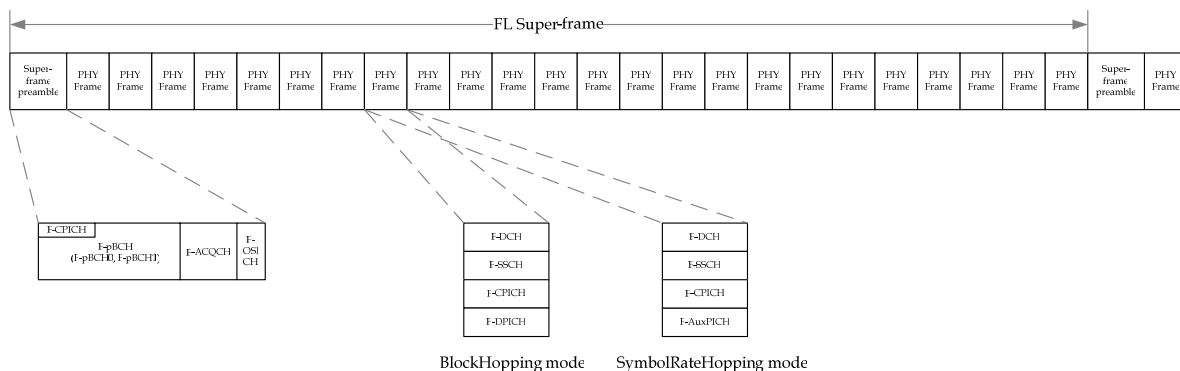
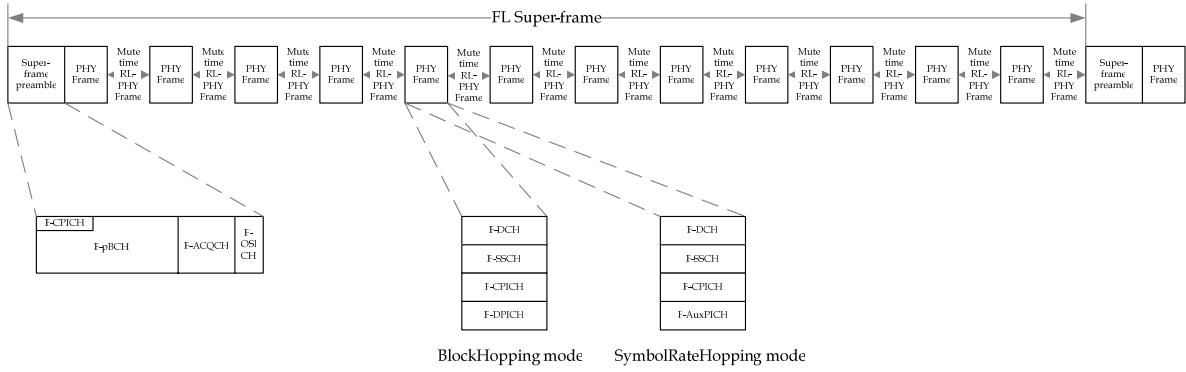


Figure 91 Forward link superframe structure: FDD

<sup>44</sup> Note that  $N_{TDD,FLPHYFrames}$  is a function of  $N_{FL\_BURST}$  and  $N_{RL\_BURST}$ .

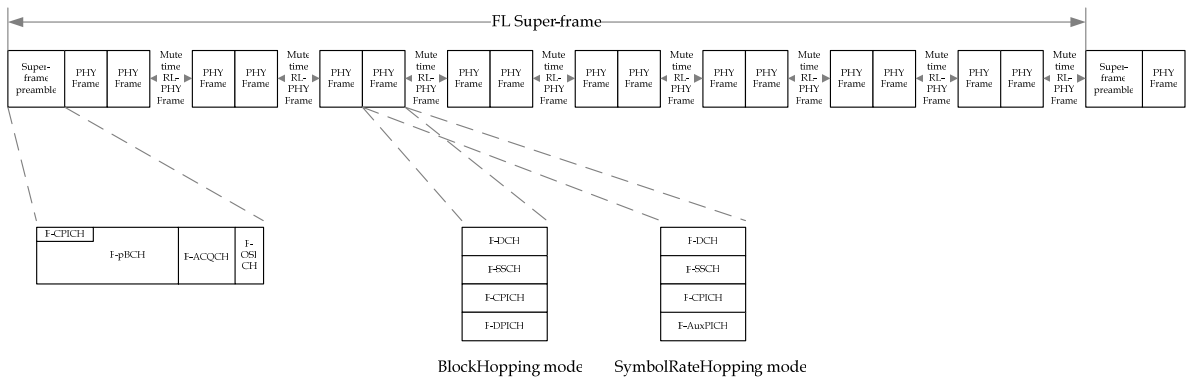




1

**Figure 92 Forward link superframe structure: TDD ( $N_{FL\_BURST} = 1, N_{RL\_BURST} = 1$ )**

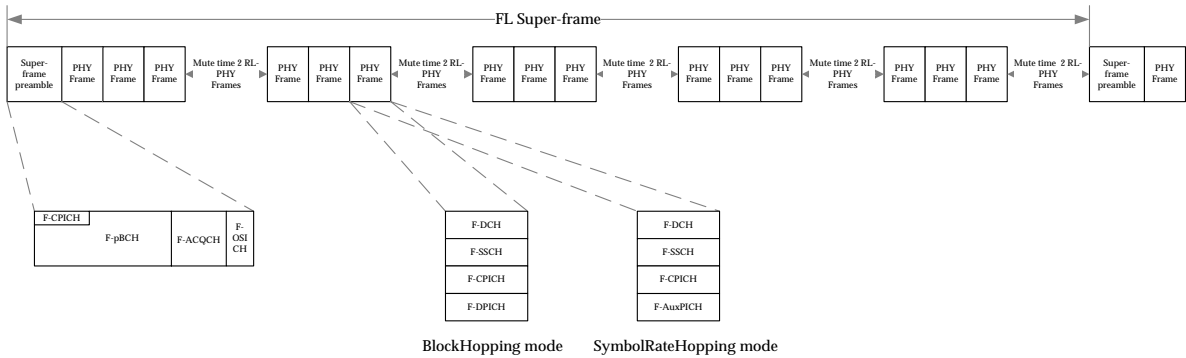
2



3

**Figure 93 Forward link superframe structure: TDD ( $N_{FL\_BURST} = 2, N_{RL\_BURST} = 1$ )**

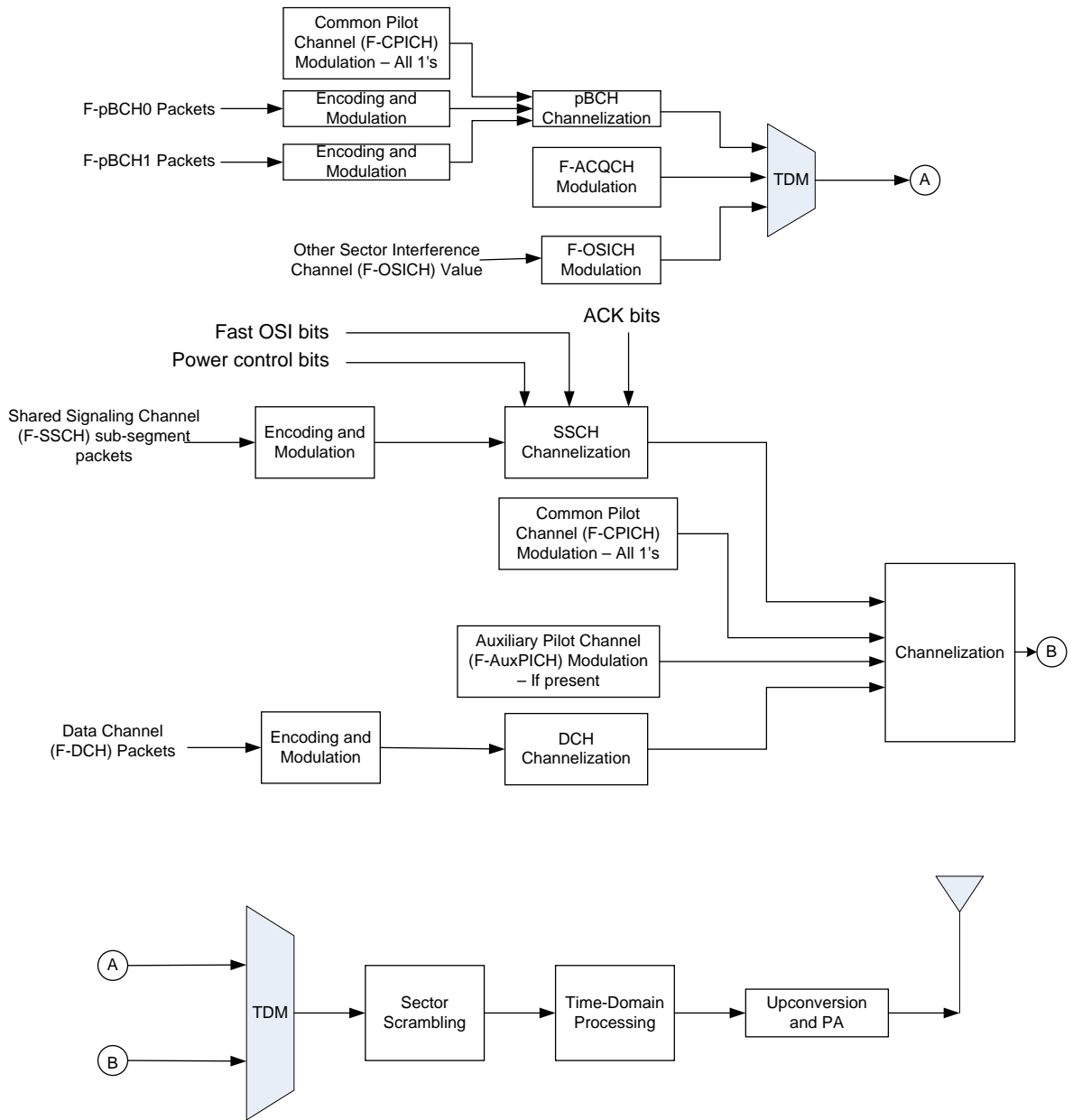
4



5

**Figure 94 Forward link superframe structure: TDD ( $N_{FL\_BURST} = 3, N_{RL\_BURST} = 2$ )**

6



1

2

**Figure 95 Forward channel structure: SymbolRateHopping mode**

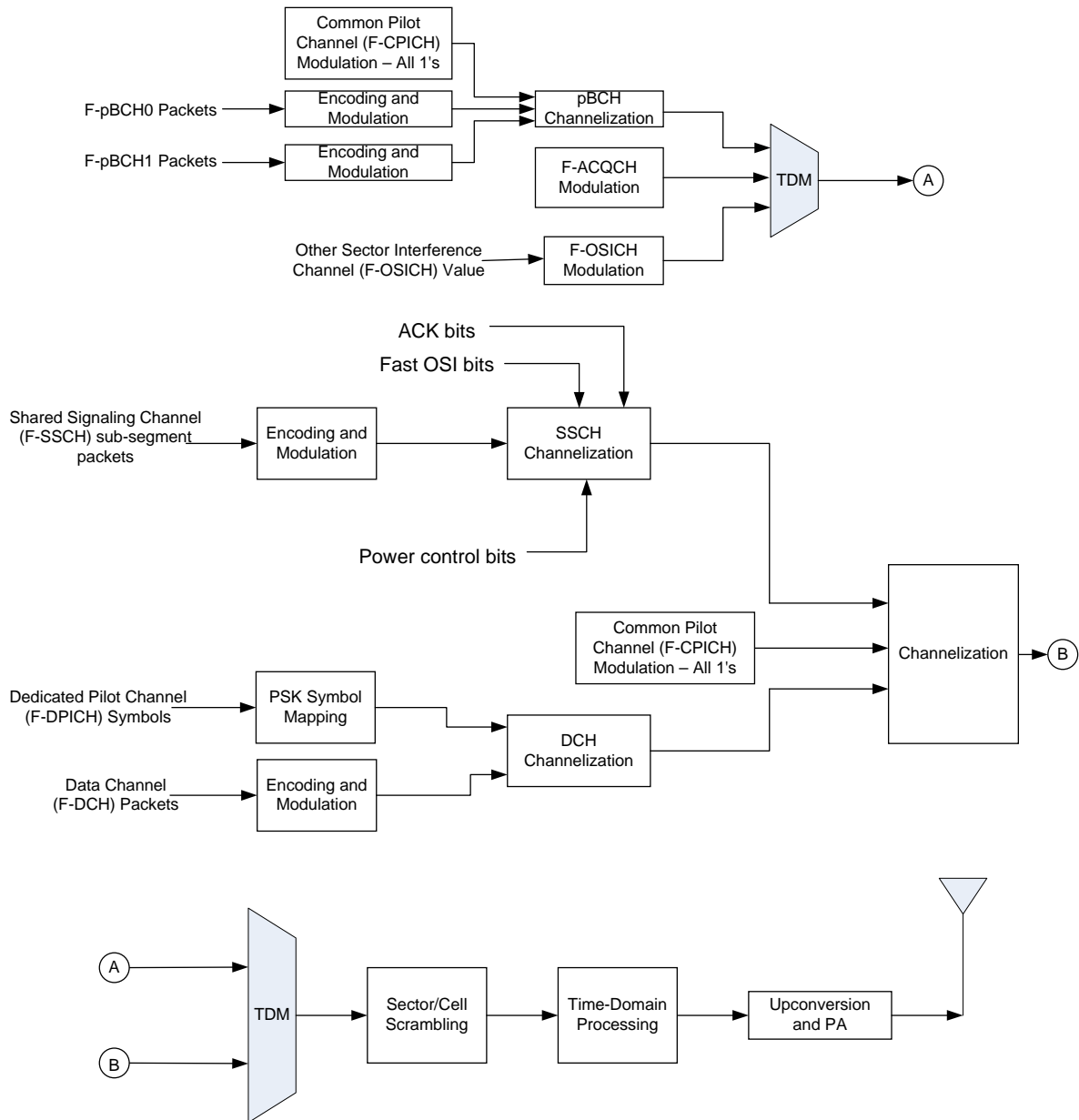


Figure 96 Forward channel structure: BlockHopping mode

### 9.3.2.2 OFDM symbol characteristics

The modulation used on the forward link is Orthogonal Frequency Division Multiplexing (OFDM). Both the superframe preamble as well as each FL PHY Frame shall be further subdivided into units of OFDM symbols. An OFDM symbol consists of  $N_{\text{FFT}}$  individually modulated subcarriers that carry complex-valued data. Complex-valued data is represented in the form  $d = (d_{\text{re}}, d_{\text{im}})$ , where  $d_{\text{re}}$  and  $d_{\text{im}}$  represent the real and imaginary components respectively. The subcarriers in each OFDM symbol shall be numbered 0 through  $N_{\text{FFT}}-1$ .

An additional indexing scheme may be used in MultiCarrierOn mode. The  $N_{\text{FFT}}$  subcarriers are split into  $N_{\text{CARRIERS}}$  contiguous groups, each of which is referred to as a carrier. Each carrier consists of

1  $N_{\text{CARRIER\_SIZE}}$  subcarriers, where  $N_{\text{CARRIER\_SIZE}} = N_{\text{FFT}} / N_{\text{CARRIERS}}$ . Each carrier has an associated index,  
 2 sometimes referred to as CarrierIndex, that ranges from 0 through  $N_{\text{CARRIERS}} - 1$ . The carrier with index  
 3  $c$  consists of subcarriers indexed  $cN_{\text{CARRIER\_SIZE}}$  through  $(c+1)N_{\text{CARRIER\_SIZE}} - 1$ . In MultiCarrierOff  
 4 mode, all  $N_{\text{FFT}}$  subcarriers belong to a single carrier having CarrierIndex 0. Furthermore, the  
 5 subcarriers within each carrier may be indexed from 0 to  $N_{\text{CARRIER\_SIZE}} - 1$  and the phrases “subcarrier  $f$   
 6 in carrier  $c$ ” and “subcarrier with index  $f$  within carrier with index  $c$ ” shall be equivalent to  
 7 “subcarrier  $cN_{\text{CARRIER\_SIZE}} + f$ .” These two subcarrier indexing schemes are used interchangeably in  
 8 the Physical Layer chapter of this specification.

### 9 **9.3.2.2.1 Guard subcarriers**

10 Some of the available subcarriers in an OFDM symbol are designated as guard subcarriers and shall  
 11 not be modulated, i.e., no energy shall be transmitted on these subcarriers. The number of guard  
 12 subcarriers in the superframe preamble and in each FL PHY Frame shall be  $N_{\text{GUARD,PR}}$  and  $N_{\text{GUARD}}$   
 13 respectively. The quantity  $N_{\text{GUARD}}$  is given by the NumGuardSubcarriers parameter which is part of  
 14 the public data of the Overhead Messages Protocol. The set of guard subcarriers in the superframe  
 15 preamble shall be the subcarriers numbered 0 through  $N_{\text{GUARD,PR}}/2 - 1$  and the subcarriers numbered  
 16  $N_{\text{FFT}} - N_{\text{GUARD,PR}}/2$  through  $N_{\text{FFT}} - 1$ . The set of guard subcarriers in each FL PHY Frame shall be the  
 17 subcarriers numbered 0 through  $N_{\text{GUARD}}/2 - 1$  and the subcarriers numbered  $N_{\text{FFT}} - N_{\text{GUARD}}/2$  through  
 18  $N_{\text{FFT}} - 1$ .

### 19 **9.3.2.2.2 Quasi-guard subcarriers**

20 In MultiCarrierOn mode, additional sub-carriers within each OFDM symbol are designated as quasi-  
 21 guard subcarriers and shall not be modulated, i.e., no energy shall be transmitted on these subcarriers.  
 22 The set of quasi-guard subcarriers in the superframe preamble shall be the subcarriers numbered  
 23  $N_{\text{CARRIER\_SIZE}} * m - N_{\text{GUARD,PR}}/2$  through  $N_{\text{CARRIER\_SIZE}} * m + N_{\text{GUARD,PR}}/2 - 1$  where  $m = 1, \dots, N_{\text{CARRIERS}} -$   
 24  $1$ . The set of quasi-guard subcarriers in each FL PHY Frame shall be the subcarriers numbered  
 25  $N_{\text{CARRIER\_SIZE}} * m - N_{\text{GUARD}}/2$  to  $N_{\text{CARRIER\_SIZE}} * m + N_{\text{GUARD}}/2 - 1$  where  $m = 1, \dots, N_{\text{CARRIERS}} - 1$ .  
 26 Subcarriers that are not guard or quasi-guard subcarriers are also referred to as usable subcarriers.

### 27 **9.3.2.2.3 OFDM symbol duration**

28 The total OFDM symbol duration, denoted by  $T_{s,PR}$  during the superframe preamble and by  $T_s$  during  
 29 each FL PHY Frame, consists of four parts:

- 30 ■ A data part with duration  $T_{\text{FFT}}$ , where  $T_{\text{FFT}} = N_{\text{FFT}} T_{\text{CHIP}}$ .
- 31 ■ A flat guard interval, also known as a cyclic prefix, with duration  $T_{\text{CP,PR}}$  in the superframe  
 32 preamble and duration  $T_{\text{CP}}$  during each FL PHY Frame. Here,  $T_{\text{CP}}$  is given by the  
 33  $\text{CPLength}$  parameter which is part of the public data of the Overhead Messages Protocol.
- 34 ■ Two windowed guard intervals of duration  $T_{\text{WGI}}$  on the two sides of the OFDM symbol.  
 35 There is an overlap of  $T_{\text{WGI}}$  between consecutive OFDM symbols (see Figure 106).

36 The effective OFDM symbol duration is  $T_{s,PR} = T_{\text{FFT}} + T_{\text{CP,PR}} + T_{\text{WGI}}$  in the superframe preamble, and  
 37  $T_s = T_{\text{FFT}} + T_{\text{CP}} + T_{\text{WGI}}$  in each FL PHY Frame. This effective symbol duration shall henceforth be  
 38 referred to as the OFDM symbol duration.

### 9.3.2.2.4 Superframe duration

The superframe preamble consists of  $N_{\text{PREAMBLE}}$  OFDM symbols. In FDD, each FL PHY Frame consists of  $N_{\text{FRAME,F}}$  OFDM symbols and the total superframe duration is given by

$$T_{\text{SUPERFRAME}} = N_{\text{PREAMBLE}}T_{s,\text{PR}} + N_{\text{FDD,FLPHYFrames}}N_{\text{FRAME,F}}T_s.$$

In TDD, each FL PHY Frame consists of  $N_{\text{FRAME,F}}$  OFDM symbols. The mute time between each set of  $N_{\text{FL\_BURST}}$  contiguous FL PHY Frames in TDD is equal to the duration of  $N_{\text{RL\_BURST}}$  contiguous RL PHY Frames plus guard times  $T_{\text{G,TDD,R}}$  and  $T_{\text{G,TDD,F}}$ . The total superframe duration is given by

$$T_{\text{SUPERFRAME}} = N_{\text{PREAMBLE}}T_{s,\text{PR}} + N_{\text{TDD,FLPHYFrames}}N_{\text{FRAME,F}}T_s + N_{\text{TDD,RLPHYFrames}}N_{\text{FRAME,R}}T_s \\ + (T_{\text{G,TDD,F}} + T_{\text{G,TDD,R}}) * (N_{\text{TDD,RLPHYFrames}}/N_{\text{RL\_BURST}})$$

### 9.3.2.2.5 Hop-port indexing

During the FL PHY Frame portion of the transmission, the subcarriers in each carrier of each OFDM symbol will also use a second indexing scheme known as hop-port indexing. In this scheme, each carrier in each OFDM symbol consists of  $Q_{\text{SDMA}}N_{\text{CARRIER\_SIZE}}$  individually-modulated hop-ports. Here  $Q_{\text{SDMA}}$  is equal to  $\text{FLNumSDMADimensions}$ , which is part of the public data of the Overhead Messages Protocol on that carrier. The hop-ports in each carrier are indexed from 0 through  $Q_{\text{SDMA}}N_{\text{CARRIER\_SIZE}} - 1$ . The hop-port with index  $p$  in the carrier with CarrierIndex  $k$  is sometimes represented by the pair  $(k,p)$ . An order is defined on the set of such pairs by saying that  $(k_0,p_0) < (k_1,p_1)$  if either of the following two conditions is satisfied:

- i.  $k_0 < k_1$ , or
- ii.  $k_0 = k_1$  and  $p_0 < p_1$ .

There is a mapping between the  $Q_{\text{SDMA}}N_{\text{CARRIER\_SIZE}}$  hop-ports and the  $N_{\text{CARRIER\_SIZE}}$  subcarriers in each carrier, called a hop-permutation. The hop-permutation may change as often as every OFDM symbol and is different for different sectors. The sequence of hop-permutations, also called the hopping sequence, is described in 9.3.2.5.1.

### 9.3.2.3 Multiple transmit antennas

Multiple transmit antennas may be present at the sector transmitter. The different transmit antennas shall have the same superframe timing (including the superframe index), the same OFDM symbol characteristics, and the same hop-permutation.

Modulation of some of the physical layer channels (for example the F-AuxPICH and the portion of the F-CPICH that lies in the FL PHY Frames) is specified separately for each transmit antenna. Here the term “transmit antenna” denotes an “effective transmit antenna” which is not necessarily the same as a physical antenna present at the sector.<sup>45</sup> The mapping between effective and physical transmit antennas is beyond the scope of this specification. Note that transmission on a single effective antenna may involve transmission on any or all of the physical antennas. The number of effective transmit antennas in the system is given by the `EffectiveNumAntennas` parameter, which is part of the

<sup>45</sup> Here, a physical antenna refers to an individual radiating element.

1 public data of the Overhead Messages Protocol. The different effective antennas in the system are  
 2 indexed 0 through EffectiveNumAntennas – 1. Any reference to the term “transmit antenna” shall  
 3 henceforth be interpreted as meaning an effective transmit antenna.

4 The modulation of some of the physical layer channels (for example all the channels in the  
 5 superframe preamble, including the portion of the F-CPICH that lies in the superframe preamble) is  
 6 specified only for a single effective antenna. If multiple effective antennas are present, the modulation  
 7 symbols corresponding to these channels shall be transmitted only from the effective antenna with  
 8 index 0.

9 Finally, the modulation of some of the physical layer channels (for example the F-DPICH) is  
 10 described in terms of a concept called tile-antennas. A tile-antenna is a linear combination of the  
 11 effective antennas present in the system. Multiple tile-antennas can be constructed using different  
 12 linear combinations of the effective antennas. The process of constructing tile-antennas, i.e., the  
 13 process of transmitting on a linear combination of effective antennas, is known as precoding. The  
 14 mapping between effective antennas and tile-antennas can be described in terms of a matrix called the  
 15 precoding matrix.

16 The physical layer also supports superposition of two waveforms on the same set of subcarriers,  
 17 potentially using different precoding matrices. This happens when the hop-permutation maps two  
 18 hop-ports to the same subcarrier, and is known as Space Division Multiple Access (SDMA).

19 The sector-specific scrambling, cell-specific scrambling and time-domain processing operations,  
 20 described in 9.3.2.6, 9.3.2.7, and 9.3.2.8, respectively, shall be identical for each of the effective  
 21 transmit antennas.

### 22 9.3.2.4 Superframe preamble modulation

23 The superframe preamble shall consist of  $N_{\text{PREAMBLE}} = 8$  OFDM symbols, which shall be indexed  
 24 from 0 through  $N_{\text{PREAMBLE}} - 1$ . The Common Pilot Channel (F-CPICH) shall be carried on the OFDM  
 25 symbols with indices 0 and 1. The Primary Broadcast Channel (F-pBCH) shall be carried on the  
 26 OFDM symbols with indices 0 through 4. The Acquisition Channel (F-ACQCH) shall be carried on  
 27 the OFDM symbols with indices 5 and 6. Finally, the Other Sector Interference Channel (F-OSICH)  
 28 shall be carried on the OFDM symbol with index 7.

#### 29 9.3.2.4.1 Offset<sub>p</sub> definition

30 A variable Offset<sub>p</sub> is defined for each carrier and each OFDM symbol in the superframe. This variable  
 31 shall be determined using a 13-bit PN-register with generator polynomial  $h(D) = 1 + D^8 + D^{11} + D^{12} +$   
 32  $D^{13}$ , which is shown in Figure 97. For the carrier with CarrierIndex  $k_c$ , the shift-register shall be  
 33 initialized to the state [1 p<sub>11</sub> p<sub>10</sub> p<sub>9</sub> p<sub>8</sub> p<sub>7</sub> p<sub>6</sub> p<sub>5</sub> p<sub>4</sub> p<sub>3</sub> p<sub>2</sub> p<sub>1</sub> p<sub>0</sub>] before the beginning of the superframe. In  
 34 SemiSynchronous mode, the bits p<sub>11</sub>, p<sub>10</sub>, p<sub>9</sub>, ..., p<sub>0</sub> are the 12 bits of the quantity (PilotPhase +  $k_c$ )  
 35 mod 4096, with p<sub>11</sub> being the MSB and p<sub>0</sub> being the LSB. In Asynchronous mode, the bits p<sub>11</sub>, p<sub>10</sub>, p<sub>9</sub>,  
 36 ..., p<sub>0</sub> are the 12 bits of the quantity (PilotPN +  $k_c$ ) mod 4096, with p<sub>11</sub> being the MSB and p<sub>0</sub> being  
 37 the LSB. Offset<sub>p</sub> shall be generated by clocking the shift-register 13 times for every even OFDM  
 38 symbol in the superframe. More precisely, for the carrier with CarrierIndex  $k_c$  and the OFDM symbol  
 39 with index  $j$  in the superframe, Offset<sub>p</sub> shall be chosen to be the value of this register after it has been  
 40 clocked  $\lfloor j/2 \rfloor * 13$  times. The value of the register shall be read from left to right; i.e., the left-most  
 41 state shown in Figure 97 shall be treated as the MSB and the right-most state shall be treated as the  
 42 LSB.

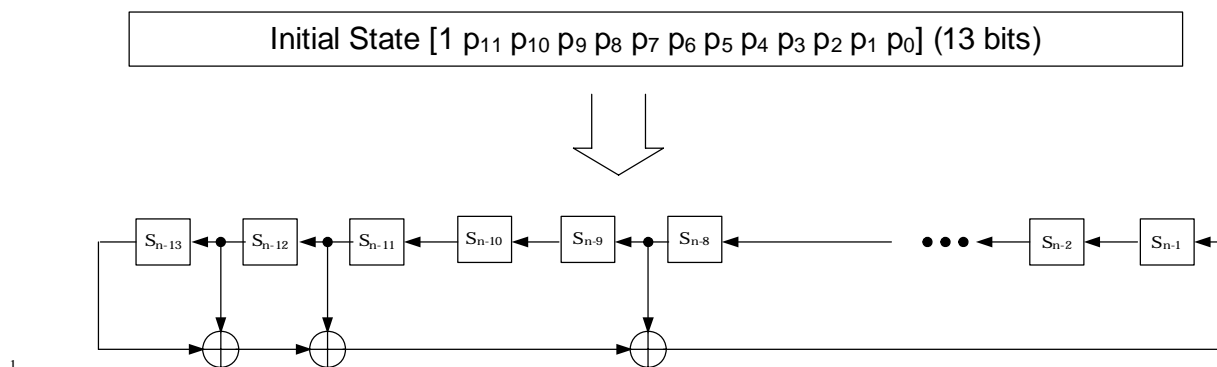


Figure 97 Offset<sub>p</sub> Determination

### 9.3.2.4.2 F-CPICH

The F-CPICH shall occupy an evenly spaced set of subcarriers in each carrier in the first two OFDM symbols of the superframe preamble. The spacing between neighboring pilot subcarriers for the superframe preamble shall be given by  $D_{p,PR} = 4$ . Let  $\text{Offset}_{p,k}$  be the value of  $\text{Offset}_p$ , as defined in 9.3.2.4.1, for the first two OFDM symbols in the superframe preamble and for the carrier with CarrierIndex  $k$ . For the OFDM symbol with index 0, the subcarrier with index  $i_{sc}$  in this OFDM symbol, where  $i_{sc}$  is such that this subcarrier lies in the carrier with CarrierIndex  $k$ , shall be occupied by the F-CPICH if it is a usable subcarrier and  $i_{sc} \bmod D_{p,PR} = \text{Offset}_{p,k} \bmod D_{p,PR}$ . For the OFDM symbol with index 1, the subcarrier with index  $i_{sc}$  in this OFDM symbol, where  $i_{sc}$  is such that this subcarrier lies in the carrier with CarrierIndex  $k$ , shall be occupied by the F-CPICH if it is a usable subcarrier and  $i_{sc} \bmod D_{p,PR} = (\text{Offset}_{p,k} + D_{p,PR}/2) \bmod D_{p,PR}$ .

Each subcarrier occupied by the F-CPICH shall be modulated with the complex value  $(\sqrt{P}, 0)$ , where  $P$  is the ratio of the power spectral density of the F-CPICH to the power spectral density of the second OFDM symbol in the F-ACQCH. This ratio is given by the PreamblePilotPower parameter, which is part of the public data of the Overhead Messages Protocol.

Any subcarrier in the superframe preamble occupied by the F-CPICH will be referred to as a pilot subcarrier.

### 9.3.2.4.3 F-pBCH

The F-pBCH shall be carried on the first five OFDM symbols in the superframe preamble. This channel consists of two sub-channels, namely F-pBCH0 and F-pBCH1.

### 9.3.2.4.3.1 F-pBCH0

One F-pBCH0 packet is transmitted over each carrier. An F-pBCH0 packet in the carrier with CarrierIndex  $k_c$  shall start in every superframe that satisfies  $(\text{PilotPhase} + k_c) \bmod N_{\text{pBCH0\_Period}} = 0$ , and shall be transmitted over a period of 16 superframes. Each F-pBCH0 packet is generated by the CC MAC Protocol and shall be appended with CRC, encoded, channel interleaved, repeated, and modulated using the procedures described in 9.2. A CRC length of  $N_{\text{CRC,pBCH}}$  shall be used for this packet. No data-scrambling operation shall be performed on this packet. QPSK modulation shall be used for all of the modulation symbols. The modulation symbols for the F-pBCH0 packet to be transmitted in the carrier with CarrierIndex  $k_c$  shall be modulated on to OFDM subcarriers using the following procedure:

1. Initialize a superframe counter  $i$  to the superframe index corresponding to the first superframe carrying the F-pBCH0 packet. Initialize a subcarrier counter  $j$  to  $N_{\text{GUARD,PR}}/2$ .
2. If the subcarrier with index  $N_{\text{CARRIER\_SIZE}}k_c + j$  in OFDM symbol index 0 of superframe index  $i$  is not a pilot subcarrier, then a QPSK modulation symbol  $s$  shall be generated using the procedure described in 9.2.6. This modulation symbol shall be modulated with energy  $P$  on the subcarrier with index  $N_{\text{CARRIER\_SIZE}}k_c + j$ , i.e., the value of the subcarrier shall be  $\sqrt{P} s$ . Here,  $P$  is the power density assigned to this packet by the CC MAC Protocol.
3. Increment  $j$ .
4. If  $j = (N_{\text{CARRIER\_SIZE}} + N_{\text{GUARD,PR}})/4$ , increment  $i$  and set  $j = N_{\text{GUARD,PR}}/2$ .
5. If  $i$  is larger than the superframe index of the last superframe carrying the F-pBCH0 packet, then stop. Otherwise repeat steps 2 through 5.

### 9.3.2.4.3.2 F-pBCH1

One F-pBCH1 packet is transmitted in each carrier in each superframe. Each F-pBCH1 packet is generated by the CC MAC Protocol and shall be appended with CRC, encoded, channel interleaved, repeated, and modulated using the procedures described in 9.2. A CRC length of  $N_{\text{CRC,pBCH}}$  shall be used for this packet. No data-scrambling operation shall be performed on this packet. QPSK modulation shall be used for all of the modulation symbols. The modulation symbols for the F-pBCH1 packet to be transmitted in the carrier with CarrierIndex  $k_c$  in a given superframe shall be modulated on to OFDM subcarriers using the following procedure:

1. At the beginning of the superframe, initialize an OFDM symbol counter  $i$  to 0 and a subcarrier counter  $j$  to  $(N_{\text{CARRIER\_SIZE}} + N_{\text{GUARD,PR}})/4$ .
2. If the subcarrier with index  $N_{\text{CARRIER\_SIZE}}k_c + j$  in the OFDM symbol with index  $i$  is not a pilot subcarrier, then a QPSK modulation symbol  $s$  shall be generated using the procedure described in 9.2.6. This modulation symbol shall be modulated with energy  $P$  on the subcarrier with index  $N_{\text{CARRIER\_SIZE}}k_c + j$ , i.e., the value of the subcarrier is  $\sqrt{P} s$ . Here,  $P$  is the power density assigned to this packet by the CC MAC Protocol.
3. Increment  $j$ .



1           4. If  $j = N_{\text{CARRIER\_SIZE}} - N_{\text{GUARD,PR}}/2$  increment  $i$  and set  $j = N_{\text{GUARD,PR}}/2$ .

2           5. If  $i = 5$ , then stop. Else repeat steps 2 through 5.

#### 3   **9.3.2.4.4 F-ACQCH**

4   The Acquisition Channel shall consist of the OFDM symbols with index 5 and 6 in each superframe  
5   preamble. In the OFDM symbol with index 5, all usable subcarriers with even-numbered indices shall  
6   be modulated with the value  $(\sqrt{2}, 0)$ . All subcarriers with odd-numbered indices shall be  
7   unmodulated; i.e., they shall have zero energy. In the OFDM symbol with index 6, all usable  
8   subcarriers shall be modulated with the value  $(1,0)$ .<sup>46</sup>

#### 9   **9.3.2.4.5 F-OSICH**

10   The F-OSICH shall be carried on the last OFDM symbol in each superframe preamble. The F-OSICH  
11   carries a three-state quantity  $x$ , taking the values 0, 1, and 2, in each carrier. The value of this quantity  
12   for each carrier is determined by the CC MAC Protocol. If  $x = 0$ , all usable subcarriers in the relevant  
13   carrier shall be modulated with the value  $(1,0)$ . If  $x = 1$ , all usable subcarriers in the relevant carrier  
14   shall be modulated with the value  $(0,1)$ . If  $x = 2$ , all usable subcarriers in the relevant carrier shall be  
15   modulated with the value  $(-1,0)$ .

#### 16   **9.3.2.5 FL PHY Frame modulation**

17   The superframe preamble shall be followed by a sequence of FL PHY Frames, each of which has an  
18   associated FL PHY Frame index, as described in 9.3.2.1. Each FL PHY Frame shall consist of  
19    $N_{\text{FRAME,F}}$  OFDM symbols, and its structure shall be as specified in Figure 95 (SymbolRateHopping  
20   mode) and Figure 96 (BlockHopping mode). Each carrier in each OFDM symbol in a FL PHY Frame  
21   consists of  $Q_{\text{SDMA}}N_{\text{CARRIER\_SIZE}}$  hop-ports, numbered from 0 through  $Q_{\text{SDMA}}N_{\text{CARRIER\_SIZE}}-1$ , as  
22   described in 9.3.2.2.5. The set of hop-ports in each carrier is mapped to the set of subcarriers in this  
23   carrier through the hop-permutation.

#### 24   **9.3.2.5.1 Hopping sequence generation**

25   The hop-permutation is used to map the set of hop-ports to the set of subcarriers. The hopping  
26   sequence will be described separately for SymbolRateHopping and BlockHopping modes.

27   In SymbolRateHopping mode, a new pseudorandom hop-permutation is generated every two OFDM  
28   symbols.

29   In BlockHopping mode, the set of non-guard hop-ports is divided into groups of  $N_{\text{BLOCK}}$  consecutive  
30   hop-ports, each of which is denoted as a block. The hop-permutation will map a block of hop-ports to  
31   a group of subcarriers with consecutive indices. This group of subcarriers will also be referred to as a  
32   block. Furthermore, the hop-permutation will remain constant for the duration of the FL PHY Frame.  
33   In this design, therefore, a group of hop-ports spanning a FL PHY Frame worth of OFDM symbols in  
34   time and  $N_{\text{BLOCK}}$  hop-ports in hop-port space are mapped to neighboring subcarriers in the time-  
35   frequency grid. This group of  $N_{\text{BLOCK}}N_{\text{FRAME,F}}$  hop-ports shall be referred to as a tile.

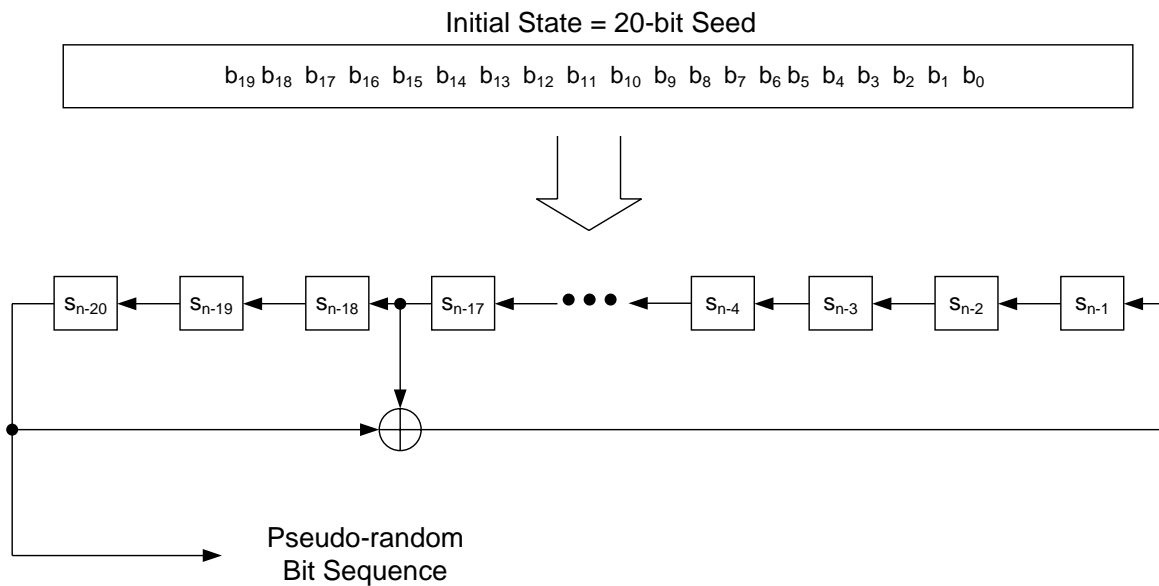
---

<sup>46</sup> All the subcarriers in the OFDM symbol with index 6 are modulated with the energy 1 because this OFDM symbol serves as a power reference for the forward link transmission. The power density to be used for all the other physical layer channels will be specified in reference to this OFDM symbol.

### 9.3.2.5.1.1 Common permutation generation algorithm

All permutations used for FL hopping shall be generated using a common permutation generation algorithm. The algorithm takes a 20-bit seed and a permutation size  $M$  as inputs and outputs a permutation of the set  $\{0, 1, \dots, M-1\}$ . The algorithm uses a linear feedback shift register to generate pseudorandom numbers, which in turn are used to generate pseudorandom permutations.

The 20-tap linear feedback shift register shall have a generator sequence of  $h(D) = 1 + D^{17} + D^{20}$ , as shown in Figure 98. The  $j$ 'th output  $s(j)$  of this shift register shall satisfy  $s(j) = s(j-17) \oplus s(j-20)$ . The initial state of the register shall generate the first output bit. A pseudorandom number  $x$  in  $\{0, 1, \dots, 2^n - 1\}$  for any  $n < 17$  can be generated by clocking the register  $n$  times, with the initial output bit being the LSB of  $x$  and the final ( $n$ 'th) output bit being the MSB of  $x$ .



**Figure 98 PN Register for generating pseudorandom bits**

The common permutation generation algorithm shall generate a permutation of size  $M$  as follows:

1. Initialization Steps:
  - a. Let  $n$  be the smallest integer such that  $M \leq 2^n$ .
  - b. Initialize an array  $A$  of size  $M$  with the numbers  $0, 1, 2, \dots, M-1$  (i.e.,  $A[0]=0, A[1]=1 \dots, A[M-1]=M-1$ )
  - c. Initialize the PN register with the 20-bit seed.
  - d. Initialize counter  $i$  to  $M-1$ .
2. Repeat the following steps until  $i=0$ .
  - a. Find the smallest  $p$  such that  $i < 2^p$ .

- 1           b. Initialize counter  $j$  to 0 and an output  $x$  to  $i+1$ .
- 2           c. Repeat the following steps until  $j=3$  or until  $x \leq i$ .
  - 3           i. Clock the PN register  $n$  times to obtain an  $n$ -bit pseudorandom number. Set  $x$  to
  - 4           be the  $p$  LSBs of that number.
  - 5           ii. Increment  $j$  by 1.
- 6           d. If  $x > i$ , set  $x = x-i$ .
- 7           e. Swap the  $i$ 'th and the  $x$ 'th elements in the array  $A$  (i.e.,  $\text{tmp} = A[x]$ ,  $A[x] = A[i]$ ,  $A[i]$
- 8            $= \text{tmp}$ .)
- 9           f. Decrement counter  $i$  by 1.

10 The resulting array  $A$  is the output permutation  $P$  i.e.,  $P(x)$  is the location of  $x$  in array  $A$ . For  
 11 example, if  $A$  reads 345201, then  $P(0)=4$ ,  $P(1)=5$ ,  $P(2)=3$ ,  $P(3)=0$ ,  $P(4)=1$ , and  $P(5)=2$ .

### 12 9.3.2.5.1.2 FL Hop Permutation Generation

13 FL Hop Permutation Generation is described in this section for both MultiCarrierOff and  
 14 MultiCarrierOn modes. In MultiCarrierOff mode, the hop permutation depends on several parameters  
 15 which are obtained from the Overhead Messages Protocol. In MultiCarrierOn mode, the hop  
 16 permutation on carrier  $c$ , where  $c$  is in  $\{0, 1, \dots, N_{\text{CARRIERS}} - 1\}$  depends on several parameters obtained  
 17 from the Overhead Messages Protocol for carrier  $c$ . These parameters may vary from carrier to  
 18 carrier.<sup>47</sup>

19 Space Division Multiple Access (SDMA) is supported on the Forward Link. There are a total of  
 20  $N_{\text{CARRIER\_SIZE}} Q_{\text{SDMA}}$  hop-ports on carrier  $c$ , which are mapped to the  $N_{\text{CARRIER\_SIZE}}$  subcarriers  
 21 corresponding to carrier  $c$ . Here  $Q_{\text{SDMA}}$  is equal to  $\text{FLNumSDMADimensions}$ , which is part of the  
 22 public data of the Overhead Messages Protocol for carrier  $c$ . The set of hop ports shall be divided into  
 23  $Q_{\text{SDMA}}$  groups, each of which has  $N_{\text{CARRIER\_SIZE}}$  hop-ports and shall be referred to as an SDMA sub-  
 24 tree<sup>48</sup>. The sub-trees shall be numbered  $\{0, 1, \dots, Q - 1\}$  where  $Q = Q_{\text{SDMA}}$ . The hop port with index  
 25  $p$ <sup>49</sup> shall belong to sub-tree with index  $q$ , where  $q = \lfloor p / N_{\text{CARRIER\_SIZE}} \rfloor$ . Note that hop-ports in  
 26 different SDMA sub-trees can get mapped to the same subcarrier.

27 The set of  $N_{\text{CARRIER\_SIZE}}$  hop-ports in each carrier in each SDMA sub-tree is divided into  $S$  subbands,  
 28 where  $S$  shall be determined by  $\text{FLNumSubbands}$ , which is part of the public data of the Overhead  
 29 Messages Protocol for carrier  $c$ . The subbands shall be numbered  $\{0, 1, \dots, S - 1\}$  and each subband

---

<sup>47</sup> A parameter that can vary from carrier to carrier should be indexed by the carrier index  $c$ . However, for convenience of notation, the index  $c$  is dropped and the parameter is assumed to correspond to the carrier of interest. For example,  $Q_{\text{SDMA}}$  should be interpreted as  $Q_{\text{SDMA}}(c)$  when generating the hop permutation for hop-ports in carrier  $c$ , and should be obtained from the Overhead Messages Protocol for carrier  $c$ .

<sup>48</sup> The term “sub-tree” is used since the  $Q_{\text{SDMA}} N_{\text{CARRIER\_SIZE}}$  hop-ports are part of a “channel tree” defined by the FTC MAC protocol.

<sup>49</sup> Here “hop-port  $p$ ” should be interpreted as “hop-port  $p$  on carrier  $c$ .” The phrase “on carrier  $c$ ” will be omitted for convenience of notation.

1 shall have  $N_{\text{SUBBAND}}$  hop-ports, where  $N_{\text{SUBBAND}} = N_{\text{CARRIER\_SIZE}} / S$ . The hop-port with index  $p$  shall  
 2 belong to subband with index  $s$ , where  $s = \lfloor (p \bmod N_{\text{CARRIER\_SIZE}}) / N_{\text{SUBBAND}} \rfloor$ .

3 Furthermore, as mentioned previously, the forward link may implement block hopping. For this  
 4 reason, the set of  $N_{\text{SUBBAND}}$  hop-ports in each subband is divided into a number of blocks<sup>50</sup>, each of  
 5 which has  $N_{\text{BLOCK}}$ <sup>51</sup> hop-ports. The blocks shall be numbered  $\{0, 1, \dots, B-1\}$  where  $B = N_{\text{SUBBAND}} /$   
 6  $N_{\text{BLOCK}}$ . The hop-port with index  $p$  shall belong to block with index  $b$ , where  
 7  $b = \lfloor (p \bmod N_{\text{SUBBAND}}) / N_{\text{BLOCK}} \rfloor$ . The index of the hop port  $p$  within the block which it belongs to  
 8 shall be denoted as  $r$ , where  $r = p \bmod N_{\text{BLOCK}}$ . Thus, there is a one-to-one correspondence between  
 9 hop-port  $p$  and the tuple  $(c, q, s, b, r)$ . For the rest of this document, the two notations are used  
 10 interchangeably and “hop-port  $(c, q, s, b, r)$ ” shall be used to refer to hop-port  $p$  on carrier  $c$ , where  
 11  $p = qN_{\text{CARRIER\_SIZE}} + sN_{\text{SUBBAND}} + bN_{\text{BLOCK}} + r$ .

12 The hop-ports within each subband shall be divided into two groups: non-guard hop-ports and guard  
 13 hop-ports. The guard hop-ports shall be mapped to either the guard subcarriers or the quasi-guard  
 14 subcarriers. The individual elements of this mapping are not specified since these hop-ports shall not  
 15 be modulated.

16 In BlockHopping mode, a hop-port  $(c, q, s, b, r)$  shall be mapped to a guard subcarrier or a quasi-  
 17 guard subcarrier either if:<sup>52</sup>

$$18 \quad b > B - 1 - \left\lfloor \frac{N_{\text{GUARD}} / N_{\text{BLOCK}}}{S} \right\rfloor$$

19 or if:

$$20 \quad b = B - 1 - \left\lfloor \frac{N_{\text{GUARD}} / N_{\text{BLOCK}}}{S} \right\rfloor \text{ and } \left| \frac{S}{2} - \frac{1}{4} - s \right| > \frac{S - [(N_{\text{GUARD}} / N_{\text{BLOCK}}) \bmod S]}{2}$$

21 The set of guard hop-ports in SymbolRateHopping mode shall be identical to the set of guard hop-  
 22 ports in BlockHopping mode. In other words, a hop port  $p$  on carrier  $c$  shall be mapped to a guard or a  
 23 quasi-guard subcarrier when its equivalent representation  $(c, q, s, b, r)$  using the value of  $N_{\text{BLOCK}}$  in  
 24 BlockHopping mode satisfies the above equations.<sup>53</sup>

---

<sup>50</sup> In SymbolRateHopping mode, contiguous hop-ports are not mapped to contiguous subcarriers. However, the term “block” can still be used if individual hop-ports and subcarriers are thought of as blocks of size 1.

<sup>51</sup> Note that the value of  $N_{\text{BLOCK}}$  used here corresponds to the Forward Link and this value may be different from the value of  $N_{\text{BLOCK}}$  used in section 9.4.1.3 which describes hop permutation generation for the Reverse Link.

<sup>52</sup> The idea behind these equations is that all subbands have approximately the same number of non-guard hop-ports. When  $(N_{\text{GUARD}} / N_{\text{BLOCK}})$  is a multiple of  $S$ , the first equation ensures that the highest numbered blocks in each subband are mapped to the guard subcarriers. In an asymmetric situation when  $(N_{\text{GUARD}} / N_{\text{BLOCK}})$  is not a multiple of  $S$ , the second equation ensures that the subbands most distant from the center of the carrier have one additional guard hop-port block.

<sup>53</sup> Note that the representation of a hop-port  $p$  as  $(c, q, s, b, r)$  is different for SymbolRateHopping and BlockHopping modes since the value of  $N_{\text{BLOCK}}$  is different. Strictly speaking, a hop-port  $p$  can be represented

1 The hop-ports that are not guard hop-ports shall be referred to as non-guard hop-ports. Note that hop-  
 2 ports in a block are either all guard hop-ports or all non-guard hop-ports. A hop-port block consisting  
 3 of only non-guard hop-ports shall be referred to as a non-guard hop-port block. The number of non-  
 4 guard hop-port blocks in subband  $s$  shall be denoted as  $B_{\text{NON-GUARD}}(s)$ . Note that  $B_{\text{NON-GUARD}}(s) \leq B$   
 5 and a hop-port  $(c, q, s, b, r)$  is non-guard if  $0 \leq b \leq B_{\text{NON-GUARD}}(s) - 1$ .

6 Furthermore, some non-guard hop-ports may be allocated to the ReservedFLBandwidth segment (as  
 7 described in 9.3.2.5.1.2.1) in any given interlace. The non-guard hop-ports not allocated to the  
 8 ReservedFLBandwidth segment in a given interlace shall be referred to as usable hop-ports<sup>54</sup> for that  
 9 interlace.

10 Let  $H^{\text{ijt}}(c, q, s, b, r)$  denote the subcarrier allocated to non-guard hop-port  $(c, q, s, b, r)$  in the OFDM  
 11 Symbol with index  $t$  in FL PHY Frame  $j$  in superframe  $i$ .  $H^{\text{ijt}}$  is referred to as the hop permutation and  
 12 shall be given by the following equation:

$$13 \quad H^{\text{ijt}}(c, q, s, b, r) = cN_{\text{CARRIER\_SIZE}} + \frac{N_{\text{GUARD}}}{2} + N_{\text{BLOCK}} H_{\text{GLOBAL}}^{\text{ijt}}(c, q, s, H_{\text{SECTOR}}^{\text{ijt}}(c, q, s, b)) + r$$

14 Here  $H_{\text{SECTOR}}^{\text{ijt}}(c, q, s, b)$  is a permutation of non-guard hop-port blocks  $b$  in the SDMA sub-tree  $q$ ,  
 15 carrier  $c$  and subband  $s$ . The generation of this permutation is described in 9.3.2.5.1.2.4.

16  $H_{\text{GLOBAL}}^{\text{ijt}}(c, q, s, b')$  is a permutation of all non-guard hop-port blocks in all subbands in carrier  $c$  and  
 17 SDMA sub-tree  $q$ . The generation of  $H^{\text{ijt}}$  is different for different values of  
 18 FLDiversityHoppingMode, which is part of the public data of the Overhead Messages Protocol for  
 19 carrier  $c$ . The generation of this permutation is described in 9.3.2.5.1.2.2 and 9.3.2.5.1.2.3.

### 20 9.3.2.5.1.2.1 ReservedFLBandwidth Segment

21 In some FL PHY Frames, a certain set of subbands may be reserved for other uses and therefore shall  
 22 not be modulated. This set of subbands is known as the ReservedFLBandwidth segment. The number  
 23 of subbands allocated to the ReservedFLBandwidth segment on a given carrier in a given FL PHY  
 24 Frame depends only on the interlace index of the FL PHY Frame. The interlace index of PHY Frame  $j$   
 25 in superframe  $i$  shall be equal to  $(iN_{\text{FL-PHY-FRAMES-IN-SUPERFRAME}} + j) \bmod N_{\text{FL-INTERLACES}}$ , where  $N_{\text{FL-}}$   
 26  $\text{INTERLACES}$  is the number of FL interlaces as specified by the lower MAC sublayer and  $N_{\text{FL-PHY-FRAMES-}}$   
 27  $\text{IN-SUPERFRAME}$  is equal to  $N_{\text{FDD,FLPHYFrames}}$  in FDD mode and  $N_{\text{TDD,FLPHYFrames}}$  in TDD mode. The set of  
 28 interlaces in which the ReservedFLBandwidth segment shall be present is specified by  
 29 FLReservedInterlaces, which is part of the public data of the Overhead Messages Protocol for carrier  
 30  $c$ . The number of subbands allocated to the ReservedFLBandwidth segment in these interlaces shall  
 31 be denoted as  $N_{\text{RESERVED-SUBBANDS}}$ , which shall be equal to NumFLReservedSubbands, which is part of  
 32 the public data of the Overhead Messages Protocol for carrier  $c$ .

---

as  $(c, q, s, b_{\text{SRH}}, r_{\text{SRH}})$  using the value of  $N_{\text{BLOCK}}$  in SymbolRateHopping mode and as  $(c, q, s, b_{\text{BH}}, r_{\text{BH}})$  using the  
 value of  $N_{\text{BLOCK}}$  in BlockHopping mode. For the purpose of computing guard hop-ports,  $(c, q, s, b_{\text{BH}}, r_{\text{BH}})$  is  
 used in both SymbolRateHopping and BlockHopping modes. For all other purposes, the notation  $(c, q, s, b, r)$   
 will refer to  $(c, q, s, b_{\text{SRH}}, r_{\text{SRH}})$  in SymbolRateHopping mode and to  $(c, q, s, b_{\text{BH}}, r_{\text{BH}})$  in BlockHopping mode.

<sup>54</sup> Note that “usable hop-ports” refer to hop-ports that can be used by the data segment. Some hop-ports which  
 are not usable are actually used by the ReservedFLBandwidth segment. Contrast this with the definition of  
 “usable subcarriers,” which are defined as subcarriers that can be used either by the data segment or  
 ReservedFLBandwidth segment.

1 A subband  $s$  in carrier  $c$  shall be allocated to the ReservedFLBandwidth segment if it satisfies the  
2 following condition

$$3 \quad s > S - 1 - N_{\text{RESERVED-SUBBANDS}}$$

4 Let  $S_{\text{MIN-RESERVED-SUBBAND}}$  be the subband with the lowest index allocated to the  
5 ReservedFLBandwidth segment in carrier  $c$ . (When  $N_{\text{RESERVED-SUBBANDS}} = 0$ ,  $S_{\text{MIN-RESERVED-SUBBAND}}$   
6 shall be set to  $S$ .) Thus all non-guard hop-ports in subbands  $\{S_{\text{MIN-RESERVED-SUBBAND}}, S_{\text{MIN-RESERVED-}}$   
7  $\text{SUBBAND} + 1, \dots, S - 1\}$  shall be allocated to the ReservedFLBandwidth segment. The number of hop-  
8 port-blocks allocated to the ReservedFLBandwidth segment in carrier  $c$  equals  $N_{\text{RESERVED-HOP-PORT-}}$   
9  $\text{BLOCKS}$ , where

$$10 \quad N_{\text{RESERVED-HOP-PORT-BLOCKS}} = \sum_{k=0}^{N_{\text{RESERVED-SUBBANDS}}-1} B_{\text{NON-GUARD}}(S_{\text{MIN-RESERVED-SUBBAND}} + k).$$

11 The contiguous set of  $N_{\text{BLOCK}}N_{\text{RESERVED-HOP-PORT-BLOCKS}}$  subcarriers indexed  $f_{\text{MIN-RESERVED}}$  to  $(f_{\text{MIN-}}$   
12  $\text{RESERVED} + N_{\text{BLOCK}}N_{\text{RESERVED-HOP-PORT-BLOCKS}} - 1)$  shall be allocated to the ReservedFLBandwidth  
13 segment, where

$$14 \quad f_{\text{MIN-RESERVED}} = cN_{\text{CARRIER\_SIZE}} + \frac{N_{\text{GUARD}}}{2} + N_{\text{BLOCK}} \sum_{k=0}^{S_{\text{MIN-RESERVED-SUBBAND}}-1} B_{\text{NON-GUARD}}(k)$$

15 These subcarriers shall not be modulated i.e., zero power shall be transmitted on them.

### 16 **9.3.2.5.1.2.2 Generation of $H_{\text{GLOBAL}}^{\text{ijt}}$ when FL Diversity Hopping Mode is off**

17 When FL Diversity Hopping Mode is off<sup>55</sup>, the permutation  $H_{\text{GLOBAL}}^{\text{ijt}}(c, q, s, b)$  shall be given by

$$18 \quad H_{\text{GLOBAL}}^{\text{ijt}}(c, q, s, b) = \left[ \sum_{P(k) < P(s)} B_{\text{NON-GUARD}}(P(k)) \right] + b$$

19 in both SymbolRateHopping and BlockHopping modes, where  $B_{\text{NON-GUARD}}(k)$  is the number of non-  
20 guard hop-port blocks in subband  $k$ . Here  $P(\cdot)$  is a permutation of size  $S_{\text{MIN-RESERVED-SUBBAND}}$  generated  
21 as follows:

- 22 1. Let  $[p_{11} p_{10} p_9 p_8 p_7 p_6 p_5 p_4 p_3 p_2 p_1 p_0]$  denote the 12-bit PilotPN of the sector, with  $p_{11}$  being  
23 the MSB and  $p_0$  being the LSB. Define an integer  $p$  to be equal to  $[p_{11} p_{10} p_9 p_8 p_7 p_6 p_5 p_4 p_3$   
24  $p_2 p_1 p_0]$  if FLIntraCellCommonHopping is off, and to be  $[p_{11} p_{10} p_9 p_8 0 0 0 p_4 p_3 p_2 p_1 p_0]$  if  
25 FLIntraCellCommonHopping is on. Here, FLIntraCellCommonHopping is part of the public  
26 data of the Overhead Messages Protocol for carrier  $c$ .
- 27 2. Set  $\text{TMP} = [(c * 4096 + p) * 2654435761] \bmod 2^{32}$ .

<sup>55</sup> When FL Diversity Hopping Mode is off, the subbands are not permuted. Therefore the hop permutation only permutes the different blocks within each subband.

3. Set  $SEED_{GLOBAL}$  to be the 20 LSBs of the bit-reversed value of TMP in a 32-bit representation, i.e.,  $SEED_{GLOBAL} = [\text{Bit-Reverse}(\text{TMP})] \bmod 2^{20}$ .
4. P is the permutation of size  $S_{MIN-RESERVED-SUBBAND}$  generated using the common permutation generation algorithm described in 9.3.2.5.1.1 with seed  $SEED_{GLOBAL}$ .

### 9.3.2.5.1.2.3 Generation of $H_{GLOBAL}^{jit}$ in BlockHopping mode when FL Diversity Hopping Mode is on

When FL Diversity Hopping Mode is on in BlockHopping mode,  $H_{GLOBAL}^{jit}(c, q, s, b)$  is a permutation of the non-guard-hop-port blocks in all subbands in carrier  $c$  of SDMA sub-tree  $q$ . The generation of  $H_{GLOBAL}^{jit}$  will be different for different values of FL Sector Hop Seed, which is part of the public data of the Overhead Messages Protocol on carrier  $c$ .  $H_{GLOBAL}^{jit}$  shall be generated as follows:

1. When FL Sector Hop Seed has a value not equal to 1111 (in binary notation), set  $TMP = [(\text{FLSectorHopSeed} * 4 * 64 * 4096 + c * 64 * 4096 + j * 4096 + (i \bmod 4096)) * 2654435761] \bmod 2^{32}$ .
2. When FL Sector Hop Seed is equal to 1111 (in binary notation), set  $TMP = [(\text{FLSectorHopSeed} * 4 * 64 * 4096 + c * 64 * 4096 + j * 4096 + P_{SECTOR}) * 2654435761] \bmod 2^{32}$ , where the 12-bit quantity  $P_{SECTOR}$  shall be computed as described in 9.3.2.5.1.2.5.
3. Set  $SEED_{GLOBAL}$  to be the 20 LSBs of the bit-reversed value of TMP in a 32-bit representation, i.e.,  $SEED_{GLOBAL} = [\text{Bit-Reverse}(\text{TMP})] \bmod 2^{20}$ .
4. Determine  $S_{MIN-RESERVED-SUBBAND}$  and  $N_{RESERVED-HOP-PORT-BLOCKS}$  for carrier  $c$  as described in 9.3.2.5.1.2.1. Determine  $B_{MIN-RESERVED-SUBBAND}$ , where
 
$$B_{MIN-RESERVED-SUBBAND} = \sum_{k=0}^{S_{MIN-RESERVED-SUBBAND}-1} B_{NON-GUARD}(k)$$
5. Generate a permutation  $\pi$  of size  $B_{MIN-RESERVED-SUBBAND}$  using the common permutation generation algorithm described in 9.3.2.5.1.1 with seed  $SEED_{GLOBAL}$ .
6.  $H_{GLOBAL}^{jit}(c, q, s, b) = P(\beta)^{56}$ , where  $\beta = b + \sum_{k=0}^{s-1} B_{NON-GUARD}(k)$  and

- a. If  $\beta \geq B_{MIN-RESERVED-SUBBAND}$  then  $P(\beta) = \beta$ .
- b. If  $\beta < B_{MIN-RESERVED-SUBBAND}$ , then  $P(\beta) = \pi(\beta)$

<sup>56</sup>  $P(\beta)$  first maps the hop port blocks allocated to the ReservedFLBandwidth segment to a contiguous set of subcarriers. The non-ReservedFLBandwidth hop port blocks are then assigned to the non-ReservedFLBandwidth subcarriers using a pseudo-random permutation  $\pi(\cdot)$

### 9.3.2.5.1.2.4 Generation of $H_{\text{GLOBAL}}^{\text{jit}}$ in SymbolRateHopping mode when FL DiversityHoppingMode is on

When FL DiversityHoppingMode is on in SymbolRateHopping mode,  $H_{\text{GLOBAL}}^{\text{jit}}(c, q, s, b)$  is a permutation of the non-guard-hop-port blocks in all subbands in carrier  $c$  of SDMA sub-tree  $q$ . The generation of  $H_{\text{GLOBAL}}^{\text{jit}}$  will be different for different values of FL SectorHopSeed, which is part of the public data of the Overhead Messages Protocol on carrier  $c$ .  $H_{\text{GLOBAL}}^{\text{jit}}$  shall be generated as follows:

1. When FL SectorHopSeed has a value not equal to 1111 (in binary notation), set  $\text{TMP} = [(\text{FLSectorHopSeed} * 4 * 64 * 4096 + c * 64 * 4096 + (i \bmod 4096)) * 2654435761] \bmod 2^{32}$ .
2. When FL SectorHopSeed is equal to 1111 (in binary notation), set  $\text{TMP} = [(\text{FLSectorHopSeed} * 4 * 64 * 4096 + c * 64 * 4096 + P_{\text{SECTOR}}) * 2654435761] \bmod 2^{32}$ , where the 12-bit quantity  $P_{\text{SECTOR}}$  shall be computed as described in 9.3.2.5.1.2.6.
3. Set  $\text{SEED}_{\text{GLOBAL}}$  to be the 20 LSBs of the bit-reversed value of TMP in a 32-bit representation, i.e.,  $\text{SEED}_{\text{GLOBAL}} = [\text{Bit-Reverse}(\text{TMP})] \bmod 2^{20}$ .
4. Determine  $S_{\text{MIN-RESERVED-SUBBAND}}$  and  $N_{\text{RESERVED-HOP-PORT-BLOCKS}}$  for carrier  $c$  as described in 9.3.2.5.1.2.1. Determine  $B_{\text{MIN-RESERVED-SUBBAND}}$ , where
 
$$B_{\text{MIN-RESERVED-SUBBAND}} = \sum_{k=0}^{S_{\text{MIN-RESERVED-SUBBAND}}-1} B_{\text{NON-GUARD}}(k)$$
5. Generate a permutation  $\pi_{\text{INITIAL}}$  of size  $B_{\text{MIN-RESERVED-SUBBAND}}$  using the common permutation generation algorithm described in 9.3.2.5.1.1 with seed  $\text{SEED}_{\text{GLOBAL}}$ .
6. Set  $k = N_{\text{FRAME,F}} * (j * N_{\text{FRAME,F}} / 2 + \lfloor t / 2 \rfloor)$ . Generate a permutation  $\pi^{57}$ , where  $\pi(x) = \pi_{\text{INITIAL}}(k + \pi_{\text{INITIAL}}(k+x))$  for all  $x$ . Here both additions are performed modulo  $B_{\text{MIN-RESERVED-SUBBAND}}$ .
7.  $H_{\text{GLOBAL}}^{\text{jit}}(c, q, s, b) = P(\beta)$ , where  $\beta = b + \sum_{k=0}^{s-1} B_{\text{NON-GUARD}}(k)$  and
  - a. If  $\beta \geq B_{\text{MIN-RESERVED-SUBBAND}}$  then  $P(\beta) = \beta$ .
  - b. If  $\beta < B_{\text{MIN-RESERVED-SUBBAND}}$ , then  $P(\beta) = \pi(\beta)$

<sup>57</sup> In SymbolRateHopping mode, a pseudo-random permutation  $\pi_{\text{INITIAL}}$  is generated at the beginning of each superframe. This permutation is then used to generate a different pseudo-random permutation  $\pi$  every two OFDM symbols.



### 9.3.2.5.1.2.5 Generation of $H_{\text{SECTOR}}^{\text{ijt}}$ in BlockHopping mode

$H_{\text{SECTOR}}^{\text{ijt}}$ (c, q, s, .) is a permutation of the non-guard hop-port blocks in subband s of carrier c of SDMA sub-tree q. The generation of  $H_{\text{SECTOR}}^{\text{ijt}}$  will be different for different values of FLIntraCellCommonHopping, which is part of the public data of the Overhead Messages Protocol on carrier c.<sup>58</sup>

The PilotPN of the sector of interest is XORed bitwise with the 12 LSBs of the superframe index i to obtain a 12-bit number  $[b_{11} b_{10} b_9 b_8 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0]$  denoted as  $P_{\text{off}}$ . The 12-bit number  $[b_{11} b_{10} b_9 b_8 i_7 i_6 i_5 b_4 b_3 b_2 b_1 b_0]$ , where  $i_7 i_6 i_5$  are the bits with indices 7,6 and 5 respectively in the superframe index i, is denoted as  $P_{\text{on}}$ .

The PilotPN is also XORed bitwise with the 12 LSBs of the superframe index (i-1) to obtain a 12-bit number  $[d_{11} d_{10} d_9 b_8 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0]$  denoted as  $P_{\text{off, MINUS}}$ . The 12-bit number  $[d_{11} d_{10} d_9 b_8 k_7 k_6 k_5 d_4 d_3 d_2 d_1 d_0]$  where  $k_7 k_6 k_5$  are the bits with indices 7,6 and 5 respectively in the superframe index (i-1), is denoted as  $P_{\text{on, MINUS}}$ .<sup>59</sup>

The permutation shall be generated as follows in FDD mode or in TDD mode when FLSectorHopSeed is not equal to 1110:

1. If FLIntraCellCommonHopping is off, set  $P_{\text{SECTOR}} = P_{\text{off}}$ . Otherwise, set  $P_{\text{SECTOR}} = P_{\text{on}}$ .
2. Find  $\text{TMP} = [(c * 16 * 64 * 4096 + s * 64 * 4096 + j * 4096 + P_{\text{SECTOR}}) * 2654435761] \bmod 2^{32}$ . Set  $\text{SEED}_{\text{SECTOR}}$  to be the 20 LSBs of the bit-reversed value of TMP in a 32-bit representation, i.e.,  $\text{SEED}_{\text{SECTOR}} = [\text{Bit-Reverse}(\text{TMP})] \bmod 2^{20}$ .
3.  $H_{\text{SECTOR}}^{\text{ijt}}$ (c, q, s, .) is the permutation of size  $B_{\text{NON-GUARD}}(s)$  generated using the common permutation generation algorithm described in 9.3.2.5.1.1 with seed  $\text{SEED}_{\text{SECTOR}}$ .

The permutation shall be generated as follows in TDD mode when FLSectorHopSeed is equal to 1110:<sup>60</sup>

1. Compute  $j_{\text{FL, BURST}} = \lfloor j / N_{\text{FL\_BURST}} \rfloor$ .

<sup>58</sup> When FLIntraCellCommonHopping is off, two sectors with different values of PilotPN have different hopping sequences. When FLIntraCellCommonHopping is on, sectors within the same cell have the same hopping sequences. For proper use of this mode, the operator should ensure that the PilotPNs of two sectors in the same cell differ only in the bits indexed 5,6, and 7.

<sup>59</sup> Note that  $P_{\text{off, MINUS}}$  and  $P_{\text{on, MINUS}}$  computed during superframe i are respectively equal to  $P_{\text{off}}$  and  $P_{\text{on}}$  computed during superframe (i-1).

<sup>60</sup> When RLSectorHopSeed and FLSectorHopSeed are both set to 1110, the FL hop permutations are “slaved” to the RL hop permutations for hop-ports on SDMA sub-tree 0 i.e., if a hop-port p is mapped to a subcarrier f on the RL in a “burst” of RL PHY Frames, then that hop port is mapped to the same subcarrier f on the FL in the subsequent burst of FL PHY Frames as well.

- 1           2. If  $j_{\text{FL, BURST}}$  is not equal to zero
- 2           a. If FLIntraCellCommonHopping is off, set  $P_{\text{SECTOR}} = P_{\text{off}}$ . Otherwise, set  $P_{\text{SECTOR}} =$   
3            $P_{\text{on}}$ .
- 4           b. Set  $j_{\text{RL, BURST}} = j_{\text{FL, BURST}} - 1$ .
- 5           3. If  $j_{\text{FL, BURST}}$  is equal to zero
- 6           a. If FLIntraCellCommonHopping is off, set  $P_{\text{SECTOR}} = P_{\text{off, MINUS}}$ . Otherwise, set  $P_{\text{SECTOR}} =$   
7            $P_{\text{on, MINUS}}$ .
- 8           b. Set  $j_{\text{RL, BURST}} = (N_{\text{TDD, FL PHYFrames}} / N_{\text{FL\_BURST}}) - 1$ . Here  $N_{\text{TDD, FL PHYFrames}}$  and  $N_{\text{FL\_BURST}}$   
9           are as defined in 9.3.2.1.
- 10          4. Find  $\text{TMP} = [(c * 16 * 64 * 4096 + s * 64 * 4096 + j_{\text{RL, BURST}} * 4096 + P_{\text{SECTOR}}) * 2654435761]$   
11           $\text{mod } 2^{32}$ . Set  $\text{SEED}_{\text{SECTOR}}$  to be the 20 LSBs of the bit-reversed value of TMP in a 32-  
12          bit representation, i.e.,  $\text{SEED}_{\text{SECTOR}} = [\text{Bit-Reverse}(\text{TMP})] \text{ mod } 2^{20}$ .
- 13          5.  $H_{\text{SECTOR}}^{\text{jit}}(c, q, s, .)$  is the permutation of size  $B_{\text{NON-GUARD}}(s)$  generated using the  
14          common permutation generation algorithm described in 9.3.2.5.1.1 with seed  
15           $\text{SEED}_{\text{SECTOR}}$ .

#### 16   **9.3.2.5.1.2.6 Generation of $H_{\text{SECTOR}}^{\text{jit}}$ in SymbolRateHopping mode**

17    $H_{\text{SECTOR}}^{\text{jit}}(c, q, s, .)$  is a permutation of the non-guard hop-port blocks in subband  $s$  of carrier  $c$  of  
18   SDMA sub-tree  $q$ . The generation of  $H_{\text{SECTOR}}^{\text{jit}}$  will be different for different values of  
19   FLIntraCellCommonHopping, which is part of the public data of the Overhead Messages Protocol on  
20   carrier  $c$ .

21   The PilotPN of the sector of interest is XORed bitwise with the 12 LSBs of the superframe index  $i$  to  
22   obtain a 12-bit number  $[b_{11} b_{10} b_9 b_8 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0]$  denoted as  $P_{\text{off}}$ . The 12-bit number  $[b_{11} b_{10} b_9$   
23    $b_8 i_7 i_6 i_5 b_4 b_3 b_2 b_1 b_0]$ , where  $i_7 i_6 i_5$  are the bits with indices 7,6 and 5 respectively in the superframe  
24   index  $i$ , is denoted as  $P_{\text{on}}$ . The permutation shall be generated as follows:

- 25          1. If FLIntraCellCommonHopping is off, set  $P_{\text{SECTOR}} = P_{\text{off}}$ . Otherwise, set  $P_{\text{SECTOR}} = P_{\text{on}}$ .
- 26          2. Find  $\text{TMP} = [(c * 16 * 64 * 4096 + s * 64 * 4096 + P_{\text{SECTOR}}) * 2654435761] \text{ mod } 2^{32}$ . Set  
27           $\text{SEED}_{\text{SECTOR}}$  to be the 20 LSBs of the bit-reversed value of TMP in a 32-bit  
28          representation, i.e.,  $\text{SEED}_{\text{SECTOR}} = [\text{Bit-Reverse}(\text{TMP})] \text{ mod } 2^{20}$ .
- 29          3. Generate a permutation  $\pi_{\text{INITIAL}}$  of size  $B_{\text{NON-GUARD}}(s)$  using the common permutation  
30          generation algorithm described in 9.3.2.5.1.1 with seed  $\text{SEED}_{\text{SECTOR}}$ .
- 31          4. Set  $k = N_{\text{FRAME, F}} * (j * N_{\text{FRAME, F}} / 2 + \lfloor t / 2 \rfloor)$ .  $H_{\text{SECTOR}}^{\text{jit}}(c, q, s, b) = \pi_{\text{INITIAL}}(k +$   
32           $\pi_{\text{INITIAL}}(k+b))$ . Here both additions are performed modulo  $B_{\text{NON-GUARD}}(s)$ .

### 9.3.2.5.2 Pilot channels

The pilot channels consist of the Common Pilot Channel (F-CPICH), the Auxiliary Pilot Channel (F-AuxPICH), and the Dedicated Pilot Channel (F-DPICH). A subcarrier occupied by the F-CPICH or the F-AuxPICH will be referred to as a pilot subcarrier. A hop-port occupied by the F-DPICH will be referred to as a DPICH hop-port.<sup>61</sup>

In SymbolRateHopping Mode, the F-CPICH shall be transmitted such that it can be used as an amplitude and phase reference for all the SISO channels. Moreover, the F-CPICH and F-AuxPICH shall be transmitted such that they can together be used as an amplitude and phase reference for demodulating F-DCH transmissions in MIMO and STTD modes.<sup>62</sup>

In BlockHopping Mode, all channels in a FL PHY Frame except the F-CPICH (namely the F-DPICH, the F-SSCH and the F-DCH) are transmitted using tile-antennas, where a tile-antenna is as defined in 9.3.2.3. These channels shall be transmitted in such a manner that the F-DPICH in each tile (where a tile is as defined in 9.3.2.5.1) can be used as an amplitude and phase reference for the modulation symbols (of the F-SSCH and the F-DCH) within that tile. Note that different precoding matrices may be used to construct the tile-antennas in different tiles, and hence the F-DPICH symbols in one tile should not be used as an amplitude or phase reference for demodulating modulation symbols in another tile. The F-CPICH modulation in BlockHopping mode is described using effective antennas instead of tile-antennas.

Modulation of all physical layers in SymbolRateHopping mode is described using effective antennas. Tile-antennas are not used in this mode.

#### 9.3.2.5.2.1 F-CPICH

##### 9.3.2.5.2.1.1 SymbolRateHopping mode

In SymbolRateHopping mode, the Common Pilot Channel (F-CPICH) shall be present in every OFDM symbol of every FL PHY Frame. The set of subcarriers occupied by the F-CPICH shall be spaced evenly in each carrier. This section describes the F-CPICH modulation procedure for a single carrier, and this procedure shall be repeated across all carriers if  $N_{\text{CARRIERS}} > 1$ .

The F-CPICH configuration is specified in terms of the following quantities, each of which can take a different value for each carrier if  $N_{\text{CARRIERS}} > 1$ .

1.  $N_p$ : This is the nominal number of pilot subcarriers in each OFDM symbol in the carrier of interest.<sup>63</sup> The value of  $N_p$  is given by the NumPilots parameter, which is part of the public data of the Overhead Messages Protocol.

---

<sup>61</sup> “Pilot subcarriers” and “DPICH hop-ports” are defined separately because F-CPICH and F-AuxPICH modulation is defined in the subcarrier domain, whereas F-DPICH modulation is defined in the hop-port domain.

<sup>62</sup> This requirement ensures that the AN uses similar processing on the pilot and data channels, i.e., if cyclic delay diversity is used for the data channels, it is also used for the pilot channels.

<sup>63</sup> The actual number of pilot subcarriers may be different due to the presence of guard subcarriers.

- 1           2.  $D_p$ : This is the spacing between neighboring pilot subcarriers, and is given by  
2            $N_{\text{CARRIER\_SIZE}}/N_p$ .
- 3           3.  $\text{Offset}_p$ : This is defined using the procedure described in 9.3.2.4.1.
- 4           4.  $N_t$ : This is the number of transmit antennas, given by the EffectiveNumAntennas  
5           parameter, which is part of the public data of the Overhead Messages Protocol.
- 6           5.  $N_{t,\text{CP}}$ : This is the number of transmit antennas to be multiplexed on the F-CPICH. This is  
7           given by the NumCommonPilotTransmitAntennas parameter, which is part of the public  
8           data of the Overhead Messages Protocol.

9 For each OFDM symbol in an FL PHY Frame, let  $j$  be the index of the OFDM symbol within the  
10 superframe (starting with index 0). The subcarrier with index  $i_{\text{sc}}$  in this OFDM symbol, where  $i_{\text{sc}}$  is  
11 such that this subcarrier lies in the carrier of interest, shall be occupied by the F-CPICH if it is a  
12 usable subcarrier, if it is not allocated to the ReservedFLBandwidth segment, and if one of the  
13 following two conditions is satisfied:

- 14           ■  $j$  is even and  $i_{\text{sc}} \bmod D_p = \text{Offset}_p \bmod D_p$ .
- 15           ■  $j$  is odd and  $i_{\text{sc}} \bmod D_p = (\text{Offset}_p + D_p/2) \bmod D_p$ .

16 Here, the ReservedFLBandwidth segment and the subcarriers allocated to it are as defined in  
17 9.3.2.5.1.2.1.

18 Each subcarrier occupied by the F-CPICH shall be modulated with the complex value  $(\sqrt{P}, 0)$  on  
19 only one of the  $N_t$  transmit antennas. No power shall be transmitted from the remaining antennas on  
20 these subcarriers. Here,  $P$  is the ratio of the power spectral density of the F-CPICH to the power  
21 spectral density of the second OFDM symbol in the F-ACQCH. This ratio is given by the  
22 CommonPilotPower parameter, which is public data of the Overhead Messages Protocol. The antenna  
23 index used to transmit an F-CPICH subcarrier in OFDM symbol  $j$  in the carrier of interest, denoted by  
24  $\text{ant}_j$ , is given by the following procedure :

- 25           1. If EnableCommonPilotStaggering is set to 0, then  $\text{ant}_j = j \bmod N_{t,\text{CP}}$ .
- 26           2. If EnableCommonPilotStaggering is set to 1, then  $\text{ant}_j = \lfloor j/2 \rfloor \bmod N_{t,\text{CP}}$ .

27 Here EnableCommonPilotStaggering is part of the public data of the Overhead Messages Protocol.

### 28 **9.3.2.5.2.1.2 BlockHopping mode**

29 In BlockHopping mode, the Common Pilot Channel (F-CPICH) shall be present in the OFDM  
30 symbols with indices 2 and 3 (where the indexing within the FL PHY Frame starts from 0) of some of  
31 the FL PHY Frames.

32 In FDD mode, an FL PHY Frame with index  $j$  within the superframe with index  $i$  shall carry the F-  
33 CPICH if the RL PHY Frame with index  $j+2$  within the superframe with index  $i$  contains a Control  
34 Segment.

In TDD mode, for each RL PHY Frame with index  $j$  within the superframe with index  $i$  containing a Control Segment, the FL PHY Frame with index  $j'$  within the superframe with index  $i'$  shall carry the F-CPICH, where  $i'$  and  $j'$  are computed as follows:

1. Define  $k = i * N_{\text{TDD,RLPHYFrames}} + j$ .
2. Define  $k' = \left( \left\lfloor k / N_{\text{RL\_BURST}} \right\rfloor + 1 \right) N_{\text{FL\_BURST}} - 2$ .
3. Define  $i' = \left\lfloor k' / N_{\text{TDD,FLPHYFrames}} \right\rfloor$  and  $j' = k' \bmod N_{\text{TDD,FLPHYFrames}}$ .

RL PHY Frame indexing and RL PHY Frame indices containing a control segment are as defined in 9.4.1.1 and 9.4.1.2.5.

This section describes the F-CPICH modulation procedure for a single carrier, and this procedure shall be repeated across all carriers if  $N_{\text{CARRIERS}} > 1$ .

The F-CPICH transmission from each transmit antenna consists of a set of evenly spaced subcarriers in each carrier, with a spacing of  $D_p = 16$ . Let  $N_t$  be the number of antennas in the carrier of interest.  $N_t$  is given by the EffectiveNumAntennas parameter, which is part of the public data of the Overhead Messages Protocol. ( $N_t$  can take a different value for each carrier.) The antenna with index  $k$ ,  $0 \leq k < N_t$ , is associated with two numbers  $a_k$  and  $b_k$ , which are as defined in columns 2 and 3 respectively of Table 100. A subcarrier with index  $i_{sc}$ , where  $i_{sc}$  is such that this subcarrier belongs to the carrier of interest, is used to transmit the F-CPICH from antenna  $k$  if it is a usable subcarrier, if it is not allocated to the ReservedFLBandwidth segment, and if it satisfies the following condition:

- In the OFDM symbol with index 2 in the FL PHY Frame,  $i_{sc}$  satisfies  $i_{sc} \bmod D_p = a_k$ .
- In the OFDM symbol with index 3 in the FL PHY Frame,  $i_{sc}$  satisfies  $i_{sc} \bmod D_p = b_k$ .

Here, the ReservedFLBandwidth segment and the subcarriers allocated to it are as defined in 9.3.2.5.1.2.1. No power is transmitted on the remaining antennas on this subcarrier.

Each subcarrier occupied by the F-CPICH shall be modulated on the appropriate antenna with the complex value  $(\sqrt{P}, 0)$ , where  $P$  is the ratio of the power spectral density of the F-CPICH to the power spectral density of the second OFDM symbol in the F-ACQCH. This ratio is given by the CommonPilotPower parameter, which is part of the public data of the Overhead Messages Protocol.

**Table 100 Values of the parameters  $a_k$  and  $b_k$ <sup>64</sup>**

Antenna Index	Frequency Interlace in OFDM Symbol 2 ( $a_k$ )	Frequency Interlace in OFDM Symbol 3 ( $b_k$ )
0	0	7
1	2	9
2	3	10

<sup>64</sup> The values of  $a_k$  and  $b_k$  are chosen so that the F-CPICH in BlockHopping mode does not collide with the F-DPICH.

Antenna Index	Frequency Interlace in OFDM Symbol 2 ( $a_k$ )	Frequency Interlace in OFDM Symbol 3 ( $b_k$ )
3	4	11
4	5	12
5	6	13
6	7	14
7	9	2

### 9.3.2.5.2.2 F-AuxPICH

The F-AuxPICH shall not be present in BlockHoppingMode. In SymbolRateHopping mode, the Auxiliary Pilot Channel (F-AuxPICH) shall be present in every OFDM symbol of every FL PHY Frame. The set of subcarriers occupied by the F-AuxPICH shall be spaced evenly in each carrier. This section describes the F-AuxPICH modulation procedure for a single carrier, and this procedure shall be repeated across all carriers if  $N_{\text{CARRIERS}} > 1$ .

The F-AuxPICH configuration is specified in terms of the following quantities, each of which can take a different value for each carrier if  $N_{\text{CARRIERS}} > 1$ .

1.  $N_p$ : This is the nominal number of pilot subcarriers in each OFDM symbol in the carrier or interest. The value of  $N_p$  is given by NumPilots, which is part of the public data of the Overhead Messages Protocol.
2.  $D_p$ : This is the spacing between neighboring pilot subcarriers, and is given by  $N_{\text{CARRIER\_SIZE}}/N_p$ .
3.  $\text{Offset}_p$ : This is defined using the procedure described in 9.3.2.4.1.
4.  $N_t$ : This is the number of transmit antennas, given by the EffectiveNumAntennas parameter, which is part of the public data of the Overhead Messages Protocol.
5.  $N_{t,\text{CP}}$ : This is the number of transmit antennas to be multiplexed on the F-CPICH. This is given by the NumCommonPilotTransmitAntennas parameter, which is part of the public data of the Overhead Messages Protocol.
6.  $N_{t,\text{Aux}}$ : This is given by  $N_t - N_{t,\text{CP}}$ . No subcarriers are occupied by the F-AuxPICH if  $N_{t,\text{Aux}} = 0$ .

For each OFDM symbol in a FL PHY Frame, let  $j$  be the index of the OFDM symbol within the superframe (starting with index 0). The subcarrier with index  $i_{\text{sc}}$  in this OFDM symbol, where  $i_{\text{sc}}$  is such that this subcarrier lies in the carrier of interest, shall be occupied by the F-AuxPICH if it is a usable subcarrier, if it is not allocated to the ReservedFLBandwidth segment, if  $N_{t,\text{Aux}} > 0$  for this carrier, and if one of the following two conditions is satisfied:

- $j$  is odd and  $i_{\text{sc}} \bmod D_p = \text{Offset}_p \bmod D_p$ .
- $j$  is even and  $i_{\text{sc}} \bmod D_p = (\text{Offset}_p + D_p/2) \bmod D_p$ .

1 Each subcarrier occupied by the F-AuxPICH shall be modulated with the complex value  $(\sqrt{P}, 0)$  on  
 2 only one of the  $N_t$  transmit antennas. No power shall be transmitted from the remaining antennas on  
 3 these subcarriers. Here,  $P$  is the ratio of the power spectral density of the F-AuxPICH to the power  
 4 spectral density of the second OFDM symbol in the F-ACQCH. This ratio is given by the  
 5 AuxPilotPower parameter, which is part of the public data of the Overhead Messages Protocol. The  
 6 antenna index used to transmit an F-AuxPICH subcarrier in OFDM symbol  $j$  in the carrier of interest,  
 7 denoted by  $\text{ant}_j$ , is given by the following procedure:

- 8 1. If EnableAuxPilotStaggering is set to 0, then  $\text{ant}_j = (j \bmod N_{t,\text{Aux}}) + N_{t,\text{CP}}$ .
- 9 2. If EnableAuxPilotStaggering is set to 1, then  $\text{ant}_j = (\lfloor j/2 \rfloor \bmod N_{t,\text{Aux}}) + N_{t,\text{CP}}$ .

10 Here, EnableAuxPilotStaggering is part of the public data of the Overhead Messages Protocol. The  
 11 ReservedFLBandwidth segment and the subcarriers allocated to it are as defined in 9.3.2.5.1.2.1.

### 12 9.3.2.5.2.3 F-DPICH

13 The Dedicated Pilot Channel (F-DPICH) shall be present in the BlockHopping mode only. The  
 14 modulation of this channel is described for a single tile, where a tile is as described in 9.3.2.5.1. The  
 15 modulation procedure shall then be repeated for each such tile in each FL PHY Frame. If multiple  
 16 tiles are mapped to the same set of subcarriers, the AN shall superimpose the F-DPICH waveforms  
 17 corresponding to these tiles. To elaborate, the AN shall compute the complex value to be transmitted  
 18 on each subcarrier on each effective transmit antenna corresponding to the different tiles. Note that  
 19 these complex values may be computed using a different precoding matrix for each data packet,  
 20 where a precoding matrix is as defined in 9.3.2.3. The AN shall then add the complex values assigned  
 21 to the same subcarrier corresponding to different data packets.

22 Each tile in each FL PHY Frame is assigned either to the F-SSCH to the F-DCH by the appropriate  
 23 MAC protocol (SS MAC in the case of SSCH and FTC MAC in the case of the F-DCH), or is not  
 24 assigned to any channel. The configuration of the F-DPICH is determined by which of these channels  
 25 this tile is assigned to and the configuration of this channel. If this tile is not assigned to either the F-  
 26 SSCH or the F-DCH, then no F-DPICH symbols shall be transmitted in this tile.

27 The F-DPICH configuration in each tile consists of the following parameters:

- 28 1. The number of tile-antennas  $n_t$  using which it is transmitted: This shall be equal to 1 if  
 29 this tile is assigned to the F-SSCH. If this tile is assigned to the F-DCH,  $n_t$  shall be equal  
 30 to the number of tile-antennas used for transmitting the F-DCH over this tile.
- 31 2. The energy per modulation symbol per tile-antenna: All the F-DPICH modulation  
 32 symbols in a given tile shall have the same energy from a given tile-antenna (this energy  
 33 may be different for different tile-antennas). The F-DPICH power density over a tile is  
 34 not specified if this tile is assigned to the F-SSCH. If the tile is assigned to the F-DCH,  
 35 the energy of each F-DPICH modulation symbol on a given tile-antenna shall be equal to  
 36 the power density used for the F-DCH on that tile for that tile-antenna.

- 1           3. F-DPICH format: The F-DPICH in a tile can have three different formats, labeled Format  
2           0, Format 1 and Format 2. Format 0 shall always be used for tiles assigned to the F-  
3           SSCH. For F-DCH tiles, the F-DPICH format to be used is determined by the FTC MAC  
4           protocol, with the following constraints:
- 5           a. Format 0 is defined only for tiles that are transmitted over up-to 3 antennas, i.e.,  $n_t \leq$   
6           3.
- 7           b. Format 1 is defined only for tiles that are transmitted over up-to 2 antennas, i.e.,  $n_t \leq$   
8           2.
- 9           4. *FLDPISectorOffset*: This is part of the public data of the Overhead Messages Protocol,  
10           and takes on integer values between 0 and 3. The value used shall correspond to the  
11           carrier containing the tile of interest.
- 12           5. *FLDPIUserOffset*: This is an integer that depends on the hop-ports contained in the tile of  
13           interest. Let  $p_{\min}$  be the smallest hop-port index contained in the tile of interest.  
14           *FLDPIUserOffset* is then given by  $\lfloor p_{\min} / N_{\text{CARRIER\_SIZE}} \rfloor$ .

15           In order to aid the description of the F-DPICH formats, the hop-ports in each tile are numbered from  
16           0 to  $N_{\text{BLOCK}}-1$  in increasing order, and the OFDM symbols in each tile from 0 to  $N_{\text{FRAME,F}}-1$  in  
17           increasing order.

#### 18           **9.3.2.5.2.3.1 F-DPICH format 0**

19           In this format, the F-DPICH occupies 18 modulation symbols in each tile in each FL PHY Frame. In  
20           this format, a hop-port with index  $i_{\text{hp}}$  of the OFDM symbol with index  $t$  (both measured within the  
21           tile) is occupied by the F-DPICH if  $i$  is in the set  $\{1,8,15\}$  and if  $t$  is in the set  $\{0,1,2,5,6,7\}$ . The set  
22           of hop-ports occupied by the F-DPICH for this format is illustrated in Figure 99.

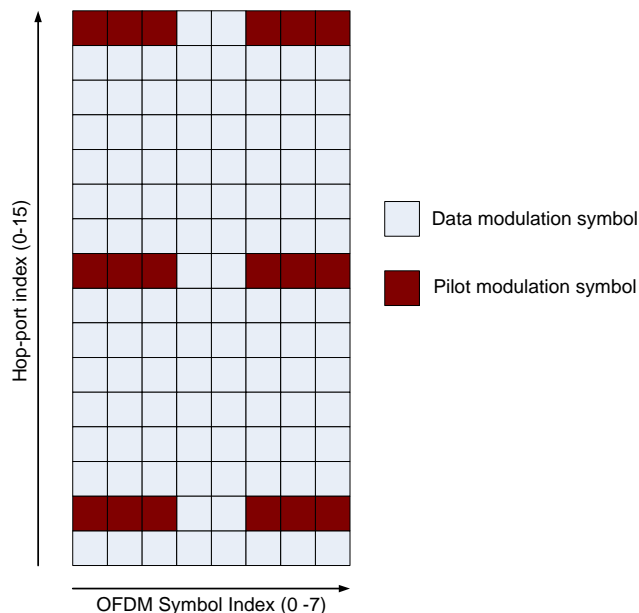
23           The complex value of the F-DPICH modulation symbol at this location on the tile-antenna with index  
24            $k$  is given by

$$25 \quad S_{i_{\text{hp}},t,k} = \sqrt{P} \exp\left(\frac{j2\pi}{3} (k + \text{FLDPISectorOffset} + \text{FLDPIUserOffset})t\right) \text{ if } t < 4, \text{ and}$$

$$26 \quad S_{i_{\text{hp}},t,k} = \sqrt{P} \exp\left(\frac{j2\pi}{3} (k + \text{FLDPISectorOffset} + \text{FLDPIUserOffset})(t-2)\right) \text{ if } t \geq 4.$$

27           where  $j$  denotes the complex number  $(0,1)$ , and  $P$  denotes the energy per modulation symbol on tile-  
28           antenna  $k$  used by the F-DPICH. Note that the value of this modulation symbol is the same for all  
29           values of  $i_{\text{hp}}$ .





1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13

**Figure 99 F-DPICH Format 0**

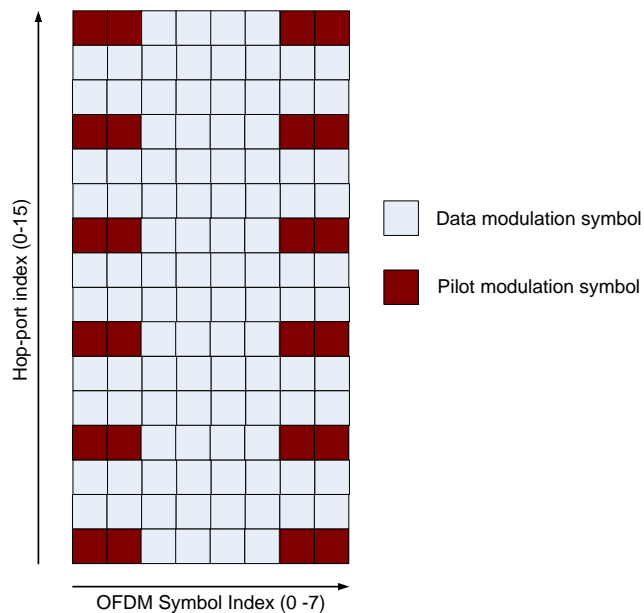
### 9.3.2.5.2.3.2 F-DPICH Format 1

In this format, the F-DPICH occupies 24 modulation symbols in each tile in each FL PHY Frame. In this format, a hop-port with index  $i_{hp}$  of the OFDM symbol with index  $t$  (both measured within the tile) is occupied by the F-DPICH if  $i_{hp}$  is in the set  $\{0,3,6,9,12,15\}$  and if  $t$  is in the set  $\{0,1,6,7\}$ . The set of hop-ports occupied by the F-DPICH for this format is illustrated in Figure 100.

The complex value of the F-DPICH modulation symbol at this location on the tile-antenna with index  $k$  is given by

$$S_{i_{hp},t,k} = \sqrt{P} \exp(j\pi(k + FLDPISectorOffset + FLDPIDUserOffset)t),$$

where  $j$  denotes the complex number  $(0,1)$ , and  $P$  denotes the energy per modulation symbol on tile-antenna  $k$  used by the F-DPICH. Note that the value of this modulation symbol is the same for all values of  $i_{hp}$ .



1  
2  
3 **Figure 100 F-DPICH Format 1**

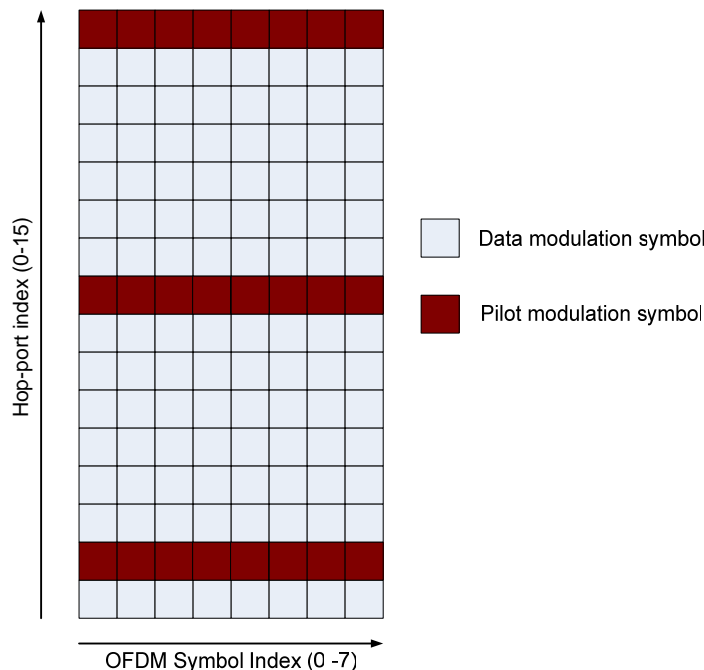
4 **9.3.2.5.2.3.3 F-DPICH Format 2**

5 In this format, the F-DPICH occupies 24 modulation symbols in each tile in each FL PHY Frame. In  
6 this format, a hop-port with index  $i_{hp}$  of the OFDM symbol with index  $t$  (both measured within the  
7 tile) is occupied by the F-DPICH if  $i_{hp}$  is in the set  $\{1,8,15\}$  and for all values of  $t$ . The set of hop-  
ports occupied by the F-DPICH for this format is illustrated in Figure 101.

8 The complex value of the F-DPICH modulation symbol at this location on the tile-antenna with index  
9  $k$  is given by

10 
$$S_{i_{hp},t,k} = \sqrt{P} \exp\left(\frac{j\pi}{2}(k + FLDPISectorOffset + FLDPIUserOffset)t\right),$$

11 where  $j$  denotes the complex number  $(0,1)$ , and  $P$  denotes the energy per modulation symbol on tile-  
12 antenna  $k$  used by the F-DPICH. Note that the value of this modulation symbol is the same for all  
13 values of  $i_{hp}$ .



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24

**Figure 101 F-DPICH Format 2**

### 9.3.2.5.3 F-SSCH

The Shared Signaling Channel (F-SSCH) shall be present in each FL PHY Frame and shall include four segments populated over hop-ports that are specified by the SS MAC Protocol. We denote by  $N_{\text{SSCH-HP}}$  the number of F-SSCH hop-ports. In the BlockHopping mode,  $N_{\text{SSCH-HP}}$  shall be an integer multiple of  $N_{\text{BLOCK}}$ . In MultiCarrierOn mode, there will be an F-SSCH (including all four segments) on every carrier. This section describes the F-SSCH modulation procedure for a single carrier.

#### 9.3.2.5.3.1 Encoded block segment

This segment shall occupy  $N_{\text{SSCH-S1}}$  modulation symbols and shall carry  $N_{\text{SSCH-M}}$  packets generated by the SS MAC Protocol. The  $i$ -th packet of size  $k_i$  (before CRC insertion) will be transmitted at power density  $P_1$  specified by the SS MAC Protocol. The number  $N_{\text{SSCH-S1}}$  is given by

$$N_{\text{SSCH-S1}} = \text{SSCHNumBlocks} * \text{SSCHModulationSymbolsPerBlock},$$

where  $\text{SSCHNumBlocks}$  is the maximum number of packets in the encoded block segment of F-SSCH ( $N_{\text{SSCH-M}} \leq \text{SSCHNumBlocks}$ ) and  $\text{SSCHModulationSymbolsPerBlock}$  is the number of modulation symbols per packet.  $\text{SSCHNumBlocks}$  and  $\text{SSCHModulationSymbolsPerBlock}$  are integers and are part of public data of the Overhead Messages Protocol. The  $i$ -th packet is appended with CRC, encoded, channel interleaved, repeated, data-scrambled, and modulated using the procedure described in 9.2 with  $\text{SSCHModulationSymbolsPerBlock}$  modulation symbols per packet. A MACID of 0 and a packet format index of 0 shall be used to generate the initial state of the data-scrambler. A CRC length of  $N_{\text{CRC,SSCH}}$  shall be used for this packet.

If  $i$ -th packet to be transmitted on the encoded block segment of F-SSCH is an Access Grant block, as specified by the SS MAC Protocol, the sequence of modulation symbols of this block shall be scrambled based on the 10-bit value of the AccessSequenceID which shall be defined by the AC

1 MAC Protocol. Let  $n$  be the number of modulation symbols corresponding to the  $i$ -th packet which is  
 2 an Access Grant block with the AccessSequenceID given by a 10-bit integer  $m$ .  $n$  is given by  $n =$   
 3  $SSCHModulationSymbolsPerBlock$ .

4 First, a sequence  $[B_{0,i}, \dots, B_{2n-1,i}]$  of  $2n$  bits shall be generated using 20-bit shift register which shall  
 5 have a generator polynomial  $h(D) = D^{20} + D^{17} + D^{12} + D^{10} + 1$ , i.e., the  $k$ -th output  $B_{k,i}$  of the register  
 6 shall satisfy  $B_{k,i} = B_{k-20,i} \oplus B_{k-17,i} \oplus B_{k-12,i} \oplus B_{k-10,i}$ . The initial state  $[B_{-1,i}, B_{-2,i}, \dots, B_{-20,i}]$  shall be given  
 7 by the  $[0\ 0\ 0\ 0\ f_5\ f_4\ f_3\ f_2\ f_1\ f_0\ a_9\ a_8\ a_7\ a_6\ a_5\ a_4\ a_3\ a_2\ a_1\ a_0]$ . Here  $[a_9\ a_8\ a_7\ a_6\ a_5\ a_4\ a_3\ a_2\ a_1\ a_0]$  is the 10-bit  
 8 AccessSequenceID, with LSB  $a_0$  and MSB  $a_9$  and  $[f_5\ f_4\ f_3\ f_2\ f_1\ f_0]$  is the 6-bit index of the FL PHY  
 9 Frame within the current superframe, with LSB  $f_0$  and MSB  $f_5$ . Next, the scrambling QPSK sequence  
 10  $[A_{0,i}, \dots, A_{n-1,i}]$  will be generated as follows:

$$A_{k,i} = e^{j\pi(2^*B_{2k,i} + B_{2k+1,i})/2}$$

12 The  $k$ -th modulation symbol of the  $i$ -th packet which is an Access Grant block shall be multiplied by  
 13 the QPSK symbol  $A_{k,i}$ ,  $0 \leq k < n$ .

14 Modulation symbols of all  $N_{SSCH-M}$  packets shall be scaled and concatenated in a sequence so that the  
 15 set of modulation symbols of the  $i$ -th packet (in the order of symbols generated by the modulator)  
 16 shall be scaled by  $\sqrt{P_i}$  and followed by the set of modulation symbols of the  $(i+1)$ -th packet (in the  
 17 order of symbols generated by the modulator) scaled by  $\sqrt{P_{i+1}}$ ,  $0 \leq i < N_{SSCH-M}$ . To this sequence, a  
 18 sequence of symbols with complex value  $(0,0)$  shall be appended so that the resulting sequence  
 19 denoted  $S_M$  is of length  $N_{SSCH-S1}$ .

### 20 9.3.2.5.3.2 Acknowledgement segment

21 This segment shall occupy  $N_{SSCH-S2} = 3N_{SSCH-ACK}$  modulation symbols where  $N_{SSCH-ACK}$  denotes the  
 22 length of the sequence of acknowledgement bits provided by the SS MAC Protocol. The SS MAC  
 23 Protocol shall provide a power density  $P_i$  ( $0 \leq i < N_{SSCH-ACK}$ ) and a valid MACID for every non-zero  
 24 bit. If the  $i$ -th bit of the sequence is non-zero, then the MACID corresponding to  $i$ -th bit shall be  
 25 converted into a scrambling sequence of three symbols  $(c_{i,0}\ c_{i,1}\ c_{i,2})$  as follows:

- 26 1. Obtain an 9-bit sequence  $(b_{i,0}\ b_{i,1}, \dots, b_{i,8})$  as 9 LSBs (starting with the LSB) of a binary  
 27 representation (from LSB to MSB) of the number  $(k * 2654435761) \bmod 2^{20}$  where  $k$  has  
 28 11-bit with representation  $[m_{10}\ m_9\ m_8\ m_7\ m_6\ m_5\ m_4\ m_3\ m_2\ m_1\ m_0]$  with LSB  $m_0$  and MSB  
 29  $m_{10}$  which is the 11-bit MACID. If the length of the MACID is less than 11 bits, then the  
 30 MACID shall be padded with 0's on the left (i.e., in the MSBs) to generate a 11-bit  
 31 MACID.
- 32 2. Map sets  $(b_{i,3j}\ b_{i,3j+1}, b_{i,3j+2})$  onto modulation symbols  $c_{i,j}$  according to Table 97 (8-PSK  
 33 mapping),  $0 \leq j < 3$ .

1 The  $i$ -th acknowledgement bit shall be mapped to a sequence  $(s_{i,0} \ s_{i,1} \ s_{i,2})$  of three modulation symbols  
2 as follows:

3 If the  $i$ -th acknowledgement bit is 0 then  $s_{i,0} = s_{i,1} = s_{i,2} = (0,0)$ ;

4 If the  $i$ -th acknowledgement bit is 1 then  $s_{i,j} = \sqrt{P_i} \ c_{i,j}$ ,  $0 \leq j \leq 2$ .

5 The resulting modulation symbols shall be concatenated into a sequence of modulation symbols

$$6 \quad S_{ACK} = (s_{0,0}, s_{0,1}, s_{0,2}, s_{1,0}, s_{1,1}, s_{1,2}, \dots, s_{i,0}, s_{i,1}, s_{i,2}, \dots, s_{N_{SSCH-ACK}-1,0}, s_{N_{SSCH-ACK}-1,1}, s_{N_{SSCH-ACK}-1,2})$$

7 of size  $N_{SSCH-S2}$ .

### 8 9.3.2.5.3.3 Power Control segment

9 This segment shall carry power control bits and CQI erasure indication bits for every access terminal  
10 with MACID  $k$  satisfying the following equation:

$$11 \quad k \bmod \text{FLPCReportInterval} = i \bmod \text{FLPCReportInterval}$$

12 where  $k$  is less or equal to  $\text{MACIDRange}$ ,  $i = s \cdot N_F + f$ , with  $N_F$  the number of FL PHY Frames in a  
13 superframe<sup>65</sup>,  $s$  the Superframe index and  $f$  the index of the current FL PHY Frame in the current  
14 Superframe starting from 0, and  $\text{FLPCReportInterval}$  is an integer and is part of the public data of the  
15 Overhead Messages Protocol.  $\text{MACIDRange}$  is an integer which is part of the public data of the  
16 Overhead Messages Protocol. The properly ordered sequence  $[a_0 \ a_1 \ a_2 \dots \ a_{N_{SSCH-S3}-1}]$  of power control  
17 bits and the corresponding sequence  $[b_0 \ b_1 \ b_2 \dots \ b_{N_{SSCH-S3}-1}]$  of CQI erasure indication bits are provided  
18 by the SS MAC protocol. Here  $N_{SSCH-S3} = \lfloor \text{MACIDRange} / \text{FLPCReportInterval} \rfloor + 1$  if  $(i \bmod$   
19  $\text{FLPCReportInterval}) < (\text{MACIDRange} \bmod \text{FLPCReportInterval})$  and  $N_{SSCH-S3} = \lfloor \text{MACIDRange} /$   
20  $\text{FLPCReportInterval} \rfloor$  otherwise. These sequences are converted into a sequence  $[c_0 \ c_1 \ c_2 \dots \ c_{N_{SSCH-S3}-1}]$   
21 of  $N_{SSCH-S3}$  modulation symbols as follows:

$$22 \quad c_k = \sqrt{\frac{P_{k,0}}{2}} (-1)^{a_k} + j \sqrt{\frac{P_{k,1}}{2}} (-1)^{b_k}, \quad 0 \leq k < N_{SSCH-S3}$$

23 where the power densities  $P_{k,0}$  and  $P_{k,1}$  assigned by the SS MAC Protocol.

24 This sequence is permuted to a sequence  $[s_0 \ s_1 \ s_2 \dots \ s_{N_{SSCH-S3}-1}]$  according to the following procedure:

- 25 1. Compute a 12-bit quantity  $B_{SECTOR}$  which is the bit-wise XOR of the 12-bit PilotPN and  
26 the 12 LSBs of the superframe index  $s$ .
- 27 2. Find  $\text{TMP} = [(7*64*4096 + f*4096 + B_{SECTOR}) * 2654435761] \bmod 2^{32}$ . Set  $\text{SEED}_{PCS}$  to  
28 be the 20 LSBs of the bit-reversed value of TMP in a 32-bit representation, i.e.,  $\text{SEED}_{PCS}$   
29  $= [\text{Bit-Reverse}(\text{TMP})] \bmod 2^{20}$ .

<sup>65</sup>  $N_F$  is equal to  $N_{FDD,FLPHYFrames}$  in FDD mode and is equal to  $N_{TDD,FLPHYFrames}$  in TDD mode.

3. Generate a permutation  $H$  of size  $N_{SSCH-S3}$  using the common permutation generation algorithm described in 9.3.2.5.1.1 with seed  $SEED_{PCS}$ .

4.  $s_{H(k)} = c_k$  for all  $k$  satisfying  $0 \leq k < N_{SSCH-S3}$ .

The sequence  $S_{PC}$  of these modulation symbols can be written as follows:

$$S_{PC} = (s_0, s_1, \dots, s_k, \dots, s_{N_{SSCH-S3}-1})$$

#### 9.3.2.5.3.4 Fast OSI Segment

The Fast OSI segment is present in each FL PHY Frame if FastOSIEnabled is set to 1. The Fast OSI segment is not present if FastOSIEnabled is set to 0. Here, FastOSIEnabled is a binary variable and is part of the public data of the Overhead Messages Protocol.

If enabled, the Fast OSI Segment carries a single three-state quantity  $x$  in each FL PHYFrame. This quantity takes the values 0, 1 and 2, and is generated by the SS MAC Protocol. The quantity  $x$  shall be mapped to complex modulation symbol  $s$  as follows:

- If  $x = 0$ , then  $s$  shall take the value  $(1,0)$ .
- If  $x = 1$ , then  $s$  shall take the value  $(0,1)$ .
- If  $x = 2$ , then  $s$  shall take the value  $(-1,0)$ .

The modulation symbol  $s$  shall be scaled to the appropriate power  $P$  determined by the SS MAC protocol and repeated  $N_{FastOSI}$  times to get the sequence  $S_{FastOSI}$ .  $S_{FastOSI}$  is therefore given by the  $N_{FastOSI}$  length sequence

$$S_{FastOSI} = (\sqrt{P}s, \sqrt{P}s, \dots, \sqrt{P}s).$$

If the Fast OSI Segment is not enabled, then  $S_{FastOSI}$  shall be set to the empty sequence.

#### 9.3.2.5.3.5 SSCH Modulation

The sequences  $S_M$ ,  $S_{ACK}$ ,  $S_{PC}$ , and  $S_{FastOSI}$  shall be concatenated into a single sequence

$$S_{SSCH} = (S_M, S_{ACK}, S_{PC}, S_{FastOSI})$$

of scaled modulation symbols of size  $N_{SSCH}$ .  $N_{SSCH}$  is given by  $N_{SSCH-S1} + N_{SSCH-S2} + N_{SSCH-S3} + N_{FastOSI}$  if the Fast OSI segment is enabled, and is given by  $N_{SSCH-S1} + N_{SSCH-S2} + N_{SSCH-S3}$  otherwise. The sequence  $S_{SSCH}$  shall be mapped onto the  $N_{SSCH-HP}$  hop-ports assigned to F-SSCH according to the following procedure:

1. Let  $p_0, p_1, \dots, p_{N_{SSCH-HP}-1}$  be the set of  $N_{SSCH-HP}$  hop-ports specified by the SS MAC Protocol. In the BlockHopping mode: generate a sequence  $q_0, q_1, \dots, q_{N_{SSCH-HP}-1}$  so that  $q_k = p_{n(k)}$ , where  $n(k) = (k \bmod M) \cdot N_{BLOCK} + \lfloor k / M \rfloor$  and  $M = N_{SSCH-HP} / N_{BLOCK}$ . In the SymbolRateHopping mode:  $q_k = p_k$ .

- 1           2. Initialize a port counter  $i$  to 0, an OFDM symbol counter  $n$  to 0, and an F-SSCH sequence  
2           index<sup>66</sup>  $k$  to 0.
- 3           3. Define  $m = (n - (i \bmod M) \lceil N_{FRAME,F} / M \rceil) \bmod N_{FRAME,F}$ .<sup>67</sup> Let  $n_{sc}$  be the subcarrier  
4           index corresponding to hop-port  $q_i$  for the  $m$ -th OFDM symbol in the FL PHY Frame. If  
5           the subcarrier with index  $n_{sc}$  is not a pilot subcarrier and if  $q_i$  is not a DPICH hop-port,  
6           then
- 7           a. modulate this subcarrier with the complex value of the  $k$ -th entry of  $S_M$  if  $k < N_{SSCH}$   
8           and modulate this subcarrier with the complex value (0,0) otherwise. The modulation  
9           shall be done on the antenna with index 0 in SymbolRateHopping mode, and on the  
10          tile-antenna with index 0 in BlockHopping mode.
- 11          b. increment  $k$ .
- 12          4. Increment  $i$ .
- 13          5. If  $i \bmod M = 0$  then
- 14          a. if  $n < N_{FRAME,F}$  then reduce  $i$  by  $M$  and increment  $n$ ;
- 15          b. otherwise set  $n$  to 0.
- 16          6. If  $i = N_{SSCH-HP}$ , then stop. Otherwise, repeat steps 3 through 6.

#### 17   **9.3.2.5.4 F-DCH**

18   The Data Channel (F-DCH) shall be present in each FL PHY Frame. The F-DCH consists of one or  
19   more packets with different target access terminals, as well as erasure sequences. Each data packet as  
20   well as erasure sequence transmission spans one or more FL PHY Frames. The set of FL PHY  
21   Frames on which this packet is transmitted is determined by the FTC MAC Protocol. Each data  
22   packet and erasure sequence is also assigned a set of hop-ports in each FL PHY Frame of  
23   transmission by the FTC MAC Protocol. Note that these hop-ports may span more than one carrier.  
24   Each data packet is further associated with a packet format index, which is also assigned by the FTC  
25   MAC Protocol.

26   If some of the hop-ports assigned to two or more data packets or erasure sequences are mapped to the  
27   same subcarriers by the hopping sequence, then the AN shall superimpose the waveforms of these  
28   data packets. To elaborate, the AN shall compute the complex value to be transmitted on each  
29   subcarrier on each effective transmit antenna corresponding to the different data packets. Note that  
30   these complex values may be computed using a different precoding matrix for each data packet,  
31   where a precoding matrix is as defined in 9.3.2.3. The AN shall then add the complex values assigned  
32   to the same subcarrier corresponding to different data packets.

---

<sup>66</sup> F-SSCH sequence index refers to the current position within the sequence  $S_{SSCH}$ .

<sup>67</sup> This equation provides a different offset for each tile in BlockHopping mode. This ensures that neighboring modulation symbols are mapped to different OFDM symbols.

1 In the following, power shall not be transmitted on an antenna (in SymbolRateHopping mode) or a  
2 tile-antenna (in BlockHopping mode) unless specifically specified.

### 3 9.3.2.5.4.1 SISO mode

4 In this mode, each F-DCH packet is generated by the FTC MAC Protocol, and is split, appended with  
5 CRC, encoded, channel interleaved, repeated, data-scrambled and modulated according to the  
6 procedure described in 9.2. A CRC length of  $N_{CRC,Data}$  is used for this packet. The MACID of the  
7 target access terminal, and the packet format index assigned to this packet, shall be used to generate  
8 the initial state of the data-scrambler described in 9.2.5. The size of the input packet generated by the  
9 FTC MAC Protocol shall be equal to  $8 \lfloor \rho n_0 N_f / 8 \rfloor - N_{CRC,Data}$ , where  $\rho$  denotes the spectral  
10 efficiency at the first transmission corresponding to the packet format of the packet (defined by the  
11 FTC MAC Protocol),  $n_0$  denotes the number of usable hop-ports assigned to this packet in the first FL  
12 PHY Frame of transmission, and  $N_f$  is equal to  $6N_{FRAME,F}$  if this packet is part of an extended duration  
13 transmission and is equal to  $N_{FRAME,F}$  otherwise. The FTC MAC protocol determines whether or not a  
14 packet is part of an extended duration transmission. Here, a usable hop-port is as defined in  
15 9.3.2.5.1.2. This packet shall be modulated on to the hop-ports assigned to this packet according to  
16 the following procedure:

- 17 1. Initialize a port counter  $i$  to 0, a frame counter  $f$  to 0, and an OFDM symbol counter  $j$   
18 to 0.
- 19 2. Arrange the set of usable hop-ports assigned to this packet in the  $f$ 'th PHY Frame of  
20 transmission in increasing order, where the ordering of hop-ports is as defined in  
21 9.3.2.2.5. Let the resulting sequence be denoted by  $(k_0, p_0), (k_1, p_1), \dots, (k_{n-1}, p_{n-1})$ , where  $n$   
22 is the total number of hop-ports assigned to this packet in the  $f$ 'th PHY Frame of  
23 transmission. The notation for a hop-port is as in 9.3.2.2.5, i.e.,  $k_i$  denotes the  
24 CarrierIndex and  $p_i$  denotes the hop-port index in that carrier.
- 25 3. Let  $n_{sc}$  be the subcarrier index corresponding to the hop-port  $(k_i, p_i)$  in the  $j$ 'th OFDM  
26 symbol in the  $f$ 'th PHY Frame of transmission. Let  $q$  be the modulation order to be used  
27 for the  $f$ 'th PHY Frame of transmission, which is a function of the packet format. If  $n_{sc}$  is  
28 not a pilot subcarrier and if  $(k_i, p_i)$  is not a DPICH hop-port, then a modulation symbol  $s$   
29 with modulation order  $q$  is generated by the modulator according to the procedure  
30 described in 9.2.6. This modulation symbol shall be modulated with energy  $P$  on hop-port  
31  $(k_i, p_i)$ , i.e., the value of the corresponding subcarrier shall be  $\sqrt{P} s$ , where  $P$  is the power  
32 density used for this assignment in the  $f$ 'th PHY Frame of transmission. The modulation  
33 shall be done on the antenna with index 0 in SymbolRateHopping mode, and on the tile-  
34 antenna with index 0 in BlockHopping mode. In SymbolRateHopping mode, the power  
35 density  $P$  is constant over all hop-ports assigned to this packet, while in BlockHopping  
36 mode, a different power density  $P$  may be used for each tile in the assignment.  
37 Determining the value of  $P$  is out of the scope of this specification.
- 38 4. Increment  $i$ . If  $i = n$ , increment  $j$  and set  $i = 0$ .
- 39 5. If  $j = N_{FRAME,F}$ , increment  $f$  and set  $j = 0$ .
- 40 6. If the last PHY Frame of transmission has been completed (as determined by the FTC  
41 MAC Protocol), then stop. Else repeat steps 2 through 6.



### 9.3.2.5.4.2 STTD mode

In the STTD mode, each F-DCH packet is generated by the FTC MAC Protocol, and is split, appended with CRC, encoded, channel interleaved, repeated, data-scrambled and modulated according to the procedure described in 9.2. A CRC length of  $N_{CRC,Data}$  is used for this packet. The MACID of the target access terminal, and the packet format index assigned to this packet, shall be used to generate the initial state of the data-scrambler described in 9.2.5. The size of the input packet generated by the FTC MAC Protocol shall be equal to  $8 \lfloor \rho n_0 N_f / 8 \rfloor - N_{CRC,Data}$ , where  $\rho$  denotes the spectral efficiency at the first transmission corresponding to the packet format of the packet (defined by the FTC MAC Protocol),  $n_0$  denotes the number of usable hop-ports assigned to this packet in the first FL PHY Frame of transmission, and  $N_f$  is equal to  $6N_{FRAME,F}$  if this packet is part of an extended duration transmission and is equal to  $N_{FRAME,F}$  otherwise. The FTC MAC protocol determines whether or not a packet is part of an extended duration transmission. Here, a usable hop-port is as defined in 9.3.2.5.1.2. This packet shall be modulated on to the hop-ports assigned to this packet according to the following procedure:

1. Initialize a port counter  $i$  to 0, a frame counter  $f$  to 0, and an OFDM symbol counter  $j$  to 0.
2. Arrange the set of usable hop-ports assigned to this packet in the  $f$ 'th PHY Frame of transmission in increasing order, where the ordering of hop-ports is as defined in 9.3.2.2.5. Let the resulting sequence be denoted by  $(k_0, p_0), (k_1, p_1), \dots, (k_{n-1}, p_{n-1})$ , where  $n$  is the total number of usable hop-ports assigned to this packet in the  $f$ 'th PHY Frame of transmission. The notation for a hop-port is as in 9.3.2.2.5, i.e.,  $k_i$  denotes the CarrierIndex and  $p_i$  denotes the hop-port index in that carrier.
3. Let  $n_{sc}$  be the subcarrier index corresponding to the hop-port  $(k_i, p_i)$  in the  $j$ 'th OFDM symbol in the  $f$ 'th PHY Frame of transmission. Let  $q$  be the modulation order to be used for the  $f$ 'th PHY Frame of transmission, which is a function of the packet format. If  $n_{sc}$  is not a pilot subcarrier and if  $(k_i, p_i)$  is not a DPICH hop-port, then a modulation symbol  $s$  with modulation order  $q$  is generated by the modulator according to the procedure described in 9.2.6. Label this modulation symbol  $s_{i,j}$ .
4. Increment  $i$ . If  $i = n$ , increment  $j$  and set  $i = 0$ .
5. If  $j = N_{FRAME,F}$ , set  $j = 0$ . Else repeat steps 3 through 5.
6. Collect the set of hop-ports in the FL PHY Frame into a list of pairs as follows:<sup>68</sup>
  - a. Start with an empty list.
  - b. For each value of  $i'$ ,  $0 \leq i' < n$ , and for each even value of  $j'$ ,  $0 \leq j' < N_{FRAME,F}$ , add the pair  $((i', j'), (i', j'+1))$  to the list if the hop-port  $(k_{i'}, p_{i'})$  is a DPICH hop-port in neither OFDM symbol  $j'$  nor OFDM symbol  $j'+1$  OFDM symbol in the FL PHY Frame.

<sup>68</sup> This list is grouping hop-ports that are adjacent to each other into pairs. The different conditions are necessary to account for F-DPICH format 0, which uses up an odd number of adjacent hop-ports.

1 c. For each value of  $i'$ ,  $0 \leq i' < n$  and for each even value of  $j'$ ,  $0 \leq j' < N_{\text{FRAME},F}$ , add the  
 2 pair  $((i', j'+1), (i', j'+2))$  to the list if the hop-port  $(k_i, p_i)$  is a DPICH hop-port in  
 3 OFDM symbol  $j'$  in the FL PHY Frame but is not a DPICH hop-port in OFDM  
 4 symbol  $j'+1$  in the FL PHY Frame.

5 7. In the following, let  $P$  be the power density to be used for this packet in the  $f$ 'th frame of  
 6 transmission. In SymbolRateHopping mode, the power density  $P$  is constant over all hop-  
 7 ports assigned to this packet, while in BlockHopping mode, a different power density  $P$   
 8 may be used for each tile in the assignment. Determining the value of  $P$  is out of the  
 9 scope of this specification.

10 Each pair in the list formed in step 6 is of the form  $((i', j'), (i', j'+1))$ . For each such pair in  
 11 the list, do the following:<sup>69</sup>

- 12 a. If the hop-port  $(k_i, p_i)$  does not correspond to a pilot subcarrier in either of OFDM  
 13 symbols  $j'$  and  $j'+1$  in the FL PHY Frame, let  $s_{i',j'}$  and  $s_{i',j'+1}$  denote the modulation  
 14 symbols allocated in step 3 to this hop-port in OFDM symbols  $j'$  and  $j'+1$   
 15 respectively. For this case, the following steps shall be carried out:
- 16 i. In SymbolRateHopping mode,  $\sqrt{P/2}s_{i',j'}$  shall be transmitted on antenna index  
 17 0 on hop-port  $(k_i, p_i)$  of OFDM symbol  $j'$ . In BlockHopping mode,  $\sqrt{P/2}s_{i',j'}$   
 18 shall be transmitted on tile-antenna index 0 on hop-port  $(k_i, p_i)$  of OFDM symbol  
 19  $j'$ .
- 20 ii. In SymbolRateHopping mode,  $\sqrt{P/2}s_{i',j'+1}$  shall be transmitted on antenna  
 21 index 1 on hop-port  $(k_i, p_i)$  of OFDM symbol  $j'$ . In BlockHopping mode,  
 22  $\sqrt{P/2}s_{i',j'+1}$  shall be transmitted on tile-antenna index 1 on hop-port  $(k_i, p_i)$  of  
 23 OFDM symbol  $j'$ .
- 24 iii. In SymbolRateHopping mode,  $-\sqrt{P/2}s_{i',j'+1}^*$  shall be transmitted on antenna  
 25 index 0 of hop-port  $(k_i, p_i)$  of OFDM symbol  $j'+1$ . In BlockHopping mode,  
 26  $-\sqrt{P/2}s_{i',j'+1}^*$  shall be transmitted on tile-antenna index 0 of hop-port  $(k_i, p_i)$   
 27 of OFDM symbol  $j'+1$ .<sup>70</sup>
- 28 iv. In SymbolRateHopping mode,  $\sqrt{P/2}s_{i',j'}^*$  shall be transmitted on the antenna  
 29 index 1 of hop-port  $(k_i, p_i)$  of OFDM symbol  $j'+1$ . In BlockHopping mode,  
 30  $\sqrt{P/2}s_{i',j'}^*$  shall be transmitted on the tile-antenna index 1 of hop-port  $(k_i, p_i)$   
 31 of OFDM symbol  $j'+1$ .

---

<sup>69</sup> The different hop-port pairs in the list formed in step 6 now undergo an STTD transformation, which maps two input symbols into two output symbols. However, in some cases, one of the hop-port pairs may map to a pilot (F-CPICH or F-AuxPICH) subcarrier. In this case, that pair of hop-ports does not undergo the STTD transformation.

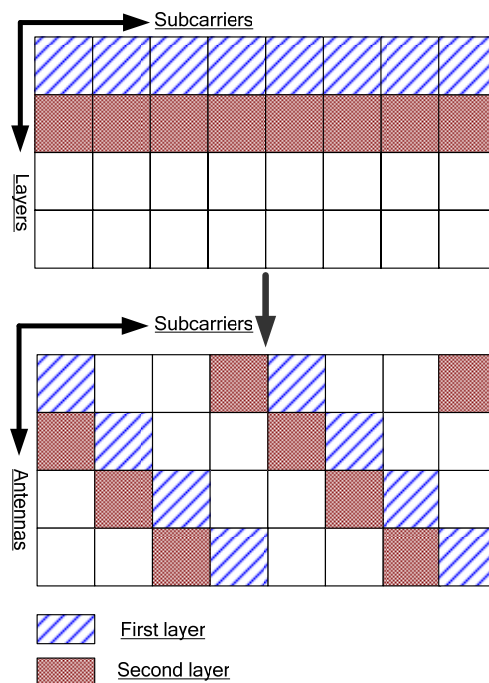
<sup>70</sup> Here,  $s^*$  denotes the complex conjugate of symbol  $s$ .

- 1           b. If the hop-port  $(k_i, p_i)$  corresponds to a pilot subcarrier in OFDM symbols  $j'$  but not  
 2           in OFDM symbol  $j'+1$  in the FL PHY Frame, let  $s_{i,j'+1}$  denote the modulation  
 3           symbols allocated in step 3 to this hop-port in OFDM symbol  $j'+1$ . In  
 4           SymbolRateHopping mode,  $\sqrt{P}s_{i,j'+1}$  shall be transmitted on antenna index 0 of  
 5           hop-port  $(k_i, p_i)$  of OFDM symbol  $j'+1$ . In BlockHopping mode,  $\sqrt{P}s_{i,j'+1}$  shall be  
 6           transmitted on tile-antenna index 0 of hop-port  $(k_i, p_i)$  of OFDM symbol  $j'+1$ . The F-  
 7           DCH shall not modulate hop-port  $(k_i, p_i)$  in OFDM symbol  $j'$ .
- 8           c. If the hop-port  $(k_i, p_i)$  corresponds to a pilot subcarrier in OFDM symbols  $j'+1$  but  
 9           not in OFDM symbol  $j'$  in the FL PHY Frame, let  $s_{i,j'}$  denote the modulation symbols  
 10           allocated in step 3 to this hop-port in OFDM symbols  $j'$ . In SymbolRateHopping  
 11           mode,  $\sqrt{P}s_{i,j'}$  shall be transmitted on antenna index 0 of hop-port  $(k_i, p_i)$  of OFDM  
 12           symbol  $j'$ . In BlockHopping mode,  $\sqrt{P}s_{i,j'}$  shall be transmitted on tile-antenna index  
 13           0 of hop-port  $(k_i, p_i)$  of OFDM symbol  $j'$ . The F-DCH shall not modulate hop-port  
 14            $(k_i, p_i)$  in OFDM symbol  $j'+1$ .
- 15           8. Increment  $f$ . If the last PHY Frame of transmission has been completed (as determined by  
 16           the FTC MAC Protocol), then stop. Else repeat steps 2 through 8.

#### 17   **9.3.2.5.4.3 MIMO MCW mode**

18   In the MCW mode, multiple codewords with, in general, different packet formats are transmitted  
 19   simultaneously on the same set of hop-ports. Each codeword (as well as the associated modulation  
 20   symbols) is denoted by a data layer. The modulation symbols corresponding to these codewords are  
 21   transmitted on the different antennas via a scheme called Cyclic Spatial Multiplexing. This scheme is  
 22   illustrated in Figure 102 for the case of  $N_t = 4$  transmit antennas,  $N_l = 2$  data layers. The figure shows  
 23   transmission on 8 consecutive hop-ports. The antennas are indexed from 0 to 3 while the hop-ports  
 24   are indexed from 0 to 7. The  $N_l$  layers are mapped to the space frequency domain in a cyclic fashion  
 25   such that each layer is transmitted from all antennas, i.e., the first modulation symbol of the first layer  
 26   is transmitted on the (antenna, hop-port) pair (0,0), the second modulation symbol on (1,1), third on  
 27   (2,2), and so on. Similarly, the first modulation symbol of the second layer is transmitted on (1,0), the  
 28   second on (2,1), the third on (3,2), and so on. The total assigned power is initially distributed equally  
 29   among the different layers. This power shall be split equally among the different antennas in  
 30   SymbolRateHopping mode. Different tile-antennas may have different powers in BlockHopping  
 31   mode.

32   The MIMO MCW mode supports early termination for each layer, i.e., after each FL PHY Frame.  
 33   The access terminal may acknowledge the first  $N_{\text{dec}}$  layers, where  $N_{\text{dec}}$  ranges from 0 to  $N_l$ . If the first  
 34    $N_{\text{dec}}$  layers have been acknowledged, then these layers are no longer transmitted and the total assigned  
 35   power is distributed equally among the remaining layers.



**Figure 102 Cyclic spatial multiplexing**

### 9.3.2.5.4.3.1 Transmitter structure

Let  $N_l$  denote the number of layers in the MCW transmission, as determined by the FTC MAC protocol. Let  $n_t$  denote the number of antennas used for the MCW transmission in SymbolRateHopping mode, and the number of tile-antennas used in BlockHopping mode. In SymbolRateHopping mode,  $n_t$  is given by the EffectiveNumAntennas parameter, which is part of the public data of the Overhead Messages Protocol. In BlockHopping mode,  $n_t$  shall be equal to  $N_l$ .

Each layer consists of a packet that is generated by the FTC MAC Protocol, and is split, appended with CRC, encoded, channel interleaved, repeated, data-scrambled and modulated according to the procedure described in 9.2. A CRC length of  $N_{CRC,Data}$  is used for this packet. The MACID of the target access terminal, and the packet format index assigned to this packet, shall be used to generate the initial state of the data-scrambler described in 9.2.5. The size of the input packet generated by the FTC MAC Protocol in each layer shall be equal to  $8 \lfloor \rho n_0 N_f / 8 \rfloor - N_{CRC,Data}$ , where  $\rho$  denotes the spectral efficiency at the first transmission corresponding to the packet format for that layer (defined by the FTC MAC Protocol),  $n_0$  denotes the number of usable hop-ports assigned to this packet in the first FL PHY Frame of transmission and  $N_f$  is equal to  $6N_{FRAME,F}$  if this packet is part of an extended duration transmission and is equal to  $N_{FRAME,F}$  otherwise. The FTC MAC protocol determines whether or not a packet is part of an extended duration transmission. Here, a usable hop-port is as defined in 9.3.2.5.1.2. The packet in the  $l^{th}$  layer,  $l$  ranging from 0 to  $N_l-1$ , shall be modulated on to the hop-ports assigned to the access terminal according to the following procedure:

1. Initialize a port counter  $i$  to 0, a frame counter  $f$  to 0, and an OFDM symbol counter  $j$  to 0. Also initialize a modulation counter  $m$  to 0.
2. Arrange the set of usable hop-ports assigned to this packet in the  $f^{th}$  PHY Frame of transmission in increasing order, where the ordering of hop-ports is as defined in

9.3.2.2.5. Let the resulting sequence be denoted by  $(k_0, p_0), (k_1, p_1), \dots, (k_{n-1}, p_{n-1})$ , where  $n$  is the total number of usable hop-ports assigned to this packet in the  $f$ 'th PHY Frame of transmission. The notation for a hop-port is as in 9.3.2.2.5, i.e.,  $k_i$  denotes the CarrierIndex and  $p_i$  denotes the hop-port index in that carrier.

3. Let  $n_{sc}$  be the subcarrier index corresponding to the hop-port  $(k_i, p_i)$  in the  $j$ 'th OFDM symbol in the  $f$ 'th PHY Frame of transmission. Let  $q$  be the modulation order to be used for the  $f$ 'th PHY Frame of transmission, which is a function of the packet format. If  $n_{sc}$  is not a pilot subcarrier and if  $(k_i, p_i)$  is not a DPICH hop-port, then a modulation symbol  $s$  with modulation order  $q$  is generated by the modulator for this packet according to the procedure described in 9.2.6. In SymbolRateHopping mode, this modulation symbol shall be modulated with energy  $P$  on hop-port  $(k_i, p_i)$  on the antenna with index  $(m+l) \bmod n_t$ , i.e., the value of the corresponding subcarrier on this antenna shall be  $\sqrt{P} s$ . Here  $P$  is the power density per antenna used for this assignment in the  $f$ 'th PHY Frame of transmission. The same power density shall be used on all antennas. In BlockHopping mode, this modulation symbol shall be modulated with energy  $P$  on hop-port  $(k_i, p_i)$  on the tile-antenna with index  $(m+l) \bmod n_t$ , i.e., the value of the corresponding subcarrier on this tile-antenna shall be  $\sqrt{P} s$ . Here  $P$  is the power density used for this assignment on tile-antenna  $(m+l) \bmod n_t$  in the  $f$ 'th PHY Frame of transmission. Different power densities may be used for different tile-antennas. Also, in SymbolRateHopping mode, the power density  $P$  is constant over all hop-ports assigned to this packet, while in BlockHopping mode, a different power density  $P$  may be used for each tile in the assignment. Determining the value of  $P$  is out of the scope of this specification.
4. Increment  $i$ . If  $n_{sc}$  is not a pilot, increment  $m$ .
5. If  $i = n$ , increment  $j$  and set  $i = 0$ .
6. If  $j = N_{FRAME,F}$ , increment  $f$  and set  $j = 0$ . Also set  $m = 0$ .
7. If the last PHY Frame of transmission for this layer has been completed (as determined by the FTC MAC Protocol), then stop. Else repeat steps 2 through 7.

#### 9.3.2.5.4.4 MIMO SCW mode

In SCW mode, the access network does not transmit multiple coded streams on the same set of subcarriers (i.e., two different packets occupy disjoint sets of subcarriers). The overall spectral efficiency is determined by the packet format and the number of layers  $N_l$ , which is defined in this case as the number of simultaneously transmitted modulation symbols (on antennas or tile-antennas) and is determined by the FTC MAC Protocol. The Cyclic Spatial Multiplexing structure described in 9.3.2.5.4.3 and illustrated in Figure 102 is also used for this mode, i.e., the set of antennas used to transmit the modulation symbols changes cyclically from subcarrier to subcarrier.

Let  $n_t$  denote the number antennas used for the SCW transmission in SymbolRateHopping mode, and the number of tile-antennas used in BlockHopping mode. In SymbolRateHopping mode,  $n_t$  shall be given by the EffectiveNumAntennas parameter, which is part of the public data of the Overhead Messages Protocol. In BlockHopping mode,  $n_t$  shall be equal to  $N_l$ .

Each F-DCH packet is generated by the FTC MAC Protocol, and is split, appended with CRC, encoded, channel interleaved, repeated, data-scrambled, and modulated according to the procedure described in 9.2. A CRC length of  $N_{CRC,Data}$  is used for this packet. The MACID of the target access

terminal, and the packet format index assigned to this packet, shall be used to generate the initial state of the data-scrambler described in 9.2.5. The size of the input packet generated by the FTC MAC Protocol shall be equal to  $8 \lfloor \rho n_0 N_f / 8 \rfloor N_l - N_{CRC,Data}$ , where  $\rho$  denotes the spectral efficiency at the first transmission corresponding to the packet format of the packet (defined by the FTC MAC Protocol),  $n_0$  denotes the number of usable hop-ports assigned to this packet in the first FL PHY Frame of transmission, and  $N_f$  is equal to  $6N_{FRAME,F}$  if this packet is part of an extended duration transmission and is equal to  $N_{FRAME,F}$  otherwise. The FTC MAC protocol determines whether or not a packet is part of an extended duration transmission. Here, a usable hop-port is as defined in 9.3.2.5.1.2. This packet shall be modulated on to the hop-ports assigned to the packet according to the following procedure:

1. Initialize a port counter  $i$  to 0, a frame counter  $f$  to 0, and an OFDM symbol counter  $j$  to 0. Also initialize a modulation symbol counter  $m$  to 0.
2. Arrange the set of usable hop-ports assigned to this packet in the  $f$ 'th PHY Frame of transmission in increasing order, where the ordering of hop-ports is as defined in 9.3.2.2.5. Let the resulting sequence be denoted by  $(k_0, p_0), (k_1, p_1), \dots, (k_{n-1}, p_{n-1})$ , where  $n$  is the total number of usable hop-ports assigned to this packet in the  $f$ 'th PHY Frame of transmission. The notation for a hop-port is as in 9.3.2.2.5, i.e.,  $k_i$  denotes the CarrierIndex and  $p_i$  denotes the hop-port index in that carrier.
3. Let  $n_{sc}$  be the subcarrier index corresponding to the hop-port  $(k_i, p_i)$  in the  $j$ 'th OFDM symbol in the  $f$ 'th PHY Frame of transmission. Let  $q$  be the modulation order to be used for the  $f$ 'th PHY Frame of transmission, which is a function of the packet format. If  $n_{sc}$  is not a pilot subcarrier and if  $(k_i, p_i)$  is not a DPICH hop-port, then generate  $N_l$  modulation symbols  $s_0, s_1, \dots, s_{N_l-1}$  with modulation order  $q$  according to the procedure described in 9.2.6.
4. In SymbolRateHopping mode, the modulation symbol  $s_l$ ,  $l$  ranging from 0 to  $N_l-1$ , is transmitted with energy  $P$  on hop-port  $(k_i, p_i)$  on the antenna with index  $(m+l) \bmod n_t$ , i.e., the value of the corresponding subcarrier shall be  $\sqrt{P} s$ . Here  $P$  is the power density per antenna used for this assignment in the  $f$ 'th PHY Frame of transmission. The same power density shall be used on all antennas. In BlockHopping mode, the modulation symbol  $s_l$ ,  $l$  ranging from 0 to  $N_l-1$ , is transmitted with energy  $P$  on hop-port  $(k_i, p_i)$  on the tile-antenna with index  $(m+l) \bmod n_t$ , i.e., the value of the corresponding subcarrier shall be  $\sqrt{P} s$ . Here  $P$  is the power density used for this assignment on tile-antenna  $(m+l) \bmod n_t$  in the  $f$ 'th PHY Frame of transmission (generated by the FTC MAC Protocol). Different power densities may be used for different tile-antennas. Also, in SymbolRateHopping mode, the power density  $P$  is constant over all hop-ports assigned to this packet, while in BlockHopping mode, a different power density  $P$  may be used for each tile in the assignment. Determining the value of  $P$  is out of the scope of this specification.
5. Increment  $i$ . If  $n_{sc}$  is not a pilot, increment  $m$ .
6. If  $i = n$ , increment  $j$  and set  $i = 0$ .
7. If  $j = N_{FRAME,F}$ , increment  $f$ . Also set  $j = m = 0$ .

- 1           8. If the last PHY Frame of transmission has been completed (as determined by the FTC  
2           MAC Protocol), then stop. Else repeat steps 2 through 8.

### 3   **9.3.2.5.4.5 Erasure sequence**

4   An erasure sequence spans one or more consecutive FL PHY Frames of transmission on a set of hop-  
5   ports determined by the FTC MAC Protocol. The erasure sequence shall be modulated on to the hop-  
6   ports assigned to this sequence according to the following procedure:

- 7           1. Construct a one-bit packet, with the bit in the packet being set to zero. This packet is  
8           encoded, channel interleaved, repeated, scrambled, and modulated according to the  
9           procedure described in 9.2.<sup>71</sup> The MAC ID of the target access terminal, and a packet  
10          format index of 0, shall be used to generate the initial seed of the scrambler. QPSK  
11          modulation shall be used for all of the modulation symbols in the packet.
- 12          2. Initialize a port counter  $i$  to 0, a frame counter  $f$  to 0, and an OFDM symbol counter  $j$   
13          to 0.
- 14          3. Arrange the set of usable hop-ports assigned to this packet in the  $f$ 'th PHY Frame of  
15          transmission in increasing order, where the ordering of hop-ports is as defined in  
16          9.3.2.2.5. Here, a usable hop-port is as defined in 9.3.2.5.1.2. Let the resulting sequence  
17          be denoted by  $(k_0, p_0), (k_1, p_1), \dots, (k_{n-1}, p_{n-1})$ , where  $n$  is the total number of usable hop-  
18          ports assigned to this packet in the  $f$ 'th PHY Frame of transmission. The notation for a  
19          hop-port is as in 9.3.2.2.5, i.e.,  $k_i$  denotes the CarrierIndex and  $p_i$  denotes the hop-port  
20          index in that carrier.
- 21          4. Let  $n_{sc}$  be the subcarrier index corresponding to the hop-port  $(k_i, p_i)$  in the  $j$ 'th OFDM  
22          symbol in the PHY Frame. If  $n_{sc}$  is not a pilot subcarrier and if  $(k_i, p_i)$  is not a DPICH hop-  
23          port, then a QPSK modulation symbol  $s$  is generated by the modulator according to the  
24          procedure described in 9.2.6. This modulation symbol shall be modulated with energy  $P$   
25          on hop-port  $(k_i, p_i)$ , i.e., the value of the corresponding subcarrier shall be  $\sqrt{P}s$ , where  $P$   
26          is the power density used for this erasure sequence. Determining the value of  $P$  is out of  
27          the scope of this specification. The modulation shall be done on the antenna with index 0  
28          in SymbolRateHopping mode, and on the tile-antenna with index 0 in BlockHopping  
29          mode.
- 30          5. Increment  $i$ . If  $i = n$ , or if  $i = N_{\text{MaxErasureHoppports}, F}$ , increment  $j$  and set  $i = 0$ .
- 31          6. If  $j = N_{\text{FRAME}, F}$ , increment  $f$  and set  $j = 0$ .
- 32          7. If the last PHY Frame of transmission has been completed (as determined by the FTC  
33          MAC Protocol), then stop. Else repeat steps 3 through 7.

---

<sup>71</sup> The operations before scrambling and modulation are all trivial operations; i.e., they result in an all-zeros sequence. The erasure sequence is equivalent to scrambling an all-zeros sequence of the required length, followed by QPSK modulation.

### 9.3.2.6 Sector-specific scrambling

Each OFDM symbol in the superframe preamble as well as in every FL PHY Frame shall be scrambled by a sector-specific scrambling sequence. The scrambling operation shall be performed independently on each carrier. The rest of this section describes the scrambling operation for the carrier  $k$ , where  $k=0,1,\dots,N_{\text{CARRIERS}}-1$ . The scrambling sequence for the carrier  $k$  consists of a complex number for every subcarrier in the carrier  $k$  in every OFDM symbol in the superframe. The scrambling operation shall consist of multiplying the unscrambled complex symbol on each subcarrier by the corresponding entry in the scrambling sequence, unless both conditions (a) and (b) are true: (a) The subcarrier corresponds to a F-DPICH hop-port (via the hop-permutation), and (b)  $\text{FLDPISectorScramble}$ , which is part of the public data of the Overhead Messages Protocol for carrier  $k$ , is set to 0. For subcarriers for which these conditions (a) and (b) are true, the scrambling operation shall consist of leaving the subcarrier unchanged; and a cell-specific scrambling sequence, as described in 9.3.2.7, shall be used to scramble the subcarrier.

Each complex number in the sector-specific scrambling sequence is generated from two bits, denoted by  $s_I$  and  $s_Q$ , using the following mapping:

1. The bit combination  $(s_I, s_Q) = (0,0)$  is mapped to the complex number  $\left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$ .

2. The bit combination  $(s_I, s_Q) = (0,1)$  is mapped to the complex number  $\left(\frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}\right)$ .

3. The bit combination  $(s_I, s_Q) = (1,0)$  is mapped to the complex number  $\left(\frac{-1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$ .

4. The bit combination  $(s_I, s_Q) = (1,1)$  is mapped to the complex number  $\left(\frac{-1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}\right)$ .

The sector-specific scrambling sequence for the carrier  $k$  shall be generated using two 20-bit registers, called the I-register and the Q-register, as shown in Figure 103 and Figure 104, respectively. The I-register shall have a generator polynomial  $h_I(D) = D^{20} + D^{19} + D^{16} + D^{14} + 1$  i.e., the  $n$ 'th output  $I(n)$  of the register shall satisfy  $I(n) = I(n-20) \oplus I(n-19) \oplus I(n-16) \oplus I(n-14)$ . The Q-register shall have a generator polynomial  $h_Q(D) = D^{20} + D^{18} + D^{15} + D^{14} + 1$  i.e., the  $n$ 'th output  $Q(n)$  of the register shall satisfy  $Q(n) = Q(n-20) \oplus Q(n-18) \oplus Q(n-15) \oplus Q(n-14)$ . Each entry in the sector-specific scrambling sequence shall be generated using  $s_I$  and  $s_Q$  bits which are respectively the outputs of the I-register and the Q-register after they have been appropriately initialized and clocked as in the following description.

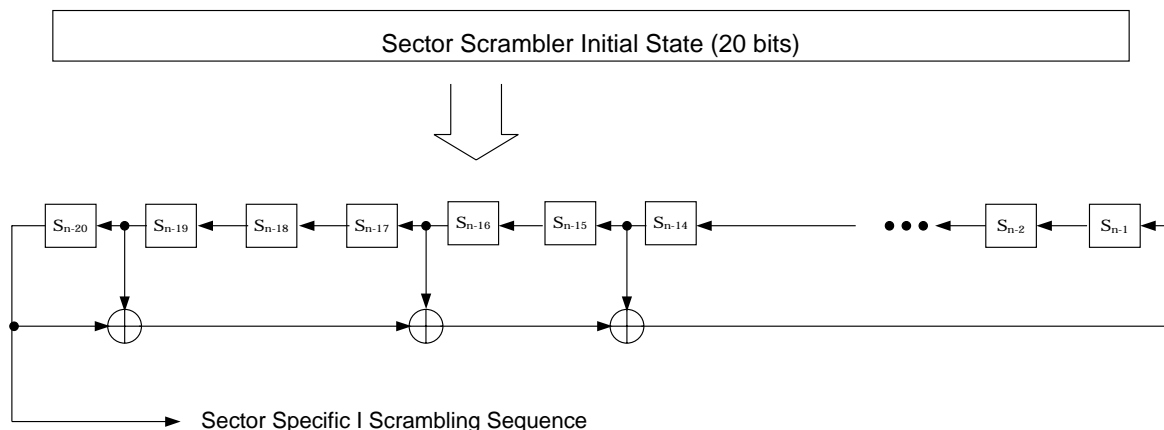
At the start of every superframe, define  $\text{PilotPNSectorScramb}$  to be equal to  $\text{PilotPhase}$  in SemiSynchronous mode and equal to  $\text{PilotPN}$  in Asynchronous mode. (Thus, for a given sector,  $\text{PilotPNSectorScramb}$  is fixed in Asynchronous mode, but changes every superframe in SemiSynchronous mode.) Let  $p_{11}, p_{10}, \dots, p_0$  be the 12 bits of  $(\text{PilotPNSectorScramb}+k) \bmod 4096$  for a given superframe, with  $p_{11}$  being the MSB and  $p_0$  being the LSB. At the beginning of each superframe, the I and Q registers shall both be initialized to the state  $[11111111p_{11}p_{10}p_9p_8p_7p_6p_5p_4p_3p_2p_1p_0]$ . The I and Q registers shall then be clocked  $5N_{\text{CARRIER\_SIZE}}$  times in the superframe to generate the  $s_I$  and  $s_Q$  bits for all of the subcarriers belonging to the carrier



1 k in the OFDM symbols with indices 0, 1, 2, 3, 4 in the superframe. The i'th entry in the scrambling  
 2 sequence (generated after i clock periods) is used to scramble the subcarrier with index i mod  
 3  $N_{CARRIER\_SIZE}$  in the carrier k, in the OFDM symbol with index  $\lfloor i / N_{CARRIER\_SIZE} \rfloor$  in the superframe.  
 4 The outputs of the I and Q registers immediately after their state has been initialized (before they are  
 5 clocked) shall be used to generate the scrambling sequence entry corresponding to the subcarrier with  
 6 index 0 in the carrier k in the OFDM symbol with index 0.

7 At the start of each of the OFDM symbols with indices 5,6,7 in the superframe, both the I and the Q  
 8 registers shall be initialized. The initialization state shall be the same for both the I and the Q  
 9 registers. The initialization state shall be the state  $[11111101p_1p_0p_1p_0p_1p_0p_1p_0p_1p_0p_1p_0]$ ,  
 10  $[11111110p_3p_2p_1p_0p_7p_6p_5p_4p_3p_2p_1p_0]$ , and  $[11111100p_{11}p_{10}p_9p_8p_7p_6p_5p_4p_3p_2p_1p_0]$ , at the start of the  
 11 OFDM symbols indexed 5,6 and 7 respectively.<sup>72</sup> For each of these OFDM symbols, the entry in the  
 12 scrambling sequence corresponding to the subcarrier with index i in the carrier k (i varying from 0 to  
 13  $N_{CARRIER\_SIZE} - 1$ ) shall be generated by clocking the I and Q registers i times, following their  
 14 initialization. The  $s_I$  and  $s_Q$  bits shall be, respectively, the outputs of the I and Q-registers. For each of  
 15 these OFDM symbols indexed 5,6,7 in the superframe, the outputs of the I and Q registers  
 16 immediately after their state has been initialized (before they are clocked) at the start of the OFDM  
 17 symbol shall be used to generate the scrambling sequence entry corresponding to the subcarrier with  
 18 index 0 in the carrier k.

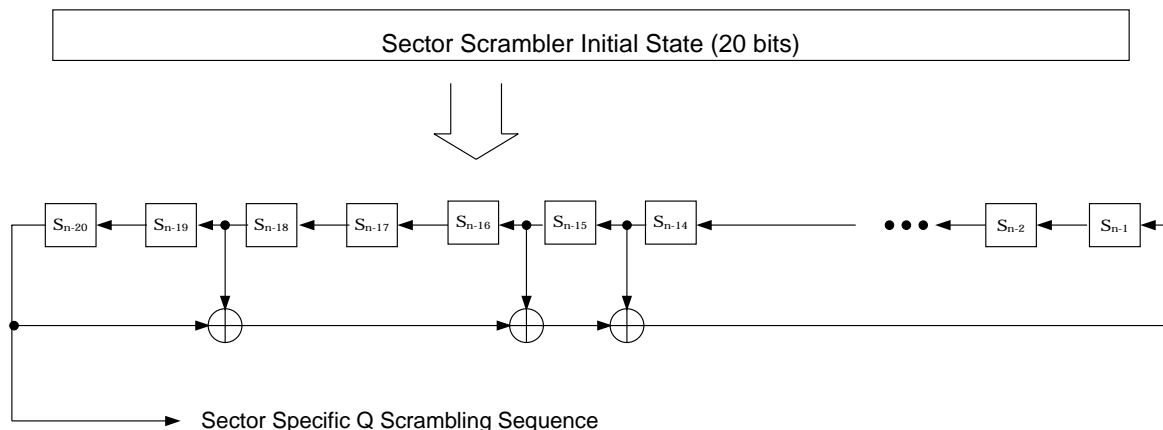
19 At the start of OFDM symbol with index 8 in the superframe, the I and Q registers shall both be  
 20 initialized to the state  $[11111000p_{11}p_{10}p_9p_8p_7p_6p_5p_4p_3p_2p_1p_0]$ . The I and Q registers shall then be  
 21 clocked  $N_{CARRIER\_SIZE}$  times for each remaining OFDM symbol in the superframe to generate the  $s_I$   
 22 and  $s_Q$  entries for all the subcarriers belonging to the carrier k in all the remaining OFDM symbols.  
 23 The  $s_I$  and  $s_Q$  entries generated from the I and Q registers after i clock periods, are used to scramble  
 24 the subcarrier with index i mod  $N_{CARRIER\_SIZE}$  in the carrier k, in the OFDM symbol with index  
 25  $\lfloor i / N_{CARRIER\_SIZE} \rfloor + 8$  in the superframe. The outputs of the I and Q registers immediately after their  
 26 state has been initialized (before they are clocked) shall be used to generate the scrambling sequence  
 27 entry corresponding to the subcarrier with index 0 in the carrier k, in the OFDM symbol with index 8.



28  
 29

**Figure 103 Sector-specific scrambler – I sequence**

<sup>72</sup> The OFDM Symbols with indices 5 and 6 are TDM pilots used in acquisition.



1  
2 **Figure 104 Sector-specific scrambler – Q sequence**

3 **9.3.2.7 Cell-specific scrambling for F-DPICH**

4 The operations in this section shall be carried out independently for each carrier, and are described for  
 5 the carrier with index  $k$ , where  $k=0,1,\dots, N_{\text{CARRIERS}} - 1$ . The operations in this section shall be carried  
 6 out for the carrier with index  $k$  if and only if  $\text{FLDPISectorScramble}$ , which is part of the public data  
 7 of the Overhead Messages Protocol for carrier  $k$ , is set to 0. A cell-specific scrambling symbol shall  
 8 be generated for each subcarrier, but only some of the generated scrambling symbols shall be used  
 9 and the rest shall be discarded. The scrambling symbols that shall be used shall be those generated for  
 10 subcarriers that correspond to F-DPICH hop-ports (via the hop-permutation), as defined in  
 11 9.3.2.5.2.3. These subcarriers are henceforth referred to as F-DPICH subcarriers. The cell-specific  
 12 scrambling sequence consists of a complex number for every subcarrier. The scrambling operation  
 13 shall consist of multiplying the unscrambled complex symbol on each F-DPICH subcarrier by the  
 14 corresponding entry in the scrambling sequence. Each complex number in the cell-specific  
 15 scrambling sequence is generated from two bits, denoted by  $s_I$  and  $s_Q$ , using the following mapping:

- 16 1. The bit combination  $(s_I, s_Q) = (0,0)$  is mapped to the complex number  $\left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$ .
- 17 2. The bit combination  $(s_I, s_Q) = (0,1)$  is mapped to the complex number  $\left(\frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}\right)$ .
- 18 3. The bit combination  $(s_I, s_Q) = (1,0)$  is mapped to the complex number  $\left(\frac{-1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$ .
- 19 4. The bit combination  $(s_I, s_Q) = (1,1)$  is mapped to the complex number  $\left(\frac{-1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}\right)$ .

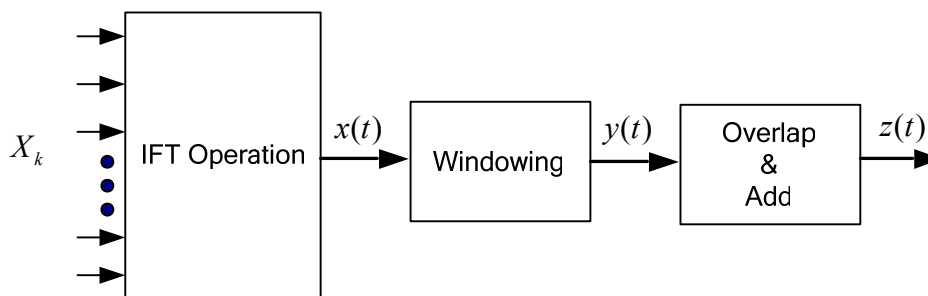
20 The cell-specific scrambling sequence for the carrier  $k$  shall be generated using two 20-bit registers,  
 21 called the I-register and the Q-register, as shown in Figure 103 and Figure 104, respectively. The I-  
 22 register shall have a generator polynomial  $h_I(D) = D^{20} + D^{19} + D^{16} + D^{14} + 1$  i.e., the  $n$ 'th output  $I(n)$   
 23 of the register shall satisfy  $I(n) = I(n-20) \oplus I(n-19) \oplus I(n-16) \oplus I(n-14)$ . The Q-register shall have a

1 generator polynomial  $h_Q(D) = D^{20} + D^{18} + D^{15} + D^{14} + 1$  i.e., the  $n$ 'th output  $Q(n)$  of the register shall  
 2 satisfy  $Q(n) = Q(n-20) \oplus Q(n-18) \oplus Q(n-15) \oplus Q(n-14)$ . Each entry in the cell-specific scrambling  
 3 sequence shall be generated using  $s_I$  and  $s_Q$  bits which are respectively the outputs of the I-register  
 4 and the Q-register after they have been appropriately initialized and clocked as in the following  
 5 description.

6 Let CellPilotPN be the 12 bit number obtained from the PilotPN by setting its 5<sup>th</sup>, 6<sup>th</sup> and 7<sup>th</sup> bits to  
 7 zero (where the bits are numbered starting from 0, with the 0<sup>th</sup> bit denoting the LSB). For the  
 8 superframe with index  $s$ , let SFInd be set equal to  $s$  in SemiSynchronous mode and set equal to zero  
 9 in Asynchronous mode. For the superframe with index  $s$ , let  $b_{11}, b_{10}, \dots, b_0$  be the 12 bits of  
 10  $(\text{CellPilotPN} + \text{SFInd} + k) \bmod 4096$ , with  $b_{11}$  being the MSB and  $b_0$  being the LSB. At the start of the  
 11 OFDM symbol with index 0 in the superframe, both the I and the Q registers shall be initialized to the  
 12 state  $[11110000b_{11}b_{10}b_9b_8b_7b_6b_5b_4b_3b_2b_1b_0]$ . The outputs of the I and Q registers after they are both  
 13 clocked  $i$  times, shall respectively be the  $s_I$  and  $s_Q$  bits used to generate a symbol  $c(i)$  in the  
 14 scrambling sequence. This symbol  $c(i)$  shall be used to scramble the subcarrier with index  $i \bmod$   
 15  $N_{\text{CARRIER\_SIZE}}$  in the carrier  $k$  in the OFDM Symbol with index  $\lfloor i / N_{\text{CARRIER\_SIZE}} \rfloor$  in the superframe,  
 16 provided this subcarrier is an F-DPICH subcarrier

### 17 9.3.2.8 Time-domain processing

18 The sequence of OFDM symbols at the output of the sector scrambler shall be converted to a complex  
 19 baseband waveform according to the procedure described in Figure 105. This procedure consists of an  
 20 Inverse Fourier Transform (IFT) operation, a windowing operation, and an overlap-and-add  
 21 operation.



22  
23 **Figure 105 Time-domain processing**

#### 24 9.3.2.8.1 Symbol start time

25 The start time  $T_{\text{START,SF}}$  of the superframe with index  $i$  is given by the product of  $i$  with the superframe  
 26 duration  $T_{\text{SUPERFRAME}}$ , where  $T_{\text{SUPERFRAME}}$  is as defined in 9.3.2.2.4.

27 In FDD, the start time of the  $k$ -th OFDM symbol in the superframe,  $k$  ranging from 0 to  $N_{\text{PREAMBLE}} +$   
 28  $N_{\text{FDD,FLPHYFrames}}N_{\text{FRAME,F}} - 1$ , is given by  $T_{\text{START,SF}} + kT_{\text{s,PR}}$  if  $k$  is less than  $N_{\text{PREAMBLE}}$ , and is given by  
 29  $T_{\text{START,SF}} + N_{\text{PREAMBLE}}T_{\text{s,PR}} + (k - N_{\text{PREAMBLE}})T_{\text{s}}$ , otherwise. Here  $T_{\text{START,SF}}$  is the start time of the  
 30 superframe, and  $N_{\text{FDD,FLPHYFrames}}$  is defined by the Lower MAC sublayer.

31 In TDD mode, the start time of the  $k$ -th OFDM symbol in the superframe,  $k$  ranging from 0 to  
 32  $N_{\text{PREAMBLE}} + N_{\text{TDD,FLPHYFrames}} * N_{\text{FRAME,F}} - 1$  is given by  $T_{\text{START,SF}} + kT_{\text{s,PR}}$  if  $k$  is less than  $N_{\text{PREAMBLE}}$ ,  
 33 and is given by  $T_{\text{START,SF}} + N_{\text{PREAMBLE}}T_{\text{s,PR}} + (k - N_{\text{PREAMBLE}})T_{\text{s}} + \lfloor (k - N_{\text{PREAMBLE}}) / (N_{\text{FL\_BURST}}$

1  $N_{\text{FRAME},F}] * (N_{\text{RL\_BURST}} N_{\text{FRAME},R} T_s + T_{G,\text{TDD},F} + T_{G,\text{TDD},R})$ , otherwise. Here  $T_{\text{START},\text{SF}}$  is the start time of  
 2 the superframe, while  $T_{s,\text{PR}}$  and  $T_s$  are as defined in 9.3.2.2.3, and  $N_{\text{TDD},\text{FLPHYFrames}}$  is defined by the  
 3 Lower MAC sublayer.

#### 4 9.3.2.8.2 IFT operation

5 Let  $X_k$  be the value of the complex modulation symbol on the  $k$ 'th subcarrier of an OFDM symbol,  $k$   
 6 ranging from 0 to  $N_{\text{FFT}}-1$ . The IFT of the OFDM symbol is given by the infinite duration signal:

$$7 \quad x(t) = \frac{1}{\sqrt{N_{\text{FFT}}}} \sum_{k=0}^{N_{\text{FFT}}-1} X_k e^{j2\pi(k - N_{\text{FFT}}/2)(t - T_{\text{CP},\text{PR}} - T_{\text{START}})/(N_{\text{FFT}}T_{\text{CHIP}})}$$

8 during the superframe preamble and by:

$$9 \quad x(t) = \frac{1}{\sqrt{N_{\text{FFT}}}} \sum_{k=0}^{N_{\text{FFT}}-1} X_k e^{j2\pi(k - N_{\text{FFT}}/2)(t - T_{\text{CP},\text{PR}} - T_{\text{START}})/(N_{\text{FFT}}T_{\text{CHIP}})}$$

10 during each FL PHY Frame, where  $T_{\text{START}}$  denotes the start time of the OFDM symbol,  $T_{\text{CP},\text{PR}}$  and  $T_{\text{CP}}$   
 11 are as defined in 9.3.2.2.3, and  $j$  denotes the complex number (0,1).

#### 12 9.3.2.8.3 Windowing

13 The signal  $x(t)$  at the output of the IFT shall be multiplied by the window function  $w(t)$ , where  $w(t)$  is  
 14 given by the equation:

$$15 \quad w(t) = \begin{cases} 0 & , t < T_{\text{START}} - T_{\text{WGI}} \\ 0.5 - 0.5 \cos\left(\frac{\pi(t + T_{\text{WGI}} - T_{\text{START}})}{T_{\text{WGI}}}\right) & , T_{\text{START}} - T_{\text{WGI}} \leq t < T_{\text{START}} \\ 1 & , T_{\text{START}} \leq t < T_{\text{START}} + T_{\text{CP},\text{PR}} + T_{\text{FFT}} \\ 0.5 + 0.5 \cos\left(\frac{\pi(t - T_{\text{START}} - T_{\text{CP},\text{PR}} - T_{\text{FFT}})}{T_{\text{WGI}}}\right) & , T_{\text{START}} + T_{\text{CP},\text{PR}} + T_{\text{FFT}} \leq t < T'_{s,\text{PR}} \\ 0 & , t \geq T'_{s,\text{PR}} \end{cases}$$

1 during the superframe preamble, and by the equation:

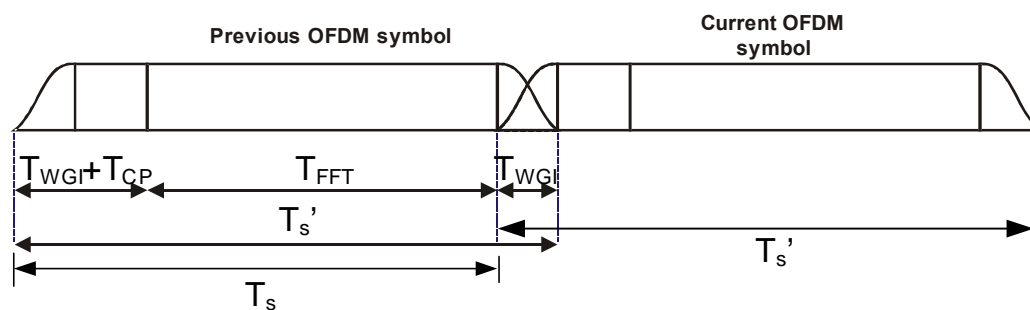
$$w(t) = \begin{cases} 0 & , t < T_{START} - T_{WGI} \\ 0.5 - 0.5 \cos\left(\frac{\pi(t + T_{WGI} - T_{START})}{T_{WGI}}\right) & , T_{START} - T_{WGI} \leq t < T_{START} \\ 1 & , T_{START} \leq t < T_{START} + T_{CP} + T_{FFT} \\ 0.5 + 0.5 \cos\left(\frac{\pi(t - T_{START} - T_{CP} - T_{FFT})}{T_{WGI}}\right) & , T_{START} + T_{CP} + T_{FFT} \leq t < T'_s \\ 0 & , t \geq T'_s \end{cases}$$

3 during each FL PHY Frame, where  $T_{START}$  denotes the start time of the OFDM symbol. The quantities  
 4  $T_{FFT}$ ,  $T_s$ ,  $T'_s$ ,  $T_{s,PR}$  and  $T_{s,PR}'$  are as defined in 9.3.2.2.3.

5 The windowed signal  $y(t)$  is given by  $y(t) = x(t)w(t)$ .

#### 6 9.3.2.8.4 Overlap and add operation

7 The windowed IFTs  $y(t)$  corresponding to all of the OFDM symbols shall be added together to create  
 8 the final complex baseband waveform  $z(t)$ . In this procedure, neighboring OFDM symbols overlap for  
 9 duration  $T_{WGI}$ , as illustrated in Figure 106.



10  
11 **Figure 106 Overlap and add operation**

### 12 9.3.3 Synchronization and timing

#### 13 9.3.3.1 Timing reference source

14 Each sector shall use a time-base reference from which all time-critical transmission components,  
 15 including superframe boundaries, PHY Frame boundaries, and superframe indices, shall be derived.  
 16 In SemiSynchronous mode, the time-base reference of any two sectors shall be time-aligned to each  
 17 other with a maximum error of PilotIncrement times  $T_{SUPERFRAME}/4$ , where PilotIncrement is a  
 18 configuration attribute of the Active Set Management Protocol.

19 There is also a notion of two sectors being synchronous with each other. If two sectors are referred to  
 20 as being synchronous with each other, their time-base references shall be time-aligned to each other  
 21 with a maximum error of  $10\mu s$ .

1 In Asynchronous mode, there is no requirement for the alignment of the time-base references of two  
2 sectors.

### 3 **9.3.3.2 Sector transmission time**

4 Each sector shall radiate the superframe boundary aligned to its time-base reference. Time  
5 measurements are made at the sector antenna connector. If a sector has multiple radiating antenna  
6 connectors for the same channel, time measurements are made at the antenna connector having the  
7 earliest radiated signal.

8 The rate of change for timing corrections shall not exceed 102 nanoseconds (ns) per 200 milliseconds  
9 (ms).

## 10 **9.4 Access terminal requirements**

### 11 **9.4.1 Modulation characteristics**

12 This section describes the transmission from an access terminal (AT) to a subset of the set of sectors  
13 in its active set, where the Active Set is public data of the Active Set Management Protocol. This  
14 subset consists of sectors that are synchronous with each other. Moreover, the subset is a maximal  
15 subset, i.e., all sectors that are synchronous with the sectors in this subset are contained in this subset.  
16 This subset will be referred to as  $AS_{\text{SYNCH}}$ . The different synchronous subsets  $AS_{\text{SYNCH}}$  can be  
17 constructed using the last instance of the Active Set Add Message. Transmission from the access  
18 terminal to two different synchronous subsets of the active set shall be independent of each other, and  
19 shall each follow the procedures specified in this section.

#### 20 **9.4.1.1 Superframe timing**

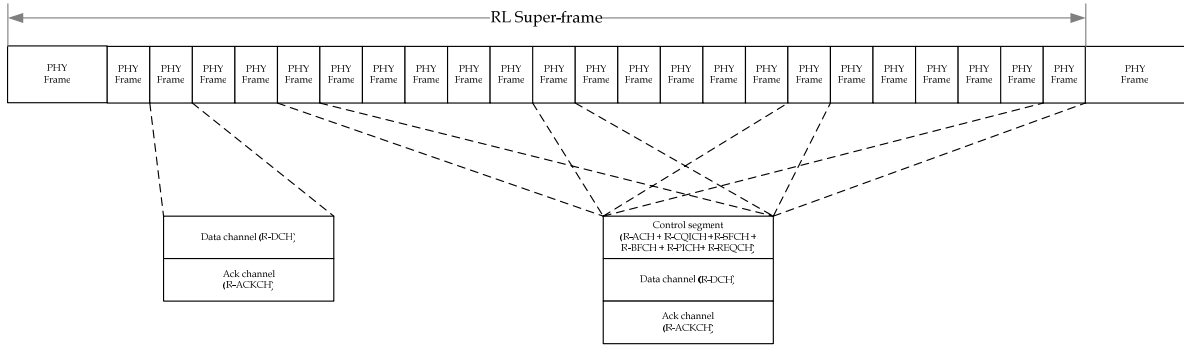
21 The reverse link transmission shall be divided into units of superframes. The duration of a reverse  
22 link superframe shall be the same as the duration of a forward link superframe, and the reverse link  
23 superframe shall be time-aligned with the forward link superframe as described in 9.4.2. Each  
24 reverse-link superframe is identified by a superframe index that is the same as the index of the time-  
25 aligned forward link superframe.

26 The structure of a reverse link superframe shall be as shown in Figure 107 for FDD and as shown in  
27 Figure 108, Figure 109, and Figure 110 for TDD with different values of  $N_{\text{FL\_BURST}}$  and  $N_{\text{RL\_BURST}}$ .  
28 Each superframe consists of  $N_{\text{FDD,RLPHYFrames}}$  RL PHY Frames in FDD and  $N_{\text{TDD,RLPHYFrames}}$  RL PHY  
29 Frames in TDD. Here  $N_{\text{FDD,RLPHYFrames}}$  and  $N_{\text{TDD,RLPHYFrames}}$ <sup>73</sup> are defined by the Lower MAC sublayer.  
30 The structure of the superframe preamble and each RL PHY Frame shall be as shown in Figure 111.

31 The PHY layer chapter of this specification uses a RL PHY Frame indexing scheme that is  
32 convenient for the descriptions herein, but is not necessarily consistent with indexing schemes used in  
33 other layers and sublayers in the specification. In this indexing scheme, the RL PHY Frames in a  
34 given superframe shall be indexed sequentially from 0 through  $N_{\text{FDD,RLPHYFrames}} - 1$  in FDD mode and  
35 from 0 through  $N_{\text{TDD,RLPHYFrames}} - 1$  in TDD mode. The RL PHY Frame index is sometimes also  
36 referred to using its 6-bit binary representation.

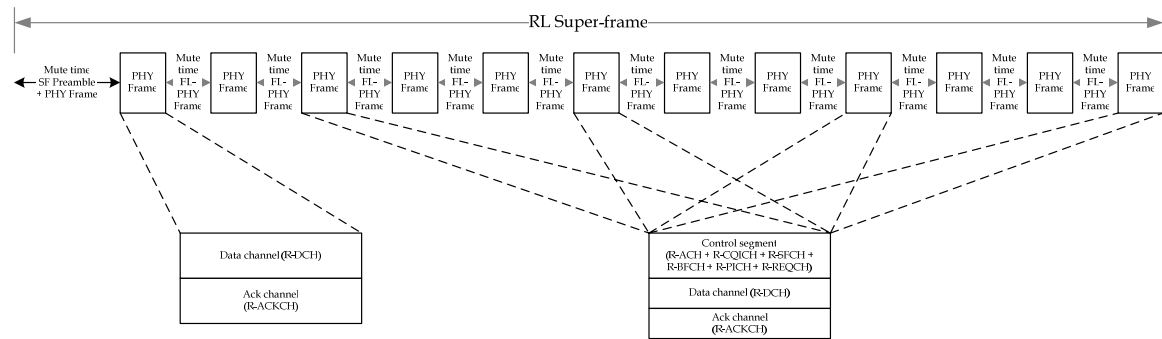
---

<sup>73</sup> Note that  $N_{\text{TDD,RLPHYFrames}}$  is a function of  $N_{\text{FL\_BURST}}$  and  $N_{\text{RL\_BURST}}$ .



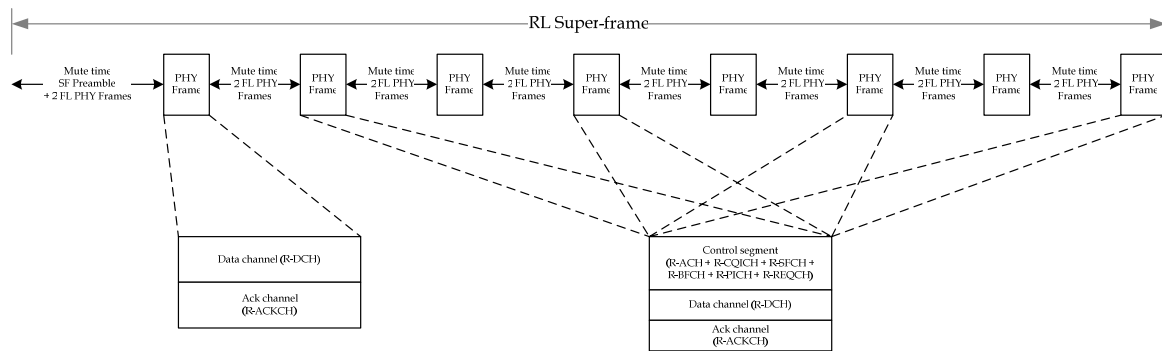
1  
2

**Figure 107 Reverse link superframe structure: FDD**



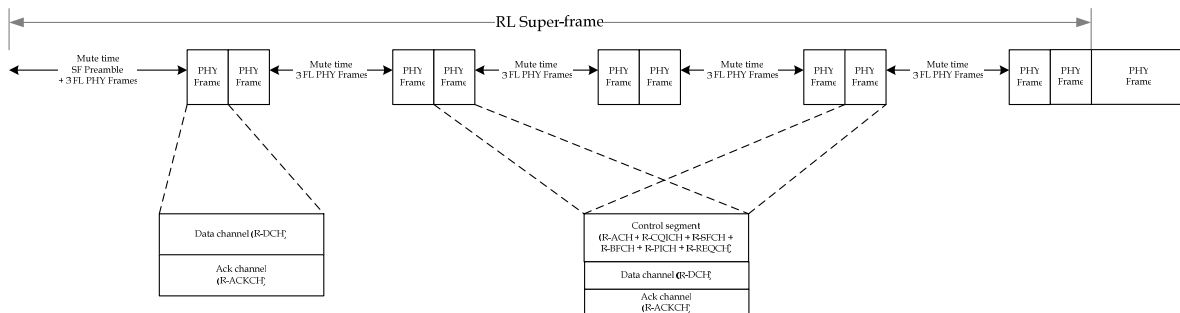
3  
4

**Figure 108 Reverse link superframe structure: TDD ( $N_{FL\_BURST} = 1, N_{RL\_BURST} = 1$ )**



5  
6

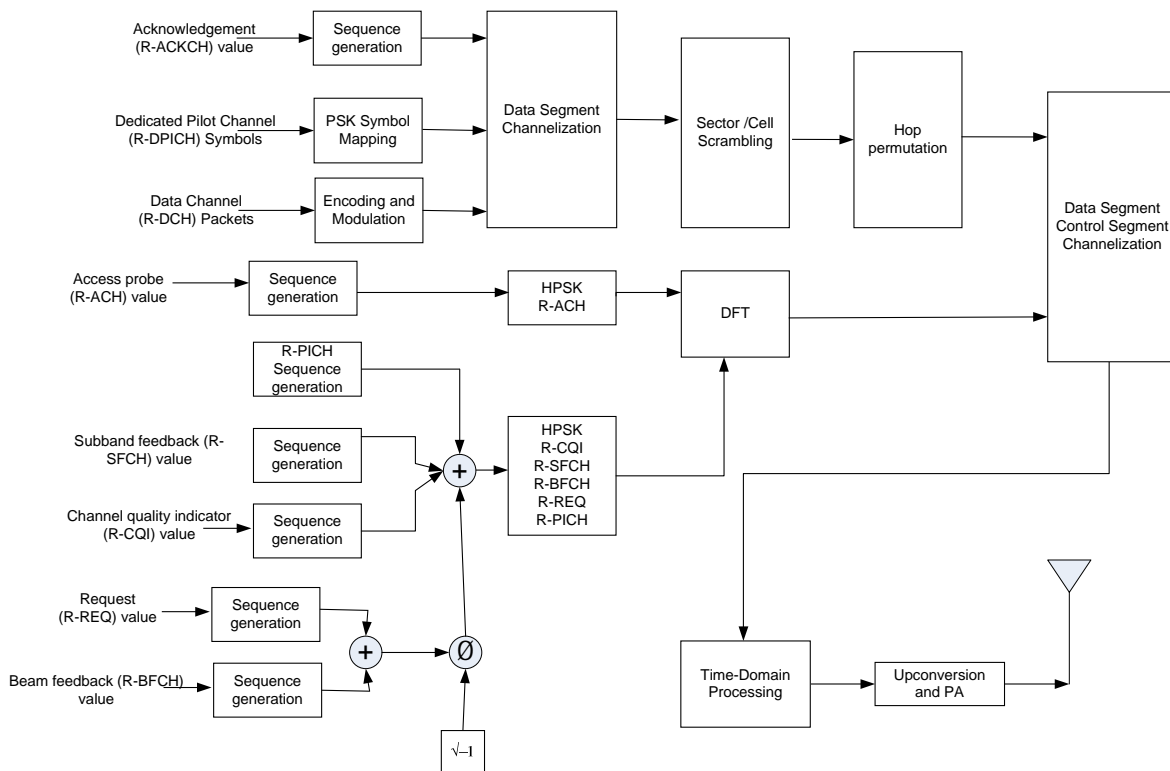
**Figure 109 Reverse link superframe structure: TDD ( $N_{FL\_BURST} = 2, N_{RL\_BURST} = 1$ )**



1

2

**Figure 110 Reverse link superframe structure ( $N_{FL\_BURST}=3, N_{RL\_BURST}=2$ )**



3

4

**Figure 111 Reverse channel structure**

5

**9.4.1.2 OFDM symbol characteristics**

6

7

8

9

10

The modulation used on the reverse link is Orthogonal Frequency Division Multiplexing (OFDM); i.e., each RL PHY Frame is subdivided into units of OFDM symbols. An OFDM symbol consists of  $N_{FFT}$  individually modulated subcarriers that carry complex-valued data. Complex-valued data is represented in the form  $d = (d_{re}, d_{im})$ , where  $d_{re}$  and  $d_{im}$  represent the real and imaginary components respectively. The subcarriers in each OFDM symbol shall be numbered 0 through  $N_{FFT}-1$ .



1 An additional indexing scheme may be used in MultiCarrierOn mode. The  $N_{\text{FFT}}$  subcarriers are split  
 2 into  $N_{\text{CARRIERS}}$  contiguous groups, each of which is referred to as a carrier. Each carrier consists of  
 3  $N_{\text{CARRIER\_SIZE}}$  subcarriers, where  $N_{\text{CARRIER\_SIZE}} = N_{\text{FFT}} / N_{\text{CARRIERS}}$ . Each carrier has an associated index,  
 4 sometimes referred to as CarrierIndex, that ranges from 0 through  $N_{\text{CARRIERS}} - 1$ . The carrier with index  
 5  $c$  consists of subcarriers indexed  $cN_{\text{CARRIER\_SIZE}}$  through  $(c+1)N_{\text{CARRIER\_SIZE}} - 1$ . In MultiCarrierOff  
 6 mode, all  $N_{\text{FFT}}$  subcarriers belong to a single carrier having CarrierIndex 0. Furthermore, the  
 7 subcarriers within each carrier may be indexed from 0 to  $N_{\text{CARRIER\_SIZE}} - 1$  and the phrases “subcarrier  $f$   
 8 in carrier  $c$ ” and “subcarrier with index  $f$  within carrier with index  $c$ ” shall be equivalent to  
 9 “subcarrier  $cN_{\text{CARRIER\_SIZE}} + f$ .” These two subcarrier indexing schemes are used interchangeably in  
 10 the Physical Layer chapter of this specification.

#### 11 9.4.1.2.1 Guard subcarriers

12 Some of the available subcarriers in an OFDM symbol are designated as guard subcarriers and shall  
 13 not be modulated; i.e., no energy shall be transmitted on these subcarriers. The number of guard  
 14 subcarriers in each OFDM symbol shall be  $N_{\text{GUARD}}$ , and the set of guard subcarriers shall be the  
 15 subcarriers numbered 0 through  $N_{\text{GUARD}}/2 - 1$  and the subcarriers numbered  $N_{\text{FFT}} - N_{\text{GUARD}}/2$  through  
 16  $N_{\text{FFT}} - 1$ .

17 The number of guard subcarriers  $N_{\text{GUARD}}$  for the reverse link shall be the same as the number of guard  
 18 subcarriers on the forward link, and is given by NumGuardSubcarriers, which is part of the public  
 19 data of the Overhead Messages Protocol for any sector. The AT shall use the value corresponding to  
 20 any sector in  $AS_{\text{SYNCH}}$ . (All sectors in  $AS_{\text{SYNCH}}$  have the same value of  $N_{\text{GUARD}}$ .)

#### 21 9.4.1.2.2 Quasi-guard subcarriers

22 In multi-carrier mode, additional sub-carriers within each OFDM symbol are designated as quasi-  
 23 guard subcarriers and shall not be modulated, i.e., no energy shall be transmitted on these subcarriers.  
 24 The set of quasi-guard subcarriers in each RL PHY Frame shall be the subcarriers numbered  
 25  $N_{\text{CARRIER\_SIZE}} * m - N_{\text{GUARD}}/2$  through  $N_{\text{CARRIER\_SIZE}} * m + N_{\text{GUARD}}/2 - 1$  where  $m = 1, \dots, N_{\text{CARRIERS}} - 1$ .

26 Any subcarrier that is not a guard or a quasi-guard subcarrier is referred to as a usable subcarrier.

#### 27 9.4.1.2.3 OFDM symbol duration

28 The total OFDM symbol duration, denoted by  $T_s$ , consists of four parts:

- 29 ■ A data part with duration  $T_{\text{FFT}}$ , where  $T_{\text{FFT}} = N_{\text{FFT}} T_{\text{CHIP}}$ .
- 30 ■ A flat guard interval, also known as a cyclic prefix. The duration of this interval shall be  
 31 given by  $T_{\text{CP}}$  for all the OFDM symbols in TDD mode, and in all but the first  $N_{\text{PREAMBLE}}$   
 32 OFDM symbols of each superframe in FDD mode. In FDD mode, the duration of the flat  
 33 guard interval in the first  $N_{\text{PREAMBLE}}$  OFDM symbols of each superframe shall be given by  
 34  $T_{\text{CP,PR}}$ . Here,  $T_{\text{CP}}$  and  $T_{\text{CP,PR}}$  are as defined in 9.3.2.2.3, while  $N_{\text{PREAMBLE}}$  is as defined in  
 35 9.3.2.2.4.
- 36 ■ Two windowed guard intervals, of duration  $T_{\text{WGI}}$  each., on the two sides of the OFDM  
 37 symbol. The windowed guard interval duration is the same as on the forward link. There  
 38 is an overlap of  $T_{\text{WGI}}$  between consecutive OFDM symbols (see Figure 118).

39 The effective OFDM symbol duration is given by  $T_{s,\text{PR}} = T_{\text{FFT}} + T_{\text{CP,PR}} + T_{\text{WGI}}$  for the first  $N_{\text{PREAMBLE}}$   
 40 OFDM symbols of each superframe in FDD mode, and by  $T_s = T_{\text{FFT}} + T_{\text{CP}} + T_{\text{WGI}}$  in all other cases.  
 41 This effective OFDM symbol duration will henceforth be referred to as the OFDM symbol duration.

#### 9.4.1.2.4 Superframe duration

In FDD mode, each RL PHY Frame consists of  $N_{\text{FRAME,R}}$  OFDM symbols, with the exception of the RL PHY Frame with index 0 in a superframe. The RL PHY Frame with index 0 in a superframe consists of  $N_{\text{PREAMBLE}}$  additional OFDM symbols, each having a duration of  $T_{s,\text{PR}}$ . The superframe duration on the reverse link is therefore equal to

$$T_{\text{SUPERFRAME}} = N_{\text{PREAMBLE}}T_{s,\text{PR}} + N_{\text{FDD,RLPHYFrames}}N_{\text{FRAME,R}}T_s.$$

This duration is exactly equal to the forward link superframe duration.

In TDD mode, each RL PHY Frame consists of  $N_{\text{FRAME,R}}$  OFDM symbols. The mute time between adjacent sets of contiguous RL PHY Frames within a superframe in TDD equals the duration of  $N_{\text{FL\_BURST}}N_{\text{FRAME,F}}$  OFDM symbols plus guard time  $T_{\text{G,TDD,F}}$  and  $T_{\text{G,TDD,R}}$ . The mute time at the beginning of the reverse superframe equals the duration of the superframe preamble  $N_{\text{PREAMBLE}}T_{s,\text{PR}}$  plus  $N_{\text{FRAME,F}}$  OFDM symbols and guard time  $T_{\text{G,TDD,F}}$ . The total superframe duration is given by

$$T_{\text{SUPERFRAME}} = N_{\text{PREAMBLE}}T_{s,\text{PR}} + N_{\text{TDD,FLPHYFrames}} * N_{\text{FRAME,F}}T_s + N_{\text{TDD,RLPHYFrames}} * N_{\text{FRAME,R}}T_s + (T_{\text{G,TDD,F}} + T_{\text{G,TDD,R}}) * (N_{\text{TDD,FLPHYFrames}}/N_{\text{FL\_BURST}}).$$

Here  $T_s$  is as defined in 9.3.2.2.4 while  $N_{\text{TDD,FLPHYFrames}}$  and  $N_{\text{TDD,RLPHYFrames}}$  are as defined by the Lower MAC sublayer.

#### 9.4.1.2.5 Control and data segments

Each RL PHY Frame in both FDD and TDD modes shall contain the Data Segment.

As shown in Figure 107, in FDD mode an RL PHY Frame with index  $j$  within the superframe shall contain the Control Segment in every carrier if  $j \bmod 6 = 5$ .

In TDD mode, an RL PHY Frame with index  $j$  within the superframe with index  $i$  shall contain the Control Segment in every carrier if

$$(i * N_{\text{TDD,RLPHYFrames}} + j) \bmod (k * N_{\text{RL\_BURST}}) = k * N_{\text{RL\_BURST}} - 1$$

where  $k$  is the smallest integer such that  $k * (N_{\text{FL\_BURST}} + N_{\text{RL\_BURST}}) \geq 6$ . The RL PHY Frames containing the control segment for TDD mode with different sets of parameters  $N_{\text{FL\_BURST}}$  and  $N_{\text{RL\_BURST}}$  are shown in Figure 108 and Figure 109. In both TDD and FDD, on the  $m^{\text{th}}$  carrier, the Control Segment occupies  $N_{\text{CTRL-SUBBANDS}}$  subbands where  $N_{\text{CTRL-SUBBANDS}}$  is equal to  $\text{NumRLControlSubbands}$ , which is part of the public data of the Overhead Message Protocol on carrier  $m$ .

The Control Segment Period is defined as 6 RL PHY Frames in FDD mode and defined as  $k * N_{\text{RL\_BURST}}$  in TDD mode.

The Data Segment carries the R-DCH, R-DPICH, and R-ACKCH while the Control Segment carries the R-CQICH, the R-SFCH, the R-BFCH, the R-REQCH, the R-PICH, and the R-ACH.

### 9.4.1.2.6 Hop-port indexing

The subcarriers in each carrier of each OFDM symbol will also use a second indexing scheme known as hop-port indexing. In this scheme, each carrier in each OFDM symbol consists of  $Q_{SDMA}N_{CARRIER\_SIZE}$  individually-modulated hop-ports. Here  $Q_{SDMA}$  is equal to  $RLNumSDMADimensions$ , which is part of the public data of the Overhead Messages Protocol on that carrier. The hop-ports in each carrier are indexed from 0 through  $Q_{SDMA}N_{CARRIER\_SIZE} - 1$ . The hop-port with index  $p$  in the carrier with CarrierIndex  $k$  is sometimes represented by the pair  $(k,p)$ . An order is defined on the set of such pairs by saying that  $(k_0,p_0) < (k_1,p_1)$  if either of the following two conditions is satisfied:

1.  $k_0 < k_1$ , or
2.  $k_0 = k_1$  and  $p_0 < p_1$ .

There is a mapping between the  $Q_{SDMA}N_{CARRIER\_SIZE}$  hop-ports and the  $N_{CARRIER\_SIZE}$  subcarriers in each carrier, called a hop-permutation, which changes every RL PHY Frame and is different for different sectors. The sequence of hop-permutations, also called the hopping sequence, is described in 9.4.1.3. The hopping permutation used for different physical layer channels may correspond to different sectors, and the sector to be used is specified individually in the description of each physical layer channel.

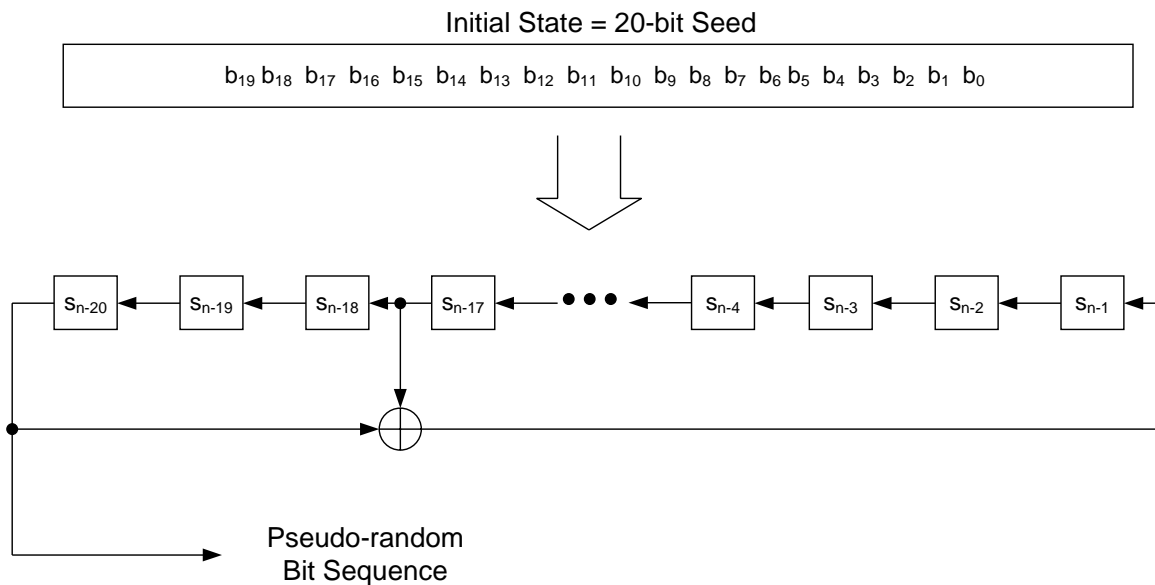
### 9.4.1.3 Hopping sequence generation

The hopping sequence will be described as a mapping from the set of hop-ports to the set of subcarriers. The reverse link implements block hopping. In this scheme, the set of non-guard hop-ports is divided into groups of  $N_{BLOCK}$  consecutive hop-ports, each of which is denoted as a block. The hop-permutation will map a block of hop-ports to a group of subcarriers with consecutive indices. This group of subcarriers will also be referred to as a block. Furthermore, the hop permutation will remain constant for the duration of the RL PHY Frame. In this design, therefore, a group of hop-ports spanning a RL PHY Frame worth of OFDM symbols in time and  $N_{BLOCK}$  hop-ports in hop-port space are mapped to neighboring subcarriers in the time-frequency grid. This group of  $N_{BLOCK}N_{FRAME,R}$  hop-ports shall be referred to as a tile for all RL PHY Frames except those with index 0 in FDD mode. RL PHY Frames with index 0 within a superframe span  $(N_{FRAME,R} + N_{PREAMBLE})$  OFDM symbols in FDD mode. For these RL PHY Frames, the group of  $N_{BLOCK}(N_{FRAME,R} + N_{PREAMBLE})$  hop-ports shall be referred to as a tile.

#### 9.4.1.3.1 Common permutation generation algorithm

Some of the permutations used for RL hopping shall be generated using a common permutation generation algorithm. The algorithm takes a 20-bit seed and a permutation size  $M$  as inputs and outputs a permutation of the set  $\{0, 1, \dots, M-1\}$ . The algorithm uses a linear feedback shift register to generate pseudorandom numbers, which in turn are used to generate pseudorandom permutations.

The 20-tap linear feedback shift register shall have a generator sequence of  $h(D) = 1 + D^{17} + D^{20}$ , as shown in Figure 112. The  $j$ 'th output  $s(j)$  of this shift register shall satisfy  $s(j) = s(j-17) \oplus s(j-20)$ . The initial state of the register shall generate the first output bit. A pseudorandom number  $x$  in  $\{0, 1, \dots, 2^n - 1\}$  for any  $n < 17$  can be generated by clocking the register  $n$  times, with the initial output bit being the LSB of  $x$  and the final ( $n$ 'th) output bit being the MSB of  $x$ .



**Figure 112 PN Register for generating pseudorandom bits**

The common permutation generation algorithm shall generate a permutation of size  $M$  as follows:

1. Initialization Steps:

- a. Let  $n$  be the smallest integer such that  $M \leq 2^n$ .
- b. Initialize an array  $A$  of size  $M$  with the numbers  $0, 1, 2, \dots, M-1$  (i.e.,  $A[0]=0, A[1]=1 \dots, A[M-1]= M-1$ )
- c. Initialize the PN register with the 20-bit seed.
- d. Initialize counter  $i$  to  $M-1$ .

2. Repeat the following steps until  $i=0$ .

- a. Find the smallest  $p$  such that  $i < 2^p$ .
- b. Initialize counter  $j$  to 0 and an output  $x$  to  $i+1$ .
- c. Repeat the following steps until  $j=3$  or until  $x \leq i$ .
- d. Clock the PN register  $n$  times to obtain an  $n$ -bit pseudorandom number. Set  $x$  to be the  $p$  LSBs of that number.
- e. Increment  $j$  by 1.
- f. If  $x > i$ , set  $x = x-i$ .
- g. Swap the  $i$ 'th and the  $x$ 'th elements in the array  $A$  (i.e.,  $\text{tmp} = A[x], A[x] = A[i], A[i] = \text{tmp}$ .)

1 h. Decrement counter  $i$  by 1.

- 2 3. The resulting array  $A$  is the output permutation  $P$ ; i.e.,  $P(x)$  is the location of  $x$  in array  $A$ .  
 3 For example, if  $A$  reads 345201, then  $P(0)=4$ ,  $P(1)=5$ ,  $P(2)=3$ ,  $P(3)=0$ ,  $P(4)=1$ , and  
 4  $P(5)=2$ .

#### 5 9.4.1.3.2 RL Hop Permutation Generation

6 RL Hop Permutation Generation is described in this section for both MultiCarrierOff and  
 7 MultiCarrierOn modes. In MultiCarrierOff mode, the hop permutation depends on several parameters  
 8 which are obtained from the Overhead Messages Protocol. In MultiCarrierOn mode, the hop  
 9 permutation on carrier  $c$ , where  $c$  is in  $\{0, 1, \dots, N_{\text{CARRIERS}} - 1\}$  depends on several parameters obtained  
 10 from the Overhead Messages Protocol for carrier  $c$ . These parameters may vary from carrier to  
 11 carrier.<sup>74</sup>

12 Space Division Multiple Access (SDMA) is supported on the Reverse Link. There are a total of  
 13  $N_{\text{CARRIER\_SIZE}} Q_{\text{SDMA}}$  hop-ports on carrier  $c$ , which are mapped to the  $N_{\text{CARRIER\_SIZE}}$  subcarriers  
 14 corresponding to carrier  $c$ . Here  $Q_{\text{SDMA}}$  is equal to  $\text{RLNumSDMADimensions}$ , which is part of the  
 15 public data of the Overhead Messages Protocol for carrier  $c$ . The set of hop ports shall be divided into  
 16  $Q_{\text{SDMA}}$  groups, each of which has  $N_{\text{CARRIER\_SIZE}}$  hop-ports and shall be referred to as an SDMA sub-  
 17 tree<sup>75</sup>. The sub-trees shall be numbered  $\{0, 1, \dots, Q - 1\}$  where  $Q = Q_{\text{SDMA}}$ . The hop port with index  
 18  $p$ <sup>76</sup> shall belong to sub-tree with index  $q$ , where  $q = \lfloor p / N_{\text{CARRIER\_SIZE}} \rfloor$ . Note that hop-ports in  
 19 different SDMA sub-trees can get mapped to the same subcarrier.

20 The set of  $N_{\text{CARRIER\_SIZE}}$  hop-ports in each carrier in each SDMA sub-tree is divided into  $S$  subbands,  
 21 where  $S$  shall be equal to  $\text{RLNumSubbands}$ , which is part of the public data of the Overhead  
 22 Messages Protocol for carrier  $c$ . The subbands shall be numbered  $\{0, 1, \dots, S - 1\}$  and each subband  
 23 shall have  $N_{\text{SUBBAND}}$  hop-ports, where  $N_{\text{SUBBAND}} = N_{\text{CARRIER\_SIZE}} / S$ . The hop-port with index  $p$  shall  
 24 belong to subband with index  $s$ , where  $s = \lfloor (p \bmod N_{\text{CARRIER\_SIZE}}) / N_{\text{SUBBAND}} \rfloor$ .

25 Furthermore, as mentioned previously, the reverse link implements block hopping. For this reason,  
 26 the set of  $N_{\text{SUBBAND}}$  hop-ports in each subband is divided into a number of blocks, each of which has  
 27  $N_{\text{BLOCK}}$ <sup>77</sup> hop-ports. The blocks shall be numbered  $\{0, 1, \dots, B - 1\}$  where  $B = N_{\text{SUBBAND}} / N_{\text{BLOCK}}$ . The  
 28 hop-port with index  $p$  shall belong to block with index  $b$ , where  $b = \lfloor (p \bmod N_{\text{SUBBAND}}) / N_{\text{BLOCK}} \rfloor$ .

29 The index of the hop port  $p$  within the block which it belongs to shall be denoted as  $r$ , where  
 30  $r = p \bmod N_{\text{BLOCK}}$ . Thus, there is a one-to-one correspondence between hop-port  $p$  and the tuple  $(c,$

<sup>74</sup> A parameter that can vary from carrier to carrier should be indexed by the carrier index  $c$ . However, for convenience of notation, the index  $c$  is dropped and the parameter is assumed to correspond to the carrier of interest. For example,  $Q_{\text{SDMA}}$  should be interpreted as  $Q_{\text{SDMA}}(c)$  when generating the hop permutation for hop-ports in carrier  $c$ , and should be obtained from the Overhead Messages Protocol for carrier  $c$ .

<sup>75</sup> The term “sub-tree” is used since the  $Q_{\text{SDMA}} N_{\text{CARRIER\_SIZE}}$  hop-ports are part of a “channel tree” defined by the RTC MAC protocol.

<sup>76</sup> Here “hop-port  $p$ ” should be interpreted as “hop-port  $p$  on carrier  $c$ .” The phrase “on carrier  $c$ ” will be omitted for convenience of notation.

<sup>77</sup> Note that the value of  $N_{\text{BLOCK}}$  used here corresponds to the Reverse Link and this value may be different from the value of  $N_{\text{BLOCK}}$  used in section 9.3.2.5.1 which describes hop permutation generation for the Forward Link.

1 q, s, b, r). For the rest of this document, the two notations are used interchangeably and “hop-port (c,  
2 q, s, b, r)” shall be used to refer to hop-port p on carrier c, where

$$3 \quad p = qN_{CARRIER\_SIZE} + sN_{SUBBAND} + bN_{BLOCK} + r.$$

4 The hop-ports within each subband shall be divided into two groups: non-guard hop-ports and guard  
5 hop-ports. The guard hop-ports shall be mapped to either the guard subcarriers or the quasi-guard  
6 subcarriers. The individual elements of this mapping are not specified since these hop-ports shall not  
7 be modulated.

8 A hop-port (c, q, s, b, r) shall be mapped to a guard subcarrier or a quasi-guard subcarrier either if:<sup>78</sup>

$$9 \quad b > B - 1 - \left\lfloor \frac{N_{GUARD} / N_{BLOCK}}{S} \right\rfloor$$

10 or if:

$$11 \quad b = B - 1 - \left\lfloor \frac{N_{GUARD} / N_{BLOCK}}{S} \right\rfloor \text{ and } \left| \frac{S}{2} - \frac{1}{4} - s \right| > \frac{S - [(N_{GUARD} / N_{BLOCK}) \bmod S]}{2}$$

12 The hop-ports that are not guard hop-ports shall be referred to as non-guard hop-ports. Note that hop-  
13 ports in a block are either all guard hop-ports or all non-guard hop-ports. A hop-port block consisting  
14 of only non-guard hop-ports shall be referred to as a non-guard hop-port block. The number of non-  
15 guard hop-port blocks in subband s shall be denoted as  $B_{NON-GUARD}(s)$ . Note that  $B_{NON-GUARD}(s) \leq B$   
16 and a hop-port (c, q, s, b, r) is non-guard if  $0 \leq b \leq B_{NON-GUARD}(s) - 1$ . Also note that  $B_{NON-GUARD}(s)$   
17 does not depend on the carrier index c.

18 Furthermore, some non-guard hop-ports may be allocated to the control segment (as described in  
19 9.4.1.3.2.1) in any given interlace. The non-guard hop-ports not allocated to the control segment in a  
20 given interlace shall be referred to as usable hop-ports<sup>79</sup> for that interlace.

21 Let  $H^{ij}(c, q, s, b, r)$  denote the subcarrier allocated to non-guard hop-port (c, q, s, b, r) in RL PHY  
22 Frame j in superframe i.  $H^{ij}$  is referred to as the hop permutation and shall be given by the following  
23 equation:

$$24 \quad H^{ij}(c, q, s, b, r) = cN_{CARRIER\_SIZE} + \frac{N_{GUARD}}{2} + N_{BLOCK} H_{GLOBAL}^{ij}(c, q, s, H_{SECTOR}^{ij}(c, q, s, b)) + r$$

---

<sup>78</sup> The idea behind these equations is that all subbands have approximately the same number of non-guard hop-ports. When  $(N_{GUARD} / N_{BLOCK})$  is a multiple of S, the first equation ensures that the highest numbered blocks in each subband are mapped to the guard subcarriers. In an asymmetric situation when  $(N_{GUARD} / N_{BLOCK})$  is not a multiple of S, the second equation ensures that the subbands most distant from the center of the carrier have one additional guard hop-port block.

<sup>79</sup> Note that “usable hop-ports” refer to hop-ports that can be used by the data segment. Some hop-ports which are not usable are actually used by the control segment. Contrast this with the definition of “usable subcarriers,” which are defined as subcarriers that can be used either by the data segment or control segment.

1 Here  $H_{\text{SECTOR}}^{ij}(c, q, s, b)$  is a permutation of non-guard hop-port blocks  $b$  in the SDMA sub-tree  $q$ ,  
 2 carrier  $c$  and subband  $s$ . The generation of this permutation is described in 9.4.1.3.2.4.

3  $H_{\text{GLOBAL}}^{ij}(c, q, s, b')$  is a permutation of all non-guard hop-port blocks in all subbands in carrier  $c$  and  
 4 SDMA sub-tree  $q$ . The generation of  $H^{ij}$  is different for different values of  
 5  $\text{RLDiversityHoppingMode}$ , which is part of the public data of the Overhead Messages Protocol for  
 6 carrier  $c$ . The generation of this permutation is described in 9.4.1.3.2.2 and 9.4.1.3.2.3.

#### 7 **9.4.1.3.2.1 Control Segment Hopping**

8 The generation of the permutation  $H_{\text{GLOBAL}}^{ij}$  depends on whether the CDM control segment is present  
 9 in RL PHY Frame  $j$ . The RL PHY Frames which contain a CDM control segment are specified in  
 10 9.4.1.2.5.

11 If the control segment is present in a RL PHY Frame, then an integer number of hop-port subbands  
 12 shall be allocated to the control segment in each carrier. This number, denoted  $N_{\text{CTRL-SUBBANDS}}$  shall be  
 13 equal to  $\text{NumRLControlSubbands}$ , which is part of the public data of the Overhead Messages  
 14 Protocol for carrier  $c$ . All non-guard hop-ports in the subbands that satisfy

$$15 \left| \frac{S}{2} - \frac{1}{4} - s \right| < \frac{N_{\text{CTRL-SUBBANDS}}}{2}$$

16 shall be allocated to the control segment. If the control segment is absent in a RL PHY Frame, then  
 17  $N_{\text{CTRL-SUBBANDS}}$  shall be equal to zero, and consequently no hop-ports shall be allocated to the control  
 18 segment.

19 Let  $S_{\text{MIN-CTRL-SUBBAND}}$  be the subband with the lowest index allocated to the control segment in  
 20 carrier  $c$ . (When  $N_{\text{CTRL-SUBBANDS}} = 0$ ,  $S_{\text{MIN-CTRL-SUBBAND}}$  shall be set to  $S/2$ .) Thus all non-guard hop-  
 21 ports in subbands  $\{S_{\text{MIN-CTRL-SUBBAND}}, S_{\text{MIN-CTRL-SUBBAND}} + 1, \dots, S_{\text{MIN-CTRL-SUBBAND}} + N_{\text{CTRL-SUBBANDS}} - 1\}$   
 22 shall be allocated to the control segment. The number of hop-port-blocks allocated to the control  
 23 segment in carrier  $c$  equals  $N_{\text{CTRL-HOP-PORT-BLOCKS}}$ , where

$$24 N_{\text{CTRL-HOP-PORT-BLOCKS}} = \sum_{k=0}^{N_{\text{CTRL-SUBBANDS}}-1} B_{\text{NON-GUARD}}(S_{\text{MIN-CTRL-SUBBAND}} + k).$$

25 These hop-ports shall be mapped to a contiguous set of subcarriers as follows.

- 26 1. Find  $\text{TMP} = [(\text{RLSectorHopSeed} * 4 * 64 * 4096 + c * 64 * 4096 + j * 4096 + (i \bmod 4096))$   
 27  $* 2654435761] \bmod 2^{32}$ . Set  $\text{SEED}_{\text{CONTROL}}$  to be the 20 LSBs of the bit-reversed value of  
 28  $\text{TMP}$  in a 32-bit representation, i.e.,  $\text{SEED}_{\text{CONTROL}} = [\text{Bit-Reverse}(\text{TMP})] \bmod 2^{20}$ . Here  
 29  $\text{RLSectorHopSeed}$  is part of the public data of the Overhead Messages Protocol for  
 30 carrier  $c$ .
- 31 2. Set  $S_{\text{SWAP-LOCATION}} = \min(\text{SEED}_{\text{CONTROL}} \bmod S, S - N_{\text{CTRL-SUBBANDS}})$ .

- 1           3. Generate a permutation  $H_{\text{SUBBAND}}^{ij}$  as described in 9.4.1.3.2.2. The contiguous set of  
 2  $N_{\text{CTRL-SUBCARRIERS}}$  subcarriers indexed  $f_{\text{MIN-CTRL}}$  to  $(f_{\text{MIN-CTRL}} + N_{\text{CTRL-SUBCARRIERS}} - 1)$  shall  
 3 be allocated to the control segment, where  $N_{\text{CTRL-SUBCARRIERS}} = N_{\text{BLOCK}} N_{\text{CTRL-HOP-PORT-}}$   
 4  $\text{BLOCKS}$  and

$$f_{\text{MIN-CTRL}} = cN_{\text{CARRIER\_SIZE}} + \frac{N_{\text{GUARD}}}{2} + N_{\text{BLOCK}} \sum_{k: H_{\text{SUBBAND}}^{ij}(k) < S_{\text{SWAP-LOCATION}}} B_{\text{NON-GUARD}}(k)$$

#### 6 **9.4.1.3.2.2 Generation of $H_{\text{GLOBAL}}^{ij}$ when RLDiversityHoppingMode is off**

7 The permutation  $H_{\text{GLOBAL}}^{ij}(c, q, s, b)$  shall be generated for carrier  $c$  as follows when  
 8 RLDiversityHoppingMode is off:

- 9           1. Determine  $S_{\text{SWAP-LOCATION}}$  and  $S_{\text{MIN-CTRL-SUBBAND}}$  for carrier  $c$  as described in 9.4.1.3.2.1.
- 10           2. When the control segment is absent in RL PHY Frame  $j$  of superframe  $i$ , generate a  
 11 permutation  $H_{\text{SUBBAND}}^{ij}$ <sup>80</sup> according to the following procedure:
- 12           a. Let  $[p_{11} p_{10} p_9 p_8 p_7 p_6 p_5 p_4 p_3 p_2 p_1 p_0]$  denote the 12-bit PilotPN of the sector, with  
 13  $p_{11}$  being the MSB and  $p_0$  being the LSB. Define an integer  $p$  to be equal to  $[p_{11} p_{10}$   
 14  $p_9 p_8 p_7 p_6 p_5 p_4 p_3 p_2 p_1 p_0]$  if RLIIntraCellCommonHopping is off, and to be  $[p_{11} p_{10}$   
 15  $p_9 p_8 0 0 0 p_4 p_3 p_2 p_1 p_0]$  if RLIIntraCellCommonHopping is on. Here,  
 16 RLIIntraCellCommonHopping is part of the public data of the Overhead Messages  
 17 Protocol for carrier  $c$ .
- 18           b. Set  $\text{TMP} = [(c * 4096 + p) * 2654435761] \bmod 2^{32}$ .
- 19           c. Set  $\text{SEED}_{\text{GLOBAL}}$  to be the 20 LSBs of the bit-reversed value of  $\text{TMP}$  in a 32-bit  
 20 representation, i.e.,  $\text{SEED}_{\text{GLOBAL}} = [\text{Bit-Reverse}(\text{TMP})] \bmod 2^{20}$ .
- 21           d.  $H_{\text{SUBBAND}}^{ij}$  is the permutation of size  $S$  generated using the common permutation  
 22 generation algorithm described in 9.4.1.3.1 with seed  $\text{SEED}_{\text{GLOBAL}}$ .
- 23           3. When the control segment is present in RL PHY Frame  $j$  of superframe  $i$ , generate a  
 24 permutation  $H_{\text{SUBBAND}}^{ij}$ <sup>81</sup> which shall satisfy  $H_{\text{SUBBAND}}^{ij}(s) = s$  except for the following  
 25 subbands:
- 26           a.  $H_{\text{SUBBAND}}^{ij}(S_{\text{MIN-CTRL-SUBBAND}} + k) = S_{\text{SWAP-LOCATION}} + k$  for  $0 \leq k \leq N_{\text{CTRL-SUBBANDS}} - 1$
- 27           b. If  $|S_{\text{SWAP-LOCATION}} - S_{\text{MIN-CTRL-SUBBAND}}| \geq N_{\text{CTRL-SUBBANDS}}$ , then  $H_{\text{SUBBAND}}^{ij}(S_{\text{SWAP-}}$   
 28  $\text{LOCATION} + k) = S_{\text{MIN-CTRL-SUBBAND}} + k$  for  $0 \leq k \leq N_{\text{CTRL-SUBBANDS}} - 1$ .

<sup>80</sup> When the control segment is absent  $H_{\text{SUBBAND}}^{ij}$  is a time-invariant sector-dependent pseudorandom permutation of subbands.

<sup>81</sup> When the control segment is present,  $H_{\text{SUBBAND}}^{ij}$  maps the control segment to a contiguous pseudo-random set of subbands. The data subbands displaced by the control segment are then mapped to the center of the carrier. All other subbands are left unchanged.



- 1 c. If  $0 \leq S_{\text{MIN-CTRL-SUBBAND}} - S_{\text{SWAP-LOCATION}} < N_{\text{CTRL-SUBBANDS}}$ , then  $H_{\text{SUBBAND}}^{ij}(S_{\text{SWAP-LOCATION}} + k) = S_{\text{SWAP-LOCATION}} + N_{\text{CTRL-SUBBANDS}} + k$  for  $0 \leq k < S_{\text{MIN-CTRL-SUBBAND}} - S_{\text{SWAP-LOCATION}}$
- 2
- 3
- 4 d. If  $0 \leq S_{\text{SWAP-LOCATION}} - S_{\text{MIN-CTRL-SUBBAND}} < N_{\text{CTRL-SUBBANDS}}$ , then  $H_{\text{SUBBAND}}^{ij}(S_{\text{SWAP-LOCATION}} + N_{\text{CTRL-SUBBANDS}} - 1 - k) = S_{\text{SWAP-LOCATION}} - 1 - k$  for  $0 \leq k < S_{\text{SWAP-LOCATION}} - S_{\text{MIN-CTRL-SUBBAND}}$
- 5
- 6

7 4. 
$$H_{\text{GLOBAL}}^{ij}(c, q, s, b) = \left[ \sum_{k: H_{\text{SUBBAND}}^{ij}(k) < H_{\text{SUBBAND}}^{ij}(s)} B_{\text{NON-GUARD}}(k) \right] + b$$

8 **9.4.1.3.2.3 Generation of  $H_{\text{GLOBAL}}^{ij}$  when RLDiversityHoppingMode is on**

9 When RLDiversityHoppingMode is on,  $H_{\text{GLOBAL}}^{ij}(c, q, s, b)$  is a permutation of the non-guard-hop-  
 10 port blocks in all subbands in carrier  $c$  of SDMA sub-tree  $q$ . The generation of  $H_{\text{GLOBAL}}^{ij}$  will be  
 11 different for different values of RLSectorHopSeed, which is part of the public data of the Overhead  
 12 Messages Protocol for carrier  $c$ .  $H_{\text{GLOBAL}}^{ij}$  shall be generated as follows:

- 13 1. Determine  $S_{\text{SWAP-LOCATION}}$ ,  $S_{\text{MIN-CTRL-SUBBAND}}$  and  $N_{\text{CTRL-HOP-PORT-BLOCKS}}$  for carrier  $c$  as  
 14 described in 9.4.1.3.2.1. Generate a permutation  $H_{\text{SUBBAND}}^{ij}$  as described in 9.4.1.3.2.2.  
 15 Determine  $B_{\text{SWAP-LOCATION}}$  and  $B_{\text{MIN-CTRL-SUBBAND}}$ , where

16 
$$B_{\text{SWAP-LOCATION}} = \sum_{k: H_{\text{SUBBAND}}^{ij}(k) < H_{\text{SUBBAND}}^{ij}(s)} B_{\text{NON-GUARD}}(k) \text{ and}$$

17 
$$B_{\text{MIN-CTRL-SUBBAND}} = \sum_{k=0}^{S_{\text{MIN-CTRL-SUBBAND}}-1} B_{\text{NON-GUARD}}(k)$$

- 18 2. If RLSectorHopSeed is not equal to 1111 (in binary notation), set  $\text{TMP} =$   
 19  $[(\text{RLSectorHopSeed} * 4 * 64 * 4096 + c * 64 * 4096 + j * 4096 + (i \bmod 4096)) * 2654435761]$   
 20  $\bmod 2^{32}$ .
- 21 3. When RLSectorHopSeed is equal to 1111, set  $\text{TMP} = [(\text{RLSectorHopSeed} * 4 * 64 * 4096 +$   
 22  $c * 64 * 4096 + j * 4096 + P_{\text{SECTOR}}) * 2654435761] \bmod 2^{32}$ , where the 12-bit quantity  $P_{\text{SECTOR}}$   
 23 shall be computed as described in 9.4.1.3.2.4.
- 24 4. Set  $\text{SEED}_{\text{GLOBAL}}$  to be the 20 LSBs of the bit-reversed value of  $\text{TMP}$  in a 32-bit  
 25 representation, i.e.,  $\text{SEED}_{\text{GLOBAL}} = [\text{Bit-Reverse}(\text{TMP})] \bmod 2^{20}$ .
- 26 5. Generate a permutation  $\pi$  of size  $\left[ \sum_{k=0}^{S-1} B_{\text{NON-GUARD}}(k) \right] - N_{\text{CTRL-HOP-PORT-BLOCKS}}$  using  
 27 the common permutation generation algorithm described in 9.4.1.3.1 with seed  
 28  $\text{SEED}_{\text{GLOBAL}}$ .

- 1           6.  $H_{\text{GLOBAL}}^{ij}(c, q, s, b) = P(\beta)^{82}$ , where  $\beta = b + \sum_{k=0}^{s-1} B_{\text{NON-GUARD}}(k)$  and
- 2           a. If  $B_{\text{MIN-CTRL-SUBBAND}} \leq \beta < B_{\text{MIN-CTRL-SUBBAND}} + N_{\text{CTRL-HOP-PORT-BLOCKS}}$ , then  $P(\beta) = (\beta -$   
3            $B_{\text{MIN-CTRL-SUBBAND}}) + B_{\text{SWAP-LOCATION}}$
- 4           b. If  $\beta < B_{\text{MIN-CTRL-SUBBAND}}$ , then  
5           i.  $P(\beta) = \pi(\beta)$  if  $\pi(\beta) < B_{\text{SWAP-LOCATION}}$   
6           ii.  $P(\beta) = \pi(\beta) + N_{\text{CTRL-HOP-PORT-BLOCKS}}$  if  $\pi(\beta) \geq B_{\text{SWAP-LOCATION}}$
- 7           c. If  $\beta \geq B_{\text{MIN-CTRL-SUBBAND}} + N_{\text{CTRL-HOP-PORT-BLOCKS}}$ , then  
8           i.  $P(\beta) = \pi(\beta - N_{\text{CTRL-HOP-PORT-BLOCKS}})$  if  $\pi(\beta - N_{\text{CTRL-HOP-PORT-BLOCKS}}) < B_{\text{SWAP-}}$   
9           LOCATION  
10           ii.  $P(\beta) = \pi(\beta - N_{\text{CTRL-HOP-PORT-BLOCKS}}) + N_{\text{CTRL-HOP-PORT-BLOCKS}}$  if  $\pi(\beta - N_{\text{CTRL-HOP-}}$   
11           PORT-BLOCKS)  $\geq B_{\text{SWAP-LOCATION}}$

#### 12 9.4.1.3.2.4 Generation of $H_{\text{SECTOR}}^{ij}$

13  $H_{\text{SECTOR}}^{ij}(c, q, s, .)$  is a permutation of the non-guard hop-port blocks in subband  $s$  of carrier  $c$  of  
14 SDMA sub-tree  $q$ . The generation of  $H_{\text{SECTOR}}^{ij}$  will be different for different values of  
15 RLIntraCellCommonHopping, which is part of the public data of the Overhead Messages Protocol for  
16 carrier  $c$ .<sup>83</sup>

17 The PilotPN of the sector of interest is XORed bitwise with the 12 LSBs of the superframe index  $i$  to  
18 obtain a 12-bit number  $[b_{11} b_{10} b_9 b_8 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0]$  denoted as  $P_{\text{off}}$ . The 12-bit number  $[b_{11} b_{10} b_9$   
19  $b_8 i_7 i_6 i_5 b_4 b_3 b_2 b_1 b_0]$ , where  $i_7 i_6 i_5$  are the bits with indices 7,6 and 5 respectively in the superframe  
20 index  $i$ , is denoted as  $P_{\text{on}}$ . The permutation shall be generated as follows:

- 21 1. If RLIntraCellCommonHopping is off, set  $P_{\text{SECTOR}} = P_{\text{off}}$ . Otherwise, set  $P_{\text{SECTOR}} = P_{\text{on}}$ .
- 22 2. In TDD mode when RLSectorHopSeed is not equal to 1110, and in FDD mode, set  $\text{TMP} = [(4 * 4 * 16 * 64 * 4096 + q * 4 * 16 * 64 * 4096 + c * 16 * 64 * 4096 + s * 64 * 4096 + j * 4096 +$   
23  $P_{\text{SECTOR}}) * 2654435761] \bmod 2^{32}$ .
- 24  
25 3. In TDD mode when RLSectorHopSeed is equal to 1110<sup>84</sup>, set  $\text{TMP} = [(q * 4 * 16 * 64 * 4096$   
26  $+ c * 16 * 64 * 4096 + s * 64 * 4096 + \lfloor j / N_{\text{RL\_BURST}} \rfloor * 4096 + P_{\text{SECTOR}}) * 2654435761] \bmod$   
27  $2^{32}$ .

<sup>82</sup>  $P(\beta)$  first maps the hop port blocks allocated to the control segment to a contiguous set of subcarriers. The non-control hop port blocks are then assigned to the non-control subcarriers using a pseudo-random permutation  $\pi(\cdot)$

<sup>83</sup> When RLIntraCellCommonHopping is off, two sectors with different values of PilotPN have different hopping sequences. When RLIntraCellCommonHopping is on, sectors within the same cell have the same hopping sequences. For proper use of this mode, the operator should ensure that the PilotPNs of two sectors in the same cell differ only in the bits indexed 5,6, and 7.

<sup>84</sup> In TDD mode when RLSectorHopSeed and FLSectorHopSeed are both set to 1110, the FL hop permutations are “slaved” to the RL hop permutations for hop-ports on SDMA sub-tree 0 i.e., if a hop-port  $p$  is mapped to a

4. Set  $SEED_{SECTOR}$  to be the 20 LSBs of the bit-reversed value of TMP in a 32-bit representation, i.e.,  $SEED_{SECTOR} = [\text{Bit-Reverse}(\text{TMP})] \bmod 2^{20}$ .
5.  $H_{SECTOR}^{ij}(c, q, s, \cdot)$  is the permutation of size  $B_{NON-GUARD}(s)$  generated using the common permutation generation algorithm described in 9.4.1.3.1 with seed  $SEED_{SECTOR}$ .

#### 9.4.1.4 R-ACKCH

The R-ACKCH is used to acknowledge FL PHY Frames transmitted on the F-DCH. For the purpose of this section, the sector of interest is the Forward Link Serving Sector (FLSS), which may or may not be different from the Reverse Link Serving Sector (RLSS). For convenience of notation, the phrase “of the Forward Link Serving Sector” shall be omitted. Sector-dependent quantities such as PilotPN, hop-permutations etc., used in this section shall be interpreted as “PilotPN of the FLSS,” “hop-permutations of the FLSS” etc.

##### 9.4.1.4.1 R-ACKCH subcarrier allocation

In each RL PHY Frame, the R-ACKCH shall be allocated a number of contiguous subcarrier groups for the duration for the RL PHY Frame. Each such group of contiguous subcarriers shall comprise of  $N_{BLOCK, R-ACKCH}$  subcarriers and shall be referred to as an R-ACKCH block. The group of  $N_{BLOCK, R-ACKCH} N_{FRAME, R}$  subcarriers spanning  $N_{BLOCK, R-ACKCH}$  subcarriers in frequency and  $N_{FRAME, R}$  OFDM Symbols in time shall be referred to as an R-ACKCH tile.

The number of R-ACKCH tiles allocated to the R-ACKCH in carrier  $c$  shall be  $N_{TILES}$ , where  $N_{TILES} = 0$  if  $NumRACKBaseNodes = 0$  and

$$N_{TILES} = \max \left( \frac{2 NumRACKBaseNodes}{N_{R-ACKCH-SUBTILE-DURATION} N_{BLOCK, R-ACKCH}}, \frac{N_{FRAME, R}}{N_{R-ACKCH-SUBTILE-DURATION}} \right)$$

otherwise. Here  $NumRACKBaseNodes$  shall be specified by the RCC MAC protocol. The constant  $N_{R-ACKCH-SUBTILE-DURATION}$  is the number of OFDM symbols allocated to each subtile, where the definition of subtile is as described in 9.4.1.4.2.

The set of R-ACKCH tiles in each carrier  $c$  of RL PHY Frame  $j$  in superframe with index  $i$  shall be indexed from 0 to  $N_{TILES} - 1$  and shall be determined according to the following procedure:

1. Compute  $P_{off}$  as described in 9.4.1.3.2.4. Compute  $TMP = [(4*4*64*4096 + c*64*4096 + j*4096 + P_{off}) * 2654435761] \bmod 2^{32}$ . Set  $SEED_{RACKCH}$  to be the 20 LSBs of the bit-reversed value of TMP in a 32-bit representation, i.e.,  $SEED_{RACKCH} = [\text{Bit-Reverse}(\text{TMP})] \bmod 2^{20}$ .
2. Generate a permutation  $H_{RACKCH-SUBBANDS}^{ij}$  of size  $S$  using the common permutation generation algorithm described in 9.4.1.3.1 with seed  $SEED_{RACKCH}$ . Here  $S = N_{CARRIER\_SIZE} / N_{SUBBAND}$  is the number of subbands in the carrier.

---

subcarrier  $f$  on the RL in a “burst” of RL PHY Frames, then that hop port is mapped to the same subcarrier  $f$  on the FL in the subsequent burst of FL PHY Frames as well.

- 1           3. Generate a permutation  $H_{\text{RACKCH-BLOCKS}}^{ij}$  of size B using the common permutation  
2           generation algorithm described in 9.4.1.3.1 with seed  $\text{SEED}_{\text{RACKCH}}$ . Here  $B = N_{\text{SUBBAND}} /$   
3            $N_{\text{BLOCK}}$  is the number of hop-port blocks in a subband.
- 4           4. Initialize counters  $i$  and  $t_{\text{TILE}}$  to 0. Also initialize counters  $j_0, j_1, \dots, j_{S-1}$  to 0.
- 5           5. Repeat the following steps till  $t_{\text{TILE}} = N_{\text{TILES}}$ .
  - 6           a. Set  $s = H_{\text{RACKCH-SUBBANDS}}^{ij}(i)$  and  $b = H_{\text{RACKCH-BLOCKS}}^{ij}(j_s)$ . Set  $\text{FLAG}_{\text{RACKCH}}$  to TRUE.
  - 7           b. Increment  $i$  and  $j_s$  by 1. If  $i = S$ , then set  $i$  to 0.
  - 8           c. Set  $\text{FLAG}_{\text{RACKCH}}$  to FALSE if any one of the following three conditions is satisfied:
    - 9           i. Hop port  $(c, 0, s, b, 0)$  is a guard hop-port.
    - 10           ii. Subband  $s$  is allocated to the control segment.
    - 11           iii. The bit with index  $s$  of the RLRestrictedSetBitmap is set to 1, Here  
12           RLRestrictedSetBitmap is part of the public data of the Overhead Messages  
13           Protocol for carrier  $c$ .
  - 14           d. If  $\text{FLAG}_{\text{RACKCH}}$  is TRUE, then
    - 15           i. Allocate the set of subcarriers  $H^{ij}(c, 0, s, b, 0)$  to  $[H^{ij}(c, 0, s, b, 0) + N_{\text{BLOCK,RACKCH}}$   
16            $-1]$  to the R-ACKCH for the duration of the RL PHY Frame for all RL PHY  
17           Frames other than those with index 0 in FDD mode. Here  $H^{ij}$  is the hop  
18           permutation for RL PHY Frame  $j$  in superframe with index  $i$ , as described in  
19           9.4.1.3.2.
    - 20           ii. For RL PHY Frames with index 0 within the superframe in FDD mode, allocate  
21           the set of subcarriers  $H^{ij}(c, 0, s, b, 0)$  to  $[H^{ij}(c, 0, s, b, 0) + N_{\text{BLOCK,RACKCH}} - 1]$  only  
22           for the OFDM symbols indexed  $N_{\text{PREAMBLE}}$  through  $N_{\text{PREAMBLE}} + N_{\text{FRAME,R}} - 1$  in  
23           the PHY Frame.
    - 24           iii. The R-ACKCH tile index of this tile shall be  $t_{\text{TILE}}$ .
    - 25           iv. Increment  $t_{\text{TILE}}$  by 1.

#### 26   9.4.1.4.2 R-ACKCH indexing

27   Each R-ACKCH tile shall further be divided into a number of R-ACKCH subtiles, each of which  
28   spans  $N_{\text{BLOCK,R-ACKCH}}$  subcarriers in frequency and  $N_{\text{R-ACKCH-SUBTILE-DURATION}}$  OFDM symbols in time.  
29   The subtiles in each R-ACKCH tile shall be indexed from 0 to  $N_{\text{SUBTILES}} - 1$ , where  $N_{\text{SUBTILES}} =$   
30    $(N_{\text{FRAME,R}} / N_{\text{R-ACKCH-SUBTILE-DURATION}})$ . The OFDM Symbol with index  $t$  shall belong to subtile with  
31   index  $k_{\text{SUBTILE}}$ , where  $k_{\text{SUBTILE}} = \lfloor t / N_{\text{R-ACKCH-SUBTILE-DURATION}} \rfloor$ . Each subtile allocated to the R-  
32   ACKCH is thus indexed by the tuple  $(c, t_{\text{TILE}}, k_{\text{SUBTILE}})$  where  $c$  is the carrier containing the tile,  $t_{\text{TILE}}$   
33   is the R-ACKCH tile index within the carrier and  $k_{\text{SUBTILE}}$  is the subtile index within that tile.

34   Each sub-tile has a total of  $L = N_{\text{BLOCK,R-ACKCH}} N_{\text{R-ACKCH-SUBTILE-DURATION}}$  subcarriers. Exponential  
35   sequences of length  $L$ , may be used to modulate these subcarriers. The combination of a sub-tile and a  
36   specific exponential sequence of length  $L$  shall be referred to as an “R-ACKCH resource.” An R-  
37   ACKCH resource indexed by the tuple  $(c, t_{\text{TILE}}, k_{\text{SUBTILE}}, \omega)$  shall correspond to the usage of  
38   exponential sequence  $E_{\omega}^L$  to modulate the subtile  $(c, t_{\text{TILE}}, k_{\text{SUBTILE}})$ . Here the sequence  $E_{\omega}^L$  is a

1 sequence of length  $L$ , whose  $i$ 'th element  $E_{\omega}^L(i)$  is given by  $E_{\omega}^L(i) = e^{-2\pi j \omega i / L}$  where  $0 \leq i < L$ , and  $j$   
 2 denotes the complex number  $(0,1)$ .

### 3 9.4.1.4.3 R-ACKCH resource assignment

4 R-ACKCH transmissions are determined by an RACKBaseNodeIndex and a corresponding  
 5 RACKVal specified by the RCC MAC protocol. An AT may be assigned zero, one or more  
 6 RACKBaseNodeIndices in any RL PHY Frame.

7 An AT which is assigned an RACKBaseNodeIndex  $D_{R-ACKCH}$  shall be assigned  $N_{SUBTILES}$  R-ACKCH  
 8 resources according to the following procedure:

- 9 1. Set  $g = \lfloor D_{R-ACKCH} / N_{TILES} \rfloor$  and  $u = D_{R-ACKCH} \bmod N_{TILES}$ .
- 10 2. Compute  $TMP = [(3*64*4*64*4096 + (g \bmod 64)*4*64*4096 + c*64*4096 + j*4096 +$   
 11  $(i \bmod 4096)) * 2654435761] \bmod 2^{32}$ . Set  $SEED_{RACKCH-ROWS}$  to be the 20 LSBs of the  
 12 bit-reversed value of TMP in a 32-bit representation, i.e.,  $SEED_{RACKCH-ROWS} = [\text{Bit-}$   
 13  $\text{Reverse}(TMP)] \bmod 2^{20}$ .
- 14 3. Generate a permutation  $H_{RACKCH-ROWS}^{ij}$  of size  $N_{TILES}$  using the common permutation  
 15 generation algorithm described in 9.4.1.3.1 with seed  $SEED_{RACKCH-ROWS}$ .
- 16 4. Compute  $TMP = [(2*64*4*64*4096 + (g \bmod 64)*4*64*4096 + c*64*4096 + j*4096 +$   
 17  $(i \bmod 4096)) * 2654435761] \bmod 2^{32}$ . Set  $SEED_{RACKCH-COLS}$  to be the 20 LSBs of the bit-  
 18 reversed value of TMP in a 32-bit representation, i.e.,  $SEED_{RACKCH-COLS} = [\text{Bit-}$   
 19  $\text{Reverse}(TMP)] \bmod 2^{20}$ .
- 20 5. Generate a permutation  $H_{RACKCH-COLS}^{ij}$  of size  $N_{SUBTILES}$  using the common permutation  
 21 generation algorithm described in 9.4.1.3.1 with seed  $SEED_{RACKCH-COLS}$ .
- 22 6. Compute  $TMP = [(1*4*64*4096 + c*64*4096 + j*4096 + (i \bmod 4096)) * 2654435761]$   
 23  $\bmod 2^{32}$ . Set  $SEED_{RACKCH-CODES}$  to be the 20 LSBs of the bit-reversed value of TMP in a  
 24 32-bit representation, i.e.,  $SEED_{RACKCH-CODES} = [\text{Bit-Reverse}(TMP)] \bmod 2^{20}$ .
- 25 7. Generate a permutation  $H_{RACKCH-CODES}^{ij}$  of size  $L/2$  using the common permutation  
 26 generation algorithm described in 9.4.1.3.1 with seed  $SEED_{RACKCH-CODES}$ .
- 27 8. Initialize a counter  $k$  to 0. Repeat the following steps until  $k = N_{SUBTILES}$ .  
 28 a. Compute  $t = (u-k) \bmod N_{TILES}$   
 29 b. Set  $t_{TILE} = H_{RACKCH-ROWS}^{ij}(t)$ ,  $k_{SUBTILE} = H_{RACKCH-COLS}^{ij}(k)$  and  $\omega = 2 * H_{RACKCH-}$   
 30  $\text{CODES}((g + t_{TILE}N_{SUBTILES} + k_{SUBTILE}) \bmod (L/2))$ .  
 31 c. Assign the R-ACKCH resource  $(c, t_{TILE}, k_{SUBTILE}, \omega)$  to the AT.  
 32 d. Increment  $k$  by 1.

#### 9.4.1.4.4 R-ACKCH modulation

An AT shall transmit a sequence  $X_{ACK}(c, t_{TILE}, k_{SUBTILE}, \omega)$  on each R-ACKCH resource  $(c, t_{TILE}, k_{SUBTILE}, \omega)$  assigned to it. The sequence  $X_{ACK}(c, t_{TILE}, k_{SUBTILE}, \omega)$  is an ON-OFF transmission specified by a bit RACKVal defined by the RCC MAC Protocol for each RACKBaseNodeIndex assigned to the AT. When RACKVal is equal to 1, the sequence  $X_{ACK}(c, t_{TILE}, k_{SUBTILE}, \omega)$  shall be

$$X_{ACK}(c, t_{TILE}, k_{SUBTILE}, \omega) = \sqrt{P_{RACKCH} N_{FFT} / N_{BLOCK-RACKCH}} E_{\omega}^L$$

where  $E_{\omega}^L$  is the exponential sequence of length L as defined in 9.4.1.4.2. and  $P_{RACKCH}$  is the power allocated to the R-ACKCH by the RCC MAC Protocol. When RACKVal is equal to 0, the sequence  $X_{ACK}(c, t_{TILE}, k_{SUBTILE}, \omega)$  shall be a sequence of L zeros.

The sequence  $X_{ACK}(c, t_{TILE}, k_{SUBTILE}, \omega)$  shall be used to modulate the L subcarriers in the subtile  $(c, t_{TILE}, k_{SUBTILE})$  according to the following procedure:

1. Initialize an OFDM symbol counter t to  $t_{START}$ , where  $t_{START}$  is the lowest indexed OFDM Symbol in the subtile. Initialize a subcarrier counter f to  $f_{START}$ , where  $f_{START}$  is the lowest indexed subcarrier in the subtile. Initialize a modulation symbol counter i to 0.
2. Repeat the following steps till  $i = L$ .
  - a. Modulate the subcarrier f in OFDM Symbol t with modulation symbol  $X_{ACK}^i(c, t_{TILE}, k_{SUBTILE}, \omega)$ . Here  $X_{ACK}^i(c, t_{TILE}, k_{SUBTILE}, \omega)$  is the  $i^{th}$  element in the sequence  $X_{ACK}(c, t_{TILE}, k_{SUBTILE}, \omega)$ .
  - b. Increment f by 1. If  $f = f_{START} + N_{BLOCK, R-ACKCH}$ , set f to  $f_{START}$  and increment t by 1.
  - c. Increment i by 1.

#### 9.4.1.5 Control segment modulation

The Control Segment carries the Access Channel (R-ACH), the Channel Quality Indicator Channel (R-CQICH), the Subband Feedback Channel (R-SFCH), the Beam Feedback Channel (R-BFCH), the Request Channel (R-REQCH) and the Pilot Channel (R-PICH).

The Control Segment is modulated in a Code Division Multiple Access (CDMA) fashion; i.e., transmissions from different access terminals are not orthogonal to each other. In MultiCarrierOff mode various channels in the Control Segment are generated in time domain, then added up and are converted to the frequency domain using a Discrete Fourier Transform (DFT) operation. The frequency domain sequence is then mapped to the subcarriers in the Control Segment assigned to the access terminal. In MultiCarrierOn mode various channels in the Control Segment are generated in time domain per carrier. These are then added up per carrier and are converted to the frequency domain using a Discrete Fourier Transform (DFT) operation, also per carrier. The frequency domain sequence is then mapped to the subcarriers in the Control Segment assigned to the access terminal.

In the frequency domain, in the  $c^{th}$  carrier, the Control Segment corresponding to each sector consists of NumRLControlSubbands contiguous subbands, where NumRLControlSubbands is public data of the Overhead Messages Protocol for carrier c for that sector. The AT uses the Control Segment of the sector in  $AS_{SYNCH}$  which has the smallest value of NumRLControlSubbands, where  $AS_{SYNCH}$  is as defined in 9.4.1. This value (the smallest value of NumRLControlSubbands) will henceforth be

referred to as  $N_{CTRL\_SUBBANDS}$ . A hopping sequence for the Control Segment, described in 9.4.1.3.2.1, is used while mapping the frequency-domain sequence to subcarriers. The Control Segment hopping sequence maps R-ACH, R-CQICH, R-BFCH, R-SFCH, R-REQCH and R-PICH to the Control Segment per carrier as specified by the RCC MAC Protocol. The MAC protocol also specifies the carrier on which the above channels are modulated and the power allocated per channel.

#### 9.4.1.5.1 Time-domain sequence generation

The following description is applicable to the time domain sequence generation in MultiCarrierOff mode and to the time domain sequence per carrier in MultiCarrierOn mode. Define  $N_{CTRL\_FFT}$ , such that  $N_{CTRL\_FFT} = A * 2^k$  where  $A = N_{SUBBAND}/128$  and  $k$  is the smallest integer such that  $2^k \geq N_{CTRL\_SUBBANDS\_AT}$ . Here,  $N_{CTRL\_SUBBANDS\_AT}$  is given by NumRLControlSubbandsUser, which is public data of the Active Set Management Protocol. Also define two integers  $f_{MIN-CTRL-AT}$  and  $N_{CTRL-SUBCARRIERS-AT}$  according to the following procedure:

1. Compute an integer  $TMP = (i_{SF} + i_f * 4096) * 2654435761 \bmod 2^{32}$ , where  $i_{SF}$  denotes the superframe index, and  $i_f$  denotes the index of the RL PHY Frame within the superframe.
2. Compute another integer  $TMP2$ , which is the bit-reversed value of  $TMP$  in a 32-bit representation.
3. Define  $N_{MIN-CTRL-SUBBAND-AT}$  as  $TMP2 + RLControlSubbandUserOffset \bmod (N_{CTRL\_SUBBANDS} - N_{CTRL\_SUBBANDS\_AT})$ , where  $RLControlSubbandUserOffset$  is public data of the Active Set Update Protocol.

4. Compute  $f_{MIN-CTRL-AT}$  as

$$f_{MIN-CTRL-AT} = f_{MIN-CTRL} + N_{BLOCK} \sum_{k=0}^{N_{MIN-CTRL-SUBBAND-AT}-1} B_{NON-GUARD}(S_{MIN-CTRL-SUBBAND} + k).$$

Here,  $f_{MIN-CTRL}$  and  $S_{MIN-CTRL-SUBBAND}$  are as defined in 9.4.1.3.2.1 and  $B_{NON-GUARD}(\cdot)$  is as defined in 9.4.1.3.2. The values of these parameters used correspond to the sector whose Control Segment is being used by the AT.

5. Compute  $N_{CTRL-SUBCARRIERS-AT}$  as

$$N_{CTRL-SUBCARRIERS-AT} = N_{BLOCK} \sum_{k=N_{MIN-CTRL-SUBBAND-AT}}^{N_{MIN-CTRL-SUBBAND-AT} + N_{CTRL\_SUBBANDS\_AT}} B_{NON-GUARD}(S_{MIN-CTRL-SUBBAND} + k).$$

The following sections discuss the modulation procedure for a specific carrier and for an individual instance of the channels specified by the AC MAC and RCC MAC Protocol. There could be more than one instance of the channels R-CQICH, R-BFCH and R-SFCH as specified by the RCC MAC Protocol. The described modulation procedure is repeated for every instance using the corresponding 10-bit value, MAC ID and PilotPN provided by the RCC MAC Protocol for these channels. In the following we assume an 11 bit MAC ID. In case the RCC MAC Protocol provides a MAC ID consisting of fewer than 11 bits, then the MAC ID is augmented to 11 bits by adding the requisite number of bits as MSBs. These added bits are set to 0.

### 9.4.1.5.1.1 Walsh sequence definition

Walsh sequences are used in the generation of the time-domain sequences for several physical layer channels carried in the Control Segment. A Walsh sequence  $W_i^N$ , where  $N$  is a power of 2 and  $i$  is a non-negative integer less than  $N$ , is a length  $N$  binary sequence taking on  $\{-1, 1\}$  which is given by the  $i$ -th column of the  $N \times N$  Hadamard matrix  $W^N$ . The  $N \times N$  Hadamard matrix  $W^N$  is conventionally defined by the following recursive relationship:

$$W^2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad W^{2N} = \begin{bmatrix} W^N & W^N \\ W^N & -W^N \end{bmatrix}$$

### 9.4.1.5.1.2 R-ACH binary sequence

The sequence  $X_{ACH}$  of R-ACH is obtained from the  $i$ -th Walsh sequence  $W_i^{1024}$  of length 1024, by repeating each entry of this sequence  $N_{CTRL\_FFT}$  times and scaling it to achieve the appropriate power  $P$ . The index  $i$  of the sequence is defined by the AC MAC Protocol. The resulting sequence is scrambled with a binary sequence  $S_{ACH}$  of length  $1024 * N_{CTRL\_FFT}$ . Let  $W_{k,i}^{1024}$  be the  $k$ -th element of the Walsh sequence  $W_i^{1024}$  with  $k$  within 0 and 1023, then define the sequence  $Z_{ACH}$  given by

$$Z_{ACH} = \sqrt{\frac{PN_{FFT}}{N_{CTRL\_SUBCARRIER}} \underbrace{\left( \underbrace{(-1)^{S_{ACH}^0} * W_{0,i}^{1024} \dots (-1)^{S_{ACH}^{N_{CTRL\_FFT}-1}} * W_{0,i}^{1024}}_{N_{CTRL\_FFT}} \dots \underbrace{(-1)^{S_{ACH}^{1024 * N_{CTRL\_FFT}}} * W_{1023,i}^{1024} \dots (-1)^{S_{ACH}^{1024 * N_{CTRL\_FFT} + N_{CTRL\_FFT} - 1}} * W_{1023,i}^{1024}}_{N_{CTRL\_FFT}} \right)}_{1024 * N_{CTRL\_FFT}}}$$

where  $P$  is the total power allocated by the AC MAC Protocol to transmit this sequence. The sequence  $X_{ACH}$  is obtained from  $Z_{ACH}$  by setting its last  $N_{CTRL\_FFT} * N_{SUBBAND}$  elements to zero.

$$X_{ACH} = \sqrt{\frac{PN_{FFT}}{N_{CTRL\_SUBCARRIER}} \underbrace{\left( \underbrace{(-1)^{S_{ACH}^0} * W_{0,i}^{1024} \dots (-1)^{S_{ACH}^{N_{CTRL\_FFT}-1}} * W_{0,i}^{1024}}_{N_{CTRL\_FFT}} \dots \underbrace{0 \dots 0}_{N_{SUBBAND} * N_{CTRL\_FFT}} \right)}_{1024 * N_{CTRL\_FFT}}}$$

The binary scrambling sequence  $S_{ACH}$  is generated as follows.

First, a binary sequence  $F_{ACH}$  of the length  $n = 512$  shall be generated using a 20-bit shift register that shall have a generator polynomial  $h(D) = D^{20} + D^{17} + D^{12} + D^{10} + 1$ ; i.e., the  $k$ -th output  $F_{ACH}^k$  of the register shall satisfy  $F_{ACH}^k = F_{ACH}^{k-20} \oplus F_{ACH}^{k-17} \oplus F_{ACH}^{k-12} \oplus F_{ACH}^{k-10}$ , for  $k=0, 1, \dots, n-1$ . Here the initial state  $[F_{ACH}^{-1}, \dots, F_{ACH}^{-20}]$  is given  $[1 \ 0 \ 0 \ f_4 \ f_3 \ f_2 \ f_1 \ f_0 \ p_{11} \ p_{10} \ p_9 \ p_8 \ p_7 \ p_6 \ p_5 \ p_4 \ p_3 \ p_2 \ p_1 \ p_0]$ . Here  $[p_{11} \ p_{10} \ p_9 \ p_8 \ p_7 \ p_6 \ p_5 \ p_4 \ p_3 \ p_2 \ p_1 \ p_0]$  is the 12-bit PilotPhase of the sector which is the target of the access probe for a given superframe, with LSB  $p_0$  and MSB  $p_{11}$  and  $[f_4 \ f_3 \ f_2 \ f_1 \ f_0]$  are the 5 LSBs of the binary representation for the index of the RL PHY Frame within the current superframe, with LSB  $f_0$ . Finally, the sequence  $S_{ACH}$  is obtained by repeating the  $n$  elements of  $F_{ACH}$   $2 * N_{CTRL\_FFT}$  times:



$$S_{ACH} = \left[ \underbrace{F_{ACH}^0 F_{ACH}^0 \dots F_{ACH}^0 F_{ACH}^0}_{2 * N_{CTRL\_FFT}} F_{ACH}^1 \dots F_{ACH}^1 F_{ACH}^1 \dots \dots \dots \underbrace{F_{ACH}^{n-1} F_{ACH}^{n-1} \dots F_{ACH}^{n-1} F_{ACH}^{n-1}}_{2 * N_{CTRL\_FFT}} \right]$$

In the absence of a R-ACH transmission request by AC MAC Protocol,  $X_{ACH}$  shall be an all-zero sequence of the same length.

#### 9.4.1.5.1.3 R-CQICH binary sequence

The sequence  $X_{CQI}$  of R-CQICH is obtained from the  $i$ -th Walsh sequence  $W_i^{1024}$  of length 1024, by repeating each entry of this sequence  $N_{CTRL\_FFT}$  times and scaling it to achieve the appropriate power  $P$ . The index  $i$  of the sequence takes on values from 0 to 1023 and is defined by ten-bit CQI value  $(b_9, b_8, b_7, b_6, \dots, b_1, b_0)$ :

$$i = \sum_{l=0}^9 b_l 2^l$$

The CQI value  $(b_9, b_8, b_7, b_6, \dots, b_1, b_0)$  is defined by the RCC MAC Protocol. The resulting sequence is scrambled with a binary sequence  $S_{CQI}$ . Let  $W_{k,i}^{1024}$  be the  $k$ -th element of the Walsh sequence  $W_i^{1024}$  with  $k$  within 0 and 1023, then  $X_{CQI}$  is given by

$$X_{CQI} = \sqrt{\frac{PN_{FFT}}{N_{CTRL\_SUBCARRISRAT}}} \left( \underbrace{(-1)^{S_{CQI}^0} * W_{0,i}^{1024} \dots (-1)^{S_{CQI}^{N_{CTRL\_FFT}-1}} * W_{0,i}^{1024}}_{N_{CTRL\_FFT}} \dots \underbrace{(-1)^{S_{CQI}^{1023 * N_{CTRL\_FFT}}} * W_{1023,i}^{1024} \dots (-1)^{S_{CQI}^{1024 * N_{CTRL\_FFT}-1}} * W_{1023,i}^{1024}}_{N_{CTRL\_FFT}} \right)$$

where  $P$  is the total power allocated by the RCC MAC Protocol to transmit this sequence. The binary scrambling sequence  $S_{CQI}$  is generated as follows. First, a binary sequence  $F_{CQI}$  of the length  $n = 512$  shall be generated using a 20-bit shift register that shall have a generator polynomial  $h(D) = D^{20} + D^{17} + D^{12} + D^{10} + 1$ ; i.e., the  $k$ -th output  $F_{CQI}^k$  of the register shall satisfy

$F_{CQI}^k = F_{CQI}^{k-20} \oplus F_{CQI}^{k-17} \oplus F_{CQI}^{k-12} \oplus F_{CQI}^{k-10}$ , for  $k=0, 1, \dots, n-1$ . Here the initial state  $[F_{CQI}^{-1}, \dots, F_{CQI}^{-20}]$  is given by  $[000f_4 f_3 f_2 f_1 f_0 p_{11} p_{10} p_9 p_8 p_7 p_6 p_5 p_4 p_3 p_2 p_1 p_0]$ . Here  $[p_{11} p_{10} p_9 p_8 p_7 p_6 p_5 p_4 p_3 p_2 p_1 p_0]$  is the 12-bit PilotPhase for the current superframe, corresponding to the PilotPN specified by the RCC MAC for this channel, with LSB  $p_0$  and MSB  $p_{11}$ . Also,  $[f_4 f_3 f_2 f_1 f_0]$  are the 5 LSBs of the binary representation for the index of the RL PHY Frame within the current superframe, with LSB  $f_0$ . Finally, the sequence  $S_{CQI}$  is obtained by repeating the  $n$  elements of  $F_{CQI}$   $2 * N_{CTRL\_FFT}$  times:

$$S_{CQI} = \left[ \underbrace{F_{CQI}^0 F_{CQI}^0 \dots F_{CQI}^0 F_{CQI}^0}_{2 * N_{CTRL\_FFT}} F_{CQI}^1 F_{CQI}^1 \dots F_{CQI}^1 F_{CQI}^1 \dots \dots \dots \underbrace{F_{CQI}^{n-1} F_{CQI}^{n-1} \dots F_{CQI}^{n-1} F_{CQI}^{n-1}}_{2 * N_{CTRL\_FFT}} \right]$$

In the absence of a R-CQICH transmission request by the RCC MAC protocol, the sequence  $X_{CQI}$  shall be the all-zero sequence of the same length.

#### 9.4.1.5.1.4 R-BFCH binary sequence

The sequence  $X_{\text{BFCH}}$  of R-BFCH is obtained from the  $i$ -th Walsh sequence  $W_i^{1024}$  of length 1024, by repeating each entry of this sequence  $N_{\text{CTRL\_FFT}}$  times and scaling it to achieve the appropriate power  $P$ . The index  $i$  of the sequence takes on values from 0 to 1023 and is defined by ten-bit value  $(b_9, b_8, b_7, b_6, \dots, b_1, b_0)$ :

$$i = \sum_{l=0}^9 b_l 2^l$$

The value  $(b_9, b_8, b_7, b_6, \dots, b_1, b_0)$  is defined by the RCC MAC Protocol. The resulting sequence is scrambled with a binary sequence  $S_{\text{BFCH}}$ . Let  $W_{k,i}^{1024}$  be the  $k$ -th element of the Walsh sequence  $W_i^{1024}$  with  $k$  within 0 and 1023, then  $X_{\text{BFCH}}$  is given by

$$X_{\text{BFCH}} = \sqrt{\frac{PN_{\text{FFT}}}{N_{\text{CTRL\_SUBCARRIERS}}}} \left( \underbrace{(-1)^{S_{\text{BFCH}}^0} * W_{0,i}^{1024} \dots (-1)^{S_{\text{BFCH}}^{N_{\text{CTRL\_FFT}}-1}} * W_{0,i}^{1024}}_{N_{\text{CTRL\_FFT}}} \dots \underbrace{(-1)^{S_{\text{BFCH}}^{1023N_{\text{CTRL\_FFT}}}} * W_{1023,i}^{1024} \dots (-1)^{S_{\text{BFCH}}^{1024N_{\text{CTRL\_FFT}}-1}} * W_{1023,i}^{1024}}_{N_{\text{CTRL\_FFT}}} \right)$$

where  $P$  is the total power allocated by the RCC MAC Protocol to transmit this sequence. The binary scrambling sequence  $S_{\text{BFCH}}$  is generated as follows. First, a binary sequence  $F_{\text{BFCH}}$  of the length  $n = 512$  shall be generated using a 20-bit shift register that shall have a generator polynomial  $h(D) = D^{20} + D^{17} + D^{12} + D^{10} + 1$ ; i.e., the  $k$ -th output  $F_{\text{BFCH}}^k$  of the register shall satisfy

$F_{\text{BFCH}}^k = F_{\text{BFCH}}^{k-20} \oplus F_{\text{BFCH}}^{k-17} \oplus F_{\text{BFCH}}^{k-12} \oplus F_{\text{BFCH}}^{k-10}$ , for  $k=0, 1, \dots, n-1$ . Here the initial state  $[F_{\text{BFCH}}^{-1}, \dots, F_{\text{BFCH}}^{-20}]$  is given by

$[001f_4 f_3 f_2 f_1 f_0 p_{11} p_{10} p_9 p_8 p_7 p_6 p_5 p_4 p_3 p_2 p_1 p_0]$  if the BFCHReportType is 1 and

$[011f_4 f_3 f_2 f_1 f_0 p_{11} p_{10} p_9 p_8 p_7 p_6 p_5 p_4 p_3 p_2 p_1 p_0]$  if the BFCHReportType is 2. Here, the BFCHReportType is a variable passed by the RCC MAC Protocol with each instance of the R-BFCH.

Here  $[p_{11} p_{10} p_9 p_8 p_7 p_6 p_5 p_4 p_3 p_2 p_1 p_0]$  is the 12-bit PilotPhase for the current superframe, corresponding to the PilotPN specified by the RCC MAC for this channel, with LSB  $p_0$  and MSB  $p_{11}$ . Also,  $[f_4 f_3 f_2 f_1 f_0]$  are the 5 LSBs of the binary representation for the index of the RL PHY Frame within the current superframe, with LSB  $f_0$ . Finally, the sequence  $S_{\text{BFCH}}$  is obtained by repeating the  $n$  elements of  $F_{\text{BFCH}}$   $2 * N_{\text{CTRL\_FFT}}$  times:

$$S_{\text{BFCH}} = \left[ \underbrace{F_{\text{BFCH}}^0 F_{\text{BFCH}}^0 \dots F_{\text{BFCH}}^0 F_{\text{BFCH}}^0}_{2 * N_{\text{CTRL\_FFT}}} F_{\text{BFCH}}^1 F_{\text{BFCH}}^1 \dots F_{\text{BFCH}}^1 F_{\text{BFCH}}^1 \dots \dots \dots \underbrace{F_{\text{BFCH}}^{n-1} F_{\text{BFCH}}^{n-1} \dots F_{\text{BFCH}}^{n-1} F_{\text{BFCH}}^{n-1}}_{2 * N_{\text{CTRL\_FFT}}} \right]$$

In the absence of a R-BFCH transmission request by the RCC MAC protocol, the sequence  $X_{\text{BFCH}}$  shall be the all-zero sequence of the same length.

### 9.4.1.5.1.5 R-SFCH binary sequence

The sequence  $X_{\text{SFCH}}$  of R-SFCH is obtained from the  $i$ -th Walsh sequence  $W_i^{1024}$  of length 1024, by repeating each entry of this sequence  $N_{\text{CTRL\_FFT}}$  times and scaling it to achieve the appropriate power  $P$ . The index  $i$  of the sequence takes on values from 0 to 1023 and is defined by ten-bit value  $(b_9, b_8, b_7, b_6, \dots, b_1, b_0)$ :

$$i = \sum_{l=0}^9 b_l 2^l$$

The value  $(b_9, b_8, b_7, b_6, \dots, b_1, b_0)$  is defined by the RCC MAC Protocol. The resulting sequence is scrambled with a binary sequence  $S_{\text{SFCH}}$ . Let  $W_{k,i}^{1024}$  be the  $k$ -th element of the Walsh sequence  $W_i^{1024}$  with  $k$  within 0 and 1023, then  $X_{\text{SFCH}}$  is given by

$$X_{\text{SFCH}} = \sqrt{\frac{PN_{\text{FFT}}}{N_{\text{CTRL\_SUBCARRIER}}}} \left( \underbrace{(-1)^{S_{\text{SFCH}}^0} W_{0,i}^{1024} \dots (-1)^{S_{\text{SFCH}}^{N_{\text{CTRL\_FFT}}-1}} W_{0,i}^{1024}}_{N_{\text{CTRL\_FFT}}} \dots \underbrace{(-1)^{S_{\text{SFCH}}^{1023 N_{\text{CTRL\_FFT}}}} W_{1023,i}^{1024} \dots (-1)^{S_{\text{SFCH}}^{1024 N_{\text{CTRL\_FFT}}-1}} W_{1023,i}^{1024}}_{N_{\text{CTRL\_FFT}}} \right)$$

where  $P$  is the total power allocated by the RCC MAC Protocol to transmit this sequence. The binary scrambling sequence  $S_{\text{SFCH}}$  are generated as follows. First, a binary sequence  $F_{\text{SFCH}}$  of the length  $n = 512$  shall be generated using a 20-bit shift register that shall have a generator polynomial  $h(D) = D^{20} + D^{17} + D^{12} + D^{10} + 1$ ; i.e., the  $k$ -th output  $F_{\text{SFCH}}^k$  of the register shall satisfy

$F_{\text{SFCH}}^k = F_{\text{SFCH}}^{k-20} \oplus F_{\text{SFCH}}^{k-17} \oplus F_{\text{SFCH}}^{k-12} \oplus F_{\text{SFCH}}^{k-10}$ , for  $k=0, 1, \dots, n-1$ . Here the initial state  $[F_{\text{SFCH}}^{-1}, \dots, F_{\text{SFCH}}^{-20}]$  is given by  $[010f_4 f_3 f_2 f_1 f_0 p_{11} p_{10} p_9 p_8 p_7 p_6 p_5 p_4 p_3 p_2 p_1 p_0]$ . Here  $[p_{11} p_{10} p_9 p_8 p_7 p_6 p_5 p_4 p_3 p_2 p_1 p_0]$  is the 12-bit PilotPhase for the current superframe, corresponding to the PilotPN specified by the RCC MAC for this channel, with LSB  $p_0$  and MSB  $p_{11}$ . Also,  $[f_4 f_3 f_2 f_1 f_0]$  are the 5 LSBs of the binary representation for the index of the RL PHY Frame within the current superframe, with LSB  $f_0$ . Finally, the sequence  $S_{\text{SFCH}}$  is obtained by repeating the  $n$  elements of  $F_{\text{SFCH}}$   $2 * N_{\text{CTRL\_FFT}}$  times:

$$S_{\text{SFCH}} = \left[ \underbrace{F_{\text{SFCH}}^0 F_{\text{SFCH}}^0 \dots F_{\text{SFCH}}^0 F_{\text{SFCH}}^0}_{2 * N_{\text{CTRL\_FFT}}} F_{\text{SFCH}}^1 F_{\text{SFCH}}^1 \dots F_{\text{SFCH}}^1 F_{\text{SFCH}}^1 \dots \dots \dots \underbrace{F_{\text{SFCH}}^{n-1} F_{\text{SFCH}}^{n-1} \dots F_{\text{SFCH}}^{n-1} F_{\text{SFCH}}^{n-1}}_{2 * N_{\text{CTRL\_FFT}}} \right]$$

In the absence of an R-SFCH transmission request by the RCC MAC protocol, the sequence  $X_{\text{SFCH}}$  shall be all-zero sequence of the same length.

### 9.4.1.5.1.6 R-REQCH binary sequence

The sequence  $X_{\text{REQ}}$  of R-REQCH is obtained from the  $i$ -th Walsh sequence  $W_i^{1024}$  of length 1024, by repeating each entry of this sequence  $N_{\text{CTRL\_FFT}}$  times and scaling it to achieve the appropriate power  $P$ . The index  $i$  of the sequence takes on values from 0 to 1023 and is defined by the ten-bit vector  $(b_9, b_8, b_7, b_6, \dots, b_1, b_0)$ :

$$i = \sum_{l=0}^9 b_l 2^l$$

The ten-bit vector  $(b_9, b_8, b_7, b_6, \dots, b_1, b_0)$  is defined by the RCC MAC Protocol. The resulting sequence is scrambled with a binary sequence  $S_{REQ}$ . Let  $W_{k,i}^{1024}$  be the  $k$ -th element of the Walsh sequence  $W_i^{1024}$  with  $k$  within 0 and 1023, then  $X_{REQ}$  is given by

$$X_{REQ} = \sqrt{\frac{PN_{FFT}}{N_{CTRL\_SUBCARRIER}} \left( \underbrace{(-1)^{S_{REQ}^0} * W_{0,i}^{1024} \dots (-1)^{S_{REQ}^{N_{CTRL\_FFT}-1}} * W_{0,i}^{1024}}_{N_{CTRL\_FFT}} \dots \underbrace{(-1)^{S_{REQ}^{1023N_{CTRL\_FFT}}} * W_{1023,i}^{1024} \dots (-1)^{S_{REQ}^{1023N_{CTRL\_FFT}+1}} * W_{1023,i}^{1024}}_{N_{CTRL\_FFT}} \right)}$$

where  $P$  is the total power allocated by the RCC MAC Protocol to transmit this sequence. The binary scrambling sequence  $S_{REQ}$  are generated as follows. First, a binary sequence  $F_{REQ}$  of the length  $n = 512$  shall be generated using a 20-bit shift register shall have a generator polynomial  $h(D) = D^{20} + D^{17} + D^{12} + D^{10} + 1$ , i.e., the  $k$ -th output  $F_{REQ}^k$  of the register shall satisfy

$F_{REQ}^k = F_{REQ}^{k-20} \oplus F_{REQ}^{k-17} \oplus F_{REQ}^{k-12} \oplus F_{REQ}^{k-10}$ , for  $k=0,1,\dots, n-1$ . Here the initial state  $[F_{REQ}^{-1}, \dots, F_{REQ}^{-20}]$  is given by  $[1\ 0\ 1\ f_4\ f_3\ f_2\ f_1\ f_0\ p_{11}\ p_{10}\ p_9\ p_8\ p_7\ p_6\ p_5\ p_4\ p_3\ p_2\ p_1\ p_0]$ .

Here  $[p_{11}\ p_{10}\ p_9\ p_8\ p_7\ p_6\ p_5\ p_4\ p_3\ p_2\ p_1\ p_0]$  is the 12-bit PilotPhase for the current superframe, corresponding to the PilotPN specified by the RCC MAC for this channel, with LSB  $p_0$  and MSB  $p_{11}$ . Also,  $[f_4\ f_3\ f_2\ f_1\ f_0]$  are the 5 LSBs of the binary representation for the index of the RL PHY Frame within the current superframe, with LSB  $f_0$ . Finally, the sequence  $S_{REQ}$  is obtained by repeating the  $n$  elements of  $F_{REQ}$   $2 * N_{CTRL\_FFT}$  times:

$$S_{REQ} = \left[ \underbrace{F_{REQ}^0\ F_{REQ}^0\ \dots\ F_{REQ}^0\ F_{REQ}^0}_{2 * N_{CTRL\_FFT}}\ F_{REQ}^1\ F_{REQ}^1\ \dots\ F_{REQ}^1\ F_{REQ}^1\ \dots\ \dots\ \underbrace{F_{REQ}^{n-1}\ F_{REQ}^{n-1}\ \dots\ F_{REQ}^{n-1}\ F_{REQ}^{n-1}}_{2 * N_{CTRL\_FFT}} \right]$$

In the absence of an R-REQCH transmission request by the RCC MAC protocol, the sequence  $X_{REQ}$  shall be all-zero sequence of the same length.

#### 9.4.1.5.1.7 R-PICH binary sequence

The sequence  $X_{PICH}$  of R-PICH is obtained from a binary sequence of length  $1024 * N_{CTRL\_FFT}$ , which takes values on  $\{-1, 1\}$ . The binary sequence is further scaled to achieve the appropriate power  $P$ , according to the following equation:

$$X_{PICH} = \sqrt{\frac{PN_{FFT}}{N_{CTRL\_SUBCARRIER}}} \left( (-1)^{S_{PICH}^0}, (-1)^{S_{PICH}^1} \dots (-1)^{S_{PICH}^{1024N_{CTRL\_FFT}-2}}, (-1)^{S_{PICH}^{1024N_{CTRL\_FFT}-1}} \right),$$

where  $P$  is the total power allocated by the RCC MAC Protocol to transmit this sequence and wherein the binary sequence  $S_{PICH}$  is generated as follows. First, a binary sequence  $F_{PICH}$  of the length  $n = 512$  which shall be generated using a 20-bit shift register which shall have a generator polynomial  $h(D) = D^{20} + D^{17} + D^{12} + D^{10} + 1$ ; i.e., the  $k$ -th output  $F_{PICH}^k$  of the register shall satisfy  $F_{PICH}^k = F_{PICH}^{k-20} \oplus F_{PICH}^{k-17} \oplus F_{PICH}^{k-12} \oplus F_{PICH}^{k-10}$ , for  $k=0,1,\dots, n-1$ . Here the initial state  $[F_{PICH}^{-1}, \dots, F_{PICH}^{-20}]$  is given by  $[1\ 1\ 0\ f_4\ f_3\ f_2\ f_1\ f_0\ p_{11}\ p_{10}\ p_9\ p_8\ p_7\ p_6\ p_5\ p_4\ p_3\ p_2\ p_1\ p_0]$ . Here  $[p_{11}\ p_{10}\ p_9\ p_8\ p_7\ p_6\ p_5\ p_4\ p_3\ p_2\ p_1\ p_0]$  is the 12-bit PilotPhase for the current superframe, corresponding to the PilotPN specified by the RCC

1 MAC for this channel, with LSB  $p_0$  and MSB  $p_{11}$ . Also,  $[f_4 f_3 f_2 f_1 f_0]$  are the 5 LSBs of the binary  
 2 representation for the index of the RL PHY Frame within the current superframe, with LSB  $f_0$ .  
 3 Finally, the sequence  $S_{PICH}$  is obtained by repeating the  $n$  elements of  $F_{PICH}$   $2*N_{CTRL\_FFT}$  times:

$$4 \quad S_{PICH} = \left[ \underbrace{F_{PICH}^0 \ F_{PICH}^0 \ \dots \ F_{PICH}^0 \ F_{PICH}^0}_{2*N_{CTRL\_FFT}} \ F_{PICH}^1 \ F_{PICH}^1 \ \dots \ F_{PICH}^1 \ F_{PICH}^1 \ \dots \dots \dots \underbrace{F_{PICH}^{n-1} \ F_{PICH}^{n-1} \ \dots \ F_{PICH}^{n-1} \ F_{PICH}^{n-1}}_{2*N_{CTRL\_FFT}} \right]$$

#### 5 **9.4.1.5.2 Multiplexing of R-CQICH, R-BFCH, R-SFCH, R-REQCH and R-PICH**

6 If multiple instances of the channels R-CQICH, R-BFCH and R-SFCH are defined by the RCC MAC  
 7 Protocol, then the resulting modulation sequences are superimposed. For example,  $X_{CQI}$  from now on  
 8 refers to the sequence obtained by superimposing the modulated sequence for every instance of the  
 9 R-CQICH channel. The same holds for  $X_{SFCH}$  and  $X_{BFCH}$ .

10 R-CQICH, R-BFCH, R-SFCH, R-REQCH, and R-PICH are I-Q multiplexed within the Control  
 11 Segment. The combined complex-valued time domain sequence  $X_{CTRL}$  of these channels is given by  
 12 the following equation:

$$13 \quad X_{CTRL} = (X_{CQICH} + X_{SFCH} + X_{PICH}) + j * (X_{REQCH} + X_{BFCH})$$

#### 14 **9.4.1.5.3 HPSK scrambling**

15 All the channels within the Control Segment with an assigned MACID undergo sector-specific and  
 16 MACID scrambling (this refers to R-CQICH, R-BFCH, R-SFCH, R-REQCH, and R-PICH). R-ACH  
 17 may undergo sector-specific scrambling or sector-specific and MACID scrambling depending on the  
 18 access sequence ID defined by the AC MAC Protocol.

19 The sector-specific scrambling sequence  $Y_{SS,ACH}$  of the length  $1024 * N_{CTRL\_FFT}$  is generated according  
 20 to:

$$21 \quad Y_{SS,ACH} = \left( Y_{SS,ACH}^0 \ Y_{SS,ACH}^1 \ \dots \ Y_{SS,ACH}^{1024*N_{CTRL\_FFT}-1} \right), \quad Y_{SS,ACH}^0 = 1, \quad Y_{SS,ACH}^k = Y_{SS,ACH}^{k-1} e^{j\pi/2 * (2*S_{SS,ACH}^{k-1})}$$

22 where the binary scrambling sequence  $S_{SS,ACH}$  of the length  $(1024 * N_{CTRL\_FFT} - 1)$  shall be generated  
 23 using a 20-bit shift register which shall have a generator polynomial  $h(D) = D^{20} + D^{17} + D^{12} + D^{10} + 1$ ,

24 i.e., the  $k$ -th output  $S_{SS,ACH}^k$  of the register shall satisfy  $S_{SS,ACH}^k = S_{SS,ACH}^{k-20} \oplus S_{SS,ACH}^{k-17} \oplus S_{SS,ACH}^{k-12} \oplus S_{SS,ACH}^{k-10}$ ,

25 where the initial state  $[S_{SS,ACH}^{-1}, \dots, S_{SS,ACH}^{-20}]$  is given by

26  $[0 \ 0 \ 1 \ 0 \ f_4 \ f_3 \ f_2 \ f_1 \ f_0 \ p_{11} \ p_{10} \ p_9 \ p_8 \ p_7 \ p_6 \ p_5 \ p_4 \ p_3 \ p_2 \ p_1 \ p_0]$ . Here  $[p_{11} \ p_{10} \ p_9 \ p_8 \ p_7 \ p_6 \ p_5 \ p_4 \ p_3 \ p_2 \ p_1 \ p_0]$  is the 12-bit

27 PilotPhase of the target sector for the access probe for the current superframe, with LSB  $p_0$  and MSB

28  $p_{11}$ . Also,  $[f_4 \ f_3 \ f_2 \ f_1 \ f_0]$  are the 5 LSBs of the binary representation for the index of the PHY Frame

29 within the current superframe, with LSB  $f_0$ .

30 The sector-specific and MACID scrambling sequence  $Y_{SM,ACH}$  of the length  $1024 * N_{CTRL\_FFT}$  is given  
 31 by:

$$32 \quad Y_{SM,ACH} = \left( Y_{SM,ACH}^0 \ Y_{SM,ACH}^1 \ \dots \ Y_{SM,ACH}^{1024*N_{CTRL\_FFT}-1} \right), \quad Y_{SM,ACH}^0 = 1, \quad Y_{SM,ACH}^k = Y_{SM,ACH}^{k-1} e^{j\pi/2 * (2*S_{SM,ACH}^{k-1})}$$

1 where the binary scrambling sequence  $S_{SM,ACH}$  of the length  $(1024 * N_{CTRL\_FFT} - 1)$  shall be generated  
 2 using a 28-bit shift register which shall have a generator polynomial  $h(D) = D^{28} + D^{25} + 1$ , i.e., the k-th  
 3 output  $S_{SM,ACH}^k$  of the register shall satisfy  $S_{SM,ACH}^k = S_{SM,ACH}^{k-28} \oplus S_{SM,ACH}^{k-25}$ . Here the initial state  
 4  $[S_{SM,ACH}^{-1}, \dots, S_{SM,ACH}^{-28}]$  is given by  
 5  $[f_4 f_3 f_2 f_1 f_0 m_{10} m_9 m_8 m_7 m_6 m_5 m_4 m_3 m_2 m_1 m_0 p_{11} p_{10} p_9 p_8 p_7 p_6 p_5 p_4 p_3 p_2 p_1 p_0]$ . Here  $[f_4 f_3 f_2 f_1 f_0]$   
 6 are the 5 LSBs of the binary representation for the index of the RL PHY Frame within the current  
 7 superframe, with LSB  $f_0$ .  $[p_{11} p_{10} p_9 p_8 p_7 p_6 p_5 p_4 p_3 p_2 p_1 p_0]$  is the 12-bit PilotPhase of the target sector  
 8 of the access probe for the current superframe, with LSB  $p_0$  and MSB  $p_{11}$ , and  
 9  $[m_{10} m_9 m_8 m_7 m_6 m_5 m_4 m_3 m_2 m_1 m_0]$  is the 11-bit MACID corresponding to the target sector of the  
 10 access probe.

11 The sector-specific and MACID scrambling sequence  $Y_{SM}$  of the length  $1024 * N_{CTRL\_FFT}$  is given by:

$$12 \quad Y_{SM} = \left( Y_{SM}^0 Y_{SM}^1 \dots Y_{SM}^{1024 * N_{CTRL\_FFT} - 1} \right), \quad Y_{SM}^0 = 1, \quad Y_{SM}^k = Y_{SM}^{k-1} e^{j\pi/2 * (2 * S_{SM}^{k-1} - 1)}$$

13 where the binary scrambling sequence  $S_{SM}$  of the length  $(1024 * N_{CTRL\_FFT} - 1)$  shall be generated using  
 14 a 28-bit shift register which shall have a generator polynomial  $h(D) = D^{28} + D^{25} + 1$ , i.e., the k-th  
 15 output  $S_{SM}^k$  of the register shall satisfy  $S_{SM}^k = S_{SM}^{k-28} \oplus S_{SM}^{k-25}$ . Here the initial state  $[S_{SM}^{-1}, \dots, S_{SM}^{-28}]$  is given  
 16 by  $[f_4 f_3 f_2 f_1 f_0 m_{10} m_9 m_8 m_7 m_6 m_5 m_4 m_3 m_2 m_1 m_0 p_{11} p_{10} p_9 p_8 p_7 p_6 p_5 p_4 p_3 p_2 p_1 p_0]$ . Here  
 17  $[p_{11} p_{10} p_9 p_8 p_7 p_6 p_5 p_4 p_3 p_2 p_1 p_0]$  is the 12-bit PilotPhase for the current superframe, corresponding to  
 18 the PilotPN specified by the RCC MAC for all channels except the R-ACH within this control  
 19 segment<sup>85</sup>, with LSB  $p_0$  and MSB  $p_{11}$ . Also,  $[f_4 f_3 f_2 f_1 f_0]$  are the 5 LSBs of the binary representation  
 20 for the index of the RL PHY Frame within the current superframe with LSB  $f_0$  and  
 21  $[m_{10} m_9 m_8 m_7 m_6 m_5 m_4 m_3 m_2 m_1 m_0]$  is the 11-bit MACID specified by the RCC MAC.

#### 22 9.4.1.5.4 Control Segment complex-valued signal in the time domain

23 The time domain sequence  $Z$  of length  $1024 * N_{CTRL\_FFT}$  transmitted over the Control Channel  
 24 segment is given by the following equation:

$$25 \quad Z = \left( Y_{SM}^0 * X_{CTRL}^0, Y_{SM}^1 * X_{CTRL}^1, \dots, Y_{SM}^{1024 * N_{CTRL\_FFT} - 1} * X_{CTRL}^{1024 * N_{CTRL\_FFT} - 1} \right) \\ + \left( Y_{SS,ACH}^0 * X_{ACH}^0, Y_{SS,ACH}^1 * X_{ACH}^1, \dots, Y_{SS,ACH}^{1024 * N_{CTRL\_FFT} - 1} * X_{ACH}^{1024 * N_{CTRL\_FFT} - 1} \right)$$

26 if the access sequence ID  $> N_{ACMP\text{SpecialSequences}}$  and the equation

$$27 \quad Z = \left( Y_{SM}^0 * X_{CTRL}^0, Y_{SM}^1 * X_{CTRL}^1, \dots, Y_{SM}^{1024 * N_{CTRL\_FFT} - 1} * X_{CTRL}^{1024 * N_{CTRL\_FFT} - 1} \right) \\ + \left( Y_{SM,ACH}^0 * X_{ACH}^0, Y_{SM,ACH}^1 * X_{ACH}^1, \dots, Y_{SM,ACH}^{1024 * N_{CTRL\_FFT} - 1} * X_{ACH}^{1024 * N_{CTRL\_FFT} - 1} \right)$$

<sup>85</sup> Note that the RCC MAC protocol specifies the same PilotPN and MACID for all channels in the control segment except possibly the R-ACH.

1 if the access sequence  $ID \leq N_{ACMPSSpecialSequences}$  where the access sequence ID is as defined by the AC  
 2 MAC Protocol and used to modulate R-ACH, and  $N_{ACMPSSpecialSequences}$  is a constant of the AC MAC  
 3 protocol.

4 In the above equations,  $X_{CTRL}^l$  and  $X_{ACH}^l$  refer to the  $l$ -th entry of the vectors  $X_{CTRL}$  and  $X_{ACH}$   
 5 respectively, with  $l$  taking values from 0 to  $1024 * N_{CTRL\_FFT} - 1$ .

#### 6 9.4.1.5.5 DFT operation

7 The following description is applicable per carrier in MultiCarrierOn mode. The scrambled sequence  
 8  $Z$  of length  $1024 * N_{CTRL\_FFT}$  generated in the previous section shall be broken up into  $N_{FRAME,R}$   
 9 different subsequences of length  $R = (N_{CTRL\_FFT}/A) * N_{SUBBAND}$  where  $A = (N_{SUBBAND}/128)$ . The first  $R$   
 10 elements of the sequence  $Z$  form the first subsequence  $Z_0$ , the next  $R$  elements form the second  
 11 sequence  $Z_1$ , etc. Each of these subsequences shall be converted to a frequency domain sequence  
 12 through a Discrete Fourier Transform (DFT) operation, modulated and transmitted on the  
 13 corresponding OFDM symbol. The DFT of an  $N$ -length sequence  $X$  with elements  $x_0, x_1, \dots, x_{N-1}$  is  
 14 given by another  $N$ -length sequence  $Y$  with elements  $y_0, y_1, \dots, y_{N-1}$ . The elements of  $Y$  are related to  
 15 the elements of  $X$  via the relationship

$$16 \quad y_i = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} x_k e^{-j2\pi(k-\frac{N}{2})i/N}.$$

17 Let  $F_l$  denote the DFT of the  $R$ -length subsequence  $Z_l$ ,  $0 \leq l < N_{FRAME,R} - 1$ . The first  $N_{CTRL-}$   
 18  $SUBCARRIERS-AT$  elements of the sequence  $F_l$  are allocated to hop ports in the Control Segment. More  
 19 precisely, the  $t$  elements of the sequence  $F_l$ ,  $t$  ranging from 0 to  $N_{CTRL-SUBCARRIERS-AT}-1$ , shall be  
 20 modulated onto subcarriers with indices ranging from  $f_{MIN-CTRL-AT}$  to  $f_{MIN-CTRL-AT} + N_{CTRL-SUBCARRIERS-}$   
 21  $AT - 1$ , of the  $l$ 'th OFDM symbol in the RL PHY Frame. Here,  $f_{MIN-CTRL-AT}$  and  $N_{CTRL-SUBCARRIERS-AT}$  are  
 22 as defined in 9.4.1.5.1. The remaining elements of the sequence  $F_l$  are discarded.

#### 23 9.4.1.6 Data segment modulation

24 The Data Segment carries the R-DPICH and the R-DCH. A subcarrier occupied by the R-DPICH  
 25 shall be referred to as a pilot subcarrier. For the purpose of this section, the sector of interest is the  
 26 Reverse Link Serving Sector (RLSS), which may or may not be different from the Forward Link  
 27 Serving Sector (FLSS). For convenience of notation, the phrase “of the Forward Link Serving  
 28 Sector” shall be omitted. Sector-dependent quantities such as PilotPN, hop-permutations etc., used in  
 29 this section shall be interpreted as “PilotPN of the RLSS,” “hop-permutations of the RLSS” etc.

##### 31 9.4.1.6.1 R-DPICH

32 The Dedicated Pilot Channel (R-DPICH) shall be present in each tile that is assigned to the R-DCH  
 33 (for this AT). The modulation of this channel is described for a single tile, where a tile is as described  
 34 in 9.4.1.3. The modulation procedure shall then be repeated for each such tile that is assigned to the  
 35 R-DCH (for this AT) in every RL PHY Frame. The configuration of the R-DPICH in a given tile is  
 36 determined by the configuration of the R-DCH in this tile. If there is no R-DCH present in a tile, R-  
 37 DPICH shall also not be transmitted.

1 The R-DPICH configuration in each tile consists of the following parameters:

- 2 1. The energy per modulation symbol: All the R-DPICH modulation symbols in a given tile  
3 shall have the same energy, which shall be the same as the energy used to transmit R-  
4 DCH modulation symbols in this tile. This energy is assigned by the RTC MAC protocol.
- 5 2. R-DPICH format: The R-DPICH in a tile can have two different formats, labeled Format  
6 0 and Format 1. The R-DPICH format to be used is determined by the RTC MAC  
7 protocol.
- 8 3. RLDPISectorOffset: This is part of the public data of the Overhead Messages Protocol,  
9 and takes on integer values between 0 and 3. The value used shall correspond to the  
10 carrier containing the tile of interest.
- 11 4. RLDPIUserOffset: This is an integer that depends on the hop-ports contained in the tile  
12 of interest. Let  $p_{\min}$  be the smallest hop-port index (within the carrier) contained in the tile  
13 of interest. RLDPIUserOffset is then given by  $\lfloor p_{\min} / N_{CARRIER\_SIZE} \rfloor$ .

14 In order to aid the description of the R-DPICH formats, the hop-ports in each tile are numbered from  
15 0 to  $N_{BLOCK}-1$  in increasing order. Also, for all RL PHY Frames, except for the RL PHY Frame with  
16 index 0 in the superframe in FDD mode, the OFDM symbols in each tile are numbered from 0 to  
17  $N_{FRAME,R}-1$  in increasing order. For the RL PHY Frame with index 0 in FDD mode, the OFDM  
18 symbols in each tile are numbered from 0 to  $N_{FRAME,R} + N_{PREAMBLE} - 1$  in increasing order.

#### 19 9.4.1.6.1.1 R-DPICH Format 0

20 In this format, the R-DPICH occupies 18 modulation symbols in a tile in all RL PHY Frames, except  
21 those with index 0 within the superframe in FDD mode. In an RL PHY Frame with index 0 within the  
22 superframe in FDD mode, the R-DPICH occupies 36 modulation symbols within each tile. A hop-  
23 port with index  $i_{hp}$  of the OFDM symbol with index  $t$  (both measured within the tile) is occupied by  
24 the R-DPICH if  $i_{hp}$  is in the set  $\{1, 8, 15\}$ , if  $t' = t \bmod N_{FRAME,R}$  is in the set  $\{0, 1, 2, 5, 6, 7\}$ , and if this  
25 hop-port is not mapped to a subcarrier assigned to the R-ACKCH of the RLSS. The set of hop-ports  
26 occupied by the R-DPICH for this format is illustrated in Figure 113, for the case when none of the  
27 hop-ports in the tile are assigned to the R-ACKCH.

28 The complex value of the R-DPICH modulation symbol at this location is given by

$$29 \quad S_{i_{hp},t} = \sqrt{P} \exp\left(\frac{j2\pi}{3}(RLDPISectorOffset + RLDPIUserOffset)t'\right) \text{ if } t' < 4, \text{ and}$$

$$30 \quad S_{i_{hp},t} = \sqrt{P} \exp\left(\frac{j2\pi}{3}(RLDPISectorOffset + RLDPIUserOffset)(t'-2)\right) \text{ if } t' \geq 4.$$

31 where  $j$  denotes the complex number  $(0,1)$  and  $P$  is the energy per modulation symbol used by the  
32 R-DPICH. Note that the value of this modulation symbol is the same for all values of  $i_{hp}$ .



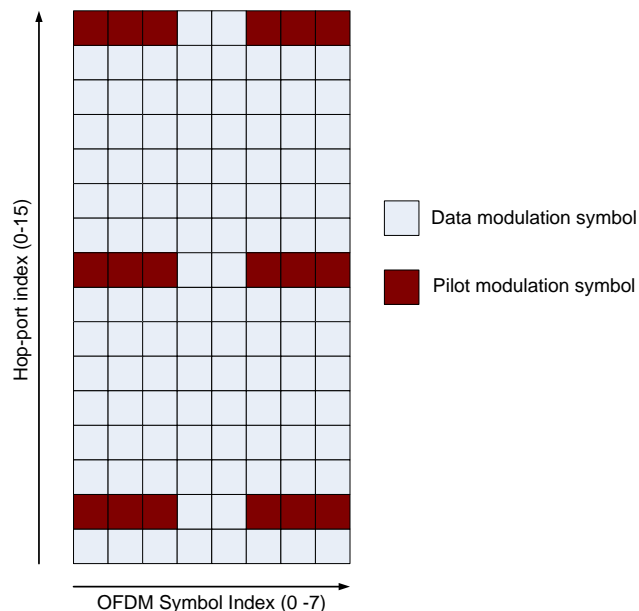


Figure 113 R-DPICH Format 0

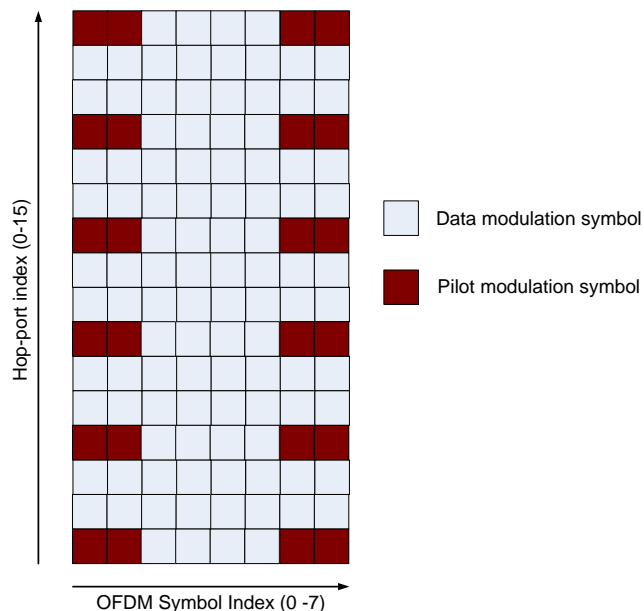
#### 9.4.1.6.1.1.2 R-DPICH Format 1

In this format, the R-DPICH occupies 24 modulation symbols in a tile in all RL PHY Frames, except those with index 0 within the superframe in FDD mode. In an RL PHY Frame with index 0 within the superframe in FDD mode, the R-DPICH occupies 48 modulation symbols in each tile. In this format, a hop-port with index  $i_{hp}$  of the OFDM symbol with index  $t$  (both measured within the tile) is occupied by the R-DPICH if  $i_{hp}$  is in the set  $\{0,3,6,9,12,15\}$ , if  $t' = t \bmod N_{FRAME,R}$  is in the set  $\{0,1,6,7\}$ , and if this hop-port is not mapped to a subcarrier assigned to the R-ACKCH of the RLSS. The set of hop-ports occupied by the R-DPICH for this format is illustrated in Figure 114, for the case when none of the hop-ports in the tile are assigned to the R-ACKCH.

The complex value of the R-DPICH modulation symbol at this location is given by

$$S_{i_{hp},t} = \sqrt{P} \exp(j\pi(RLDPISectorOffset + RLDPIUserOffset) t'),$$

where  $j$  denotes the complex number  $(0,1)$  and  $P$  is the energy per modulation symbol used by the R-DPICH. Note that the value of this modulation symbol is the same for all values of  $i_{hp}$ .



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

**Figure 114 R-DPICH Format 1**

#### 9.4.1.6.2 R-DCH

The R-DCH consists of either a data packet or an erasure sequence, both of which can span one or more RL PHY Frames. The set of RL PHY Frames on which this packet or erasure sequence is transmitted is determined by the RTC MAC Protocol. Each data packet and erasure sequence is also assigned a set of hop-ports in each PHY Frame of transmission by the RTC MAC Protocol. Note that this set of hop-ports can span multiple carriers. Each data packet is further associated with a packet format index, which is also assigned by the RTC MAC Protocol.

##### 9.4.1.6.2.1 Data packet transmission

Each R-DCH packet is generated by the RTC MAC Protocol, and is split, appended with CRC, encoded, channel interleaved, repeated, data-scrambled and modulated according to the procedure described in 9.2. A CRC length of  $N_{CRC,Data}$  is used for this packet. The MACID of the access terminal corresponding to its Reverse Link Serving Sector (RLSS), and the packet format index assigned to this packet, shall be used to generate the initial state of the data-scrambler described in 9.2.5. The size of the input packet generated by the RTC MAC Protocol shall be equal to  $8 \lfloor \rho n_0 N_f / 8 \rfloor - N_{CRC,Data}$ , where  $\rho$  denotes the spectral efficiency at the first transmission corresponding to the packet format of the packet (defined by the RTC MAC Protocol),  $n_0$  denotes the number of usable hop-ports assigned to this packet in the first RL PHY Frame of transmission, and  $N_f$  is equal to  $6N_{FRAME,R}$  if this packet is part of an extended duration transmission and is equal to  $N_{FRAME,R}$  otherwise. The RTC MAC protocol determines whether or not a packet is part of an extended duration transmission. Here, a usable hop-port is as defined in 9.4.1.3.2. This packet shall be modulated on to the hop-ports assigned to this packet according to the following procedure:

1. Initialize a port counter  $i$  to 0, a frame counter  $f$  to 0, and an OFDM symbol counter  $j$  to 0.

- 1           2. Arrange the set of usable hop-ports assigned to this packet in the  $f$ 'th PHY Frame of  
 2 transmission in increasing order, where the ordering of hop-ports is as defined in  
 3 9.4.1.2.6. Let the resulting sequence be denoted by  $(k_0, p_0), (k_1, p_1), \dots, (k_{n-1}, p_{n-1})$ , where  $n$   
 4 is the total number of usable hop-ports assigned to this packet in the  $f$ 'th PHY Frame of  
 5 transmission. The notation for a hop-port is as in 9.3.2.2.5, i.e.,  $k_i$  denotes the  
 6 CarrierIndex and  $(k_i, p_i)$  denotes the hop-port index in that carrier.
- 7           3. Let  $n_{sc}$  be the subcarrier index corresponding to the hop-port  $(k_i, p_i)$  in the  $j$ 'th OFDM  
 8 symbol in the  $f$ 'th RL PHY Frame of transmission. Let  $q$  be the modulation order to be  
 9 used for the  $f$ 'th PHY Frame of transmission, which is a function of the packet format. If  
 10  $n_{sc}$  is not assigned to the R-ACKCH of the RLSS and if  $(k_i, p_i)$  is not a DPICH hop-port,  
 11 then a modulation symbol  $s$  with modulation order  $q$  is generated by the modulator  
 12 according to the procedure described in 9.2.6. This modulation symbol shall be  
 13 modulated with energy  $\frac{PN_{FFT}}{n}$  on hop-port  $(k_i, p_i)$ , i.e., the value of the corresponding  
 14 subcarrier shall be  $\sqrt{PN_{FFT} / n} s$ , where  $P$  is the power specified for this assignment in  
 15 the  $f$ 'th PHY Frame of transmission (generated by the RTC MAC Protocol).
- 16           4. Increment  $i$ . If  $i = n$ , increment  $j$  and set  $i = 0$
- 17           5. Increment  $f$  and set  $j = 0$  if any of the following two conditions is satisfied:
- 18           □ If this is an RL PHY Frame with index 0 within the superframe and the duplexing  
 19 mode is FDD, and if  $j = N_{FRAME,R} + N_{PREAMBLE}$ .
- 20           □ For any other RL PHY Frame (including all RL PHY Frames in TDD mode), if  $j =$   
 21  $N_{FRAME,R}$ .
- 22           6. If the last RL PHY Frame of transmission has been completed (as determined by the RTC  
 23 MAC Protocol), then stop. Else repeat steps 2 through 6.

#### 24 9.4.1.6.2.2 Erasure sequence

25 An erasure sequence spans one or more consecutive RL PHY Frames of transmission on a set of hop-  
 26 ports determined by the RTC MAC Protocol. The erasure sequence shall be modulated on to the hop-  
 27 ports assigned to this sequence according to the following procedure:

- 28           1. Construct a one-bit packet, with the bit in the packet being set to zero. This packet is  
 29 encoded, channel interleaved, repeated, scrambled, and modulated according to the  
 30 procedure described in 9.2<sup>86</sup>. The MAC ID of the access terminal corresponding to its RL  
 31 Serving Sector, and a packet format index of 0 shall be used to generate the initial seed of  
 32 the scrambler. QPSK modulation shall be used for all of the modulation symbols in the  
 33 packet.
- 34           2. Initialize a port counter  $i$  to 0, an OFDM symbol counter  $j$  to 0, and a PHY Frame counter  
 35  $f$  to 0.

---

<sup>86</sup> The operations before scrambling and modulation are all trivial operations, i.e., they result in an all-zeros sequence. The erasure sequence is equivalent to scrambling an all-zeros sequence of the required length, followed by QPSK modulation.

- 1           3. Arrange the set of usable hop-ports assigned to this packet in the  $f$ 'th PHY Frame of  
2 transmission in increasing order, where the ordering of hop-ports is as defined in  
3 9.3.2.2.5. Let the resulting sequence be denoted by  $(k_0, p_0), (k_1, p_1), \dots, (k_{n-1}, p_{n-1})$ , where  $n$   
4 is the total number of usable hop-ports assigned to this packet in the  $f$ 'th PHY Frame of  
5 transmission. The notation for a hop-port is as in 9.3.2.2.5, i.e.,  $k_i$  denotes the  
6 CarrierIndex and  $p_i$  denotes the hop-port index in that carrier.
- 7           4. Let  $n_{sc}$  be the subcarrier index corresponding to the hop-port  $p_i$  in the  $j$ 'th OFDM symbol  
8 in the  $f$ 'th RL PHY Frame of transmission. If  $n_{sc}$  is not assigned to the R-ACKCH of the  
9 RLSS and if  $(k_i, p_i)$  is not a DPICH hop-port, then a QPSK modulation symbol  $s$  is  
10 generated by the modulator according to the procedure described in 9.2.6. This  
11 modulation symbol shall be modulated with energy  $PN_{FFT}/n$  on hop-port  $(k_i, p_i)$ . That is,  
12 the value of the corresponding subcarrier shall be  $\sqrt{PN_{FFT}/n} s$ , where  $P$  is the assigned  
13 power to this erasure sequence (generated by the RTC MAC Protocol).
- 14           5. Increment  $i$ . If  $i = n$ , or if  $i = N_{MaxErasureHopports,R}$ , increment  $j$  and set  $i = 0$ .
- 15           6. Increment  $f$  and set  $j = 0$  if any of the following two conditions is satisfied:
- 16           □ If this is an RL PHY Frame with index 0 within the superframe and the duplexing  
17 mode is FDD, and if  $j = N_{FRAME,R} + N_{PREAMBLE}$ .
- 18           □ For any other RL PHY Frame (including all RL PHY Frames in TDD mode), if  $j =$   
19  $N_{FRAME,R}$ .
- 20           7. If the last RL PHY Frame of transmission has been completed (as determined by the RTC  
21 MAC Protocol), then stop. Else repeat steps 3 through 7.

#### 22 **9.4.1.6.3 Sector-specific scrambling**

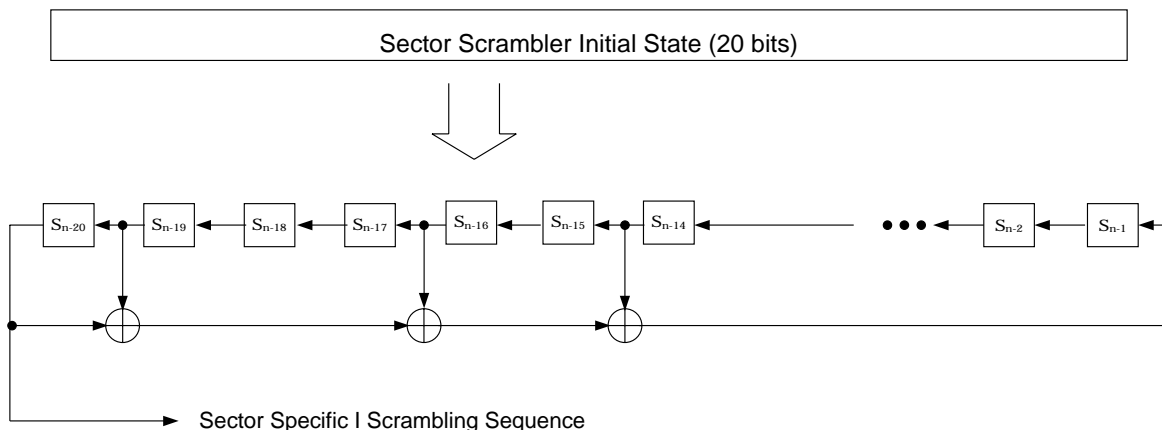
23 Each OFDM symbol in the superframe shall be scrambled by a sector-specific scrambling sequence.  
24 The scrambling operation shall be performed independently on each carrier. The rest of this section  
25 describes the scrambling operation for the carrier  $k$ , where  $k=0,1,\dots, N_{CARRIERS} -1$ . The scrambling  
26 sequence for the carrier  $k$  consists of a complex number for every subcarrier in the carrier  $k$  in every  
27 OFDM symbol in the superframe. The scrambling operation shall consist of multiplying the  
28 unscrambled complex symbol on each subcarrier by the corresponding entry in the scrambling  
29 sequence, unless the subcarrier is allocated to the control segment, or both conditions (a) and (b) are  
30 true: (a) The subcarrier corresponds to a R-DPICH hop-port (via the hop-permutation), and (b)  
31 RLDPISectorScramble, which is part of the public data of the Overhead Messages Protocol for carrier  
32  $k$ , is set to 0. For subcarriers for which these conditions (a) and (b) are true, the scrambling operation  
33 shall consist of leaving the subcarrier unchanged; and a cell-specific scrambling sequence, as  
34 described in 9.4.1.6.4, shall be used to scramble the subcarrier. If the subcarrier is allocated to the  
35 control segment, the scrambling operation shall consist of leaving the subcarrier unchanged.

1 Each complex number in the sector-specific scrambling sequence is generated from two bits, denoted  
2 by  $s_I$  and  $s_Q$ , using the following mapping:

- 3 1. The bit combination  $(s_I, s_Q) = (0,0)$  is mapped to the complex number  $\left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$ .
- 4 2. The bit combination  $(s_I, s_Q) = (0,1)$  is mapped to the complex number  $\left(\frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}\right)$ .
- 5 3. The bit combination  $(s_I, s_Q) = (1,0)$  is mapped to the complex number  $\left(\frac{-1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$ .
- 6 4. The bit combination  $(s_I, s_Q) = (1,1)$  is mapped to the complex number  $\left(\frac{-1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}\right)$ .

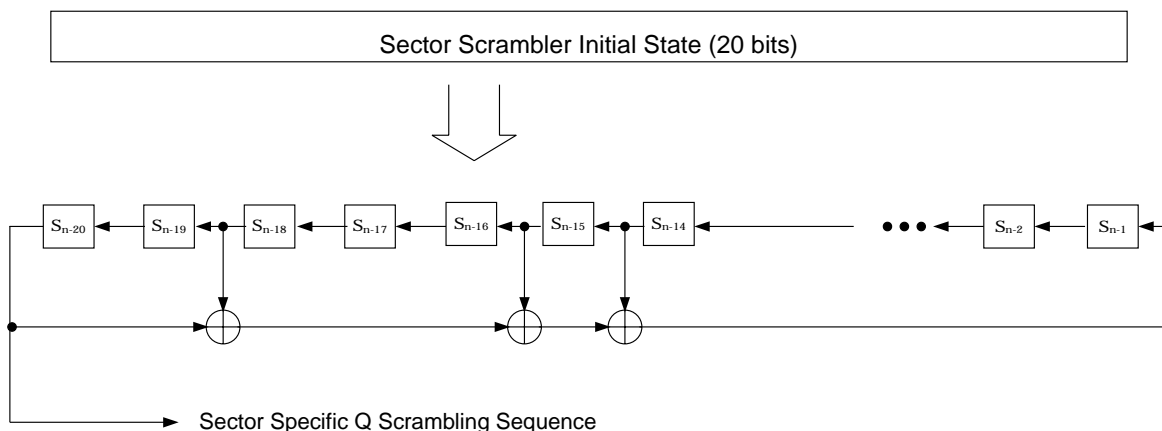
7 The sector-specific scrambling sequence for the carrier  $k$  shall be generated using two 20-bit registers,  
8 called the I-register and the Q-register, as shown in Figure 115 and Figure 116, respectively. The  
9 I-register shall have a generator polynomial  $h_I(D) = D^{20} + D^{19} + D^{16} + D^{14} + 1$  i.e., the  $n$ 'th output  $I(n)$   
10 of the register shall satisfy  $I(n) = I(n-20) \oplus I(n-19) \oplus I(n-16) \oplus I(n-14)$ . The Q-register shall have a  
11 generator polynomial  $h_Q(D) = D^{20} + D^{18} + D^{15} + D^{14} + 1$  i.e., the  $n$ 'th output  $Q(n)$  of the register shall  
12 satisfy  $Q(n) = Q(n-20) \oplus Q(n-18) \oplus Q(n-15) \oplus Q(n-14)$ . Each entry in the sector-specific scrambling  
13 sequence shall be generated using  $s_I$  and  $s_Q$  bits which are respectively the outputs of the I-register  
14 and the Q-register after they have been appropriately initialized and clocked as in the following  
15 description.

16 At the start of every superframe, define `PilotPNSectorScramb` to be equal to `PilotPhase` in  
17 `SemiSynchronous` mode and equal to `PilotPN` in `Asynchronous` mode. (Thus, for a given sector,  
18 `PilotPNSectorScramb` is fixed in `Asynchronous` mode, but changes every superframe in  
19 `SemiSynchronous` mode.) Let  $p_{11}, p_{10}, \dots, p_0$  be the 12 bits of  $(\text{PilotPNSectorScramb} + k) \bmod 4096$  for  
20 a given superframe, with  $p_{11}$  being the MSB and  $p_0$  being the LSB. At the beginning of each  
21 superframe, the I and Q registers shall both be initialized to the state  
22  $[11111111p_{11}p_{10}p_9p_8p_7p_6p_5p_4p_3p_2p_1p_0]$ . The I and Q registers shall then be clocked  $N_{\text{CARRIER\_SIZE}}$  times  
23 for each OFDM symbol in the superframe to generate the  $s_I$  and  $s_Q$  bits for all of the subcarriers  
24 belonging to the carrier  $k$  in every OFDM symbol. The  $i$ 'th entry in the scrambling sequence  
25 (generated after  $i$  clock periods) is used to scramble the subcarrier with index  $i \bmod N_{\text{CARRIER\_SIZE}}$  in  
26 the carrier  $k$ , in the OFDM symbol with index  $\lfloor i / N_{\text{CARRIER\_SIZE}} \rfloor$  in the superframe. The outputs of  
27 the I and Q registers immediately after their state has been initialized (before they are clocked) shall  
28 be used to generate the scrambling sequence entry corresponding to the subcarrier with index 0 in the  
29 carrier  $k$  in the OFDM symbol with index 0.



1  
2

**Figure 115 Sector-specific scrambler for the data segments – I sequence**



3  
4

**Figure 116 Sector-specific scrambler for the data segments – Q sequence**

**9.4.1.6.4 Cell-specific scrambling for R-DPICH**

6 The operations in this section shall be carried out independently for each carrier, and shall be  
 7 described for the carrier with index  $k$ , where  $k=0,1,\dots,N_{\text{CARRIERS}}-1$ . The operations in this section  
 8 shall be carried out for the carrier with index  $k$  if and only if  $\text{RLDPISectorScramble}$ , which is part of  
 9 the public data of the Overhead Messages Protocol for carrier  $k$ , is set to 0. A cell-specific scrambling  
 10 symbol shall be generated for each subcarrier, but only some of the generated scrambling symbols  
 11 shall be used and the rest shall be discarded. The scrambling symbols that shall be used shall be those  
 12 generated for subcarriers that correspond to R-DPICH hop-ports (via the hop-permutation), as defined  
 13 in 9.4.1.6.1. These subcarriers are henceforth referred to as R-DPICH subcarriers. The cell-specific  
 14 scrambling sequence consists of a complex number for every subcarrier. The scrambling operation  
 15 shall consist of multiplying the unscrambled complex symbol on each R-DPICH subcarrier by the  
 16 corresponding entry in the scrambling sequence. Each complex number in the cell-specific  
 17 scrambling sequence is generated from two bits, denoted by  $s_I$  and  $s_Q$ , using the following mapping:

- 18 1. The bit combination  $(s_I, s_Q) = (0,0)$  is mapped to the complex number  $\left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$ .

- 1           2. The bit combination  $(s_I, s_Q) = (0,1)$  is mapped to the complex number  $\left(\frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}\right)$ .
- 2           3. The bit combination  $(s_I, s_Q) = (1,0)$  is mapped to the complex number  $\left(\frac{-1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$ .
- 3           4. The bit combination  $(s_I, s_Q) = (1,1)$  is mapped to the complex number  $\left(\frac{-1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}\right)$ .

4           The cell-specific scrambling sequence for the carrier  $k$  shall be generated using two 20-bit registers,  
 5           called the I-register and the Q-register, as shown in Figure 115 and Figure 116, respectively. The I-  
 6           register shall have a generator polynomial  $h_I(D) = D^{20} + D^{19} + D^{16} + D^{14} + 1$  i.e., the  $n$ 'th output  $I(n)$   
 7           of the register shall satisfy  $I(n) = I(n-20) \oplus I(n-19) \oplus I(n-16) \oplus I(n-14)$ . The Q-register shall have a  
 8           generator polynomial  $h_Q(D) = D^{20} + D^{18} + D^{15} + D^{14} + 1$  i.e., the  $n$ 'th output  $Q(n)$  of the register shall  
 9           satisfy  $Q(n) = Q(n-20) \oplus Q(n-18) \oplus Q(n-15) \oplus Q(n-14)$ . Each entry in the cell-specific scrambling  
 10          sequence shall be generated using  $s_I$  and  $s_Q$  bits which are respectively the outputs of the I-register  
 11          and the Q-register after they have been appropriately initialized and clocked as in the following  
 12          description.

13          Let CellPilotPN be the 12 bit number obtained from the PilotPN by setting its 5<sup>th</sup>, 6<sup>th</sup> and 7<sup>th</sup> bits to  
 14          zero (where the bits are numbered starting from 0, with the 0<sup>th</sup> bit denoting the LSB). For the  
 15          superframe with index  $s$ , let SFInd be set equal to  $s$  in SemiSynchronous mode and set equal to zero  
 16          in Asynchronous mode. For the superframe with index  $s$ , let  $b_{11}, b_{10}, \dots, b_0$  be the 12 bits of  
 17           $(\text{CellPilotPN} + \text{SFInd} + k) \bmod 4096$ , with  $b_{11}$  being the MSB and  $b_0$  being the LSB. At the start of the  
 18          OFDM symbol with index 0 in the superframe, both the I and the Q registers shall be initialized to the  
 19          state  $[11110000b_{11}b_{10}b_9b_8b_7b_6b_5b_4b_3b_2b_1b_0]$ . The outputs of the I and Q registers after they are both  
 20          clocked  $i$  times, shall respectively be the  $s_I$  and  $s_Q$  bits used to generate a symbol  $c(i)$  in the  
 21          scrambling sequence. This symbol  $c(i)$  shall be used to scramble the subcarrier with index  $i \bmod$   
 22           $N_{\text{CARRIER\_SIZE}}$  in the carrier  $k$  in the OFDM Symbol with index  $\lfloor i / N_{\text{CARRIER\_SIZE}} \rfloor$  in the superframe,  
 23          provided this subcarrier is an R-DPICH subcarrier.

### 9.4.1.7 Time-domain processing

The sequence of OFDM symbols at the output of the sector scrambler shall be converted to a baseband waveform according to the procedure described in Figure 117. This procedure consists of an Inverse Fourier Transform (IFT) operation, a windowing operation, and an overlap-and-add operation.

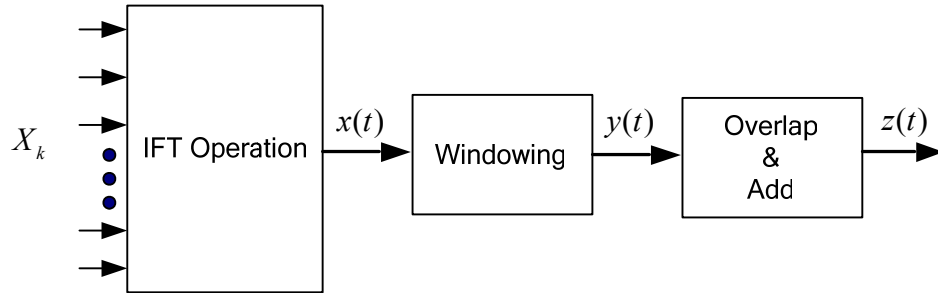


Figure 117 Time-domain processing

#### 9.4.1.7.1 Symbol start time

The start time of the superframe with index  $i$  is given by the product of  $i$  with the superframe duration  $T_{\text{SUPERFRAME}}$ , where  $T_{\text{SUPERFRAME}}$  is as defined in 9.4.1.2.4.

In FDD, the start time of the  $k$ -th OFDM symbol in the superframe,  $k$  ranging from 0 to  $N_{\text{PREAMBLE}} + N_{\text{FDD,RLPHYFrames}} N_{\text{FRAME,R}} - 1$ , is given by  $T_{\text{START,SF}} + kT_{\text{s,PR}}$  if  $k$  is less than  $N_{\text{PREAMBLE}}$ , and is given by  $T_{\text{START,SF}} + N_{\text{PREAMBLE}}T_{\text{s,PR}} + (k - N_{\text{PREAMBLE}})T_{\text{s}}$ , otherwise. Here  $T_{\text{START,SF}}$  is the start time of the superframe,  $T_{\text{s}}$  and  $T_{\text{s,PR}}$  are as defined in 9.4.1.2.3, and  $N_{\text{FDD,RLPHYFrames}}$  is defined by the Lower MAC sublayer.

In TDD, the start time of the  $k$ -th OFDM symbol in the superframe,  $k$  ranging from 0 to  $N_{\text{TDD,RLPHYFrames}} * N_{\text{FRAME,R}} - 1$ , is given by  $T_{\text{START,SF}} + N_{\text{PREAMBLE}}T_{\text{s,PR}} + N_{\text{FL\_BURST}} N_{\text{FRAME,F}}T_{\text{s}} + T_{\text{G,TDD,F}} + kT_{\text{s}} + \lfloor k / (N_{\text{RL\_BURST}} N_{\text{FRAME,R}}) \rfloor * (N_{\text{FL\_BURST}} N_{\text{FRAME,F}}T_{\text{s}} + T_{\text{G,TDD,F}} + T_{\text{G,TDD,R}})$ , where  $T_{\text{START,SF}}$  is the start time of the superframe,  $T_{\text{s}}$  is as defined in 9.4.1.2.3, and  $N_{\text{TDD,RLPHYFrames}}$  is defined by the Lower MAC sublayer.

#### 9.4.1.7.2 IFT operation

Let  $X_k$  be the value of the complex modulation symbol on the  $k$ 'th subcarrier of an OFDM symbol,  $k$  ranging from 0 to  $N_{\text{FFT}} - 1$ . The IFT of the OFDM symbol is given by the infinite duration signal:

$$x(t) = \frac{1}{\sqrt{N_{\text{FFT}}}} \sum_{k=0}^{N_{\text{FFT}}-1} X_k e^{j2\pi(k - N_{\text{FFT}}/2)(t - T_{\text{CP,PR}} - T_{\text{START}})/(N_{\text{FFT}}T_{\text{CHIP}})}$$

during the first  $N_{\text{PREAMBLE}}$  OFDM symbols in the superframe preamble in FDD, and in all other cases (including all the OFDM symbols for TDD) by the equation:

$$x(t) = \frac{1}{\sqrt{N_{\text{FFT}}}} \sum_{k=0}^{N_{\text{FFT}}-1} X_k e^{j2\pi(k - N_{\text{FFT}}/2)(t - T_{\text{CP}} - T_{\text{START}})/(N_{\text{FFT}}T_{\text{CHIP}})}$$



1 where  $T_{START}$  denotes the start time of the OFDM symbol,  $T_{CP}$  and  $T_{CP,R}$  are as defined in 9.4.1.2.2,  
 2 and  $j$  denotes the complex number (0,1).

### 3 9.4.1.7.3 Windowing

4 The signal  $x(t)$  at the output of the IFT shall be multiplied by the window function  $w(t)$ , where  $w(t)$  is  
 5 given by the equation:

$$6 \quad w(t) = \begin{cases} 0 & , t < T_{START} - T_{WGI} \\ 0.5 - 0.5 \cos\left(\frac{\pi(t + T_{WGI} - T_{START})}{T_{WGI}}\right) & , T_{START} - T_{WGI} \leq t < T_{START} \\ 1 & , T_{START} \leq t < T_{START} + T_{CP,PR} + T_{FFT} \\ 0.5 + 0.5 \cos\left(\frac{\pi(t - T_{START} - T_{CP,PR} - T_{FFT})}{T_{WGI}}\right) & , T_{START} + T_{CP,PR} + T_{FFT} \leq t < T'_{s,PR} \\ 0 & , t \geq T'_{s,PR} \end{cases}$$

7 during the first  $N_{PREMABLE}$  OFDM symbols in the superframe preamble in FDD, and in all other cases  
 8 (including all the OFDM symbols for TDD) by the equation:

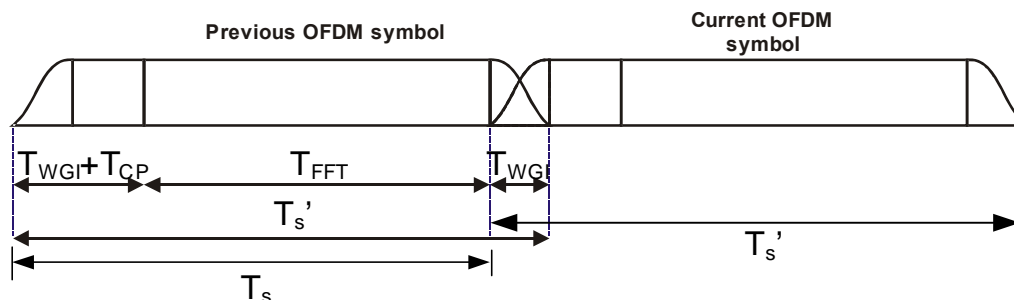
$$9 \quad w(t) = \begin{cases} 0 & , t < T_{START} - T_{WGI} \\ 0.5 - 0.5 \cos\left(\frac{\pi(t + T_{WGI} - T_{START})}{T_{WGI}}\right) & , T_{START} - T_{WGI} \leq t < T_{START} \\ 1 & , T_{START} \leq t < T_{START} + T_{CP} + T_{FFT} \\ 0.5 + 0.5 \cos\left(\frac{\pi(t - T_{START} - T_{CP} - T_{FFT})}{T_{WGI}}\right) & , T_{START} + T_{CP} + T_{FFT} \leq t < T'_s \\ 0 & , t \geq T'_s \end{cases}$$

10 during each RL PHY Frame, where  $T_{START}$  denotes the start time of the OFDM symbol. The  
 11 quantities  $T_{FFT}$ ,  $T_s$ ,  $T'_s$ ,  $T_{s,PR}$  and  $T'_{s,PR}$  are as defined in 9.4.1.2.2.

12 The windowed signal  $y(t)$  is given by  $y(t) = x(t)w(t)$ .

1 **9.4.1.7.4 Overlap and add operation**

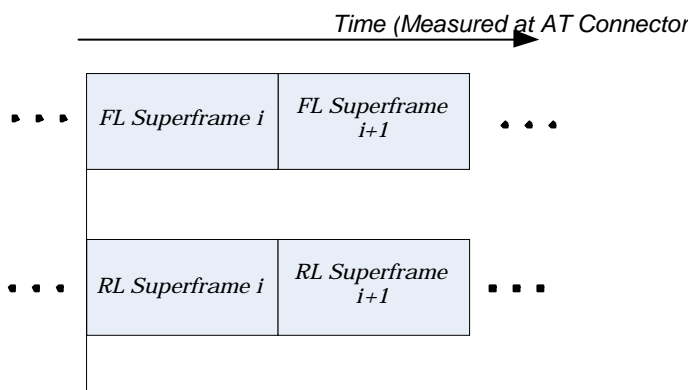
2 The windowed IFTs  $y(t)$  corresponding to all of the OFDM symbols shall be added together to create  
 3 the final complex baseband waveform  $z(t)$ . In this procedure, neighboring OFDM symbols overlap for  
 4 a duration  $T_{WGI}$  as illustrated in Figure 118.



5  
6 **Figure 118 Overlap and add operation**

7 **9.4.2 Synchronization and timing**

8 The access terminal shall establish a time reference that is used to derive timing for all time-critical  
 9 transmission components, including superframe boundaries, PHY Frame boundaries, etc. The access  
 10 terminal initial time reference shall be established from the acquired Acquisition Channel  
 11 (F-ACQCH) and from the SystemTimeLSB field transmitted as part of the Primary Broadcast  
 12 Channel (F-pBCH). The initial access terminal time reference shall coincide with the time of  
 13 occurrence, as measured at the access terminal antenna connector, of the earliest arriving multipath  
 14 component of the forward link waveform. To elaborate, the beginning of the reverse link superframe  
 15 with index  $i$  shall coincide with the beginning of the forward link superframe with index  $i$ , where the  
 16 beginning of both superframes are measured at the access terminal antenna connector. The inaccuracy  
 17 in this time-alignment shall be within  $\pm 1\mu s$ .



18  
19 **Figure 119 Relationship between forward link and reverse link timings**

20 After the initial time reference has been established, the access terminal shall advance and retard  
 21 timing in response to the AccessGrant message of the SS MAC Protocol and the TimingCorrection  
 22 message of this protocol, as specified in the following. The TimingCorrection message shall be

1 declared in error if it contains two SectorIDs within the same synchronous subgroup. Otherwise, the  
2 access terminal shall use the TimingCorrection field corresponding to the SectorID in AS<sub>SYNCH</sub>. If no  
3 such SectorID is present, the access terminal shall not retard or advance the timing for that AS<sub>SYNCH</sub>.

4 To advance timing by a period of  $k$  chips, the access terminal shall move its time reference earlier by  
5 a period of  $kT_c$ , where  $T_c$  is the chip duration. To retard timing by a period of  $k$  chips, the access  
6 terminal shall move its time reference later by a period of  $kT_c$ , where  $T_c$  is the chip duration.

7 If the Reverse Link Serving Sector (RLSS) is contained within AS<sub>SYNCH</sub>, the access terminal shall also  
8 move its time reference when the RLSS changes from one sector to another sector within AS<sub>SYNCH</sub>. If,  
9 at the access terminal connector, the time of arrival of the superframe boundary from the new RLSS  
10 is later than the time of arrival of the superframe boundary from the previous RLSS (for the same  
11 superframe index), then the access terminal shall move its time reference earlier (i.e., advance its  
12 timing) by the difference in the times of arrival between the two sectors. If the time of arrival of the  
13 superframe boundary from the new RLSS is earlier than the time of arrival of the superframe  
14 boundary from the previous RLSS (for the same superframe index), then the access terminal shall  
15 move its time reference later (i.e., retard its timing) by the difference in the times of arrival between  
16 the two sectors.<sup>87</sup>

17 The access terminal shall maintain independent time references for transmission to asynchronous  
18 APs.

19

---

<sup>87</sup> This ensures that the time of arrival of the access terminal signal at the RLSS remains unchanged, even though the RLSS itself changes. The assumption is that the forward link propagation delay is the same as the reverse link propagation delay.

## 10 Common Algorithms and Data Structures

### 10.1 Channel band record

The channel band record defines an access network (AN) channel frequency and the system bandwidth on that frequency. This record contains the following fields:

Field	Length (bits)
ChannelBandRecordType	1
ReverseLinkCenterFrequencyIncluded	1
ForwardLinkSystemBandwidthIncluded	1
ReverseLinkSystemBandwidthIncluded	1
ChannelBandRecordLength	4
ChannelBandRecord	ChannelBandRecordLength × 8

#### ChannelBandRecordType

The access network shall set this field to an identifier for the channel band record type according to Table 101.

**Table 101 ChannelBandRecordType for channel band record**

Value	ChannelBandRecordType
'0'	Band Class
'1'	Frequency Specified

#### ReverseLinkCenterFrequencyIncluded

The access network shall set this field to '0' if the value of the ChannelBandRecordType is '0', or if the value of the center frequency of the reverse link channel is not included in the ChannelBandRecord. Otherwise, the access network shall set this field to '1'.

#### ForwardLinkSystemBandwidthIncluded

The access network shall set this field to '0' if the value of the ChannelBandRecordType is '0', or if the value of the system bandwidth of the forward link channel is not included in the ChannelBandRecord. Otherwise, the access network shall set this field to '1'.

#### ReverseLinkSystemBandwidthIncluded

The access network shall set this field to '0' if the value of the ChannelBandRecordType is '0', if the value of the system bandwidth of the reverse link channel is the same as the value of the system bandwidth of the forward link channel, or if the value of the system bandwidth of the reverse link channel is not included in the ChannelBandRecord. Otherwise, the access network shall set this field to '1'.

1 ChannelBandRecordLength

2 The access network shall set this field to the length of the  
3 ChannelBandRecord field in units of octets.

4 ChannelBandRecord If ChannelBandRecordType is '0', then the access network shall set this  
5 record as defined in 10.1.1. If ChannelBandRecordType is '1', then the  
6 sender shall set this record as defined in 10.1.2.

7 **10.1.1 Definition of ChannelBandRecord record for Band Class**

Field	Length (bits)
BandClass	5
ChannelBandNumber	11

9 BandClass The access network shall set this field to the band class number  
10 corresponding to the frequency assignment of the channel specified by this  
11 record.

12 ChannelBandNumber The access network shall set this field to the channel number corresponding  
13 to the frequency assignment of the channel specified by this record.

14 **10.1.2 Definition of ChannelBandRecord record for Frequency Specified**

Field	Length (bits)
ForwardLinkCenterFrequency	24
ReverseLinkCenterFrequency	0 or 24
ForwardLinkSystemBandwidth	0 or 16
ReverseLinkSystemBandwidth	0 or 16

16 ForwardLinkCenterFrequency  
17 The access network shall set this field to the value of the center frequency of  
18 the forward link channel in units of kHz.

19 ReverseLinkCenterFrequency  
20 If ReverseLinkCenterFrequencyIncluded is '0', then the access network shall  
21 omit this field. Otherwise, the access network shall set this field to the value  
22 of the center frequency of the reverse link channel in units of kHz.

23 ForwardLinkSystemBandwidth  
24 If ForwardLinkSystemBandwidthIncluded is '0', then the access network  
25 shall omit this field. Otherwise, the access network shall set this field to the  
26 value of the system bandwidth of the forward link channel in units of kHz.

27 ReverseLinkSystemBandwidth  
28 If ReverseLinkSystemBandwidthIncluded is '0', then the access network  
29 shall omit this field. Otherwise, the access network shall set this field to the  
30 value of the system bandwidth of the reverse link channel in units of kHz.

## 10.2 Access terminal identifier record

The Access Terminal Identifier record provides a unicast, multicast, or broadcast access terminal (AT) address. This record contains the following fields:

Field	Length (bits)
ATIType	2
ATI	0 or 32

**ATIType** Access Terminal Identifier Type. This field shall be set to the type of the ATI, as shown in Table 102.

**ATI** Access Terminal Identifier. The field is included only if ATIType is not equal to '00'. This field shall be set as shown in Table 102.

**Table 102 ATIType field encoding**

ATIType	ATIType Description	ATI Length (bits)
'00'	Broadcast ATI (BATI)	0
'01'	Multicast ATI (MATI)	32
'10'	Unicast ATI	32
'11'	Reserved	N/A

## 10.3 Attribute record

The attribute record defines a value for a given attribute.

An attribute can be one of the following two types:

- Simple attribute if it contains a single value.
- Complex attribute if it contains multiple values that together form a complex value for a particular attribute identifier.

The type of the attribute is determined by the attribute identifier.

The format of a simple attribute is given by:

Field	Length (bits)
Length	8
AttributeID	Protocol Specific
AttributeValue	Attribute dependent
Reserved	Variable

**Length** Length in octets of the attribute record, excluding the Length field.

1	AttributeID	Attribute identifiers are unique in the context of the protocol being
2		configured.
3	AttributeValue	A suggested value for the attribute. In general, attribute value lengths are an
4		integer number of octets. Attribute values have an explicit or implicit length
5		indication (e.g., fixed length or null terminated strings) so that the recipient
6		can successfully parse the record.
7	Reserved	The length of this field is the smallest value that will make the attribute
8		record octet aligned. The sender shall set this field to zero. The receiver shall
9		ignore this field.

10 The format of a complex attribute is given by:

11

Field	Length (bits)
Length	8
AttributeID	Protocol Specific

An appropriate number of instances of the following record:

AttributeValue	Attribute dependent
----------------	---------------------

Reserved	Variable
----------	----------

12

Length Length in octets of the attribute record, excluding the Length field.

13

AttributeID Attribute identifiers are unique in the context of the protocol being

14

15

AttributeValue A suggested value for the attribute. In general, attribute value lengths are an

16

17

18

19

Reserved The length of this field is the smallest value that will make the attribute

20

21

## 22 10.4 Hash function

23

The hash function takes three arguments:

24

**Key** This argument shall be 32 bits; typically, the access terminal's ATI.

25

**N** The number of resources.

26

**Decorrelate** An argument used to de-correlate values obtained for different applications

27

for the same access terminal.

1 Define:

- 2 ■ Word  $L$  to be bits 0-15 of  $Key$
- 3 ■ Word  $H$  to be bits 16-31 of  $Key$

4 where bit 0 is the least significant bit of  $Key$ .

5 The hash value is computed as follows<sup>88</sup>:

$$6 \quad R(L, H, N, Decorrelate) = \lfloor N \times ((40503 \times (L \oplus H \oplus Decorrelate)) \bmod 2^{16}) / 2^{16} \rfloor.$$

## 7 **10.5 Computation of the CRC bits**

8 This section describes the computation of CRC bits for a stream of data bits  $\{b_{N-1}, b_{N-2}, \dots, b_1, b_0\}$ .  
 9 Here  $b_0$  is the initial bit in the stream and  $b_{N-1}$  is the final bit.

10 The CRC shall be calculated using the standard CRC-CCITT generator polynomial:

$$11 \quad g(x) = x^{24} + x^{23} + x^6 + x^5 + x + 1$$

12 The CRC shall be equal to the value computed according to the following procedure as shown in  
 13 Figure 120:

- 14 ■ All shift-register elements shall be initialized to '0's.
- 15 ■ The register shall be clocked once for each data bit. The bit stream shall be read starting  
 16 from  $b_0$  and ending in  $b_{N-1}$ .
- 17 ■ When all the data bits are exhausted, the values remaining in the shift registers  $\{c_0, c_1, \dots,$   
 18  $c_{23}\}$  constitute the 24-bit CRC. The data stream padded with a 24-bit CRC shall be  $\{c_0,$   
 19  $c_1, \dots, c_{22}, c_{23}, b_{N-1}, b_{N-2}, \dots, b_1, b_0\}$  i.e.,  $b_0$  is the initial bit and  $c_0$  is the last bit in the  
 20 padded stream.
- 21 ■ If less than a 24-bit CRC is to be used, the most significant bits of the 24-bit CRC shall  
 22 be used. The data stream padded with a  $M$ -bit CRC ( $M < 24$ ) CRC shall be  $\{c_{24-M}, c_{24-M+1},$   
 23  $\dots, c_{23}, b_{N-1}, b_{N-2}, \dots, b_1, b_0\}$  i.e.,  $b_0$  is the initial bit and  $c_{24-M}$  is the last bit in the padded  
 24 stream.

---

<sup>88</sup> This formula is adapted from Knuth, D. N., *Sorting and Searching*, vol. 3 of *The Art of Computer Programming*, 3 vols. (Reading, MA: Addison-Wesley, 1973), pp. 508-513. The symbol  $\oplus$  represents bitwise exclusive-or function (or modulo 2 addition) and the symbol  $\lfloor \rfloor$  represents the "largest integer smaller than" function.



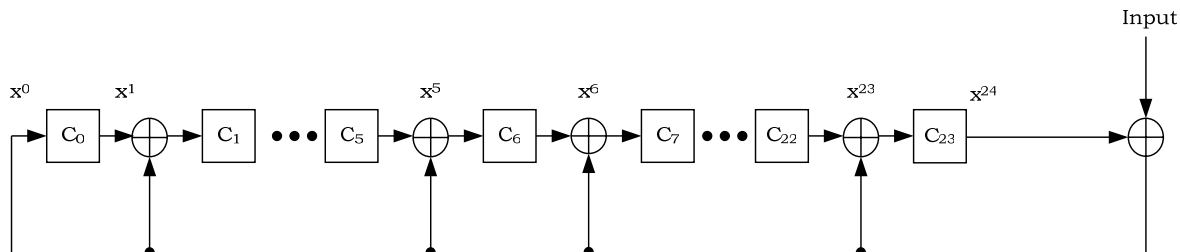


Figure 120 CRC calculation

## 10.6 Pseudorandom number generator

### 10.6.1 General procedures

When an access terminal is required to use the pseudo random number generator described in this section, then the access terminal shall implement the linear congruential generator defined by:

$$z_n = a \times z_{n-1} \text{ mod } m$$

where  $a = 7^5 = 16807$  and  $m = 2^{31} - 1 = 2147483647$ .  $z_n$  is the output of the generator.<sup>89</sup>

The access terminal shall initialize the random number generator as defined in 10.6.2.

The access terminal shall compute a new  $z_n$  for each subsequent use.

The access terminal shall use the value  $u_n = z_n / m$  for those applications that require a binary fraction  $u_n$ ,  $0 < u_n < 1$ .

The access terminal shall use the value  $k_n = \lfloor N \times z_n / m \rfloor$  for those applications that require a small integer  $k_n$ ,  $0 \leq k_n \leq N-1$ .

### 10.6.2 Initialization

The access terminal shall initialize the random number generator by setting  $z_0$  to

$$z_0 = (\text{HardwareID} \oplus \chi) \text{ mod } m$$

where HardwareID is the least 32 bits of the hardware identifier associated with the access terminal, and  $\chi$  is a 32-bit, time-varying physical measure available to the access terminal. If the initial value so produced is found to be zero, the access terminal shall repeat the procedure with a different value of  $\chi$ .

<sup>89</sup> This generator has full period, ranging over all integers from 1 to  $m-1$ ; the values 0 and  $m$  are never produced. Several suitable implementations can be found in Park, Stephen K. and Miller, Keith W., "Random Number Generators: Good Ones are Hard to Find," *Communications of the ACM*, vol. 31, no. 10, October 1988, pp. 1192-1201.

## 10.7 Sequence number

When the order in which protocol messages are delivered is important, air interface protocols use a sequence number to verify this order.

The sequence number has  $s$  bits. The sequence space is  $2^s$ . All operations and comparisons performed on sequence numbers shall be carried out in unsigned modulo  $2^s$  arithmetic. For any message sequence number  $N$ , the sequence numbers in the range  $[N+1, N+2^{s-1}-1]$  shall be considered greater than  $N$ , and the sequence numbers in the range  $[N-2^{s-1}, N-1]$  shall be considered smaller than  $N$ .

### 10.7.1 Sequence number initialization

Upon entering into the initial state, the sequence number on the sender side shall be initialized to 0. The sequence number on the receiver,  $V(R)$ , shall be initialized to  $2^{s-1}$ .

### 10.7.2 Sequence number validation

The receiver of the message maintains a receive pointer  $V(R)$  whose initialization is defined in 10.7.1. When a message arrives, the receiver compares the sequence number of the message with  $V(R)$ . If the sequence number is greater than  $V(R)$ , the message is considered a valid message and  $V(R)$  is set to this sequence number. Otherwise, the message is considered an invalid message.

## 10.8 TransactionID number

A TransactionID is used in some protocol messages to associate a response message with an initiating message. The initiator sets the value of the TransactionID field in the message. The responder uses the same TransactionID in the response message to associate the response message with the corresponding initiating message. A TransactionID shall not be used to order protocol messages at the responder.

### 10.8.1 TransactionID initiating procedures

The initiator shall maintain a separate list of the TransactionIDs for each initiating message for which it is expecting a response message. The initiator shall set the TransactionID field for each message sent to a value which shall meet the following requirements:

- The initiator is not expecting the TransactionID in a response message.
- The TransactionID has not been used in the last 8 response messages received for that initiating message.

## 10.9 Generic Attribute Update Protocol

### 10.9.1 Introduction

The Generic Attribute Update Protocol provides a means to update and query the values of protocol attributes. The protocol defines an AttributeUpdateRequest message, an AttributeUpdateAccept message, and an AttributeUpdateReject message to negotiate a mutually acceptable configuration. The protocol defines an AttributeValueRequest message and an AttributeValueResponse message to query the value of an attribute.

1 The initiator uses the AttributeUpdateRequest message to provide the responder with a proposed  
2 value for each attribute. The responder uses the AttributeUpdateAccept message to accept the  
3 proposed values. If the responder is an access network and if any of the attribute values in the  
4 received AttributeUpdateRequest message is not acceptable to it, then the access network sends the  
5 AttributeUpdateReject message, and the access terminal and access network continue to use the  
6 previously negotiated values for the attributes.

7 The access terminal shall not send an AttributeUpdateReject message.

8 After receiving an AttributeValueRequest message, the receiver shall send an  
9 AttributeValueResponse message within time  $T_{\text{Turnaround}}$ , unless specified otherwise by the protocol  
10 which uses the Generic Attribute Update Protocol.

## 11 **10.9.2 Procedures**

### 12 **10.9.2.1 Initiator requirements**

13 The initiator shall not send an AttributeUpdateRequest message if the value of the ConfigurationLock  
14 public data of the Session Configuration Protocol is Locked.

15 The initiator shall include one attribute value for each attribute included in the  
16 AttributeUpdateRequest message.

17 After sending an AttributeUpdateRequest message, the initiator should continue to use previously  
18 negotiated values for attributes listed in the message until it receives either an AttributeUpdateAccept  
19 message or an AttributeUpdateReject message. However, the initiator should be prepared for the  
20 responder to begin using attribute values proposed by the initiator in the AttributeUpdateRequest  
21 message.

22 If the initiator receives an AttributeUpdateAccept message, then it shall pair the received message  
23 with the associated AttributeUpdateRequest message using the TransactionID field of the messages.  
24 The initiator shall use the attribute values in the AttributeUpdateRequest message as the configured  
25 attribute values. If the access terminal receives an AttributeUpdateReject message, then it shall use  
26 the previously configured values of the attributes included in the corresponding  
27 AttributeUpdateRequest message.

28 If the initiator does not receive the corresponding AttributeUpdateAccept or AttributeUpdateReject  
29 message in response to the AttributeUpdateRequest message, it should re-transmit the  
30 AttributeUpdateRequest message.

31 While the initiator is waiting for a response to an AttributeUpdateRequest message, it shall not  
32 transmit another AttributeUpdateRequest message with a different TransactionID field that requests  
33 reconfiguration of an attribute included in the original AttributeUpdateRequest message.

### 10.9.2.2 Responder requirements

After receiving an AttributeUpdateRequest message, the responder shall respond within time  $T_{\text{Turnaround}}$ , unless specified otherwise by the protocol which uses the Generic Attribute Update Protocol.

If the responder is an access terminal, then:

- The responder shall send an AttributeUpdateAccept message.
- Upon sending an AttributeUpdateAccept message, the responder shall begin using the accepted attribute values.

If the responder is an access network, then:

- If the responder finds the proposed value for each attribute in the AttributeUpdateRequest message to be acceptable, then the responder shall send an AttributeUpdateAccept message. Upon sending an AttributeUpdateAccept message, the responder shall begin using the accepted attribute values.
- If the responder does not recognize an attribute or does not find a proposed attribute value to be acceptable, then it shall send an AttributeUpdateReject message.
- If the responder sends an AttributeUpdateReject message, then it shall continue to use the previously configured values of the attributes found in the corresponding AttributeUpdateRequest message.

### 10.9.3 Message formats

No protocol or transport shall define a message with the same MessageID value as the AttributeUpdateRequest, AttributeUpdateAccept, AttributeUpdateReject, AttributeValueRequest, AttributeValueResponse messages of the Generic Attribute Update Protocol.

#### 10.9.3.1 AttributeUpdateRequest

The sender sends an AttributeUpdateRequest message to offer an attribute-value for a given attribute.

Field	Length (bits)
MessageID	8
TransactionID	8
One or more instances of the following record	
AttributeRecord	Attribute dependent

**MessageID** The sender shall set this field to 0x52. The value of this field is the same for all protocols using this message.

**TransactionID** The sender shall set this field according to 10.8.

**AttributeRecord** The format of this record is specified in 10.3.

<b>Channels</b>	FTC    RTC	<b>SLP</b>	Reliable
<b>Addressing</b>	Unicast	<b>Security</b>	Required

1

### 2 10.9.3.2 AttributeUpdateAccept

3 The sender sends an AttributeUpdateAccept message in response to an AttributeUpdateRequest  
4 message to accept the offered attribute value.

5

Field	Length (bits)
MessageID	8
TransactionID	8

6 MessageID                    The sender shall set this field to 0x53. The value of this field is the same for  
7 all protocols using this message.

8 TransactionID                The sender shall set this value to the TransactionID field of the  
9 corresponding AttributeUpdateRequest message.

10

<b>Channels</b>	FTC    RTC	<b>SLP</b>	Reliable
<b>Addressing</b>	Unicast	<b>Security</b>	Required

### 11 10.9.3.3 AttributeUpdateReject

12 The access network sends an AttributeUpdateReject message in response to an  
13 AttributeUpdateRequest message to reject the offered attribute value.

14

Field	Length (bits)
MessageID	Protocol dependent
TransactionID	8

15 MessageID                    The sender shall set this field to 0x54. The value of this field is the same for  
16 all protocols using this message.

17 TransactionID                The sender shall set this value to the TransactionID field of the  
18 corresponding AttributeUpdateRequest message.

19

<b>Channels</b>	FTC	<b>SLP</b>	Reliable
<b>Addressing</b>	Unicast	<b>Security</b>	Required

20

### 10.9.3.4 AttributeValueRequest

The sender sends an AttributeValueRequest message to request the value of an attribute.

Field	Length (bits)
MessageID	8
TransactionID	8
AttributeCount	8

AttributeCount instances of the following field:

{	
AttributeID	16
}	

**MessageID** The sender shall set this field to 0x55.

**TransactionID** The sender shall set this field according to 10.8.

**AttributeCount** The sender shall set this field to the number of AttributeID fields included in this message.

**AttributeID** The sender shall set this field to the AttributeID for which this request is generated.

<b>Channels</b>	FTC    RTC	<b>SLP</b>	Reliable
<b>Addressing</b>	Unicast	<b>Security</b>	Required

### 10.9.3.5 AttributeValueResponse

The sender sends an AttributeValueResponse message in response to an AttributeValueRequest message.

Field	Length (bits)
MessageID	8
TransactionID	8
AttributeCount	8

AttributeCount instances of the following field:

{	
AttributeRecord	Attribute dependent
}	

**MessageID** The sender shall set this field to 0x56.

**TransactionID** The sender shall set this value to the TransactionID field of the corresponding AttributeValueRequest message.

1 AttributeCount The sender shall set this field to the number of AttributeID fields included in  
2 this message.

3 AttributeRecord The format of this record is specified in 10.3.  
4

<b>Channels</b>	FTC RTC	<b>SLP</b>	Reliable
<b>Addressing</b>	Unicast	<b>Security</b>	Required

5

#### 6 10.9.4 Protocol numeric constants

7

Constant	Meaning	Value
$T_{\text{Turnaround}}$	Maximum time to respond to an AttributeUpdateRequest message.	2 sec

#### 8 10.10 Session state information record

9 The Session State Information is to be used for transferring the session parameters corresponding to  
10 the InUse and Suspended protocol and transport instances from a source access network to a target  
11 access network. Session parameters are the attributes and the internal parameters that define the state  
12 of each protocol. The format of this record is shown in Table 103. If an attribute is not contained in  
13 the Session State Information record, the target access network shall assume that the missing  
14 attributes have the default values (specified for each attribute in each protocol). The sender shall  
15 include all of the Parameter Records associated with the ProtocolType and ProtocolSubtype in the  
16 same Session State Information Record.

17

**Table 103 Format of the session state information record**

Field	Length (bits)
FormatID	8
ProtocolType	8
ProtocolSubtype	16
One or more instances of the following Parameter Record:	
ParameterType	8
ParameterType-specific record	Variable

18 FormatID This field identifies the format of the rest of the fields in this record and shall  
19 be set to 0x00.

20 ProtocolType This field shall be set the Type value (see Table 9) for the protocol associated  
21 with the encapsulated session parameters.

22 ProtocolSubtype This field shall be set to the protocol subtype value (see Table 105) for the  
23 protocol associated with the encapsulated session parameters.

24 ParameterType This field shall be set according to Table 104.

**Table 104 Encoding of the DataType field**

Field Value	Meaning
0x00	The ParameterType-specific record consists of a Complex or a Simple Attribute as defined in 10.3.
All other values	ParameterType-specific records are protocol dependent.

**ParameterType-specific record**

If the ParameterType field is set to 0x00, then this record shall be set to the simple or complex attribute (see 10.3) associated with the protocol identified by the (ProtocolType, ProtocolSubtype) pair. Otherwise, the structure of this record shall be as specified by the protocol which is identified by the (ProtocolType, ProtocolSubtype) pair.

**10.11 SectorID provisioning**

The SectorID may have the format of an IPv6 address from one of the following four address pools: Global Unicast, Unique-Local Unicast, Link-Local Unicast, and Reserved defined in RFC 3513 [21] and RFC 4193 [23].

**10.11.1 SectorID construction**

The access network shall construct the SectorID to be either a Globally Unique SectorID or a Locally Unique SectorID as described in the following:

- If a Globally Unique SectorID is used, the SectorID is universally unique by construction.
- If a Locally Unique SectorID is used, it is the responsibility of the access network to ensure the uniqueness of the SectorID throughout the access networks that the access terminal can visit while maintaining the session.

The methods for constructing the SectorID are beyond the scope of this specification.



## 11 Assigned Names and Numbers

### 11.1 Protocols

Table 105 shows the Protocol Types and Protocol Subtypes assigned to the protocols defined in this specification.

**Table 105 Protocol types and subtypes**

Protocol Type		Protocol Subtype	
Name	ID	Name	ID
Physical Layer	0x00	Default Physical Layer	0x0000
Control Channel MAC	0x01	Default Control Channel MAC	0x0000
Access Channel MAC	0x02	Default Access Channel MAC	0x0000
Forward Traffic Channel MAC	0x03	Default Forward Traffic Channel MAC	0x0000
Reverse Traffic Channel MAC	0x04	Default Reverse Traffic Channel MAC	0x0000
Reverse Control Channel MAC	0x05	Default Reverse Control Channel MAC	0x0000
Shared Signaling MAC	0x06	Default Shared Signaling MAC	0x0000
Air Link Management	0x07	Default Air Link Management	0x0000
Initialization State	0x08	Default Initialization State	0x0000
Idle State	0x09	Default Idle State	0x0000
Connected State	0x0a	Default Connected State	0x0000
Active Set Management	0x0b	Default Active Set Management	0x0000
Overhead Messages	0x0c	Overhead Messages	0x0000
Authentication	0x0d	Default Authentication	0x0000
Encryption	0x0e	Default Encryption	0x0000
Encryption	0x0e	Generic Encryption	0x0001
Security	0x0f	Default Security	0x0000
Key Exchange	0x10	Default Key Exchange	0x0000
Session Management	0x11	Default Session Management	0x0000
Address Management	0x12	Default Address Management	0x0000
Session Configuration	0x13	Default Session Configuration	0x0000
Capabilities Discovery	0x14	Default Capabilities Discovery	0x0000
Inter RAT Protocol	0x15	Default Inter RAT Protocol	0x0000
Packet Consolidation	0x16	Default Packet Consolidation	0x0000
Transport 0	0x17	Transport Subtype per 11.2	See 11.2
Transport 1	0x18	Transport Subtype per 11.2	See 11.2
Transport 2	0x19	Transport Subtype per 11.2	See 11.2
Transport 3	0x1a	Transport Subtype per 11.2	See 11.2

Protocol Type		Protocol Subtype	
Name	ID	Name	ID
Transport 4	0x1b	Transport Subtype per 11.2	See 11.2
Transport 5	0x1c	Transport Subtype per 11.2	See 11.2
Transport 6	0x1d	Transport Subtype per 11.2	See 11.2
Transport 7	0x1e	Transport Subtype per 11.2	See 11.2

1

## 11.2 Transport subtype assignments

2

A transport subtype identifies the transport that is bound to a Transport of the Packet Consolidation Protocol. Table 106 shows the transport subtype defined in this specification.

3

4

5

**Table 106 Transport subtypes assignments**

Value	Name
0x0000	Default Signaling Transport
0x0001	Default Data Transport
0x0002	Test Transport
0xffff	Transport not used
All other values reserved	

6

## 11.3 Messages

7

8

Protocol/Transport		Message	
Subtype Name	Type ID	Name	ID
Default Data	0x18 – 0x1e	ActivateRoute	0x02
Default Data	0x18 – 0x1e	ActivateRouteAck	0x03
Default Active Set Management	0x0b	ActiveSetAssignment	0x03
Default Active Set Management	0x0b	ActiveSetComplete	0x04
Default Key Exchange	0x10	ANKeyComplete	0x03
Default Key Exchange	0x10	ATKeyComplete	0x04
All subtypes with configurable attributes	N/A	AttributeUpdateAccept	0x53
All subtypes with configurable attributes	N/A	AttributeUpdateReject	0x54
All subtypes with configurable attributes	N/A	AttributeUpdateRequest	0x52
All subtypes with configurable attributes	N/A	AttributeValueRequest	0x55
All subtypes with configurable attributes	N/A	AttributeValueResponse	0x56
Default Capabilities Discovery	0x14	CapabilitiesRequest	0x00
Default Capabilities Discovery	0x14	CapabilitiesResponse	0x01
Default Connected State	0x0a	ChannelMeasurementReport	0x08

Protocol/Transport		Message	
Subtype Name	Type ID	Name	ID
Default Connected State	0x0a	ChannelMeasurementReportRequest	0x07
Default Session Configuration	0x13	ConfigurationAccept	0x0a
Default Session Configuration	0x13	ConfigurationReject	0x0b
Default Session Configuration	0x13	ConfigurationRequest	0x09
Default Connected State	0x0a	ConnectionClose	0x00
Default Idle State	0x09	ConnectionOpenRequest	0x00
Default Idle State	0x09	ConnectionOpenResponse	0x01
Overhead Messages	0x0c	EncapsulatedQuickChannelInfo	0x02
Overhead Messages	0x0c	ExtendedChannelInfo	0x00
Default Data	0x18 – 0x1e	FlowQoSdetect	0x0d
Default Data	0x18 – 0x1e	FwdReservationAck	0x0c
Default Data	0x18 – 0x1e	FwdReservationOff	0x0b
Default Data	0x18 – 0x1e	FwdReservationOn	0x0a
Default Address Management	0x12	HardwareIDRequest	0x03
Default Address Management	0x12	HardwareIDResponse	0x04
Default Inter RAT	0x15	InterRATBlob	0x00
Default Inter RAT	0x15	InterRATIDRequest	0x01
Default Inter RAT	0x15	InterRATIDResponse	0x02
Default Session Management	0x11	KeepAliveRequest	0x02
Default Session Management	0x11	KeepAliveResponse	0x03
Default Key Exchange	0x10	KeyChangeAck	0x07
Default Key Exchange	0x10	KeyChangeInitiateRequest	0x05
Default Key Exchange	0x10	KeyChangeRequest	0x06
Default Key Exchange	0x10	KeyInitiateRequest	0x00
Default Key Exchange	0x10	KeyRequest	0x01
Default Key Exchange	0x10	KeyResponse	0x02
Default Session Configuration	0x13	LockConfiguration	0x05
Default Session Configuration	0x13	LockConfigurationAck	0x06
Default Connected State	0x0a	MIMOREquest	0x01
Default Reverse Traffic Channel MAC	0x04	OSIReport	0x01
Default Reverse Traffic Channel MAC	0x04	OSIReportRequest	0x00
Default Idle State	0x09	PageUATI	0x02
Default Active Set Management	0x0b	PilotReport	0x00
Default Active Set Management	0x0b	PilotReportRequest	0x06
Default Idle State	0x09	PreferredChannelRequest	0x03
Default Signaling	0x17	ReceiverStatus	0x04

Protocol/Transport		Message	
Subtype Name	Type ID	Name	ID
Default Air Link Management	0x07	Redirect	0x00
Default Data	0x18 – 0x1e	ReservationAccept	0x06
Default Data	0x18 – 0x1e	ReservationOffRequest	0x05
Default Data	0x18 – 0x1e	ReservationOnRequest	0x04
Default Data	0x18 – 0x1e	ReservationReject	0x07
Default Active Set Management	0x0b	ResetReport	0x05
Default Signaling	0x17	ResetRxAck	0x01
Default Signaling	0x17	ResetRxRequest	0x00
Default Signaling	0x17	ResetTxAck	0x03
Default Signaling	0x17	ResetTxIndication	0x02
Default Data	0x18 – 0x1e	RestartNetworkInterface	0x0e
Default Data	0x18 – 0x1e	RestartNetworkInterfaceAck	0x0f
Default Data	0x18 – 0x1e	RevReservationOff	0x09
Default Data	0x18 – 0x1e	RevReservationOn	0x08
Default Data	0x18 – 0x1e	RouteSelect	0x00
Default Data	0x18 – 0x1e	RouteSelectAck	0x01
Overhead Messages	0x0c	SectorParameters	0x01
Default Connected State	0x0a	SelectedInterlaceAck	0x04
Default Connected State	0x0a	SelectedInterlaceAssignment	0x03
Default Connected State	0x0a	SelectedInterlaceRequest	0x02
Default Session Management	0x11	SessionClose	0x01
Default Session Management	0x11	SessionOpen	0x00
Default Physical Layer	0x00	TimingCorrection	0x02
Default Session Configuration	0x13	TokenAssignment	0x03
Default Session Configuration	0x13	TokenComplete	0x04
Default Session Configuration	0x13	TokensSupportedRequest	0x00
Default Session Configuration	0x13	TokensSupportedResponse	0x01
Default Session Configuration	0x13	TokenUpdateRequest	0x02
Default Connected State	0x0a	TuneAwayRequest	0x05
Default Connected State	0x0a	TuneAwayResponse	0x06
Default Address Management	0x12	UATIAssignment	0x01
Default Address Management	0x12	UATIComplete	0x02
Default Address Management	0x12	UATIUpdateRequest	0x00
Default Session Configuration	0x13	UnlockConfiguration	0x07
Default Session Configuration	0x13	UnlockConfigurationAck	0x08
Default Active Set Management	0x0b	VCQIRreportMIMO	0x02
Default Active Set Management	0x0b	VCQIRreportSISO	0x01

## 11.4 Other RAT Types

A RAT Type ID identifies the type of the radio access technology. Table 107 shows the RAT types defined in this specification.

**Table 107 RAT Types and IDs**

RAT Type ID	RAT Type	RAT ID
0x00	The technology specified in this document	UATI as specified in 2.3
0x01	L2TP	N/A
0x02	802.11	48-bit extended unique identifier (EUI-48)
0x03	CDMA2000 1x	IMSI <sup>90</sup>
0x04	CDMA2000 1xEV	UATI <sup>91</sup>
0x05	GSM	IMSI <sup>92</sup>
0x06	WCDMA	IMSI <sup>92</sup>
All other values	Reserved	N/A

## 11.5 Session Configuration Tokens

### 11.5.1 SessionConfigurationToken 0x0000

Table 108 shows the Protocol Types and Subtypes for SessionConfigurationToken 0x0000. All attributes for SessionConfigurationToken 0x0000 shall be set to the default values defined by the protocol or transport Subtype.

**Table 108 Protocol types and subtypes**

Protocol Type		Protocol Subtype	
Name	ID	Name	ID
Physical Layer	0x00	Default Physical Layer	0x0000
Control Channel MAC	0x01	Default Control Channel MAC	0x0000
Access Channel MAC	0x02	Default Access Channel MAC	0x0000
Forward Traffic Channel MAC	0x03	Default Forward Traffic Channel MAC	0x0000
Reverse Traffic Channel MAC	0x04	Default Reverse Traffic Channel MAC	0x0000
Reverse Control Channel MAC	0x05	Default Reverse Control Channel MAC	0x0000
Shared Signaling MAC	0x06	Default Shared Signaling MAC	0x0000
Air Link Management	0x07	Default Air Link Management	0x0000

<sup>90</sup> 3GPP2 C.S0005-D v1.0, Upper Layer (Layer 3) Signaling Specification for cdma2000<sup>®</sup> Spread Spectrum Systems.

<sup>91</sup> 3GPP2 C.S0024-A v2.0, High Rate Packet Data Air Interface Specification.

<sup>92</sup> 3GPP TS 23.003: Numbering, addressing and identification.

Protocol Type		Protocol Subtype	
Name	ID	Name	ID
Initialization State	0x08	Default Initialization State	0x0000
Idle State	0x09	Default Idle State	0x0000
Connected State	0x0a	Default Connected State	0x0000
Active Set Management	0x0b	Default Active Set Management	0x0000
Overhead Messages	0x0c	Overhead Messages	0x0000
Authentication	0x0d	Default Authentication	0x0000
Encryption	0x0e	Default Encryption	0x0000
Security	0x0f	Default Security	0x0000
Key Exchange	0x10	Default Key Exchange	0x0000
Session Management	0x11	Default Session Management	0x0000
Address Management	0x12	Default Address Management	0x0000
Session Configuration	0x13	Default Session Configuration	0x0000
Capabilities Discovery	0x14	Default Capabilities Discovery	0x0000
Inter RAT Protocol	0x15	Default Inter RAT Protocol	0x0000
Packet Consolidation	0x16	Default Packet Consolidation	0x0000
Transport 0	0x17	Default Signaling Transport	0x0000
Transport 1	0x18	Default Data Transport	0x0001
Transport 2	0x19	Transport not used	0xffff
Transport 3	0x1a	Transport not used	0xffff
Transport 4	0x1b	Transport not used	0xffff
Transport 5	0x1c	Transport not used	0xffff
Transport 6	0x1d	Transport not used	0xffff
Transport 7	0x1e	Transport not used	0xffff

1

## 2 11.5.2 SessionConfigurationToken 0x0001

3 Table 109 shows the Protocol Types and Subtypes for SessionConfigurationToken 0x0001. All  
4 attributes for SessionConfigurationToken 0x0001 shall be set to the default values defined by the  
5 protocol or transport Subtype except for the attributes defined in Table 110.

6

**Table 109 Protocol types and subtypes**

Protocol Type		Protocol Subtype	
Name	ID	Name	ID
Physical Layer	0x00	Default Physical Layer	0x0000
Control Channel MAC	0x01	Default Control Channel MAC	0x0000
Access Channel MAC	0x02	Default Access Channel MAC	0x0000
Forward Traffic Channel MAC	0x03	Default Forward Traffic Channel MAC	0x0000

Protocol Type		Protocol Subtype	
Name	ID	Name	ID
Reverse Traffic Channel MAC	0x04	Default Reverse Traffic Channel MAC	0x0000
Reverse Control Channel MAC	0x05	Default Reverse Control Channel MAC	0x0000
Shared Signaling MAC	0x06	Default Shared Signaling MAC	0x0000
Air Link Management	0x07	Default Air Link Management	0x0000
Initialization State	0x08	Default Initialization State	0x0000
Idle State	0x09	Default Idle State	0x0000
Connected State	0x0a	Default Connected State	0x0000
Active Set Management	0x0b	Default Active Set Management	0x0000
Overhead Messages	0x0c	Overhead Messages	0x0000
Authentication	0x0d	Default Authentication	0x0000
Encryption	0x0e	Generic Encryption	0x0001
Security	0x0f	Default Security	0x0000
Key Exchange	0x10	Default Key Exchange	0x0000
Session Management	0x11	Default Session Management	0x0000
Address Management	0x12	Default Address Management	0x0000
Session Configuration	0x13	Default Session Configuration	0x0000
Capabilities Discovery	0x14	Default Capabilities Discovery	0x0000
Inter RAT Protocol	0x15	Default Inter RAT Protocol	0x0000
Packet Consolidation	0x16	Default Packet Consolidation	0x0000
Transport 0	0x17	Default Signaling Transport	0x0000
Transport 1	0x18	Default Data Transport	0x0001
Transport 2	0x19	Transport not used	0xffff
Transport 3	0x1a	Transport not used	0xffff
Transport 4	0x1b	Transport not used	0xffff
Transport 5	0x1c	Transport not used	0xffff
Transport 6	0x1d	Transport not used	0xffff
Transport 7	0x1e	Transport not used	0xffff

1 For this SessionConfigurationToken, the following attributes shall be set to the specified values:

2 **Table 110 Configuration attributes that shall be set to non-default values**

Protocol Subtype		Configuration Attribute		
Name	ID	Name	ID	Value
Default Security	0x0000	SecurityEnabled	0x00	0x01

3

### 11.5.3 SessionConfigurationToken 0x0002

Table 111 shows the Protocol Types and Subtypes for SessionConfigurationToken 0x0002. All attributes for SessionConfigurationToken 0x0002 shall be set to the default values defined by the protocol or transport Subtype except for the attributes defined in Table 112.

**Table 111 Protocol types and subtypes**

Protocol Type		Protocol Subtype	
Name	ID	Name	ID
Physical Layer	0x00	Default Physical Layer	0x0000
Control Channel MAC	0x01	Default Control Channel MAC	0x0000
Access Channel MAC	0x02	Default Access Channel MAC	0x0000
Forward Traffic Channel MAC	0x03	Default Forward Traffic Channel MAC	0x0000
Reverse Traffic Channel MAC	0x04	Default Reverse Traffic Channel MAC	0x0000
Reverse Control Channel MAC	0x05	Default Reverse Control Channel MAC	0x0000
Shared Signaling MAC	0x06	Default Shared Signaling MAC	0x0000
Air Link Management	0x07	Default Air Link Management	0x0000
Initialization State	0x08	Default Initialization State	0x0000
Idle State	0x09	Default Idle State	0x0000
Connected State	0x0a	Default Connected State	0x0000
Active Set Management	0x0b	Default Active Set Management	0x0000
Overhead Messages	0x0c	Overhead Messages	0x0000
Authentication	0x0d	Default Authentication	0x0000
Encryption	0x0e	Generic Encryption	0x0001
Security	0x0f	Default Security	0x0000
Key Exchange	0x10	Default Key Exchange	0x0000
Session Management	0x11	Default Session Management	0x0000
Address Management	0x12	Default Address Management	0x0000
Session Configuration	0x13	Default Session Configuration	0x0000
Capabilities Discovery	0x14	Default Capabilities Discovery	0x0000
Inter RAT Protocol	0x15	Default Inter RAT Protocol	0x0000
Packet Consolidation	0x16	Default Packet Consolidation	0x0000
Transport 0	0x17	Default Signaling Transport	0x0000
Transport 1	0x18	Default Data Transport	0x0001
Transport 2	0x19	Transport not used	0xffff
Transport 3	0x1a	Transport not used	0xffff
Transport 4	0x1b	Transport not used	0xffff
Transport 5	0x1c	Transport not used	0xffff
Transport 6	0x1d	Transport not used	0xffff
Transport 7	0x1e	Transport not used	0xffff



1 For this SessionConfigurationToken, the following attributes shall be set to the specified values:

2 **Table 112 Configuration attributes that shall be set to non-default values**

Protocol Subtype		Configuration Attribute		
Name	ID	Name	ID	Value
Default Authentication	0x0000	AuthenticationMode	0x00	0x01
Default Security	0x0000	SecurityEnabled	0x00	0x01

### 3 11.6 Flow profile identifier assignments

4 The FlowProfileID identifies the service needs for an application flow. In order to utilize the  
5 FlowProfileID as a short hand method for specifying all relevant air interface parameters necessary to  
6 support a particular multimedia packet data service, one must explicitly define the service for which  
7 flow requirements are being indicated by the identifier.  
8

9 Each flow profile is identified by a unique FlowProfileID to facilitate proper processing within the  
10 access network and access terminals. FlowProfileID may be used for standard services, as well as for  
11 proprietary (non-standard) services. The FlowProfileID format is shown in Table 113.

12 **Table 113 FlowProfileID format**

Field	Length (bits)
FlowProfileIDType	4
FlowProfileIDLength	4
FlowProfileNum	FlowProfileIDLength × 8

13 FlowProfileIDType This field shall be set to an identifier for the FlowProfileIDType according to  
14 Table 15.

15 **Table 114 FlowProfileIDType values**

Value	Description
0x0	Standard FlowProfileNum according to 11.6.1
0x1	cdma2000 FlowProfileNum according to [24]
0x2	Proprietary FlowProfileNum
All other values	Reserved

16 FlowProfileIDLength This field shall be set to the length of the FlowProfileID field in units of  
17 octets.

18 FlowProfileNum If FlowProfileIDType is 0x0, then this field shall be set as defined in 11.6.1.  
19 If FlowProfileIDType is 0x1, then this field shall be set as defined in [24]. If  
20 FlowProfileIDType is 0x2, the setting of this field is beyond the scope of this  
21 specification.

## 11.6.1 Flow Profile Identifier Assignments

The FlowProfileIDType of standard FlowProfileNum is set to 0x0. The characteristics associated with each FlowProfileNum are beyond the scope of this specification.

### 11.6.1.1 Generic data service flow profile identifier assignments

Table 9 shows the Generic data FlowProfileNum assignments defined for this specification.

**Table 115 Generic data service profile identifier assignments**

FlowProfileNum (Decimal)	FlowProfileNum (Hexadecimal)	Flow Description
0	0x0000	Best effort
1	0x0001	Streaming 32 kbps
2	0x0002	Streaming 64 kbps
3	0x0003	Streaming 96 kbps
4	0x0004	Streaming 128 kbps
5	0x0005	Minimum Acceptable User Data Rate of 32kbps, max. latency <sup>93</sup> is 100msec, 1% avg. data loss rate <sup>94</sup> .
6	0x0006	Minimum Acceptable User Data Rate of 64kbps, max. latency is 100msec, 1% avg. data loss rate.
7	0x0007	Minimum Acceptable User Data Rate of 128kbps, max. latency is 100msec, 1% avg. data loss rate.
8	0x0008	Minimum Acceptable User Data Rate of 256kbps, max. latency is 100msec, 1% avg. data loss rate.
9	0x0009	Minimum Acceptable User Data Rate of 512kbps, max. latency is 100msec, 1% avg. data loss rate.
10	0x000a	Minimum Acceptable User Data Rate of 1024kbps, max. latency is 100msec, 1% avg. data loss rate.
11	0x000b	Minimum Acceptable User Data Rate of 2048kbps, max. latency is 100msec, 1% avg. data loss rate.
12	0x000c	Minimum Acceptable User Data Rate of 4096kbps, max. latency is 100msec, 1% avg. data loss rate.
13	0x000d	Minimum Acceptable User Data Rate of 32kbps, max. latency is 100msec, 0.01% avg. data loss rate.
14	0x000e	Minimum Acceptable User Data Rate of 64kbps, max. latency is 100msec, 0.01% avg. data loss rate.
15	0x000f	Minimum Acceptable User Data Rate of 128kbps, max. latency is 100msec, 0.01% avg. data loss rate.

<sup>93</sup> Maximum latency is defined to be the maximum amount of time allowed from the time that an octet of user data is submitted to the transmitting RLP until the receiving RLP either delivers the octet or aborts its delivery.

<sup>94</sup> Data loss rate is defined as the ratio of the number of lost data octets to the number of transmitted data octets, measured above RLP.

FlowProfileNum (Decimal)	FlowProfileNum (Hexadecimal)	Flow Description
16	0x0010	Minimum Acceptable User Data Rate of 256kbps, max. latency is 100msec, 0.01% avg. data loss rate.
17	0x0011	Minimum Acceptable User Data Rate of 512kbps, max. latency is 100msec, 0.01% avg. data loss rate.
18	0x0012	Minimum Acceptable User Data Rate of 1024kbps, max. latency is 100msec, 0.01% avg. data loss rate.
19	0x0013	Minimum Acceptable User Data Rate of 2048kbps, max. latency is 100msec, 0.01% avg. data loss rate.
20	0x0014	Minimum Acceptable User Data Rate of 4096kbps, max. latency is 100msec, 0.01% avg. data loss rate.
21 – 255	0x0015 – 0x00ff	Reserved

1

### 2 11.6.1.2 Speech service flow profile identifier assignments

3 Table 116 shows the Speech Service FlowProfileNum assignments defined for this specification.

4

**Table 116 Speech service profile identifier assignments**

FlowProfileNum (Decimal)	FlowProfileNum (Hexadecimal)	Flow Description
256	0x0100	G.711 at 64 kbps <sup>95</sup>
257	0x0101	G.722.1 at 24 kbps <sup>96</sup>
258	0x0102	G.722.1 at 32 kbps <sup>96</sup>
259	0x0103	G.723.1 at 5.3 kbps <sup>97</sup>
260	0x0104	G.723.1 at 6.3 kbps <sup>97</sup>
261	0x0105	G.726 at 16 kbps <sup>98</sup>
262	0x0106	G.726 at 24 kbps <sup>98</sup>
263	0x0107	G.726 at 32 kbps <sup>98</sup>
264	0x0108	G.726 at 40 kbps <sup>98</sup>
265	0x0109	G.728 at 16 kbps <sup>99</sup>

<sup>95</sup> ITU-T Recommendation G.711: "Pulse code modulation (PCM) of voice frequencies".

<sup>96</sup> ITU-T Recommendation G.722.1: "Low-complexity coding at 24 and 32 kbit/s for hands-free operation in systems with low frame loss."

<sup>97</sup> ITU-T Recommendation G.723.1: "Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s."

<sup>98</sup> ITU-T Recommendation G.726: "40, 32, 24, 16 kbit/s adaptive differential pulse code modulation (ADPCM)."

<sup>99</sup> ITU-T Recommendation G.728: "Coding of speech at 16 kbit/s using low-delay code excited linear prediction."

FlowProfileNum (Decimal)	FlowProfileNum (Hexadecimal)	Flow Description
266	0x010a	G.729 at 8 kbps <sup>100</sup>
267	0x010b	AMR at 12.2 kbps (Also known as GSM-EFR which is the TIA TDMA-US1 Enhanced speech codec) <sup>101</sup>
268	0x010c	AMR at 10.2 kbps <sup>101</sup>
269	0x010d	AMR at 7.95 kbps <sup>101</sup>
270	0x010e	AMR at 7.40 kbps (Also known as TIA/EIA IS-641 TDMA Enhanced Full Rate Speech Codec) <sup>101</sup>
271	0x010f	AMR at 6.70 kbps (Also known as PDC-EFR which is the ARIB 6.7 kbps Enhanced Full Rate Speech Codec) <sup>101</sup>
272	0x0110	AMR at 5.90 kbps <sup>101</sup>
273	0x0111	AMR at 5.15 kbps <sup>101</sup>
274	0x0112	AMR at 4.75 kbps <sup>101</sup>
275	0x0113	AMR at 1.80 kbps (SID) <sup>101</sup>
276	0x0114	EVRC <sup>102</sup>
277	0x0115	SMV <sup>103</sup>
278 – 511	0x116 – 0x01ff	Reserved

1

2 **11.6.1.3 Audio service flow profile identifier assignments**

3 The audio media service is specified using the “Streaming” (high latency, low error rate) traffic class.  
4 Table 117 shows the Audio Service FlowProfileNum assignments defined for this specification.

5

**Table 117 Audio service profile identifier assignments**

FlowProfileNum (Decimal)	FlowProfileNum (Hexadecimal)	Flow Description
512	0x0200	Streaming Audio 16k
513	0x0201	Streaming Audio 24k
514	0x0202	Streaming Audio 32k
515	0x0203	Streaming Audio 48k
516	0x0204	Streaming Audio 64k

<sup>100</sup> ITU-T Recommendation G.729: “Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear-prediction (CS-ACELP).”

<sup>101</sup> 3GPP TS 26.071: "Adaptive Multi-Rate (AMR) Speech Codec; General Description."

<sup>102</sup> 3GPP2 C.S0014-A v1.0, Enhanced Variable Rate Codec, Speech Service Option 3 for Wideband Spread Spectrum Digital Systems.

<sup>103</sup> 3GPP2 C.S0030-0 v2.0, Selectable Mode Vocoder Service Option 56 for Wideband Spread Spectrum Communication Systems.

FlowProfileNum (Decimal)	FlowProfileNum (Hexadecimal)	Flow Description
517	0x0205	Streaming Audio 128k
518	0x0206	Streaming Audio 256k
519	0x0207	Streaming Audio 512k
520	0x0208	Streaming Audio 1024k
521 – 767	0x0209 – 0x02ff	Reserved

1

#### 2 11.6.1.4 Video service flow profile identifier assignments

3 Video media services are specified using the “Conversational” (low latency, low error rate) and  
 4 “Streaming” (high latency, low error rate) traffic classes. Table 118 shows the Video Service  
 5 FlowProfileNum assignments defined for this specification.

6

**Table 118 Video service profile identifier assignments**

FlowProfileNum (Decimal)	FlowProfileNum (Hexadecimal)	Flow Description
768	0x0300	Conversational Interactive Video 24k
769	0x0301	Conversational Interactive Video 32k
770	0x0302	Conversational Interactive Video 40k
771	0x0303	Conversational Interactive Video 48k
772	0x0304	Conversational Interactive Video 56k
773	0x0305	Conversational Interactive Video 64k
774	0x0306	Conversational Interactive Video 128k
775	0x0307	Conversational Interactive Video 256k
776	0x0308	Conversational PtT Video 24k
777	0x0309	Conversational PtT Video 32k
778	0x030a	Conversational PtT Video 40k
779	0x030b	Conversational PtT Video 48k
780	0x030c	Conversational PtT Video 56k
781	0x030d	Conversational PtT Video 64k
782	0x030e	Conversational PtT Video 128k
783	0x030f	Conversational PtT Video 256k
784	0x0310	Streaming Video 24k
785	0x0311	Streaming Video 48k
786	0x0312	Streaming Video 64k
787	0x0313	Streaming Video 96k
788	0x0314	Streaming Video 120k
789	0x0315	Streaming Video 128k
790	0x0316	Streaming Video 256k

FlowProfileNum (Decimal)	FlowProfileNum (Hexadecimal)	Flow Description
791	0x0317	Streaming Video 512k
792	0x0318	Streaming Video 1024k
793	0x0319	Streaming Video 2000k
794	0x031a	Streaming Video 4000k
795	0x031b	Streaming Video 8000k
796	0x031c	Streaming Video 16000k
797	0x031d	Streaming Video 20000k
798 – 1023	0x031e – 0x03ff	Reserved

1

### 11.6.1.5 Text service flow profile identifier assignments

2

3 The text media service is specified using the “Streaming” (high latency, low error rate) traffic class.  
4 Table 119 shows the Text Service FlowProfileNum assignments defined for this specification.

4

5 **Table 119 Text service profile identifier assignments**

FlowProfileNum (Decimal)	FlowProfileNum (Hexadecimal)	Flow Description
1024	0x0400	Streaming Text
1025 – 1279	0x0400 – 0x04ff	Reserved

6

### 11.6.1.6 Signaling service flow profile identifier assignments

7

8 The signaling media service is specified using the “Conversational” (low latency, low error rate),  
9 “Streaming” (high latency, low error rate), and “Interactive” (low latency, medium error rate) traffic  
10 classes. Table 120 shows the Signaling Service FlowProfileNum assignments defined for this  
11 specification.

11

12 **Table 120 Signaling service profile identifier assignments**

FlowProfileNum (Decimal)	FlowProfileNum (Hexadecimal)	Flow Description
1280	0x0500	Conversational Media Control Signaling
1281	0x0501	Streaming Media Control Signaling
1282	0x0502	Interactive Media Control Signaling
1283 – 1535	0x0503 – 0x05ff	Reserved

13

### 11.6.1.7 Gaming Service Flow Profile Identifier Assignments

The gaming media service is specified using the “Interactive” (low latency, medium error rate) traffic class. Table 121 shows the Gaming Service FlowProfileNum assignments defined for this specification.

**Table 121 Gaming service profile identifier assignments**

<b>FlowProfileNum (Decimal)</b>	<b>FlowProfileNum (Hexadecimal)</b>	<b>Flow Description</b>
1536	0x0600	Interactive Gaming
1537 – 1791	0x0601 – 0x06ff	Reserved

6

## 12 Precoding and SDMA Codebooks

Precoding and SDMA are defined in Chapter 9 as a mapping between effective antennas and tile antennas. A particular mapping is defined by a precoding matrix. The columns of the precoding matrix define a set of spatial beams that can be used by AN. The AN uses only one column, of the precoding matrix in SISO transmission, and multiple columns in STTD or MIMO transmissions.

The AT may choose to feedback a preferred precoding matrix to be used by AN for future transmissions. The set of such precoding matrices forms a codebook. In this chapter a number of precoding/SDMA codebooks are listed and the corresponding values of BFCHBeamCodeBookIndex are defined in the Overhead Messages protocol. Some of the precoding matrices in a codebook are grouped into clusters. Matrices in a single cluster typically span only part of the space. If the AT feeds back a beam index within a cluster, the AN treats this as an indication that it may schedule other ATs on different clusters.

The BeamIndex field, in the BFCHBeamIndex report, indexes a beam in a codebook specified by BFCHBeamCodeBookIndex. The BeamIndex may indicate one or more of the following: A no preferred precoding or SDMA matrix, a preferred SISO precoding or SDMA transmission on a spatial beam, a preferred STTD precoding or SDMA transmission on two spatial beams, and a preferred MIMO precoding or SDMA transmission on a set of spatial beams (more than one column of the precoding matrix).

It shall be understood that, if the codebook supports MIMO, only the 0<sup>th</sup> to (SpatialOrder-1)<sup>th</sup> columns of the precoding matrix are to be used, where SpatialOrder is defined in the Forward Traffic Channel MAC protocol. If a spatial beam  $\mathbf{w} = [w_0 \quad w_1 \quad \cdots \quad w_{N_{EFT\_TX\_ANT}-1}]^T$ , is used to transmit a modulation symbol  $s$ , then  $w_j s$  is transmitted on effective antenna  $j$ , where  $N_{EFT\_TX\_ANT}$  is equal to the parameter EffectiveNumAntenna maintained in the overhead messages protocol, and T is the matrix transpose.

In the sequel  $i$  indicates the imaginary part in a complex number.

### 12.1 BFCHBeamCodeBookIndex = 0000

This codebook is only valid for SISO transmission and  $N_{EFT\_TX\_ANT}=4$ .

BeamIndex=0: The AT does not prefer a specific precoding matrix and rather prefers a random matrix that is chosen by AN. The random matrix may change at a rate chosen by AN.

#### 12.1.1 Cluster 1 :

BeamIndex=1:  $[0.5 \ 0.5i \ -0.5 \ -0.5i]^T$

#### 12.1.2 Cluster 2:

BeamIndex=2:  $[0.5 \ -0.5i \ -0.5 \ 0.5i]^T$



## 12.2 BFCHBeamCodeBookIndex = 0001

This codebook is valid for SISO and MIMO transmissions, and only if  $N_{\text{EFT\_TX\_ANT}}=4$ .

Let **BeamMat**=

$$\begin{bmatrix} 0.5000 & 0.5000 & 0.5000 & 0.5000 \\ 0.3536 + 0.3536i & -0.3536 + 0.3536i & 0.3536 - 0.3536i & -0.3536 - 0.3536i \\ 0.0000 + 0.5000i & -0.0000 - 0.5000i & 0.0000 - 0.5000i & -0.0000 + 0.5000i \\ -0.3536 + 0.3536i & 0.3536 + 0.3536i & -0.3536 - 0.3536i & 0.3536 - 0.3536i \end{bmatrix}$$

The codebook is defined as follows:

BeamIndex=0: The AT does not prefer a specific precoding matrix and rather prefers a random matrix that is chosen by AN. The random matrix may change at a rate chosen by AN.

BeamIndex=1: Zero<sup>th</sup> column in **BeamMat**

BeamIndex=2: First column in **BeamMat**

BeamIndex=3: Second column in **BeamMat**

BeamIndex=4: Third column in **BeamMat**

### 12.2.1 Precoding Matrices

BeamIndex=5 to BeamIndex=35: If the BeamIndex is equal to  $j$ , define  $\text{seed}_{k,j} = (2\pi [\text{BIT\_REVERSE}([(4*j+k)*2654435761] \bmod 2^{32})] \bmod 2^{20}) / 2^{20}$ ,  $k=0,1,2,3$ . The corresponding precoding matrix,  $\mathbf{U}_j$ , is a random unitary matrix defined as  $\mathbf{U}_j = \mathbf{\Lambda}_j \mathbf{D}$ , where  $\mathbf{\Lambda}_j$  is

a diagonal matrix on the form  $\begin{bmatrix} e^{i\phi_0} & 0 & 0 & 0 \\ 0 & e^{i\phi_1} & 0 & 0 \\ 0 & 0 & e^{i\phi_2} & 0 \\ 0 & 0 & 0 & e^{i\phi_3} \end{bmatrix}$ ,  $\phi_k = \text{seed}_{k,j}$  is a uniform random variable

between 0 and  $2\pi$ , and  $\mathbf{D}$  is the 4x4 DFT matrix, i.e.,

$$\mathbf{D} = \{D_{m,n}, m, n = 0, \dots, 3\}, D_{m,n} = \frac{1}{\sqrt{4}} e^{j2\pi mn/4}.$$

### 12.2.2 Cluster 1:

BeamIndex=36 to BeamIndex=49: If the BeamIndex is equal to  $j$ , define  $\text{seed}_{k,j} = (2\pi [\text{BIT\_REVERSE}([(2*j+k)*2654435761] \bmod 2^{32})] \bmod 2^{20}) / 2^{20}$ ,  $k=0,1$ . The corresponding precoding matrix is defined as  $\mathbf{BeamMat}(:,0:1)*\mathbf{U}_j$ , where  $\mathbf{BeamMat}(:,0:1)$  is the zero<sup>th</sup> and first columns of  $\mathbf{BeamMat}$ ,  $\mathbf{U}_j = \mathbf{\Lambda}_j \mathbf{D}$ ,  $\mathbf{\Lambda}_j = \begin{bmatrix} e^{j\phi_0} & \mathbf{0} \\ \mathbf{0} & e^{j\phi_1} \end{bmatrix}$ ,  $\phi_k = \text{seed}_{k,j}$  is a uniform random variable between 0 and  $2\pi$ , and  $\mathbf{D}$  is the 2x2 DFT matrix, i.e.,

$$\mathbf{D} = \{D_{m,n}, m, n = 0, 1\}, D_{m,n} = \frac{1}{\sqrt{2}} e^{j2\pi mn/2}.$$

### 12.2.3 Cluster 2:

BeamIndex=50 to BeamIndex=63: If the BeamIndex is equal to  $j$ , define  $\text{seed}_{k,j} = (2\pi [\text{BIT\_REVERSE}([(2*j+k)*2654435761] \bmod 2^{32})] \bmod 2^{20}) / 2^{20}$ ,  $k=0,1$ . The corresponding precoding matrix is defined as  $\mathbf{BeamMat}(:,2:3)*\mathbf{U}_j$ , where  $\mathbf{BeamMat}(:,2:3)$  is the second and third columns of  $\mathbf{BeamMat}$ ,  $\mathbf{U}_j = \mathbf{\Lambda}_j \mathbf{D}$ ,  $\mathbf{\Lambda}_j = \begin{bmatrix} e^{j\phi_0} & \mathbf{0} \\ \mathbf{0} & e^{j\phi_1} \end{bmatrix}$ ,  $\phi_k = \text{seed}_{k,j}$  is a uniform random variable between 0 and  $2\pi$ , and  $\mathbf{D}$  is the 2x2 DFT matrix, i.e.,

$$\mathbf{D} = \{D_{m,n}, m, n = 0, 1\}, D_{m,n} = \frac{1}{\sqrt{2}} e^{j2\pi mn/2}.$$

15

## 1 13 MAC and PHY MIB

### 2 13.1 Overview

3 This chapter defines a Management Information Base (MIB) module for managing the MAC and  
 4 PHY. Managed objects are accessed via a virtual information store, termed the Management  
 5 Information Base or MIB. MIB objects are generally accessed through the Simple Network  
 6 Management Protocol (SNMP). The objects in this MIB are defined using the mechanisms specified  
 7 in the Structure of Management Information (SMI). The MIB module specified is compliant to  
 8 SMIV2 which is described in RFC 2578 [18], RFC 2579 [19], and RFC 2580 [20].

### 9 13.2 MIB Structure

10 The MIB structure is based on the architecture reference model in Figure 1 and the layering  
 11 architecture for the air interface in Figure 2 describes the layering architecture for the air interface.  
 12 The MIB object is composed of two groups:

- 13 ■ dot20An: This group contains managed objects defined for the access network.
- 14 ■ dot20Cmn: This group contains managed objects defined for the access network and the  
 15 access terminal.

### 16 13.3 Definition

```

17
18 IEEE802dot20-MIB DEFINITIONS ::= BEGIN
19
20 IMPORTS
21     ifIndex
22     FROM IF-MIB
23     MODULE-COMPLIANCE, OBJECT-GROUP
24     FROM SNMPv2-CONF
25     Counter32, Counter64, Integer32, MODULE-IDENTITY, OBJECT-IDENTITY,
26     OBJECT-TYPE, transmission
27     FROM SNMPv2-SMI
28     RowPointer, RowStatus, TEXTUAL-CONVENTION, TruthValue
29     FROM SNMPv2-TC
30     ;
31
32 ieee802dot20 MODULE-IDENTITY
33     LAST-UPDATED "200512301948Z" -- December 30, 2005
34     ORGANIZATION
35         "IEEE 802.20"
36     CONTACT-INFO
37         "Contact: J. Tomcik
38         Postal: 5775 Morehouse Dr
39         San Diego, CA, 92121, USA
40         Tel: 858-658-3231
41         Fax: 858-658-3231
42         E-mail: jtomcik@qualcomm.com"
43     DESCRIPTION
44         "The MIB module for IEEE 802.20 entities.
45         (The transmission oid used for this MIB needs to be updated
46         when a valid one is obtained from IANA along with the new
47         802.20 ifType)"
48     ::= { transmission 9999 }
```

```

1
2 Dot20AnCarrierConfigEntry ::= SEQUENCE
3 {
4     dot20AnCarrierID                Integer32,
5     dot20AnFLReservedInterlaces     INTEGER,
6     dot20AnNumFLReservedSubbands    Integer32,
7     dot20AnFLFirstRestrSetSubband   Integer32,
8     dot20AnFLNumRestrSetSubbands     Integer32,
9     dot20AnFLChannelTreeIndex       Integer32,
10    dot20AnFLSectorHopSeed           Integer32,
11    dot20AnFLIntraCellCommonHopping TruthValue,
12    dot20AnFLDPISectorOffset         Integer32,
13    dot20AnFLDPISectorScramble       Integer32,
14    dot20AnFLNumSDMADimensions       Integer32,
15    dot20AnFLNumSubbands              Integer32,
16    dot20AnFLDiversityHoppingMode    TruthValue,
17    dot20AnNumPilots                  Integer32,
18    dot20AnEffectiveNumAntennas       Integer32,
19    dot20AnNumCmnPilotTxAntennas     Integer32,
20    dot20AnEnableCmnPilotStaggering  TruthValue,
21    dot20AnEnableAuxPilotStaggering  TruthValue,
22    dot20AnSSCHNumHopports           Integer32,
23    dot20AnSSCHNumBlocks              Integer32,
24    dot20AnSSCHModSymbolsPerBlock    Integer32,
25    dot20AnPilotPN                    Integer32,
26    dot20AnRLChannelTreeIndex        Integer32,
27    dot20AnRLSectorHopSeed           Integer32,
28    dot20AnRLIntraCellCommonHopping TruthValue,
29    dot20AnBFCHBeamCodeBookIndex     Integer32,
30    dot20AnRPICHEEnabled              TruthValue,
31    dot20AnRLDPISectorOffset         Integer32,
32    dot20AnRLDPISectorScramble       Integer32,
33    dot20AnRLNumSDMADimensions       Integer32,
34    dot20AnRLRestrictedSetBitmap      Integer32,
35    dot20AnRLNumSubbands              Integer32,
36    dot20AnRLDiversityHoppingMode    TruthValue,
37    dot20AnNumRLControlSubbands       Integer32,
38    dot20AnRACKBandwidthFactor        Integer32,
39    dot20AnHalfDuplexModeSupported    TruthValue,
40    dot20AnRevLinkSilenceDuration     Integer32,
41    dot20AnRevLinkSilencePeriod       Integer32,
42    dot20AnTransmitPower              Integer32,
43    dot20AnCommonPilotPower           Integer32,
44    dot20AnAuxPilotPower              Integer32,
45    dot20AnPreamblePilotPower         Integer32,
46    dot20AnMACIDRange                 INTEGER,
47    dot20AnFLPCReportInterval         Integer32,
48    dot20AnRLCtrlPCMode               INTEGER,
49    dot20AnCtrlAccessOffset           Integer32,
50    dot20AnBFCHPowerOffset            Integer32,
51    dot20AnSFCHPowerOffset            Integer32,
52    dot20AnPICHPowerOffset            Integer32,
53    dot20AnReqChannelGain0            Integer32,
54    dot20AnReqChannelGain1            Integer32,
55    dot20AnReqChannelGain2            Integer32,
56    dot20AnReqChannelGain3            Integer32,
57    dot20AnErasureGain0               Integer32,
58    dot20AnErasureGain1               Integer32,
59    dot20AnErasureGain2               Integer32,
60    dot20AnErasureGain3               Integer32,
61    dot20AnAccessCycleDuration         Integer32,
62    dot20AnAccessSequencePartition    Integer32,
63    dot20AnMaxProbesPerSequence        Integer32,
64    dot20AnProbeRampUpStepSize        Integer32,
65    dot20AnFastOSIEnabled              TruthValue,
66    dot20AnRDCHInitialPacketFormat    Integer32,
67    dot20AnPilotThreshold1            Integer32,
68    dot20AnPilotThreshold2            Integer32,

```

```

1      dot20AnOpenLoopAdjust      Integer32,
2      dot20AnAccessRetryPersistence0 Integer32,
3      dot20AnAccessRetryPersistence1 Integer32,
4      dot20AnAccessRetryPersistence2 Integer32,
5      dot20AnAccessRetryPersistence3 Integer32,
6      dot20AnSynchGroup          Integer32,
7      dot20AnCarrierConfigRowStatus RowStatus
8  }
9
10     Dot20AnChannelBandsEntry ::= SEQUENCE
11     {
12         dot20AnChannelBandIndex      Integer32,
13         dot20AnChannelBandRecordType INTEGER,
14         dot20AnChannelBandClass      Integer32,
15         dot20AnChannelNumber         Integer32,
16         dot20AnChannelFwdLinkCenterFreq Integer32,
17         dot20AnChannelRevLinkCenterFreq Integer32,
18         dot20AnChannelFwdLinkSystemBw Integer32,
19         dot20AnChannelRevLinkSystemBw Integer32,
20         dot20AnChannelBandStatus     RowStatus
21     }
22
23     Dot20AnIdleStateStatsEntry ::= SEQUENCE
24     {
25         dot20AnAccessAttemptCounts   Counter32,
26         dot20AnAccessAttemptFailCounts Counter32,
27         dot20AnPageAttemptCounts     Counter32,
28         dot20AnPageFailureCounts     Counter32
29     }
30
31     Dot20AnNeighborCarriersEntry ::= SEQUENCE
32     {
33         dot20AnNeighborSectorIndex   Integer32,
34         dot20AnNeighborCarrierID     Integer32,
35         dot20AnNeighborSectorID     OCTET STRING,
36         dot20AnNeighborPilotPN      Integer32,
37         dot20AnNeighborTransmitPower Integer32,
38         dot20AnNghbrGloballySynch   TruthValue,
39         dot20AnNghbrSynchGroup      Integer32,
40         dot20AnNeighborCarrierStatus RowStatus
41     }
42
43     Dot20AnNeighborListEntry ::= SEQUENCE
44     {
45         dot20AnNeighborIndex         Integer32,
46         dot20AnNeighborCarrierPointer RowPointer,
47         dot20AnNeighborRowStatus     RowStatus
48     }
49
50     Dot20AnOtherTechNghbrsEntry ::= SEQUENCE
51     {
52         dot20AnOtherTechnologyIndex   Integer32,
53         dot20AnTechnologyType         INTEGER,
54         dot20AnTechNghbrListLength   Integer32,
55         dot20AnTechnologyNeighborList OCTET STRING,
56         dot20AnOtherTechNghbrRowStatus RowStatus
57     }
58
59     Dot20AnSectorConfigEntry ::= SEQUENCE
60     {
61         dot20AnSectorID              OCTET STRING,
62         dot20AnChannelBandRef        Integer32,
63         dot20AnMaximumRevision       Integer32,
64         dot20AnMinimumRevision       Integer32,
65         dot20AnNumCarriers           Integer32,
66         dot20AnCPLength              Integer32,
67         dot20AnPrNumGuardSubcarriers Integer32,
68         dot20AnNumGuardSubcarriers   Integer32,

```

```

1      dot20AnBlockHoppingEnabled      TruthValue,
2      dot20AnNFLBurst                 Integer32,
3      dot20AnNRLBurst                 Integer32,
4      dot20AnCountryCode              Integer32,
5      dot20AnSubnetMask               OCTET STRING,
6      dot20AnLatitude                 Integer32,
7      dot20AnLongitude                Integer32,
8      dot20AnRegistrationRadius       Integer32,
9      dot20AnLeapSeconds              Integer32,
10     dot20AnLocalTimeOffset          Integer32,
11     dot20AnRegistrationZoneIncluded TruthValue,
12     dot20AnRegistrationZoneCode     Integer32,
13     dot20AnRegistrationZoneMaxAge   Integer32,
14     dot20AnSynchronousSystem       TruthValue,
15     dot20AnPhysicalMode             INTEGER,
16     dot20AnSectorConfigRowStatus    RowStatus
17 }
18
19 Dot20AnSectorToIfIndexEntry ::= SEQUENCE
20 {
21     dot20AnIfChannelBandRef Integer32
22 }
23
24 Dot20CmnAuthStatsEntry ::= SEQUENCE
25 {
26     dot20CmnAuthFailureCounts Counter64,
27     dot20CmnAuthSuccessCounts Counter64
28 }
29
30 Dot20CmnDataTransportStatsEntry ::= SEQUENCE
31 {
32     dot20CmnRlpTxBytes           Counter64,
33     dot20CmnRlpReTxBytes        Counter64,
34     dot20CmnRlpTxDropBytes      Counter64,
35     dot20CmnRlpTxStatus         Counter64,
36     dot20CmnRlpRxBytes         Counter64,
37     dot20CmnRlpRxStatus        Counter64,
38     dot20CmnRlpTxPackets        Counter64,
39     dot20CmnRlpReTxPackets      Counter64,
40     dot20CmnRlpTxrDropPackets   Counter64,
41     dot20CmnRlpRxPackets        Counter64,
42     dot20CmnRlpTxNAKTimeouts    Counter64,
43     dot20CmnActiveReservationsCounts Counter64,
44     dot20CmnIdleReservationsCounts Counter64,
45     dot20CmnReservationOpenCounts Counter64,
46     dot20CmnReservationCloseCounts Counter64,
47     dot20CmnReservationFailCounts Counter64
48 }
49
50 Dot20CmnLMACPacketStatsEntry ::= SEQUENCE
51 {
52     dot20CmnPacketFormatIndex Integer32,
53     dot20CmnARQAttemptsIndex Integer32,
54     dot20CmnFwdTxPacketCounts Counter64,
55     dot20CmnRevRxPacketCounts Counter64
56 }
57
58 Dot20CmnLMACStatsEntry ::= SEQUENCE
59 {
60     dot20CmnFLABCounts          Counter64,
61     dot20CmnRLABCounts          Counter64,
62     dot20CmnAccessGrantCounts Counter64
63 }
64
65 Dot20CmnSessionConfigTokenEntry ::= SEQUENCE
66 {
67     dot20CmnSessionTokenIndex Integer32,
68     dot20CmnSessionConfToken Integer32,

```

```

1      dot20CmnSessionConfTokenStatus RowStatus
2  }
3
4  Dot20CmnSigTransportStatsEntry ::= SEQUENCE
5  {
6      dot20CmnSlpTxBytes          Counter64,
7      dot20CmnSlpReTxBytes       Counter64,
8      dot20CmnSlpTxDroppedBytes  Counter64,
9      dot20CmnSlpTxStatus        Counter64,
10     dot20CmnSlpRxBytes         Counter64,
11     dot20CmnSlpRxStatus        Counter64,
12     dot20CmnSlpTxPackets       Counter64,
13     dot20CmnSlpReTxPackets     Counter64,
14     dot20CmnSlpTxDropPackets  Counter64,
15     dot20CmnSlpRxPackets       Counter64,
16     dot20CmnSlpTxACKTimeouts   Counter64
17 }
18
19 dot20An OBJECT-IDENTITY
20     STATUS          current
21     DESCRIPTION
22         "AN specific configuration and statistics."
23     ::= { ieee802dot20 1 }
24
25 dot20AnMac OBJECT-IDENTITY
26     STATUS          current
27     DESCRIPTION
28         "MAC layer objects"
29     ::= { dot20An 1 }
30
31 dot20AnLowerMACControl OBJECT IDENTIFIER ::= { dot20AnMac 3 }
32
33 dot20AnIdleState OBJECT IDENTIFIER ::= { dot20AnLowerMACControl 1 }
34
35 dot20AnIdleStateStatsTable OBJECT-TYPE
36     SYNTAX          SEQUENCE OF Dot20AnIdleStateStatsEntry
37     MAX-ACCESS      not-accessible
38     STATUS          current
39     DESCRIPTION
40         "This table provides one row of Idle State protocol statistics
41         per 802.20 interface (i.e. sector for a specific ChannelBand)
42         and carrier."
43     ::= { dot20AnIdleState 1 }
44
45 dot20AnIdleStateStatsEntry OBJECT-TYPE
46     SYNTAX          Dot20AnIdleStateStatsEntry
47     MAX-ACCESS      not-accessible
48     STATUS          current
49     DESCRIPTION
50         "An Entry (conceptual row) in the IdleStateStats table. This
51         table is indexed by ifIndex and CarrierID. ifIndex: Each IEEE
52         802.20 interface (uniquely identified by SectorID) is
53         represented by an ifEntry. In the case of a multicarrier
54         Sector, the carrierID indentifies one specific carrier."
55     INDEX
56         { ifIndex, dot20AnCarrierID }
57     ::= { dot20AnIdleStateStatsTable 1 }
58
59 dot20AnAccessAttemptCounts OBJECT-TYPE
60     SYNTAX          Counter32
61     MAX-ACCESS      read-only
62     STATUS          current
63     DESCRIPTION
64         "Number of Access Attempts"
65     ::= { dot20AnIdleStateStatsEntry 1 }
66
67 dot20AnAccessAttemptFailCounts OBJECT-TYPE
68     SYNTAX          Counter32

```

```

1     MAX-ACCESS      read-only
2     STATUS          current
3     DESCRIPTION
4         "Number of Failed Access Attempts"
5     ::= { dot20AnIdleStateStatsEntry 2 }
6
7     dot20AnPageAttemptCounts OBJECT-TYPE
8     SYNTAX          Counter32
9     MAX-ACCESS      read-only
10    STATUS          current
11    DESCRIPTION
12        "Number of Page Attempts"
13    ::= { dot20AnIdleStateStatsEntry 3 }
14
15    dot20AnPageFailureCounts OBJECT-TYPE
16    SYNTAX          Counter32
17    MAX-ACCESS      read-only
18    STATUS          current
19    DESCRIPTION
20        "Number of Failed Page Attempts"
21    ::= { dot20AnIdleStateStatsEntry 4 }
22
23    dot20AnOverheadMessages OBJECT IDENTIFIER ::= { dot20AnLowerMACControl 4 }
24
25    dot20AnSectorConfigTable OBJECT-TYPE
26    SYNTAX          SEQUENCE OF Dot20AnSectorConfigEntry
27    MAX-ACCESS      not-accessible
28    STATUS          current
29    DESCRIPTION
30        "This table provides one row per 802.20 interface, i.e. sector
31         for a specific ChannelBand. This table's attributes specify the
32         configuration of the corresponding sector, and can be used to
33         populate fields in SystemInfo block and SectorParameters
34         message."
35    ::= { dot20AnOverheadMessages 1 }
36
37    dot20AnSectorConfigEntry OBJECT-TYPE
38    SYNTAX          Dot20AnSectorConfigEntry
39    MAX-ACCESS      not-accessible
40    STATUS          current
41    DESCRIPTION
42        "An Entry (conceptual row) in the SectorConfig table. This
43         table is indexed by IfIndex. ifIndex: Each IEEE 802.20
44         interface (uniquely identified by SectorID) is represented by
45         an ifEntry."
46    INDEX
47        { ifIndex }
48    ::= { dot20AnSectorConfigTable 1 }
49
50    dot20AnSectorID OBJECT-TYPE
51    SYNTAX          OCTET STRING (SIZE(16))
52    MAX-ACCESS      read-write
53    STATUS          current
54    DESCRIPTION
55        "Sector Address Identifier. The access network shall set the
56         value of the SectorID according to the rules specified in IEEE
57         802.20 AIS. The access terminal shall not assume anything about
58         the format of the SectorID other than that it uniquely
59         identifies the sector."
60    ::= { dot20AnSectorConfigEntry 1 }
61
62    dot20AnChannelBandRef OBJECT-TYPE
63    SYNTAX          Integer32
64    MAX-ACCESS      read-write
65    STATUS          current
66    DESCRIPTION
67        "The reference to the ChannelBand defined in ChannelBands table
68         using this value as index (dot20AnChannelBandIndex)"

```



```

1      ::= { dot20AnSectorConfigEntry 2 }
2
3      dot20AnMaximumRevision OBJECT-TYPE
4          SYNTAX      Integer32 (0..15)
5          MAX-ACCESS  read-write
6          STATUS      current
7          DESCRIPTION
8              "This attribute shall be set to the maximum revision number
9              that the sector can support."
10     ::= { dot20AnSectorConfigEntry 3 }
11
12     dot20AnMinimumRevision OBJECT-TYPE
13         SYNTAX      Integer32 (0..15)
14         MAX-ACCESS  read-write
15         STATUS      current
16         DESCRIPTION
17             "This attribute shall be set to the minimum revision number
18             that the sector can support."
19     ::= { dot20AnSectorConfigEntry 4 }
20
21     dot20AnNumCarriers OBJECT-TYPE
22         SYNTAX      Integer32 (1..4)
23         MAX-ACCESS  read-only
24         STATUS      current
25         DESCRIPTION
26             "This attribute shall determine the number of carriers
27             available at this sector."
28     ::= { dot20AnSectorConfigEntry 5 }
29
30     dot20AnCPLength OBJECT-TYPE
31         SYNTAX      Integer32 (0..3)
32         MAX-ACCESS  read-write
33         STATUS      current
34         DESCRIPTION
35             "This attribute's value noted n shall determine the cyclic
36             prefix length in units of chips. The cyclic prefix length shall
37             take the value  $NFFT \cdot (1+n) / 16$ ."
38     ::= { dot20AnSectorConfigEntry 6 }
39
40     dot20AnPrNumGuardSubcarriers OBJECT-TYPE
41         SYNTAX      Integer32 (0..7)
42         MAX-ACCESS  read-only
43         STATUS      current
44         DESCRIPTION
45             "This attribute shall determine the preamble's number of guard
46             subcarriers as defined in 802.20 Physical layer
47             specification."
48     ::= { dot20AnSectorConfigEntry 7 }
49
50     dot20AnNumGuardSubcarriers OBJECT-TYPE
51         SYNTAX      Integer32 (0..7)
52         MAX-ACCESS  read-only
53         STATUS      current
54         DESCRIPTION
55             "This attribute shall determine the number of guard subcarriers
56             as defined in 802.20 Physical layer specification."
57     ::= { dot20AnSectorConfigEntry 8 }
58
59     dot20AnBlockHoppingEnabled OBJECT-TYPE
60         SYNTAX      TruthValue
61         MAX-ACCESS  read-write
62         STATUS      current
63         DESCRIPTION
64             "This attribute shall be set to true if block hopping is
65             enabled. This attribute shall be set to false if symbol rate
66             hopping is disabled"
67     ::= { dot20AnSectorConfigEntry 9 }
68

```

```

1 dot20AnNFLBurst OBJECT-TYPE
2     SYNTAX      Integer32 (1..4)
3     MAX-ACCESS  read-write
4     STATUS      current
5     DESCRIPTION
6         "This attribute shall determine the number of forward link PHY
7         Frames that comprise a forward link burst in TDD mode."
8     ::= { dot20AnSectorConfigEntry 10 }
9
10 dot20AnNRLBurst OBJECT-TYPE
11     SYNTAX      Integer32 (1..4)
12     MAX-ACCESS  read-write
13     STATUS      current
14     DESCRIPTION
15         "This attribute shall determine the number of reverse link PHY
16         Frames that comprise a reverse link burst in TDD mode."
17     ::= { dot20AnSectorConfigEntry 11 }
18
19 dot20AnCountryCode OBJECT-TYPE
20     SYNTAX      Integer32 (0..999)
21     MAX-ACCESS  read-write
22     STATUS      current
23     DESCRIPTION
24         "This attribute shall be set to the three digit Mobile Country
25         Code associated with this sector (as specified in ITU-T
26         Recommendation E.212, Identification Plan for Land Mobile
27         Stations)."
28     ::= { dot20AnSectorConfigEntry 12 }
29
30 dot20AnSubnetMask OBJECT-TYPE
31     SYNTAX      OCTET STRING (SIZE(16))
32     MAX-ACCESS  read-write
33     STATUS      current
34     DESCRIPTION
35         "Sector Subnet identifier."
36     ::= { dot20AnSectorConfigEntry 13 }
37
38 dot20AnLatitude OBJECT-TYPE
39     SYNTAX      Integer32 (-1296000..1296000)
40     MAX-ACCESS  read-write
41     STATUS      current
42     DESCRIPTION
43         "The latitude of the sector. This attribute shall be set to
44         this sector's latitude in units of 0.25 second, expressed as a
45         two's complement signed number with positive numbers signifying
46         North latitudes. This attribute shall be set to a value in the
47         range 1296000 to 1296000 inclusive (corresponding to a range of
48         -90 to +90)."

```

```

1         "If access terminals are to perform distance based
2         registration, this attribute shall be set to the non-zero
3         distance beyond which the access terminal is to send a new
4         PilotReport message. If access terminals are not to perform
5         distance based registration, This attribute shall be set to
6         0."
7         ::= { dot20AnSectorConfigEntry 16 }
8
9     dot20AnLeapSeconds OBJECT-TYPE
10        SYNTAX      Integer32 (0..255)
11        MAX-ACCESS  read-write
12        STATUS      current
13        DESCRIPTION
14            "The number of leap seconds that have occurred since the start
15            of system time."
16        ::= { dot20AnSectorConfigEntry 17 }
17
18    dot20AnLocalTimeOffset OBJECT-TYPE
19        SYNTAX      Integer32 (0..2047)
20        MAX-ACCESS  read-write
21        STATUS      current
22        DESCRIPTION
23            "This attribute shall be set to the offset of the local time
24            from System Time. This value will be in units of minutes,
25            expressed as a two's complement signed number."
26        ::= { dot20AnSectorConfigEntry 18 }
27
28    dot20AnRegistrationZoneIncluded OBJECT-TYPE
29        SYNTAX      TruthValue
30        MAX-ACCESS  read-write
31        STATUS      current
32        DESCRIPTION
33            "This attribute shall be set to true if the
34            RegistrationZoneCode and RegistrationZoneMaxAge are included."
35        ::= { dot20AnSectorConfigEntry 19 }
36
37    dot20AnRegistrationZoneCode OBJECT-TYPE
38        SYNTAX      Integer32 (0..4095)
39        MAX-ACCESS  read-write
40        STATUS      current
41        DESCRIPTION
42            "The zone code value for this sector."
43        ::= { dot20AnSectorConfigEntry 20 }
44
45    dot20AnRegistrationZoneMaxAge OBJECT-TYPE
46        SYNTAX      Integer32 (0..15)
47        MAX-ACCESS  read-write
48        STATUS      current
49        DESCRIPTION
50            "The Max Age value for this sector."
51        ::= { dot20AnSectorConfigEntry 21 }
52
53    dot20AnSynchronousSystem OBJECT-TYPE
54        SYNTAX      TruthValue
55        MAX-ACCESS  read-write
56        STATUS      current
57        DESCRIPTION
58            "This attribute shall be set to true if all sectors in the
59            deployment are synchronous. This attribute shall be set to
60            false otherwise."
61        ::= { dot20AnSectorConfigEntry 22 }
62
63    dot20AnPhysicalMode OBJECT-TYPE
64        SYNTAX      INTEGER {
65            tdd(1),
66            fdd(2)
67        }
68        MAX-ACCESS  read-create

```

```

1      STATUS          current
2      DESCRIPTION
3      "The Physical mode used for this sector."
4      ::= { dot20AnSectorConfigEntry 23 }
5
6      dot20AnSectorConfigRowStatus OBJECT-TYPE
7      SYNTAX          RowStatus
8      MAX-ACCESS      read-create
9      STATUS          current
10     DESCRIPTION
11     "The status column used for creating, modifying, and deleting
12     instances of the columnar objects in the SectorConfig Table. If
13     the implementor of this MIB has chosen not to implement
14     'dynamic assignment' of sectors, this attribute is not useful
15     and should return noSuchName upon SNMP request."
16     DEFVAL          { active }
17     ::= { dot20AnSectorConfigEntry 24 }
18
19     dot20AnCarrierConfigTable OBJECT-TYPE
20     SYNTAX          SEQUENCE OF Dot20AnCarrierConfigEntry
21     MAX-ACCESS      not-accessible
22     STATUS          current
23     DESCRIPTION
24     "This table provides one row per 802.20 carrier of a sector for
25     a specific ChannelBand. This table's attributes specify the
26     configuration of the corresponding carrier and can be used to
27     populate fields in SystemInfo block, QuickChannelInfo block
28     and ExtendedChannelInfo message."
29     ::= { dot20AnOverheadMessages 2 }
30
31     dot20AnCarrierConfigEntry OBJECT-TYPE
32     SYNTAX          Dot20AnCarrierConfigEntry
33     MAX-ACCESS      not-accessible
34     STATUS          current
35     DESCRIPTION
36     "An Entry (conceptual row) in the AnCarrierConfig table. This
37     table is indexed by ifIndex and CarrierID. ifIndex: Each IEEE
38     802.20 interface (uniquely identified by SectorID) is
39     represented by an ifEntry. In the case of a multicarrier
40     Sector, the carrierID indentifies one specific carrier."
41     INDEX
42     { ifIndex, dot20AnCarrierID }
43     ::= { dot20AnCarrierConfigTable 1 }
44
45     dot20AnCarrierID OBJECT-TYPE
46     SYNTAX          Integer32 (0..3)
47     MAX-ACCESS      read-only
48     STATUS          current
49     DESCRIPTION
50     "This attribute shall be set to the CarrierID of the carrier
51     this block is transmitted on. The CarrierID of a carrier is
52     unique within the corresponding sector."
53     ::= { dot20AnCarrierConfigEntry 1 }
54
55     dot20AnFLReservedInterlaces OBJECT-TYPE
56     SYNTAX          INTEGER {
57         none(1),
58         zero(2),
59         zeroToOne(3),
60         zeroToTwo(4),
61         zeroToThree(5),
62         zeroToFour(6),
63         zeroToFive(7),
64         zeroToSix(8),
65         zeroToSeven(9),
66         zeroToEight(10),
67         zeroToNine(11),
68         zeroToTen(12),

```

```

1         zeroToEleven(13),
2         zeroToTwelve(14),
3         zeroAndThree(15),
4         zeroAndSix(16)
5     }
6     MAX-ACCESS    read-write
7     STATUS        current
8     DESCRIPTION
9         "This attribute shall determine which interlaces contain
10        reserved bandwidth on the forward link."
11    ::= { dot20AnCarrierConfigEntry 2 }
12
13    dot20AnNumFLReservedSubbands OBJECT-TYPE
14        SYNTAX      Integer32 (0..15)
15        MAX-ACCESS  read-write
16        STATUS      current
17        DESCRIPTION
18            "This attribute shall determine the number of subbands
19            allocated to the ReservedFLBandwidth segment. The
20            interpretation of this field is used by the Physical Layer to
21            govern FL PHY Frame Modulation."
22        ::= { dot20AnCarrierConfigEntry 3 }
23
24    dot20AnFLFirstRestrSetSubband OBJECT-TYPE
25        SYNTAX      Integer32 (0..15)
26        MAX-ACCESS  read-write
27        STATUS      current
28        DESCRIPTION
29            "This attribute shall be set to the index of the first
30            restricted subband on the forward link."
31        ::= { dot20AnCarrierConfigEntry 4 }
32
33    dot20AnFLNumRestrSetSubbands OBJECT-TYPE
34        SYNTAX      Integer32 (0..3)
35        MAX-ACCESS  read-write
36        STATUS      current
37        DESCRIPTION
38            "This attribute shall be set to the number of restricted
39            subbands on the forward link. This attribute shall be set to 0
40            if no subbands are restricted. Otherwise, subbands
41            FLFirstRestrictedSetSubband through
42            (FLFirstRestrictedSetSubband+FLNumRestrictedSetSubbands-1) shall
43            be considered to be restricted subbands, with possible rollover
44            at subband zero."
45        ::= { dot20AnCarrierConfigEntry 5 }
46
47    dot20AnFLChannelTreeIndex OBJECT-TYPE
48        SYNTAX      Integer32 (0..15)
49        MAX-ACCESS  read-write
50        STATUS      current
51        DESCRIPTION
52            "FL Channel Tree Index used by the Lower MAC Sublayer for this
53            carrier."
54        ::= { dot20AnCarrierConfigEntry 6 }
55
56    dot20AnFLSectorHopSeed OBJECT-TYPE
57        SYNTAX      Integer32 (0..15)
58        MAX-ACCESS  read-write
59        STATUS      current
60        DESCRIPTION
61            "FL Sector Hop seed used by the PHY Layer to determine the
62            hopping pattern for this carrier."
63        ::= { dot20AnCarrierConfigEntry 7 }
64
65    dot20AnFLIntraCellCommonHopping OBJECT-TYPE
66        SYNTAX      TruthValue
67        MAX-ACCESS  read-write
68        STATUS      current

```

```

1      DESCRIPTION
2      "This attribute is set to True if FL Intra Cell common hopping
3      used by the PHY Layer is enabled. This attribute is set to
4      False if FL Intra Cell common hopping is disabled."
5      ::= { dot20AnCarrierConfigEntry 8 }
6
7      dot20AnFLDPISectorOffset OBJECT-TYPE
8      SYNTAX      Integer32 (0..3)
9      MAX-ACCESS  read-write
10     STATUS      current
11     DESCRIPTION
12     "This attribute shall be set to the relative offset of F-DPICH
13     pilots."
14     ::= { dot20AnCarrierConfigEntry 9 }
15
16     dot20AnFLDPISectorScramble OBJECT-TYPE
17     SYNTAX      Integer32 (0..1)
18     MAX-ACCESS  read-write
19     STATUS      current
20     DESCRIPTION
21     "This attribute shall determine the scrambling of pilots as
22     defined by the Physical Layer sector and cell specific
23     scrambling."
24     ::= { dot20AnCarrierConfigEntry 10 }
25
26     dot20AnFLNumSDMADimensions OBJECT-TYPE
27     SYNTAX      Integer32 (1..4)
28     MAX-ACCESS  read-write
29     STATUS      current
30     DESCRIPTION
31     "This attribute shall determine the number of spatial
32     dimensions on the forward link."
33     ::= { dot20AnCarrierConfigEntry 11 }
34
35     dot20AnFLNumSubbands OBJECT-TYPE
36     SYNTAX      Integer32 (0..1)
37     MAX-ACCESS  read-write
38     STATUS      current
39     DESCRIPTION
40     "This attribute shall determine the number of subbands on the
41     forward link. If equal to 0, the number of subbands is equal to
42     NCARRIER_SIZE/128 and if equal to 1, the number of subbands is
43     equal to NCARRIER_SIZE/256."
44     ::= { dot20AnCarrierConfigEntry 12 }
45
46     dot20AnFLDiversityHoppingMode OBJECT-TYPE
47     SYNTAX      TruthValue
48     MAX-ACCESS  read-write
49     STATUS      current
50     DESCRIPTION
51     "This attribute shall be used by the Physical Layer to
52     determine the hop pattern for the sector. This attribute shall
53     be set to true if DiversityHoppingMode is On, false if not."
54     ::= { dot20AnCarrierConfigEntry 13 }
55
56     dot20AnNumPilots OBJECT-TYPE
57     SYNTAX      Integer32 (0..1)
58     MAX-ACCESS  read-write
59     STATUS      current
60     DESCRIPTION
61     "This attribute shall determine the nominal number of pilots in
62     F-CPICH as being NCARRIER_SIZE/16 or NCARRIER_SIZE/8, depending
63     on whether the attribute is set to 0 or 1, respectively."
64     ::= { dot20AnCarrierConfigEntry 14 }
65
66     dot20AnEffectiveNumAntennas OBJECT-TYPE
67     SYNTAX      Integer32 (1..8)
68     MAX-ACCESS  read-write

```

```

1     STATUS          current
2     DESCRIPTION
3         "This attribute shall determine the effective number of
4         antennas. This attribute shall be set to three or below when
5         the BlockHoppingEnabled field of the SystemInfo block is set to
6         0."
7     ::= { dot20AnCarrierConfigEntry 15 }
8
9     dot20AnNumCmnPilotTxAntennas OBJECT-TYPE
10    SYNTAX          Integer32 (1..2)
11    MAX-ACCESS      read-write
12    STATUS          current
13    DESCRIPTION
14        "This attribute shall determine the number of common pilot
15        transmit antennas."
16    ::= { dot20AnCarrierConfigEntry 16 }
17
18    dot20AnEnableCmnPilotStaggering OBJECT-TYPE
19    SYNTAX          TruthValue
20    MAX-ACCESS      read-write
21    STATUS          current
22    DESCRIPTION
23        "This attribute shall be set to true if common pilot staggering
24        is enabled on this sector, false if not."
25    ::= { dot20AnCarrierConfigEntry 17 }
26
27    dot20AnEnableAuxPilotStaggering OBJECT-TYPE
28    SYNTAX          TruthValue
29    MAX-ACCESS      read-write
30    STATUS          current
31    DESCRIPTION
32        "This attribute shall be set to true if auxiliary pilot
33        staggering is enabled on this sector, false if not."
34    ::= { dot20AnCarrierConfigEntry 18 }
35
36    dot20AnSSCHNumHopports OBJECT-TYPE
37    SYNTAX          Integer32 (0..7)
38    MAX-ACCESS      read-write
39    STATUS          current
40    DESCRIPTION
41        "This attribute shall determine the number of hop-ports
42        allocated to F-SSCH. This attribute shall be interpreted as
43        indicated in the 802.20 AIS spec (QuickChannelInfo block
44        description)."
45    ::= { dot20AnCarrierConfigEntry 19 }
46
47    dot20AnSSCHNumBlocks OBJECT-TYPE
48    SYNTAX          Integer32 (0..7)
49    MAX-ACCESS      read-write
50    STATUS          current
51    DESCRIPTION
52        "This attribute's value noted n shall determine the number of
53        blocks carried by the F-SSCH. The number of F-SSCH blocks shall
54        be equal to 2*(n+1)."
55    ::= { dot20AnCarrierConfigEntry 20 }
56
57    dot20AnSSCHModSymbolsPerBlock OBJECT-TYPE
58    SYNTAX          Integer32 (0..3)
59    MAX-ACCESS      read-write
60    STATUS          current
61    DESCRIPTION
62        "This attribute shall determine the number of modulation
63        symbols for each block carried by the F-SSCH. This attribute
64        shall be interpreted as indicated in 802.20 AIS spec
65        (QuickChannelInfo block description)."
66    ::= { dot20AnCarrierConfigEntry 21 }
67
68    dot20AnPilotPN OBJECT-TYPE

```

```

1      SYNTAX      Integer32 (0..4095)
2      MAX-ACCESS  read-write
3      STATUS      current
4      DESCRIPTION
5          "This attribute shall be set to the PilotPN of the sector."
6      ::= { dot20AnCarrierConfigEntry 22 }
7
8  dot20AnRLChannelTreeIndex OBJECT-TYPE
9      SYNTAX      Integer32 (0..15)
10     MAX-ACCESS  read-write
11     STATUS      current
12     DESCRIPTION
13         "RL Channel Tree Index used by the Lower MAC Sublayer."
14     ::= { dot20AnCarrierConfigEntry 23 }
15
16  dot20AnRLSectorHopSeed OBJECT-TYPE
17     SYNTAX      Integer32 (0..15)
18     MAX-ACCESS  read-write
19     STATUS      current
20     DESCRIPTION
21         "RL Hop Seed used by the PHY Layer to determine the hopping
22         pattern."
23     ::= { dot20AnCarrierConfigEntry 24 }
24
25  dot20AnRLIntraCellCommonHopping OBJECT-TYPE
26     SYNTAX      TruthValue
27     MAX-ACCESS  read-write
28     STATUS      current
29     DESCRIPTION
30         "This attribute is set to True if RL Intra Cell common hopping
31         used by the PHY Layer is enabled. This attribute is set to
32         False if RL Intra Cell common hopping is disabled."
33     ::= { dot20AnCarrierConfigEntry 25 }
34
35  dot20AnBFCHBeamCodeBookIndex OBJECT-TYPE
36     SYNTAX      Integer32 (0..15)
37     MAX-ACCESS  read-write
38     STATUS      current
39     DESCRIPTION
40         "This attribute shall refer to the code book index, the code
41         book comprising of transmit weights for SDMA and precoding."
42     ::= { dot20AnCarrierConfigEntry 26 }
43
44  dot20AnRPICHEnabled OBJECT-TYPE
45     SYNTAX      TruthValue
46     MAX-ACCESS  read-write
47     STATUS      current
48     DESCRIPTION
49         "This attribute shall be set to true if RPICH enabled on this
50         sector, false if not."
51     ::= { dot20AnCarrierConfigEntry 27 }
52
53  dot20AnRLDPISectorOffset OBJECT-TYPE
54     SYNTAX      Integer32 (0..3)
55     MAX-ACCESS  read-write
56     STATUS      current
57     DESCRIPTION
58         "Relative offset of reverse pilots"
59     ::= { dot20AnCarrierConfigEntry 28 }
60
61  dot20AnRLDPISectorScramble OBJECT-TYPE
62     SYNTAX      Integer32 (0..1)
63     MAX-ACCESS  read-write
64     STATUS      current
65     DESCRIPTION
66         "This attribute shall determine the scrambling of pilots as
67         defined by the Physical Layer sector and cell specific
68         scrambling."

```



```

1      ::= { dot20AnCarrierConfigEntry 29 }
2
3      dot20AnRLNumSDMADimensions OBJECT-TYPE
4          SYNTAX      Integer32 (1..4)
5          MAX-ACCESS  read-write
6          STATUS      current
7          DESCRIPTION
8              "This attribute shall determine the number of spatial
9              dimensions on the reverse link."
10     ::= { dot20AnCarrierConfigEntry 30 }
11
12     dot20AnRLRestrictedSetBitmap OBJECT-TYPE
13         SYNTAX      Integer32 (0..65535)
14         MAX-ACCESS  read-write
15         STATUS      current
16         DESCRIPTION
17             "Bit position j in this bitfield shall be set to 1 if subband j
18             is restricted on the reverse link."
19     ::= { dot20AnCarrierConfigEntry 31 }
20
21     dot20AnRLNumSubbands OBJECT-TYPE
22         SYNTAX      Integer32 (0..1)
23         MAX-ACCESS  read-write
24         STATUS      current
25         DESCRIPTION
26             "This attribute shall determine the number of subbands on the
27             reverse link. If equal to 0, the number of subbands is equal to
28             NCARRIER_SIZE/128 and if equal to 1, the number of subbands is
29             equal to NCARRIER_SIZE/256."
30     ::= { dot20AnCarrierConfigEntry 32 }
31
32     dot20AnRLDiversityHoppingMode OBJECT-TYPE
33         SYNTAX      TruthValue
34         MAX-ACCESS  read-write
35         STATUS      current
36         DESCRIPTION
37             "This attribute shall be used by the Physical Layer to
38             determine the hop pattern for the sector. This attribute shall
39             be set to true if DiversityHoppingMode is On, and to false if
40             DiversityHoppingMode is Off."
41     ::= { dot20AnCarrierConfigEntry 33 }
42
43     dot20AnNumRLControlSubbands OBJECT-TYPE
44         SYNTAX      Integer32 (1..8)
45         MAX-ACCESS  read-write
46         STATUS      current
47         DESCRIPTION
48             "This attribute shall determine the number of control subbands
49             on the reverse link."
50     ::= { dot20AnCarrierConfigEntry 34 }
51
52     dot20AnRACKBandwidthFactor OBJECT-TYPE
53         SYNTAX      Integer32 (1..4)
54         MAX-ACCESS  read-write
55         STATUS      current
56         DESCRIPTION
57             "This attribute shall determine the bandwidth reduction on the
58             R-ACKCH."
59     ::= { dot20AnCarrierConfigEntry 35 }
60
61     dot20AnHalfDuplexModeSupported OBJECT-TYPE
62         SYNTAX      TruthValue
63         MAX-ACCESS  read-write
64         STATUS      current
65         DESCRIPTION
66             "This attribute shall be set to True if the access network
67             supports half duplex terminals, and shall be set to False
68             otherwise. If half-duplex terminals are supported, the access

```

```

1         network should assign MAC IDs and channel assignments in a
2         manner that enables half-duplex terminal operation. A
3         half-duplex access terminal is not required to monitor forward
4         link transmissions on a PHY Frame where it is scheduled to make
5         a reverse link transmission."
6     ::= { dot20AnCarrierConfigEntry 36 }
7
8 dot20AnRevLinkSilenceDuration OBJECT-TYPE
9     SYNTAX      Integer32 (0..15)
10    MAX-ACCESS   read-write
11    STATUS       current
12    DESCRIPTION
13        "This attribute's value noted n shall determine the duration of
14        the Reverse Link Silence Interval. The Reverse Link Silence
15        duration shall be equal to 2^n PHY Frames. In a region with
16        asynchronous sectors, this attribute shall be set to a value
17        larger than the timing offset between sectors."
18    ::= { dot20AnCarrierConfigEntry 37 }
19
20 dot20AnRevLinkSilencePeriod OBJECT-TYPE
21    SYNTAX      Integer32 (0..15)
22    MAX-ACCESS   read-write
23    STATUS       current
24    DESCRIPTION
25        "This attribute's value noted n shall determine the periodicity
26        of occurrence the Reverse Link Silence Interval. The reverse
27        link silence interval shall take the value
28        ReverseLinkSilencePeriod = (1+n)*144000. The Reverse Link
29        Silence Interval is defined as the time interval of duration
30        ReverseLinkSilenceDuration RL PHY Frames that starts at
31        superframe index m that satisfies the following equation: m mod
32        (ReverseLinkSilencePeriod) = 0"
33    ::= { dot20AnCarrierConfigEntry 38 }
34
35 dot20AnTransmitPower OBJECT-TYPE
36    SYNTAX      Integer32 (0..63)
37    MAX-ACCESS   read-write
38    STATUS       current
39    DESCRIPTION
40        "This attribute shall be set to the transmit power of the
41        sector in units of dBm"
42    ::= { dot20AnCarrierConfigEntry 39 }
43
44 dot20AnCommonPilotPower OBJECT-TYPE
45    SYNTAX      Integer32 (0..15)
46    MAX-ACCESS   read-write
47    STATUS       current
48    DESCRIPTION
49        "The attribute's value noted n shall determine the power
50        spectral density of the F-CPICH during the FL PHY frame
51        relative to the F-ACQCH. The pilot power density shall be equal
52        to (-4 + n*0.5) dB."
53    ::= { dot20AnCarrierConfigEntry 40 }
54
55 dot20AnAuxPilotPower OBJECT-TYPE
56    SYNTAX      Integer32 (0..15)
57    MAX-ACCESS   read-write
58    STATUS       current
59    DESCRIPTION
60        "The attribute's value noted n shall determine the power
61        spectral density of the F-AuxPICH relative to the F-ACQCH. The
62        pilot power density shall be equal to (-4 + n*0.5) dB."
63    ::= { dot20AnCarrierConfigEntry 41 }
64
65 dot20AnPreamblePilotPower OBJECT-TYPE
66    SYNTAX      Integer32 (0..15)
67    MAX-ACCESS   read-write
68    STATUS       current

```

```

1      DESCRIPTION
2          "The attribute's value noted n shall determine the power
3          spectral density of the F-CPICH during the superframe preamble
4          relative to the F-ACQCH. The pilot power density shall be equal
5          to (-4 + n*0.5) dB."
6      ::= { dot20AnCarrierConfigEntry 42 }
7
8      dot20AnMACIDRange OBJECT-TYPE
9          SYNTAX      INTEGER {
10             upTo63(1),
11             upTo127(2),
12             upTo255(3),
13             upTo511(4),
14             upTo1023(5),
15             upTo2047(6)
16         }
17          MAX-ACCESS      read-write
18          STATUS          current
19          DESCRIPTION
20             "This attribute shall be set to indicate the range of assigned
21             MACID values in the sector."
22      ::= { dot20AnCarrierConfigEntry 43 }
23
24      dot20AnFLPCReportInterval OBJECT-TYPE
25          SYNTAX      Integer32 (1..16)
26          MAX-ACCESS      read-write
27          STATUS          current
28          DESCRIPTION
29             "FLPC Report Interval determines the periodicity at which power
30             control commands are sent to the access terminal in PHY
31             Frames."
32      ::= { dot20AnCarrierConfigEntry 44 }
33
34      dot20AnRLCtrlPCMode OBJECT-TYPE
35          SYNTAX      INTEGER {
36             upDown(1),
37             erasureBased(2)
38         }
39          MAX-ACCESS      read-write
40          STATUS          current
41          DESCRIPTION
42             "This attribute shall determine the closed loop power control
43             mode of the sector."
44      ::= { dot20AnCarrierConfigEntry 45 }
45
46      dot20AnCtrlAccessOffset OBJECT-TYPE
47          SYNTAX      Integer32 (0..7)
48          MAX-ACCESS      read-write
49          STATUS          current
50          DESCRIPTION
51             "This attribute shall be set to the initial gain of the R-CQICH
52             over the R-ACH in units of dB expressed in 2's complement
53             notation."
54      ::= { dot20AnCarrierConfigEntry 46 }
55
56      dot20AnBFCHPowerOffset OBJECT-TYPE
57          SYNTAX      Integer32 (0..15)
58          MAX-ACCESS      read-write
59          STATUS          current
60          DESCRIPTION
61             "This attribute shall be set to power offset of the R-BFCH
62             relative to the R-CQICH in units of dB expressed in 2's
63             complement notation."
64      ::= { dot20AnCarrierConfigEntry 47 }
65
66      dot20AnSFCHPowerOffset OBJECT-TYPE
67          SYNTAX      Integer32 (0..15)
68          MAX-ACCESS      read-write

```

```

1      STATUS      current
2      DESCRIPTION
3          "This attribute shall be set to power offset of the R-SFCH
4          relative to the R-CQICHin units of dB expressed in 2's
5          complement notation."
6      ::= { dot20AnCarrierConfigEntry 48 }
7
8      dot20AnPICHPowerOffset OBJECT-TYPE
9          SYNTAX      Integer32 (0..15)
10         MAX-ACCESS  read-write
11         STATUS      current
12         DESCRIPTION
13             "This attribute shall be set to power offset of the R-PICH
14             relative to the R-CQICHin units of dB expressed in 2's
15             complement notation."
16         ::= { dot20AnCarrierConfigEntry 49 }
17
18         dot20AnReqChannelGain0 OBJECT-TYPE
19             SYNTAX      Integer32 (0..15)
20             MAX-ACCESS  read-write
21             STATUS      current
22             DESCRIPTION
23                 "This attribute shall be set to power offset of the R-REQCH
24                 relative to the R-CQICHin units of dB expressed in 2's
25                 complement notation."
26             ::= { dot20AnCarrierConfigEntry 50 }
27
28         dot20AnReqChannelGain1 OBJECT-TYPE
29             SYNTAX      Integer32 (0..15)
30             MAX-ACCESS  read-write
31             STATUS      current
32             DESCRIPTION
33                 "This attribute shall be set to power offset of the R-REQCH
34                 relative to the R-CQICHin units of dB expressed in 2's
35                 complement notation."
36             ::= { dot20AnCarrierConfigEntry 51 }
37
38         dot20AnReqChannelGain2 OBJECT-TYPE
39             SYNTAX      Integer32 (0..15)
40             MAX-ACCESS  read-write
41             STATUS      current
42             DESCRIPTION
43                 "This attribute shall be set to power offset of the R-REQCH
44                 relative to the R-CQICHin units of dB expressed in 2's
45                 complement notation."
46             ::= { dot20AnCarrierConfigEntry 52 }
47
48         dot20AnReqChannelGain3 OBJECT-TYPE
49             SYNTAX      Integer32 (0..15)
50             MAX-ACCESS  read-write
51             STATUS      current
52             DESCRIPTION
53                 "This attribute shall be set to power offset of the R-REQCH
54                 relative to the R-CQICHin units of dB expressed in 2's
55                 complement notation."
56             ::= { dot20AnCarrierConfigEntry 53 }
57
58         dot20AnErasureGain0 OBJECT-TYPE
59             SYNTAX      Integer32 (0..15)
60             MAX-ACCESS  read-write
61             STATUS      current
62             DESCRIPTION
63                 "This attribute's value noted n shall determine the transmit
64                 power of erasure sequences for different assignment sizes. The
65                 transmit power shall be equal to n-4 dB."
66             ::= { dot20AnCarrierConfigEntry 54 }
67
68         dot20AnErasureGain1 OBJECT-TYPE

```

```

1      SYNTAX      Integer32 (0..15)
2      MAX-ACCESS  read-write
3      STATUS      current
4      DESCRIPTION
5          "This attribute's value noted n shall determine the transmit
6          power of erasure sequences for different assignment sizes. The
7          transmit power shall be equal to n-4 dB."
8      ::= { dot20AnCarrierConfigEntry 55 }
9
10     dot20AnErasureGain2 OBJECT-TYPE
11         SYNTAX      Integer32 (0..15)
12         MAX-ACCESS  read-write
13         STATUS      current
14         DESCRIPTION
15             "This attribute's value noted n shall determine the transmit
16             power of erasure sequences for different assignment sizes. The
17             transmit power shall be equal to n-4 dB."
18         ::= { dot20AnCarrierConfigEntry 56 }
19
20     dot20AnErasureGain3 OBJECT-TYPE
21         SYNTAX      Integer32 (0..15)
22         MAX-ACCESS  read-write
23         STATUS      current
24         DESCRIPTION
25             "This attribute's value noted n shall determine the transmit
26             power of erasure sequences for different assignment sizes. The
27             transmit power shall be equal to n-4 dB."
28         ::= { dot20AnCarrierConfigEntry 57 }
29
30     dot20AnAccessCycleDuration OBJECT-TYPE
31         SYNTAX      Integer32 (1..4)
32         MAX-ACCESS  read-write
33         STATUS      current
34         DESCRIPTION
35             "This attribute shall determine the duration of the access
36             cycle in units of Control Segment Periods (as defined by the
37             802.20 AIS spec Physical Layer)."
38         ::= { dot20AnCarrierConfigEntry 59 }
39
40     dot20AnAccessSequencePartition OBJECT-TYPE
41         SYNTAX      Integer32 (0..31)
42         MAX-ACCESS  read-write
43         STATUS      current
44         DESCRIPTION
45             "This attribute shall indicate the partition of the access
46             sequence space to allow the access terminal to signal pilot
47             power and buffer status information with the access sequence."
48         ::= { dot20AnCarrierConfigEntry 60 }
49
50     dot20AnMaxProbesPerSequence OBJECT-TYPE
51         SYNTAX      Integer32 (1..16)
52         MAX-ACCESS  read-write
53         STATUS      current
54         DESCRIPTION
55             "This attribute shall determine the maximum number of probe
56             sequences that can be part of one access sequence."
57         ::= { dot20AnCarrierConfigEntry 61 }
58
59     dot20AnProbeRampUpStepSize OBJECT-TYPE
60         SYNTAX      Integer32 (0..15)
61         MAX-ACCESS  read-write
62         STATUS      current
63         DESCRIPTION
64             "This attribute's value noted n shall determine the power ramp
65             up used for probes within a probe sequence and shall indicate a
66             ramp up value of 0.5*(1+n) dB."
67         ::= { dot20AnCarrierConfigEntry 62 }
68

```

```

1 dot20AnFastOSIEnabled OBJECT-TYPE
2     SYNTAX      TruthValue
3     MAX-ACCESS  read-write
4     STATUS      current
5     DESCRIPTION
6         "This field shall be set to true if the F-SSCH transmitted by
7         this sector contains a Fast OSI Segment. This field shall be
8         set to false if the F-SSCH transmitted by this sector does not
9         contain a Fast OSI Segment."
10    ::= { dot20AnCarrierConfigEntry 63 }
11
12 dot20AnRDCHInitialPacketFormat OBJECT-TYPE
13     SYNTAX      Integer32 (0..63)
14     MAX-ACCESS  read-write
15     STATUS      current
16     DESCRIPTION
17         "This attribute shall be set to the packet format that is used
18         on the first transmission the access terminal makes on the
19         R-DCH after getting an access grant."
20    ::= { dot20AnCarrierConfigEntry 64 }
21
22 dot20AnPilotThreshold1 OBJECT-TYPE
23     SYNTAX      Integer32 (0..3)
24     MAX-ACCESS  read-write
25     STATUS      current
26     DESCRIPTION
27         "This attribute's value noted n shall determine PilotThreshold1
28         used by the Access Channel MAC Protocol. The value shall be -2n
29         dB."
30    ::= { dot20AnCarrierConfigEntry 65 }
31
32 dot20AnPilotThreshold2 OBJECT-TYPE
33     SYNTAX      Integer32 (0..3)
34     MAX-ACCESS  read-write
35     STATUS      current
36     DESCRIPTION
37         "This attribute's value noted n shall determine PilotThreshold2
38         used by the Access Channel MAC Protocol. The value shall be -2n
39         dB."
40    ::= { dot20AnCarrierConfigEntry 66 }
41
42 dot20AnOpenLoopAdjust OBJECT-TYPE
43     SYNTAX      Integer32 (0..255)
44     MAX-ACCESS  read-write
45     STATUS      current
46     DESCRIPTION
47         "This attribute's value noted n shall determine the nominal
48         power to be used by access terminal in the open loop power
49         estimate. The value of nominal power shall be 70+n dB."
50    ::= { dot20AnCarrierConfigEntry 67 }
51
52 dot20AnAccessRetryPersistence0 OBJECT-TYPE
53     SYNTAX      Integer32 (0..7)
54     MAX-ACCESS  read-write
55     STATUS      current
56     DESCRIPTION
57         "This attribute shall determine the persistence probability for
58         determining access sequence backoff. If this attribute's value
59         is set to n, the access terminal shall use 2^-n as the retry
60         persistence."
61    ::= { dot20AnCarrierConfigEntry 68 }
62
63 dot20AnAccessRetryPersistence1 OBJECT-TYPE
64     SYNTAX      Integer32 (0..7)
65     MAX-ACCESS  read-write
66     STATUS      current
67     DESCRIPTION
68         "This attribute shall determine the persistence probability for

```

```

1         determining access sequence backoff. If this attribute's value
2         is set to n, the access terminal shall use 2^-n as the retry
3         persistence."
4     ::= { dot20AnCarrierConfigEntry 69 }
5
6 dot20AnAccessRetryPersistence2 OBJECT-TYPE
7     SYNTAX      Integer32 (0..7)
8     MAX-ACCESS  read-write
9     STATUS      current
10    DESCRIPTION
11        "This attribute shall determine the persistence probability for
12        determining access sequence backoff. If this attribute's value
13        is set to n, the access terminal shall use 2^-n as the retry
14        persistence."
15    ::= { dot20AnCarrierConfigEntry 70 }
16
17 dot20AnAccessRetryPersistence3 OBJECT-TYPE
18     SYNTAX      Integer32 (0..7)
19     MAX-ACCESS  read-write
20     STATUS      current
21     DESCRIPTION
22        "This attribute shall determine the persistence probability for
23        determining access sequence backoff. If this attribute's value
24        is set to n, the access terminal shall use 2^-n as the retry
25        persistence."
26    ::= { dot20AnCarrierConfigEntry 71 }
27
28 dot20AnSynchGroup OBJECT-TYPE
29     SYNTAX      Integer32 (0..2147483647)
30     MAX-ACCESS  read-write
31     STATUS      current
32     DESCRIPTION
33        "This attribute specifies the synchronization group to which
34        this carrier belongs to. All carriers (local, i.e. defined in
35        the CarrierConfig table, or remote, i.e. defined in the
36        NeighborCarriers table) which are synchronous with this carrier
37        should belong to the same group. The value 0 indicates that the
38        synchronization for this carrier is unknown."
39    ::= { dot20AnCarrierConfigEntry 72 }
40
41 dot20AnCarrierConfigRowStatus OBJECT-TYPE
42     SYNTAX      RowStatus
43     MAX-ACCESS  read-create
44     STATUS      current
45     DESCRIPTION
46        "The status column used for creating, modifying, and deleting
47        instances of the columnar objects in the CarrierConfig Table.
48        If the implementor of this MIB has chosen not to implement
49        'dynamic assignment' of carriers, this attribute is not useful
50        and should return noSuchName upon SNMP request."
51     DEFVAL      { active }
52    ::= { dot20AnCarrierConfigEntry 73 }
53
54 dot20AnChannelBandsTable OBJECT-TYPE
55     SYNTAX      SEQUENCE OF Dot20AnChannelBandsEntry
56     MAX-ACCESS  not-accessible
57     STATUS      current
58     DESCRIPTION
59        "This table provides one row per 802.20 ChannelBand. This
60        table's attributes specify the ChannelBand record of a
61        particular ChannelBand which may be used for a sector defined
62        in the SectorConfig table, or as a neighbor to one sector
63        defined in the SectorConfig table."
64    ::= { dot20AnOverheadMessages 4 }
65
66 dot20AnChannelBandsEntry OBJECT-TYPE
67     SYNTAX      Dot20AnChannelBandsEntry
68     MAX-ACCESS  not-accessible

```

```

1      STATUS      current
2      DESCRIPTION
3          "An Entry (conceptual row) in the ChannelBands table. This
4          table is indexed by ChannelBandIndex."
5      INDEX
6          { dot20AnChannelBandIndex }
7      ::= { dot20AnChannelBandsTable 1 }
8
9      dot20AnChannelBandIndex OBJECT-TYPE
10     SYNTAX      Integer32 (1..2147483647)
11     MAX-ACCESS  not-accessible
12     STATUS      current
13     DESCRIPTION
14         "Index of the ChannelBand within the ChannelBands table."
15     ::= { dot20AnChannelBandsEntry 1 }
16
17     dot20AnChannelBandRecordType OBJECT-TYPE
18     SYNTAX      INTEGER {
19         frequencySpecified(1),
20         bandClass(2)
21     }
22     MAX-ACCESS  read-write
23     STATUS      current
24     DESCRIPTION
25         "ChannelBand Record Type for this ChannelBand. If equal to
26         bandclass then ChannelBandClass and ChannelNumber are used to
27         specify the Channel record for this channel. If equal to
28         frequencySpecified then ChannelFwdLinkCenterFreq,
29         ChannelRevLinkCenterFreq, ChannelFwdLinkSystemBw amd
30         ChannelRevLinkSystemBw are used to specify the ChannelBand
31         record for this ChannelBand."
32     ::= { dot20AnChannelBandsEntry 2 }
33
34     dot20AnChannelBandClass OBJECT-TYPE
35     SYNTAX      Integer32 (0..31)
36     MAX-ACCESS  read-write
37     STATUS      current
38     DESCRIPTION
39         "This attribute shall be set to the band class number
40         corresponding to the frequency assignment of the ChannelBand
41         specified by this record."
42     ::= { dot20AnChannelBandsEntry 3 }
43
44     dot20AnChannelNumber OBJECT-TYPE
45     SYNTAX      Integer32 (0..2047)
46     MAX-ACCESS  read-write
47     STATUS      current
48     DESCRIPTION
49         "This attribute shall be set to the Channel number
50         corresponding to the frequency assignment of the ChannelBand
51         specified by this record."
52     ::= { dot20AnChannelBandsEntry 4 }
53
54     dot20AnChannelFwdLinkCenterFreq OBJECT-TYPE
55     SYNTAX      Integer32 (0..16777215)
56     MAX-ACCESS  read-write
57     STATUS      current
58     DESCRIPTION
59         "This attribute shall be set to the value of the center
60         frequency of the forward link channel band in units of KHz. The
61         value 0 indicates that the center frequency of the forward link
62         channel is unspecified."
63     ::= { dot20AnChannelBandsEntry 5 }
64
65     dot20AnChannelRevLinkCenterFreq OBJECT-TYPE
66     SYNTAX      Integer32 (0..16777215)
67     MAX-ACCESS  read-write
68     STATUS      current

```



```

1      DESCRIPTION
2      "This attribute shall be set to the value of the center
3      frequency of the reverse link channel band in units of KHz. The
4      value 0 indicates that the center frequency of the reverse link
5      channel is unspecified."
6      ::= { dot20AnChannelBandsEntry 6 }
7
8      dot20AnChannelFwdLinkSystemBw OBJECT-TYPE
9          SYNTAX      Integer32 (0..65535)
10         MAX-ACCESS   read-write
11         STATUS      current
12         DESCRIPTION
13             "The access network shall set this field to the value of the
14             system bandwidth of the forward link channel in units of kHz.
15             If this value is 0, then the forward link channel's system
16             bandwidth is unspecified."
17         ::= { dot20AnChannelBandsEntry 7 }
18
19         dot20AnChannelRevLinkSystemBw OBJECT-TYPE
20             SYNTAX      Integer32 (0..65535)
21             MAX-ACCESS   read-write
22             STATUS      current
23             DESCRIPTION
24                 "The access network shall set this field to the value of the
25                 system bandwidth of the reverse link channel in units of kHz.
26                 If this value is 0, then the reverse link channel's system
27                 bandwidth is unspecified."
28             ::= { dot20AnChannelBandsEntry 8 }
29
30         dot20AnChannelBandStatus OBJECT-TYPE
31             SYNTAX      RowStatus
32             MAX-ACCESS   read-create
33             STATUS      current
34             DESCRIPTION
35                 "The status column used for creating, modifying, and deleting
36                 instances of the columnar objects in the ChannelBands Table.
37                 If the implementor of this MIB has chosen not to implement
38                 'dynamic assignment' of ChannelBands, this attribute is not
39                 useful and should return noSuchName upon SNMP request."
40             DEFVAL      { active }
41             ::= { dot20AnChannelBandsEntry 9 }
42
43         dot20AnNeighborCarriersTable OBJECT-TYPE
44             SYNTAX      SEQUENCE OF Dot20AnNeighborCarriersEntry
45             MAX-ACCESS   not-accessible
46             STATUS      current
47             DESCRIPTION
48                 "This table provides one row per 802.20 neighbor carrier of a
49                 neighbor sector. This table's attributes specify the sector and
50                 carrier parameters of a particular neighbor carrier which may
51                 be used as a neighbor to one sector defined in the SectorConfig
52                 table."
53             ::= { dot20AnOverheadMessages 5 }
54
55         dot20AnNeighborCarriersEntry OBJECT-TYPE
56             SYNTAX      Dot20AnNeighborCarriersEntry
57             MAX-ACCESS   not-accessible
58             STATUS      current
59             DESCRIPTION
60                 "An Entry (conceptual row) in the AnNeighborCarriers table.
61                 This table is indexed by ChannelBandIndex, NeighborSectorIndex,
62                 NeighborCarrierID."
63             INDEX
64                 { dot20AnChannelBandIndex, dot20AnNeighborSectorIndex,
65                   dot20AnNeighborCarrierID }
66             ::= { dot20AnNeighborCarriersTable 1 }
67
68         dot20AnNeighborSectorIndex OBJECT-TYPE

```

```

1      SYNTAX      Integer32 (1..2147483647)
2      MAX-ACCESS  not-accessible
3      STATUS      current
4      DESCRIPTION
5          "Index of the Neighbor Sector for this Neighbor Carrier within
6          the ChannelBand."
7      ::= { dot20AnNeighborCarriersEntry 1 }
8
9  dot20AnNeighborCarrierID OBJECT-TYPE
10     SYNTAX      Integer32 (0..3)
11     MAX-ACCESS  read-only
12     STATUS      current
13     DESCRIPTION
14         "CarrierID of the Neighbor Carrier for this Neighbor Sector.
15         The CarrierID is unique within the NeighborSector."
16     ::= { dot20AnNeighborCarriersEntry 2 }
17
18 dot20AnNeighborSectorID OBJECT-TYPE
19     SYNTAX      OCTET STRING (SIZE(16))
20     MAX-ACCESS  read-write
21     STATUS      current
22     DESCRIPTION
23         "Sector Address Identifier. The access network shall set the
24         value of the SectorID according to the rules specified in IEEE
25         802.20 AIS. The access terminal shall not assume anything about
26         the format of the SectorID other than that it uniquely
27         identifies the sector."
28     ::= { dot20AnNeighborCarriersEntry 3 }
29
30 dot20AnNeighborPilotPN OBJECT-TYPE
31     SYNTAX      Integer32 (0..4095)
32     MAX-ACCESS  read-write
33     STATUS      current
34     DESCRIPTION
35         "This attribute shall be set to the PilotPN of a neighboring
36         sector that the access terminal should add to its Neighbor
37         Set."
38     ::= { dot20AnNeighborCarriersEntry 4 }
39
40 dot20AnNeighborTransmitPower OBJECT-TYPE
41     SYNTAX      Integer32 (0..63)
42     MAX-ACCESS  read-write
43     STATUS      current
44     DESCRIPTION
45         "This attribute shall be set to the transmit power of the
46         sector in units of dBm."
47     ::= { dot20AnNeighborCarriersEntry 5 }
48
49 dot20AnNghbrGloballySynch OBJECT-TYPE
50     SYNTAX      TruthValue
51     MAX-ACCESS  read-write
52     STATUS      current
53     DESCRIPTION
54         "This attribute shall be set to true if the sector transmitting
55         this pilot is synchronous to system time."
56     ::= { dot20AnNeighborCarriersEntry 6 }
57
58 dot20AnNgbhrSynchGroup OBJECT-TYPE
59     SYNTAX      Integer32 (0..2147483647)
60     MAX-ACCESS  read-write
61     STATUS      current
62     DESCRIPTION
63         "This attribute specifies the synchronization group to which
64         this carrier belongs to. All carriers (local, i.e. defined in
65         the CarrierConfig table, or remote, i.e. defined in the
66         NeighborCarriers table) which are synchronous with this carrier
67         should belong to the same group. The value 0 indicates that the
68         synchronization for this carrier is unknown."

```

```

1      ::= { dot20AnNeighborCarriersEntry 7 }
2
3  dot20AnNeighborCarrierStatus OBJECT-TYPE
4      SYNTAX          RowStatus
5      MAX-ACCESS      read-create
6      STATUS          current
7      DESCRIPTION
8          "The status column used for creating, modifying, and deleting
9          instances of the columnar objects in the NeighborCarriers
10         Table. If the implementor of this MIB has chosen not to
11         implement 'dynamic assignment' of neighbor carriers this
12         attribute is not useful and should return noSuchName upon SNMP
13         request."
14     DEFVAL          { active }
15     ::= { dot20AnNeighborCarriersEntry 8 }
16
17 dot20AnOtherTechNghbrsTable OBJECT-TYPE
18     SYNTAX          SEQUENCE OF Dot20AnOtherTechNghbrsEntry
19     MAX-ACCESS      not-accessible
20     STATUS          current
21     DESCRIPTION
22         "This table provides one row per other technology neighbor
23         channel. This table's attributes specify the technology type
24         and neighborlist of a particular neighbor channel which may be
25         used by one sector defined in the SectorConfig table for
26         inter-technology handoff."
27     ::= { dot20AnOverheadMessages 6 }
28
29 dot20AnOtherTechNghbrsEntry OBJECT-TYPE
30     SYNTAX          Dot20AnOtherTechNghbrsEntry
31     MAX-ACCESS      not-accessible
32     STATUS          current
33     DESCRIPTION
34         "An Entry (conceptual row) in the AnOtherTechNghbrs table. This
35         table is indexed by Sector (ifIndex) and OtherTechnologyIndex"
36     INDEX
37         { ifIndex, dot20AnOtherTechnologyIndex }
38     ::= { dot20AnOtherTechNghbrsTable 1 }
39
40 dot20AnOtherTechnologyIndex OBJECT-TYPE
41     SYNTAX          Integer32 (1..2147483647)
42     MAX-ACCESS      not-accessible
43     STATUS          current
44     DESCRIPTION
45         "The neighbor other technology entry index"
46     ::= { dot20AnOtherTechNghbrsEntry 1 }
47
48 dot20AnTechnologyType OBJECT-TYPE
49     SYNTAX          INTEGER {
50         l2tp(1),
51         w802dot11(2),
52         cdma20001x(3),
53         cdma20001xe(4),
54         gsm(5),
55         wcdma(6)
56     }
57     MAX-ACCESS      read-write
58     STATUS          current
59     DESCRIPTION
60         "This attribute shall be set to the type of other technology"
61     ::= { dot20AnOtherTechNghbrsEntry 2 }
62
63 dot20AnTechNghbrListLength OBJECT-TYPE
64     SYNTAX          Integer32 (0..255)
65     MAX-ACCESS      read-write
66     STATUS          current
67     DESCRIPTION
68         "This attribute shall be set the length, in bytes, of the

```

```

1      neighbor list information for the other technology."
2      ::= { dot20AnOtherTechNghbrsEntry 3 }
3
4  dot20AnTechnologyNeighborList OBJECT-TYPE
5      SYNTAX      OCTET STRING (SIZE(256))
6      MAX-ACCESS  read-write
7      STATUS      current
8      DESCRIPTION
9          "This attribute shall be set to the neighbor list information
10         for the other technology."
11     ::= { dot20AnOtherTechNghbrsEntry 4 }
12
13  dot20AnOtherTechNghbrRowStatus OBJECT-TYPE
14      SYNTAX      RowStatus
15      MAX-ACCESS  read-create
16      STATUS      current
17      DESCRIPTION
18          "The status column used for creating, modifying, and deleting
19         instances of the columnar objects in the OtherTechNghbrs Table.
20         If the implementor of this MIB has chosen not to implement
21         'dynamic assignment' of other technology neighbors, this
22         attribute is not useful and should return noSuchName upon SNMP
23         request."
24      DEFVAL      { active }
25     ::= { dot20AnOtherTechNghbrsEntry 5 }
26
27  dot20AnNeighborListTable OBJECT-TYPE
28      SYNTAX      SEQUENCE OF Dot20AnNeighborListEntry
29      MAX-ACCESS  not-accessible
30      STATUS      current
31      DESCRIPTION
32          "This table defines the neighbor lists for the sectors defined
33         in the SectorConfig table. Each row in this table indexed per
34         sector (ifIndex) specifies a pointer to a neighbor carrier of
35         this sector."
36     ::= { dot20AnOverheadMessages 7 }
37
38  dot20AnNeighborListEntry OBJECT-TYPE
39      SYNTAX      Dot20AnNeighborListEntry
40      MAX-ACCESS  not-accessible
41      STATUS      current
42      DESCRIPTION
43          "An Entry (conceptual row) in the AnNeighborList table. This
44         table is indexed by Sector (ifIndex) and NeighborIndex indexing
45         each neighbor carrier for a particular Sector."
46      INDEX
47          { ifIndex, dot20AnNeighborIndex }
48     ::= { dot20AnNeighborListTable 1 }
49
50  dot20AnNeighborIndex OBJECT-TYPE
51      SYNTAX      Integer32 (1..32)
52      MAX-ACCESS  not-accessible
53      STATUS      current
54      DESCRIPTION
55          "This index identifies one neighbor carrier for a Sector."
56     ::= { dot20AnNeighborListEntry 1 }
57
58  dot20AnNeighborCarrierPointer OBJECT-TYPE
59      SYNTAX      RowPointer
60      MAX-ACCESS  read-create
61      STATUS      current
62      DESCRIPTION
63          "This attribute points to an instance of carrier in
64         CarrierConfig table or in NeighborCarriers table. This carrier
65         is defined as a neighbor of the sector identified by the
66         ifIndex of this attribute's entry."
67     ::= { dot20AnNeighborListEntry 2 }
68

```

```

1 dot20AnNeighborRowStatus OBJECT-TYPE
2     SYNTAX          RowStatus
3     MAX-ACCESS      read-create
4     STATUS          current
5     DESCRIPTION
6         "The status column used for creating, modifying, and deleting
7         instances of the columnar objects in the NeighborList Table.
8         If the implementor of this MIB has chosen not to implement
9         'dynamic assignment' of neighbor list entries this attribute is
10        not useful and should return noSuchName upon SNMP request."
11    DEFVAL          { active }
12    ::= { dot20AnNeighborListEntry 3 }
13
14 dot20AnSectorToIfIndexTable OBJECT-TYPE
15     SYNTAX          SEQUENCE OF Dot20AnSectorToIfIndexEntry
16     MAX-ACCESS      not-accessible
17     STATUS          current
18     DESCRIPTION
19         "This table can be used to find the ifIndex of an 802.20
20         interface based on its SectorID and ChannelBand information
21         (reverse mapping of the Sector Config table)."
22    ::= { dot20An 2 }
23
24 dot20AnSectorToIfIndexEntry OBJECT-TYPE
25     SYNTAX          Dot20AnSectorToIfIndexEntry
26     MAX-ACCESS      not-accessible
27     STATUS          current
28     DESCRIPTION
29         "An Entry (conceptual row) in the AnSectorToIfIndex table."
30     INDEX
31         { dot20AnSectorID, ifIndex }
32    ::= { dot20AnSectorToIfIndexTable 1 }
33
34 dot20AnIfChannelBandRef OBJECT-TYPE
35     SYNTAX          Integer32
36     MAX-ACCESS      read-write
37     STATUS          current
38     DESCRIPTION
39         "The reference to the ChannelBand defined in ChannelBands table
40         (dot20AnChannelBandIndex)"
41    ::= { dot20AnSectorToIfIndexEntry 1 }
42
43 dot20Cmn OBJECT-IDENTITY
44     STATUS          current
45     DESCRIPTION
46         "Common configuration and statistics."
47    ::= { ieee802dot20 2 }
48
49 dot20CmnMac OBJECT-IDENTITY
50     STATUS          current
51     DESCRIPTION
52         "MAC layer objects"
53    ::= { dot20Cmn 1 }
54
55 dot20CmnSessionControl OBJECT IDENTIFIER ::= { dot20CmnMac 1 }
56
57 dot20CmnSessionMgtProtocol OBJECT IDENTIFIER ::= { dot20CmnSessionControl 1 }
58
59 dot20CmnSessionOpenCounts OBJECT-TYPE
60     SYNTAX          Counter64
61     MAX-ACCESS      read-only
62     STATUS          current
63     DESCRIPTION
64         "Number of session opened"
65    ::= { dot20CmnSessionMgtProtocol 1 }
66
67 dot20CmnSessionCloseCounts OBJECT-TYPE
68     SYNTAX          Counter64

```

```

1      MAX-ACCESS      read-only
2      STATUS          current
3      DESCRIPTION
4          "Number of session closed"
5      ::= { dot20CmnSessionMgtProtocol 2 }
6
7  dot20CmnSessionFailureCounts OBJECT-TYPE
8      SYNTAX          Counter64
9      MAX-ACCESS      read-only
10     STATUS          current
11     DESCRIPTION
12         "Number of session open/close failures"
13     ::= { dot20CmnSessionMgtProtocol 3 }
14
15 dot20CmnSessionConfigProtocol OBJECT IDENTIFIER ::= { dot20CmnSessionControl 2
16 }
17
18 dot20CmnSessionConfigTokenTable OBJECT-TYPE
19     SYNTAX          SEQUENCE OF Dot20CmnSessionConfigTokenEntry
20     MAX-ACCESS      not-accessible
21     STATUS          current
22     DESCRIPTION
23         "This table provides one row per supported session
24         configuration token."
25     ::= { dot20CmnSessionConfigProtocol 1 }
26
27 dot20CmnSessionConfigTokenEntry OBJECT-TYPE
28     SYNTAX          Dot20CmnSessionConfigTokenEntry
29     MAX-ACCESS      not-accessible
30     STATUS          current
31     DESCRIPTION
32         "An Entry (conceptual row) in the SessionConfigToken table.
33         This table is indexed by TokenIndex."
34     INDEX
35         { dot20CmnSessionTokenIndex }
36     ::= { dot20CmnSessionConfigTokenTable 1 }
37
38 dot20CmnSessionTokenIndex OBJECT-TYPE
39     SYNTAX          Integer32 (1..2147483647)
40     MAX-ACCESS      not-accessible
41     STATUS          current
42     DESCRIPTION
43         "The index of a supported session configuration token."
44     ::= { dot20CmnSessionConfigTokenEntry 1 }
45
46 dot20CmnSessionConfToken OBJECT-TYPE
47     SYNTAX          Integer32 (0..65535)
48     MAX-ACCESS      read-create
49     STATUS          current
50     DESCRIPTION
51         "The value of a supported session configuration token."
52     ::= { dot20CmnSessionConfigTokenEntry 2 }
53
54 dot20CmnSessionConfTokenStatus OBJECT-TYPE
55     SYNTAX          RowStatus
56     MAX-ACCESS      read-create
57     STATUS          current
58     DESCRIPTION
59         "The status column used for creating, modifying, and deleting
60         instances of the columnar objects in the SessionConfigToken
61         Table. If the implementor of this MIB has chosen not to
62         implement 'dynamic assignment' of session configuration tokens,
63         this attribute is not useful and should return noSuchName upon
64         SNMP request."
65     DEFVAL          { active }
66     ::= { dot20CmnSessionConfigTokenEntry 3 }
67
68 dot20CmnSecurityCcontrol OBJECT IDENTIFIER ::= { dot20CmnMac 2 }

```

```

1
2 dot20CmnKeyExchangeProtocol OBJECT IDENTIFIER ::= { dot20CmnSecurityControl 1
3 }
4
5 dot20CmnKeyExchangeAttemptCounts OBJECT-TYPE
6     SYNTAX      Counter64
7     MAX-ACCESS  read-only
8     STATUS      current
9     DESCRIPTION
10      "Number of key exchanges attempts"
11     ::= { dot20CmnKeyExchangeProtocol 1 }
12
13 dot20CmnKeyExchangeFailureCounts OBJECT-TYPE
14     SYNTAX      Counter64
15     MAX-ACCESS  read-only
16     STATUS      current
17     DESCRIPTION
18      "Number of key exchanges failures"
19     ::= { dot20CmnKeyExchangeProtocol 2 }
20
21 dot20CmnLowerMACControl OBJECT IDENTIFIER ::= { dot20CmnMac 3 }
22
23 dot20CmnConnectedState OBJECT IDENTIFIER ::= { dot20CmnLowerMACControl 1 }
24
25 dot20CmnActiveConnectionCounts OBJECT-TYPE
26     SYNTAX      Counter64
27     MAX-ACCESS  read-only
28     STATUS      current
29     DESCRIPTION
30      "Number of current active connections"
31     ::= { dot20CmnConnectedState 1 }
32
33 dot20CmnConnectionAttemptCounts OBJECT-TYPE
34     SYNTAX      Counter64
35     MAX-ACCESS  read-only
36     STATUS      current
37     DESCRIPTION
38      "Number of connection attempts"
39     ::= { dot20CmnConnectedState 2 }
40
41 dot20CmnConnectionFailureCounts OBJECT-TYPE
42     SYNTAX      Counter64
43     MAX-ACCESS  read-only
44     STATUS      current
45     DESCRIPTION
46      "Number of connection failures during connection attempt."
47     ::= { dot20CmnConnectedState 3 }
48
49 dot20CmnConnectionDropCounts OBJECT-TYPE
50     SYNTAX      Counter64
51     MAX-ACCESS  read-only
52     STATUS      current
53     DESCRIPTION
54      "Number of dropped connection after a connection has been
55      established."
56     ::= { dot20CmnConnectedState 4 }
57
58 dot20CmnConnectionReleaseCounts OBJECT-TYPE
59     SYNTAX      Counter64
60     MAX-ACCESS  read-only
61     STATUS      current
62     DESCRIPTION
63      "Number of connection release after a connection has been
64      established."
65     ::= { dot20CmnConnectedState 5 }
66
67 dot20CmnConvergence OBJECT IDENTIFIER ::= { dot20CmnMac 4 }
68

```

```

1 dot20CmnSignalingTransport OBJECT IDENTIFIER ::= { dot20CmnConvergence 1 }
2
3 dot20CmnSigTransportStatsTable OBJECT-TYPE
4     SYNTAX      SEQUENCE OF Dot20CmnSigTransportStatsEntry
5     MAX-ACCESS  not-accessible
6     STATUS      current
7     DESCRIPTION
8         "This table provides one row of Signaling Transport statistics
9         per 802.20 interface."
10    ::= { dot20CmnSignalingTransport 1 }
11
12 dot20CmnSigTransportStatsEntry OBJECT-TYPE
13     SYNTAX      Dot20CmnSigTransportStatsEntry
14     MAX-ACCESS  not-accessible
15     STATUS      current
16     DESCRIPTION
17         "An Entry (conceptual row) in the SigTransportStats table. This
18         table is indexed by IfIndex. ifIndex: Each IEEE 802.20
19         interface is represented by an ifEntry."
20     INDEX
21         { ifIndex }
22    ::= { dot20CmnSigTransportStatsTable 1 }
23
24 dot20CmnSlpTxBytes OBJECT-TYPE
25     SYNTAX      Counter64
26     MAX-ACCESS  read-only
27     STATUS      current
28     DESCRIPTION
29         "Number of SLP bytes transmitted"
30    ::= { dot20CmnSigTransportStatsEntry 1 }
31
32 dot20CmnSlpReTxBytes OBJECT-TYPE
33     SYNTAX      Counter64
34     MAX-ACCESS  read-only
35     STATUS      current
36     DESCRIPTION
37         "Number of SLP bytes retransmitted"
38    ::= { dot20CmnSigTransportStatsEntry 2 }
39
40 dot20CmnSlpTxDroppedBytes OBJECT-TYPE
41     SYNTAX      Counter64
42     MAX-ACCESS  read-only
43     STATUS      current
44     DESCRIPTION
45         "Number of SLP bytes dropped before transmission"
46    ::= { dot20CmnSigTransportStatsEntry 3 }
47
48 dot20CmnSlpTxStatus OBJECT-TYPE
49     SYNTAX      Counter64
50     MAX-ACCESS  read-only
51     STATUS      current
52     DESCRIPTION
53         "Number of ReceiverStatus messages transmitted"
54    ::= { dot20CmnSigTransportStatsEntry 4 }
55
56 dot20CmnSlpRxBytes OBJECT-TYPE
57     SYNTAX      Counter64
58     MAX-ACCESS  read-only
59     STATUS      current
60     DESCRIPTION
61         "Number of SLP Bytes received"
62    ::= { dot20CmnSigTransportStatsEntry 5 }
63
64 dot20CmnSlpRxStatus OBJECT-TYPE
65     SYNTAX      Counter64
66     MAX-ACCESS  read-only
67     STATUS      current
68     DESCRIPTION

```



```

1      "Number of ReceiverStatus messages received"
2      ::= { dot20CmnSigTransportStatsEntry 6 }
3
4  dot20CmnSlpTxPackets OBJECT-TYPE
5      SYNTAX      Counter64
6      MAX-ACCESS  read-only
7      STATUS      current
8      DESCRIPTION
9          "Number of SLP Packets transmitted"
10     ::= { dot20CmnSigTransportStatsEntry 7 }
11
12  dot20CmnSlpReTxPackets OBJECT-TYPE
13     SYNTAX      Counter64
14     MAX-ACCESS  read-only
15     STATUS      current
16     DESCRIPTION
17         "Number of SLP Packets retransmitted"
18     ::= { dot20CmnSigTransportStatsEntry 8 }
19
20  dot20CmnSlpTxDropPackets OBJECT-TYPE
21     SYNTAX      Counter64
22     MAX-ACCESS  read-only
23     STATUS      current
24     DESCRIPTION
25         "Number of SLP Packets dropped before transmission"
26     ::= { dot20CmnSigTransportStatsEntry 9 }
27
28  dot20CmnSlpRxPackets OBJECT-TYPE
29     SYNTAX      Counter64
30     MAX-ACCESS  read-only
31     STATUS      current
32     DESCRIPTION
33         "Number of SLP Packets received"
34     ::= { dot20CmnSigTransportStatsEntry 10 }
35
36  dot20CmnSlpTxACKTimeouts OBJECT-TYPE
37     SYNTAX      Counter64
38     MAX-ACCESS  read-only
39     STATUS      current
40     DESCRIPTION
41         "Number of ACK Timeouts"
42     ::= { dot20CmnSigTransportStatsEntry 11 }
43
44  dot20CmnDataTransport OBJECT IDENTIFIER ::= { dot20CmnConvergence 2 }
45
46  dot20CmnDataTransportStatsTable OBJECT-TYPE
47     SYNTAX      SEQUENCE OF Dot20CmnDataTransportStatsEntry
48     MAX-ACCESS  not-accessible
49     STATUS      current
50     DESCRIPTION
51         "This table provides one row of Data Transport statistics per
52         802.20 interface"
53     ::= { dot20CmnDataTransport 1 }
54
55  dot20CmnDataTransportStatsEntry OBJECT-TYPE
56     SYNTAX      Dot20CmnDataTransportStatsEntry
57     MAX-ACCESS  not-accessible
58     STATUS      current
59     DESCRIPTION
60         "An Entry (conceptual row) in the DataTransportStats table.
61         This table is indexed by IfIndex. ifIndex: Each IEEE 802.20
62         interface is represented by an ifEntry."
63     INDEX
64         { ifIndex }
65     ::= { dot20CmnDataTransportStatsTable 1 }
66
67  dot20CmnRlpTxBytes OBJECT-TYPE
68     SYNTAX      Counter64

```

```

1      MAX-ACCESS      read-only
2      STATUS          current
3      DESCRIPTION
4          "Number of RLP bytes of payload transmitted"
5      ::= { dot20CmnDataTransportStatsEntry 1 }
6
7      dot20CmnRlpReTxBytes OBJECT-TYPE
8          SYNTAX      Counter64
9          MAX-ACCESS  read-only
10         STATUS      current
11         DESCRIPTION
12             "Number of RLP bytes of payload retransmitted"
13         ::= { dot20CmnDataTransportStatsEntry 2 }
14
15         dot20CmnRlpTxDropBytes OBJECT-TYPE
16             SYNTAX      Counter64
17             MAX-ACCESS  read-only
18             STATUS      current
19             DESCRIPTION
20                 "Number of RLP bytes of dropped before transmission"
21             ::= { dot20CmnDataTransportStatsEntry 3 }
22
23         dot20CmnRlpTxStatus OBJECT-TYPE
24             SYNTAX      Counter64
25             MAX-ACCESS  read-only
26             STATUS      current
27             DESCRIPTION
28                 "Number of RLP ReceiverStatus messages transmitted"
29             ::= { dot20CmnDataTransportStatsEntry 4 }
30
31         dot20CmnRlpRxBytes OBJECT-TYPE
32             SYNTAX      Counter64
33             MAX-ACCESS  read-only
34             STATUS      current
35             DESCRIPTION
36                 "Number of RLP bytes of payload received"
37             ::= { dot20CmnDataTransportStatsEntry 5 }
38
39         dot20CmnRlpRxStatus OBJECT-TYPE
40             SYNTAX      Counter64
41             MAX-ACCESS  read-only
42             STATUS      current
43             DESCRIPTION
44                 "Number of RLP ReceiverStatus messages received"
45             ::= { dot20CmnDataTransportStatsEntry 6 }
46
47         dot20CmnRlpTxPackets OBJECT-TYPE
48             SYNTAX      Counter64
49             MAX-ACCESS  read-only
50             STATUS      current
51             DESCRIPTION
52                 "Number of RLP Packets transmitted"
53             ::= { dot20CmnDataTransportStatsEntry 7 }
54
55         dot20CmnRlpReTxPackets OBJECT-TYPE
56             SYNTAX      Counter64
57             MAX-ACCESS  read-only
58             STATUS      current
59             DESCRIPTION
60                 "Number of RLP Packets retransmitted"
61             ::= { dot20CmnDataTransportStatsEntry 8 }
62
63         dot20CmnRlpTxrDropPackets OBJECT-TYPE
64             SYNTAX      Counter64
65             MAX-ACCESS  read-only
66             STATUS      current
67             DESCRIPTION
68                 "Number of RLP Packets dropped before transmission"

```

```

1      ::= { dot20CmnDataTransportStatsEntry 9 }
2
3  dot20CmnRlpRxPackets OBJECT-TYPE
4      SYNTAX          Counter64
5      MAX-ACCESS      read-only
6      STATUS          current
7      DESCRIPTION
8          "Number of RLP Packets received"
9      ::= { dot20CmnDataTransportStatsEntry 10 }
10
11 dot20CmnRlpTxNAKTimeouts OBJECT-TYPE
12     SYNTAX          Counter64
13     MAX-ACCESS      read-only
14     STATUS          current
15     DESCRIPTION
16         "Number of NAK Timeouts"
17     ::= { dot20CmnDataTransportStatsEntry 11 }
18
19 dot20CmnActiveReservationsCounts OBJECT-TYPE
20     SYNTAX          Counter64
21     MAX-ACCESS      read-only
22     STATUS          current
23     DESCRIPTION
24         "Number of Active Reservations"
25     ::= { dot20CmnDataTransportStatsEntry 12 }
26
27 dot20CmnIdleReservationsCounts OBJECT-TYPE
28     SYNTAX          Counter64
29     MAX-ACCESS      read-only
30     STATUS          current
31     DESCRIPTION
32         "Number of Idle Reservations"
33     ::= { dot20CmnDataTransportStatsEntry 13 }
34
35 dot20CmnReservationOpenCounts OBJECT-TYPE
36     SYNTAX          Counter64
37     MAX-ACCESS      read-only
38     STATUS          current
39     DESCRIPTION
40         "Number of Reservations Open requests"
41     ::= { dot20CmnDataTransportStatsEntry 14 }
42
43 dot20CmnReservationCloseCounts OBJECT-TYPE
44     SYNTAX          Counter64
45     MAX-ACCESS      read-only
46     STATUS          current
47     DESCRIPTION
48         "Number of Reservations Close requests"
49     ::= { dot20CmnDataTransportStatsEntry 15 }
50
51 dot20CmnReservationFailCounts OBJECT-TYPE
52     SYNTAX          Counter64
53     MAX-ACCESS      read-only
54     STATUS          current
55     DESCRIPTION
56         "Number of Failed Reservations requests"
57     ::= { dot20CmnDataTransportStatsEntry 16 }
58
59 dot20CmnSecurity OBJECT IDENTIFIER ::= { dot20CmnMac 5 }
60
61 dot20CmnAuthProtocol OBJECT IDENTIFIER ::= { dot20CmnSecurity 1 }
62
63 dot20CmnAuthStatsTable OBJECT-TYPE
64     SYNTAX          SEQUENCE OF Dot20CmnAuthStatsEntry
65     MAX-ACCESS      not-accessible
66     STATUS          current
67     DESCRIPTION
68         "This table provides one row of Authentication protocol

```

```

1      statistics per 802.20 interface"
2      ::= { dot20CmnAuthProtocol 1 }
3
4  dot20CmnAuthStatsEntry OBJECT-TYPE
5      SYNTAX      Dot20CmnAuthStatsEntry
6      MAX-ACCESS  not-accessible
7      STATUS      current
8      DESCRIPTION
9          "An Entry (conceptual row) in the AuthStats table. This table
10         is indexed by IfIndex. ifIndex: Each IEEE 802.20 interface is
11         represented by an ifEntry."
12      INDEX
13          { ifIndex }
14      ::= { dot20CmnAuthStatsTable 1 }
15
16  dot20CmnAuthFailureCounts OBJECT-TYPE
17      SYNTAX      Counter64
18      MAX-ACCESS  read-only
19      STATUS      current
20      DESCRIPTION
21          "Number of Authentication failures"
22      ::= { dot20CmnAuthStatsEntry 1 }
23
24  dot20CmnAuthSuccessCounts OBJECT-TYPE
25      SYNTAX      Counter64
26      MAX-ACCESS  read-only
27      STATUS      current
28      DESCRIPTION
29          "Number of successful Authentications"
30      ::= { dot20CmnAuthStatsEntry 2 }
31
32  dot20CmnLowerMAC OBJECT IDENTIFIER ::= { dot20CmnMac 6 }
33
34  dot20CmnLMACPacketStatsTable OBJECT-TYPE
35      SYNTAX      SEQUENCE OF Dot20CmnLMACPacketStatsEntry
36      MAX-ACCESS  not-accessible
37      STATUS      current
38      DESCRIPTION
39          "This table provides one row of Lower MAC protocol statistics
40         per 802.20 interface, packet format and nb of ARQ attempts
41         needed in order to successfully transmit/receive a packet."
42      ::= { dot20CmnLowerMAC 1 }
43
44  dot20CmnLMACPacketStatsEntry OBJECT-TYPE
45      SYNTAX      Dot20CmnLMACPacketStatsEntry
46      MAX-ACCESS  not-accessible
47      STATUS      current
48      DESCRIPTION
49          "An Entry (conceptual row) in the LMACPacketStats table. This
50         table is indexed by IfIndex, PacketFormatIndex and
51         ARQAttemptsIndex."
52      INDEX
53          { ifIndex, dot20CmnPacketFormatIndex, dot20CmnARQAttemptsIndex }
54      ::= { dot20CmnLMACPacketStatsTable 1 }
55
56
57  dot20CmnPacketFormatIndex OBJECT-TYPE
58      SYNTAX      Integer32 (0..15)
59      MAX-ACCESS  not-accessible
60      STATUS      current
61      DESCRIPTION
62          "The packet format index as defined in 802.20 AIS spec."
63      ::= { dot20CmnLMACPacketStatsEntry 1 }
64
65  dot20CmnARQAttemptsIndex OBJECT-TYPE
66      SYNTAX      Integer32 (0..15)
67      MAX-ACCESS  not-accessible
68      STATUS      current

```

```

1      DESCRIPTION
2          "Number of ARQ attempts that were needed in order to transmit
3          or receive a packet. Index 0 means that the packets failed to
4          be transmitted/received."
5      ::= { dot20CmnLMACPacketStatsEntry 2 }
6
7      dot20CmnFwdTxPacketCounts OBJECT-TYPE
8          SYNTAX      Counter64
9          MAX-ACCESS  read-only
10         STATUS      current
11         DESCRIPTION
12             "Number of transmitted packets"
13         ::= { dot20CmnLMACPacketStatsEntry 3 }
14
15         dot20CmnRevRxPacketCounts OBJECT-TYPE
16             SYNTAX      Counter64
17             MAX-ACCESS  read-only
18             STATUS      current
19             DESCRIPTION
20                 "Number of received packets"
21             ::= { dot20CmnLMACPacketStatsEntry 4 }
22
23         dot20CmnLMACStatsTable OBJECT-TYPE
24             SYNTAX      SEQUENCE OF Dot20CmnLMACStatsEntry
25             MAX-ACCESS  not-accessible
26             STATUS      current
27             DESCRIPTION
28                 "This table provides one row of Lower MAC protocol statistics
29                 per 802.20 interface and packet formats."
30             ::= { dot20CmnLowerMAC 2 }
31
32         dot20CmnLMACStatsEntry OBJECT-TYPE
33             SYNTAX      Dot20CmnLMACStatsEntry
34             MAX-ACCESS  not-accessible
35             STATUS      current
36             DESCRIPTION
37                 "An Entry (conceptual row) in the LMACStats table. This table
38                 is indexed by IfIndex, PacketFormatIndex."
39             INDEX
40                 { ifIndex, dot20CmnPacketFormatIndex }
41             ::= { dot20CmnLMACStatsTable 1 }
42
43         dot20CmnFLABCounts OBJECT-TYPE
44             SYNTAX      Counter64
45             MAX-ACCESS  read-only
46             STATUS      current
47             DESCRIPTION
48                 "Number of FL SSCH assignments"
49             ::= { dot20CmnLMACStatsEntry 1 }
50
51         dot20CmnRLABCounts OBJECT-TYPE
52             SYNTAX      Counter64
53             MAX-ACCESS  read-only
54             STATUS      current
55             DESCRIPTION
56                 "Number of RL SSCH assignments"
57             ::= { dot20CmnLMACStatsEntry 2 }
58
59         dot20CmnAccessGrantCounts OBJECT-TYPE
60             SYNTAX      Counter64
61             MAX-ACCESS  read-only
62             STATUS      current
63             DESCRIPTION
64                 "Number of Access Grants"
65             ::= { dot20CmnLMACStatsEntry 3 }
66
67         dot20Conformance OBJECT IDENTIFIER ::= { ieee802dot20 4 }
68

```

```

1 dot20Groups OBJECT IDENTIFIER ::= { dot20Conformance 1 }
2
3 dot20CmnSessionMgtPGroup OBJECT-GROUP
4   OBJECTS
5     { dot20CmnSessionCloseCounts, dot20CmnSessionFailureCounts,
6       dot20CmnSessionOpenCounts }
7   STATUS      current
8   DESCRIPTION
9     "The session management protocol statistics"
10  ::= { dot20Groups 1 }
11
12 dot20CmnSessionConfigPGroup OBJECT-GROUP
13   OBJECTS
14     { dot20CmnSessionConfToken }
15   STATUS      current
16   DESCRIPTION
17     "The session configuration protocol configuration"
18  ::= { dot20Groups 2 }
19
20 dot20CmnSessionConfigPGroup2 OBJECT-GROUP
21   OBJECTS
22     { dot20CmnSessionConfTokenStatus }
23   STATUS      current
24   DESCRIPTION
25     "This group should be implemented if assignment of tokens is
26     performed through snmp."
27  ::= { dot20Groups 3 }
28
29 dot20CmnKeyExchangePGroup OBJECT-GROUP
30   OBJECTS
31     { dot20CmnKeyExchangeAttemptCounts,
32       dot20CmnKeyExchangeFailureCounts }
33   STATUS      current
34   DESCRIPTION
35     "The key exchange protocol statistics"
36  ::= { dot20Groups 4 }
37
38 dot20CmnConnectedStatePGroup OBJECT-GROUP
39   OBJECTS
40     { dot20CmnActiveConnectionCounts,
41       dot20CmnConnectionAttemptCounts, dot20CmnConnectionDropCounts,
42       dot20CmnConnectionFailureCounts, dot20CmnConnectionReleaseCounts
43     }
44   STATUS      current
45   DESCRIPTION
46     "The connected state protocol statistics"
47  ::= { dot20Groups 5 }
48
49 dot20CmnSigTransportGroup OBJECT-GROUP
50   OBJECTS
51     { dot20CmnSlpReTxBytes, dot20CmnSlpReTxPackets,
52       dot20CmnSlpRxBytes, dot20CmnSlpRxPackets, dot20CmnSlpRxStatus,
53       dot20CmnSlpTxACKTimeouts, dot20CmnSlpTxBytes,
54       dot20CmnSlpTxDropPackets, dot20CmnSlpTxDroppedBytes,
55       dot20CmnSlpTxPackets, dot20CmnSlpTxStatus }
56   STATUS      current
57   DESCRIPTION
58     "The signaling transport statistics"
59  ::= { dot20Groups 6 }
60
61 dot20CmnDataTransportGroup OBJECT-GROUP
62   OBJECTS
63     { dot20CmnActiveReservationsCounts,
64       dot20CmnIdleReservationsCounts, dot20CmnReservationCloseCounts,
65       dot20CmnReservationFailCounts, dot20CmnReservationOpenCounts,
66       dot20CmnRevRxPacketCounts, dot20CmnRlpReTxPackets,
67       dot20CmnRlpReTxBytes, dot20CmnRlpRxPackets, dot20CmnRlpRxBytes,
68       dot20CmnRlpRxStatus, dot20CmnRlpTxNAKTimeouts,

```

```

1         dot20CmnRlpTxPackets, dot20CmnRlpTxBytes,
2         dot20CmnRlpTxDropBytes, dot20CmnRlpTxStatus,
3         dot20CmnRlpTxrDropPackets }
4     STATUS         current
5     DESCRIPTION
6         "The data transport statistics"
7     ::= { dot20Groups 7 }
8
9     dot20CmnAuthGroup OBJECT-GROUP
10    OBJECTS
11        { dot20CmnAuthFailureCounts, dot20CmnAuthSuccessCounts }
12    STATUS         current
13    DESCRIPTION
14        "The authentication protocol statistics"
15    ::= { dot20Groups 8 }
16
17    dot20CmnLowerMACGroup OBJECT-GROUP
18    OBJECTS
19        { dot20CmnAccessGrantCounts, dot20CmnFLABCounts,
20          dot20CmnFwdTxPacketCounts, dot20CmnRLABCounts,
21          dot20CmnRevRxPacketCounts }
22    STATUS         current
23    DESCRIPTION
24        "The lower mac sublayer statistics"
25    ::= { dot20Groups 9 }
26
27    dot20AnIdleStatePGroup OBJECT-GROUP
28    OBJECTS
29        { dot20AnAccessAttemptCounts, dot20AnAccessAttemptFailCounts,
30          dot20AnPageAttemptCounts, dot20AnPageFailureCounts }
31    STATUS         current
32    DESCRIPTION
33        "The An idle state protocol statistics"
34    ::= { dot20Groups 10 }
35
36    dot20AnOverheadGroup OBJECT-GROUP
37    OBJECTS
38        { dot20AnAccessCycleDuration, dot20AnAccessRetryPersistence0,
39          dot20AnAccessRetryPersistence1, dot20AnAccessRetryPersistence2,
40          dot20AnAccessRetryPersistence3, dot20AnAccessSequencePartition,
41          dot20AnAuxPilotPower, dot20AnBFCHBeamCodeBookIndex,
42          dot20AnBFCHPowerOffset, dot20AnBlockHoppingEnabled,
43          dot20AnCPLength, dot20AnCarrierID, dot20AnChannelBandClass,
44          dot20AnChannelFwdLinkCenterFreq, dot20AnChannelFwdLinkSystemBw,
45          dot20AnChannelNumber, dot20AnChannelBandRecordType,
46          dot20AnChannelBandRef, dot20AnChannelRevLinkCenterFreq,
47          dot20AnChannelRevLinkSystemBw, dot20AnCommonPilotPower,
48          dot20AnCountryCode, dot20AnCtrlAccessOffset,
49          dot20AnEffectiveNumAntennas, dot20AnEnableAuxPilotStaggering,
50          dot20AnEnableCmnPilotStaggering, dot20AnErasureGain0,
51          dot20AnErasureGain1, dot20AnErasureGain2, dot20AnErasureGain3,
52          dot20AnFLChannelTreeIndex, dot20AnFLDPISectorOffset,
53          dot20AnFLDPISectorScramble, dot20AnFLDiversityHoppingMode,
54          dot20AnFLFirstRestrSetSubband, dot20AnFLIntraCellCommonHopping,
55          dot20AnFLNumRestrSetSubbands, dot20AnFLNumSDMADimensions,
56          dot20AnFLNumSubbands, dot20AnFLPCReportInterval,
57          dot20AnFLReservedInterlaces, dot20AnFLSectorHopSeed,
58          dot20AnFastOSIEnabled, dot20AnHalfDuplexModeSupported,
59          dot20AnIfChannelBandRef, dot20AnLatitude, dot20AnLeapSeconds,
60          dot20AnLocalTimeOffset, dot20AnLongitude, dot20AnMACIDRange,
61          dot20AnMaxProbesPerSequence, dot20AnMaximumRevision,
62          dot20AnMinimumRevision, dot20AnNFLBurst, dot20AnNRLBurst,
63          dot20AnNeighborCarrierID, dot20AnNeighborCarrierPointer,
64          dot20AnNeighborPilotPN, dot20AnNeighborSectorID,
65          dot20AnNeighborTransmitPower, dot20AnNghbrSynchGroup,
66          dot20AnNghbrGloballySynch, dot20AnNumCarriers,
67          dot20AnNumCmnPilotTxAntennas, dot20AnNumFLReservedSubbands,
68          dot20AnNumGuardSubcarriers, dot20AnNumPilots,

```

```

1      dot20AnNumRLControlSubbands, dot20AnRACKBandwidthFactor,
2      dot20AnOpenLoopAdjust, dot20AnPICHPowerOffset,
3      dot20AnPhysicalMode, dot20AnPilotPN, dot20AnPilotThreshold1,
4      dot20AnPilotThreshold2, dot20AnPrNumGuardSubcarriers,
5      dot20AnPreamblePilotPower, dot20AnProbeRampUpStepSize,
6      dot20AnRDCHInitialPacketFormat, dot20AnRLChannelTreeIndex,
7      dot20AnRLCtrlPCMode, dot20AnRLDPISectorOffset,
8      dot20AnRLDPISectorScramble, dot20AnRLDiversityHoppingMode,
9      dot20AnRLIntraCellCommonHopping, dot20AnRLNumSDMADimensions,
10     dot20AnRLNumSubbands, dot20AnRLRestrictedSetBitmap,
11     dot20AnRLSectorHopSeed, dot20AnRPICHEnabled,
12     dot20AnRegistrationRadius, dot20AnRegistrationZoneCode,
13     dot20AnRegistrationZoneIncluded, dot20AnRegistrationZoneMaxAge,
14     dot20AnReqChannelGain0, dot20AnReqChannelGain1,
15     dot20AnReqChannelGain2, dot20AnReqChannelGain3,
16     dot20AnRevLinkSilenceDuration, dot20AnRevLinkSilencePeriod,
17     dot20AnSFCHPowerOffset, dot20AnSSCHModSymbolsPerBlock,
18     dot20AnSSCHNumBlocks, dot20AnSSCHNumHopports, dot20AnSectorID,
19     dot20AnSubnetMask, dot20AnSynchGroup, dot20AnSynchronousSystem,
20     dot20AnTechNghbrListLength, dot20AnTechnologyNeighborList,
21     dot20AnTechnologyType, dot20AnTransmitPower }
22     STATUS          current
23     DESCRIPTION
24         "The overhead messages protocol configuration"
25     ::= { dot20Groups 11 }
26
27 dot20AnOverheadGroup2 OBJECT-GROUP
28     OBJECTS
29         { dot20AnCarrierConfigRowStatus, dot20AnChannelBandStatus,
30         dot20AnNeighborCarrierStatus, dot20AnNeighborRowStatus,
31         dot20AnOtherTechNghbrRowStatus, dot20AnSectorConfigRowStatus }
32     STATUS          current
33     DESCRIPTION
34         "The overhead messages protocol configuration This group should
35         be implemented if assignment of 802.20 interfaces is to be
36         preformed through snmp."
37     ::= { dot20Groups 12 }
38
39 dot20Compliances OBJECT IDENTIFIER ::= { dot20Conformance 2 }
40
41 dot20AnCompliance MODULE-COMPLIANCE
42     STATUS          current
43     DESCRIPTION
44         "The compliance statement for SNMPv2 entities that implement
45         the IEEE 802.20 MIB for the An."
46     MODULE          IEEE802dot20-MIB
47         MANDATORY-GROUPS
48             { dot20AnIdleStatePGroup, dot20AnOverheadGroup,
49             dot20CmnAuthGroup, dot20CmnConnectedStatePGroup,
50             dot20CmnDataTransportGroup, dot20CmnKeyExchangePGroup,
51             dot20CmnLowerMACGroup, dot20CmnSessionConfigPGroup,
52             dot20CmnSessionMgtPGroup, dot20CmnSigTransportGroup }
53         GROUP          dot20AnOverheadGroup2
54         DESCRIPTION
55             "This group is required only if 'dynamic assignment' of
56             rows in the OverheadGroup tables is supported."
57         GROUP          dot20CmnSessionConfigPGroup2
58         DESCRIPTION
59             "This group is only implemented if 'dynamic assignment' of
60             rows in the SessionConfigP group tables is supported."
61     ::= { dot20Compliances 1 }
62
63     END

```



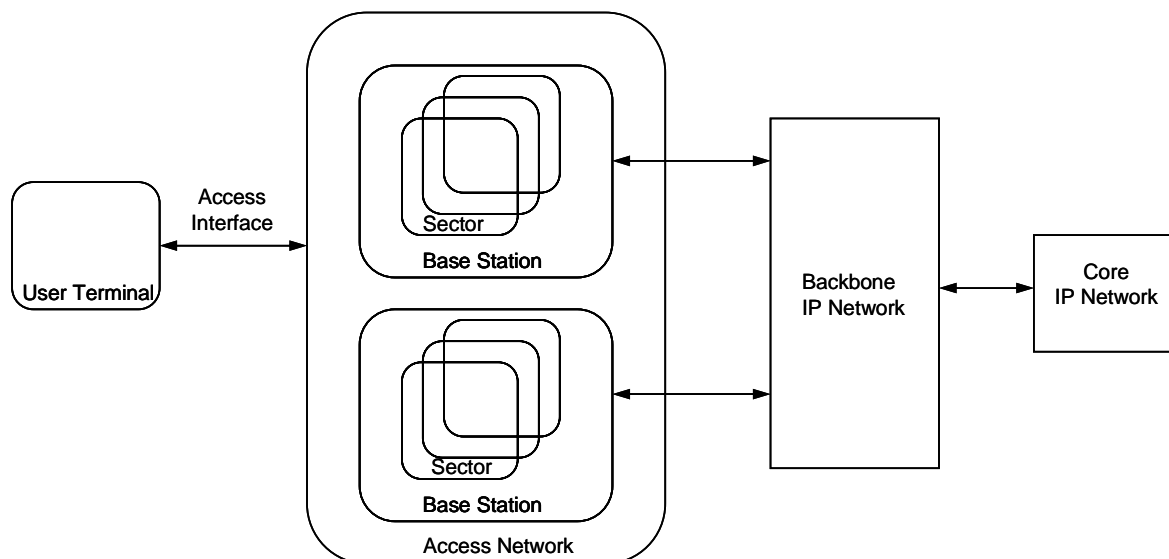
## 14 System Overview of 625k-MC (625kiloHertz-spaced MultiCarrier) Mode

### 14.1 Scope

This specification defines the physical layer, medium access control, and the signaling layer for 625k-MC, see 1.1 of ATIS-PP-0700004-2005, High Capacity-Spatial Division Multiple Access (HC-SDMA), September 2005. For brevity, references to the baseline specification: ATIS-PP-0700004-2005, High Capacity-Spatial Division Multiple Access (HC-SDMA), September 2005 will be denoted as “HC-SDMA [25]” in this document. Unless otherwise specified in this document, the requirements of HC-SDMA [25] shall apply to the 625k-MC mode of 802.20.

### 14.2 Architecture Reference Model

The architecture reference model for 625k-MC Mode is presented in Figure 121. The reference model includes the air interface between the user terminal and the access network. While the protocols specified in HC-SDMA [25] are referred to as basic protocols, this document defines only the additional specifications that are needed to supplement the protocols, and are referred to as enhanced specifications with reference to Base-Draft Specifications of HC-SDMA [25].



**Figure 121 Architecture reference model**

The functional units of the reference architecture in Figure 121 are:

**Access Network (AN)** The network equipment providing IP layer connectivity between an IP network (typically the Internet) and the user terminals.

**Base Station (BS)** The device in the access network that communicates over the air interface, via one or more sectors, with the user terminals. Base Stations coordinate the management of the air interface attributes.

- 1 User Terminal (UT) A device providing data connectivity to an end user device (EUD) user. A  
 2 user terminal may be connected to a computing device such as a laptop  
 3 personal computer or it may be a self-contained data device such as a  
 4 personal digital assistant.
- 5 Sector One set of physical layer channels transmitted between base station and the  
 6 user terminals within a given frequency assignment. A sector consists of a  
 7 reverse link radio channel and a forward link radio channel.

### 8 **14.3 Acronyms**

9 *Acronyms as specified in Section 1.2 of HC-SDMA [25] with additional acronyms as underlined*  
 10 *below:*

11 <b><u>BCMCS</u></b>	<b><u>Broadcast and Multicast Services</u></b>
<b><u>SMB</u></b>	<b><u>Short Message Broadcast</u></b>
<b><u>VoIP</u></b>	<b><u>Voice over Internet Protocol</u></b>

### 12 **14.4 Conventions**

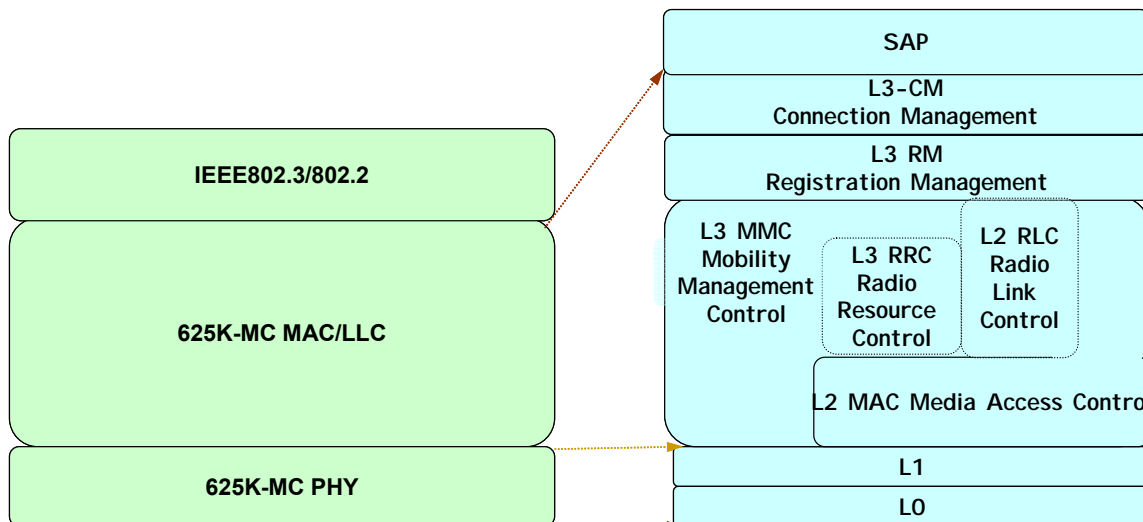
13 The terminologies and conventions used in this specification shall be as specified in 1.3 of  
 14 HC-SDMA [25].

### 15 **14.5 625k-MC Application Overview**

16 The 625k-MC application overview is as specified in 1.4 of HC-SDMA [25].

## 14.6 625k-MC Protocol Overview

625k-MC mode's RF/PHY/MAC/LLC specifications are based on the L0/L1/L2/L3 specifications defined in HC-SDMA [25]. The 625k-MC PHY/MAC/LLC Protocol Reference model is as shown in Figure 122.



**Figure 122 625k-MC PHY/MAC/LLC Protocol Reference**

The 625k-MC Physical layer (PHY) specifications are based on the definitions of L1 (Layer 1) and L0 (Layer 0) protocols of HC-SDMA [25]. The 625k-MC Medium Access Control (MAC) specifications are based on the definitions of the L2-MAC (Layer 2-MAC) protocol of HC-SDMA [25]. The 625k-MC Logical Link Control (LLC) specifications are based on the definitions of L2-RLC (Layer 2-Radio Link Control), Layer L3 (L3-RRC, L3-MMC, L3-CM, L3-RM) of HC-SDMA [25].

### 14.6.1 625k-MC Protocol Features

The basic protocol features of 625k-MC shall be as defined in 1.5.1 of HC-SDMA [25].

### 14.6.2 625k-MC Protocol Reference Model and Interfaces

The 625k-MC protocol reference model and communication interface between protocol layers shall be as defined in 1.6 of HC-SDMA [25].

1 **15 625k-MC Spectral Layout Terminology and Requirements**

2 The 625k-MC spectral layout terminology and requirements shall be as defined in Chapter 2 of  
3 HC-SDMA [25].

## 1 **16 625k-MC Slot and Frame Structure**

2 The 625k-MC mode MAC frame and slot structures for each “carrier” whose RF channel bandwidth  
3 is 625 kHz, are specified in this chapter. The RF carrier frequency in the carrier allocation shall be a  
4 consecutive set of frequencies separated by 625 kHz.

### 5 **16.1 Overview**

6 As defined in 3.1 of HC-SDMA [25].

### 7 **16.2 RF Channel and Frame Structure**

8 As defined in 3.2 of HC-SDMA [25].

### 9 **16.3 Burst Formats**

#### 10 **16.3.1 Frequency Synchronization**

11 As defined in 3.3.1 of HC-SDMA [25].

#### 12 **16.3.2 Timing Synchronization**

13 As defined in 3.3.2 of HC-SDMA [25].

#### 14 **16.3.3 Broadcast Burst**

15 As defined in 3.3.3 of HC-SDMA [25].

#### 16 **16.3.4 Page Burst**

17 As defined in 3.3.4 of HC-SDMA [25].

#### 18 **16.3.5 Configuration Request Burst**

19 As defined in 3.3.5 of HC-SDMA [25].

#### 20 **16.3.6 Standard Uplink Burst (FACCH, RACH& TCH)**

21 As defined in 3.3.6 of HC-SDMA [25].

#### 22 **16.3.7 Standard Downlink Burst (CM, SMB, AA & TCH)**

23 As defined in 3.3.7 of HC-SDMA [25] with *the following underlined text included before the end of*  
24 *this section:*

25 The 625k-MC Short Message Broadcast (SMB) burst shall adhere to the standard downlink burst  
26 format.

## 17 625k-MC Modulation and Channel Coding

### 17.1 625k-MC Modulation and Channel Coding Overview

Chapter 4 of HC-SDMA [25] gives the baseline capabilities for 625k-MC Modulation and channel coding. The capability therein described is further enhanced by the requirements of this section as underlined below:

625k-MC supports 11 Modulation Classes (ModClasses). ModClasses 0-8 for the downlink and ModClasses 0-7 for uplink shall be as specified in the 4.2 of HC-SDMA [25]. Table 122 shows the updated Table 4.1 of HC-SDMA [25] and adds ModClasses 9-10 for the downlink and 8-10 for the uplink.

**Table 122 User data throughput at various ModClasses**

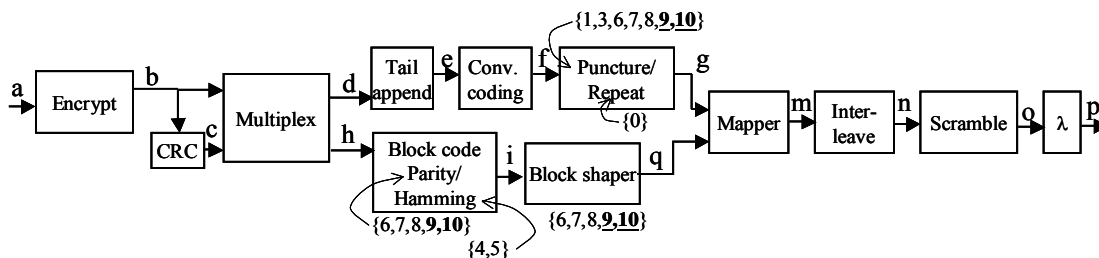
ModClass	Single stream downlink throughput (kbps) <u>per carrier</u>	Aggregated 3 stream downlink throughput (kbps) <u>per carrier</u>	Single stream uplink throughput (kbps) <u>per carrier</u>	Aggregated 3 stream uplink throughput (kbps) <u>per carrier</u>
0	35.2	105.6	6.4	19.2
1	49.6	148.8	12.8	38.4
2	81.6	244.8	25.6	76.8
3	126.4	379.2	43.2	129.6
4	161.6	484.8	57.6	172.8
5	198.4	595.2	72.0	216.0
6	262.4	787.2	97.6	292.8
7	307.2	921.6	115.2	345.6
8	353.6	1060.8	<u>132.8</u>	<u>398.4</u>
9	<u>377.6</u>	<u>1132.8</u>	<u>142.4</u>	<u>427.2</u>
<u>10</u>	<u>497.6</u>	<u>1492.8</u>	<u>190.4</u>	<u>571.2</u>

11

## 17.2 Standard Modulation and Coding

As specified in Section 4.2 of HC-SDMA[25] *with the additional text and tables as underlined below*.

The 625k-MC physical layer supports different data rates by selecting among various coding and modulation schemes. Figure 123 illustrates the coded modulation system that achieves rates from approximately  $\frac{1}{2}$  to 5.5 bits/symbol. Table 123 lists the modulation and signal sets together with the associated parameters for puncturing, shaping and block coding for ModClasses 9 and 10, in addition to those for modClasses 0-8 as shown in Table 4.2 of HC-SDMA[25].



**Figure 123** Block diagram for error control coding scheme. The notation  $\{i, j\}$  indicates that a block is active only for modulation classes  $i$  and  $j$ .

Coding is provided by a rate-1/2 convolutional code with 256 states combined in some cases with a block code. In some ModClasses, periodic puncturing is applied to increase the rate of the convolutional code to  $\frac{2}{3}$ ,  $\frac{3}{4}$  or  $\frac{5}{6}$ . Repetition is used in ModClass 0 to construct RA, AA, CM and SMB bursts (and is not used in ModClass 0 to construct CR and TCH bursts). ModClasses 6, 7 and 8 employ four-dimensional block shaping to generate 12-, 16- and 24-QAM signal set respectively. ModClasses 9 and 10 employ block shaping to generate 32-, and 64-QAM, respectively.

The operations of each block in Figure 123 are detailed in the subsequent sections. In the descriptions that follow,  $\{a_1, a_2, \dots, a_{N_a}\}$  denotes the sequence of  $N_a$  bits input to the encrypt block,  $\{b_1, b_2, \dots, b_{N_b}\}$  denotes the  $N_b$  bits input to the CRC block. The block sizes  $\{N_{a_1}, N_{b_2}, \dots, N_{q,j}\}$  are listed in Table 125 and Table 126. The block sizes for CR, CM, RA and AA are as shown in Table 4.6 to 4.9 of HC-SDMA [25]. The block sizes for SMB are as shown in Table 127.

**Table 123** Modulation and Coding Rates

ModClass	Bits/Sym	Signal Set	Puncture	Shaper	Block Code
0	0.5	BPSK	Repeat	-	-
1	0.67	BPSK	1 of 4	-	-
2	1	QPSK	-	-	-
3	1.5	QPSK	2 of 6	-	-
4	2	8-PSK	-	-	(64,57)
5	2.5	8-PSK	-	-	(64,57)
6	3	12-QAM	2 of 6	3/4	(48,47)
7	3.5	16-QAM	2 of 6	4/4	(64,63)
8	4	24-QAM	2 of 6	5/4	(80,79)

ModClass	Bits/Sym	Signal Set	Puncture	Shaper	Block Code
<u>9</u>	<u>4.5</u>	<u>32-QAM</u>	<u>2 of 6</u>	<u>5/5</u>	<u>(80,79)</u>
<u>10</u>	<u>5.5</u>	<u>64-QAM</u>	<u>2 of 5</u>	<u>6/6</u>	<u>(80,79)</u>
<u>11-15</u>	<u>RESERVED</u>				

1

2

**Table 124 ModClass versus Burst Type**

Logical Channel	Burst Type	ModClass
<u>TCH-uplink</u>	<u>Standard uplink</u>	<u>0-10</u>
<u>TCH-downlink</u>	<u>Standard downlink</u>	<u>0-10</u>
RACH-uplink	Standard uplink	0
RACH-downlink	Standard downlink	0
CCH-uplink	Configuration Request	0
CCH-downlink	Standard downlink	0
BCH and PCH	Broadcast and Page	See Section 4.3 Broadcast channel Modulation and Coding on page 4.22 of HC-SDMA [25]

3

4

**Table 125 Block Lengths in Downlink Traffic Burst**

ModClass	Nab	Nc	Nd	Ne	Nf	Ng	Nh	Ni	Nq	Nm	Nmnop
0	206	16	222	230	460	460	0	0	0	460	460
1	282	16	298	306	613	460	0	0	0	460	460
2	436	16	452	460	920	920	0	0	0	920	460
3	666	16	682	690	1380	920	0	0	0	920	460
4	840	16	452	460	920	920	404	460	460	460	460
5	1021	16	222	230	460	460	815	920	920	460	460
6	1341	16	682	690	1380	920	675	690	920	920	460
7	1571	16	682	690	1380	920	905	920	920	920	460
8	1801	16	682	690	1380	920	1135	1150	920	920	460
<u>9</u>	<u>1919</u>	<u>16</u>	<u>1027</u>	<u>1035</u>	<u>2070</u>	<u>1380</u>	<u>908</u>	<u>920</u>	<u>920</u>	<u>920</u>	<u>460</u>
<u>10</u>	<u>2523</u>	<u>16</u>	<u>950</u>	<u>958</u>	<u>1916</u>	<u>1150</u>	<u>1589</u>	<u>1610</u>	<u>1610</u>	<u>920</u>	<u>460</u>

5



1 **Table 126 Block Lengths in Uplink Traffic Burst**

ModClass	Nab	Nc	Nd	Ne	Nf	Ng	Nh	Ni	Nq	Nm	Nmnop
0	67	16	83	91	182	182	0	0	0	182	182
1	67	16	113	121	182	182	0	0	0	182	182
2	158	16	174	182	364	364	0	0	0	364	182
3	249	16	265	273	364	364	0	0	0	364	182
4	319	16	174	182	364	364	161	182	182	182	182
5	389	16	83	91	182	182	322	364	364	182	182
6	516	16	265	273	364	364	267	273	364	364	182
7	607	16	265	273	364	364	358	364	364	364	182
<u>8</u>	<u>698</u>	<u>16</u>	<u>265</u>	<u>273</u>	<u>546</u>	<u>364</u>	<u>449</u>	<u>455</u>	<u>364</u>	<u>364</u>	<u>182</u>
<u>9</u>	<u>744</u>	<u>16</u>	<u>401</u>	<u>409</u>	<u>819</u>	<u>546</u>	<u>359</u>	<u>364</u>	<u>364</u>	<u>364</u>	<u>182</u>
<u>10</u>	<u>984</u>	<u>16</u>	<u>371</u>	<u>379</u>	<u>758</u>	<u>455</u>	<u>629</u>	<u>937</u>	<u>637</u>	<u>364</u>	<u>182</u>

2  
3 **Table 127 Block Lengths in Short Message Broadcast**

<u>ModClass</u>	<u>Nab</u>	<u>Nc</u>	<u>Nd</u>	<u>Ne</u>	<u>Nf</u>	<u>Ng</u>	<u>Nh</u>	<u>Ni</u>	<u>Nq</u>	<u>Nm</u>	<u>Nmnop</u>
<u>0</u>	<u>105</u>	<u>16</u>	<u>121</u>	<u>129</u>	<u>258</u>	<u>460</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>460</u>	<u>460</u>

4  
5 **17.2.1 Encryption**

6 As defined in 4.2.1 of HC-SDMA [25].

7 **17.2.2 Cyclic redundancy check**

8 As defined in 4.2.2 of HC-SDMA [25].

9 **17.2.3 Multiplexing**

10 As defined in 4.2.3 of HC-SDMA [25].

11 **17.2.4 Tail append**

12 As defined in 4.2.4 of HC-SDMA [25].

13 **17.2.5 Convolutional Encoding**

14 As defined in 4.2.5 of HC-SDMA [25].

## 17.2.6 Puncturing and repeating

Puncturing and repeating for modClasses 0-8 is as specified in 4.2.6 of HC-SDMA[25] *with the following underlined text included at the end of the section:*

- For modClass 9, the coded outputs are punctured by a periodic puncturing pattern that deletes two bits from every block of six as that for modClass 3, 5, 7 and 8.
- For modulation class 10, the coded outputs are punctured by a periodic puncturing pattern that deletes two bits from every block of five.

<b>Input</b>	$f_1$ $f_3$ $f_4$	$f_6$ $f_8$ $f_9$	$f_{11}$ $f_{13}$ $f_{14}$	...
<b>Output</b>	$g_1$ $g_2$ $g_3$	$g_4$ $g_5$ $g_6$	$g_7$ $g_8$ $g_9$	...

## 17.2.7 Block Coding

As defined in 4.2.7 of HC-SDMA [25].

### 17.2.7.1 Extended Hamming Code

As defined in 4.2.7.1 of HC-SDMA [25].

### 17.2.7.2 Parity check Code

The parity check code for ModClasses 6, 7 and 8 shall be as defined in 4.2.7.2 of HC-SDMA [25] *with the following underlined text included at the end of the section:*

The parity check code specified in 4.2.7.2 of HC-SDMA [25] for ModClass 8 with input block size 79 shall be used for ModClasses 9 and 10.

## 17.2.8 Block Shaper

ModClasses 6-10 employ block shaping. The block shaper processes a binary-valued input sequence into a ternary-valued output sequence. The block shaping for ModClasses 6, 7 and 8 are as defined in 4.2.8 of HC-SDMA [25]. *The following underlined text and tables are included at the end of the section:*

The input sequence is divided into blocks of size B=5 and 6 bits for ModClasses 9 and 10 respectively. ModClasses 9 and 10 use 5/5 and 6/6 block shaper as shown in Table 128 and Table 129, respectively.

**Table 128 Mapping for Rate 5/5 Block shaper**

$i_{1+5l}$ $i_{2+5l}$ $i_{3+5l}$ $i_{4+5l}$ $i_{5+5l}$	$q_{1+5l}$ $q_{i_{2+5l}}$ $q_{3+5l}$ $q_{i_{4+5l}}$ $q_{i_{5+5l}}$
$x_1$ $x_2$ $x_3$ $x_4$ $x_5$	$x_1$ $x_2$ $x_3$ $x_4$ $x_5$

**Table 129 Mapping for Rate 6/6 Block shaper**

$i_{1+6l}$ , $i_{2+6l}$ , $i_{3+6l}$ , $i_{4+6l}$ , $i_{5+6l}$ , $i_{5+6l}$	$q_{1+6l}$ , $q_{2+6l}$ , $q_{3+6l}$ , $q_{4+6l}$ , $q_{5+6l}$ , $q_{5+6l}$
$X_1$ , $X_2$ , $X_3$ , $X_4$ , $X_5$ , $X_6$	$X_1$ , $X_2$ , $X_3$ , $X_4$ , $X_5$ , $X_6$

**17.2.9 Symbol Mapping**

Symbol mapping for ModClasses 2, 3, 6, 7, and 8 shall be as defined in 4.2.9 of HC-SDMA [25]. *The following underlined text and tables are included at the end of the section 4.2.9 of HC-SDMA [25]:*

For modclasses 9 and 10, pairs of outputs from the lookup table are multiplexed into a single complex output symbol as specified in 4.2.9 of HC-SDMA [25]. The symbol mapper for modclass 9 is shown in Table 130.

**Table 130 Symbol mapper for Modulation classes 9**

$q_{2k-1}$	$q_{2k}$	$q_{3k-2}$	$q_{3k-1}$	$q_{3k}$	$\tilde{m}$	
					$2k-1$	$2k$
<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>-3</u>	<u>5</u>
<u>0</u>	<u>1</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>-1</u>	<u>5</u>
<u>1</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>-5</u>	<u>3</u>
<u>1</u>	<u>1</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>-5</u>	<u>1</u>
<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>1</u>	<u>-3</u>	<u>3</u>
<u>0</u>	<u>1</u>	<u>0</u>	<u>0</u>	<u>1</u>	<u>-1</u>	<u>3</u>
<u>1</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>1</u>	<u>-3</u>	<u>1</u>
<u>1</u>	<u>1</u>	<u>0</u>	<u>0</u>	<u>1</u>	<u>-1</u>	<u>1</u>
<u>0</u>	<u>0</u>	<u>0</u>	<u>1</u>	<u>0</u>	<u>-3</u>	<u>-5</u>
<u>0</u>	<u>1</u>	<u>0</u>	<u>1</u>	<u>0</u>	<u>-1</u>	<u>-5</u>
<u>1</u>	<u>0</u>	<u>0</u>	<u>1</u>	<u>0</u>	<u>-5</u>	<u>-3</u>
<u>1</u>	<u>1</u>	<u>0</u>	<u>1</u>	<u>0</u>	<u>-5</u>	<u>-1</u>
<u>0</u>	<u>0</u>	<u>0</u>	<u>1</u>	<u>1</u>	<u>-3</u>	<u>-3</u>
<u>0</u>	<u>1</u>	<u>0</u>	<u>1</u>	<u>1</u>	<u>-1</u>	<u>-3</u>
<u>1</u>	<u>0</u>	<u>0</u>	<u>1</u>	<u>1</u>	<u>-3</u>	<u>-1</u>
<u>1</u>	<u>1</u>	<u>0</u>	<u>1</u>	<u>1</u>	<u>-1</u>	<u>-1</u>
<u>0</u>	<u>0</u>	<u>1</u>	<u>0</u>	<u>0</u>	<u>3</u>	<u>5</u>
<u>0</u>	<u>1</u>	<u>1</u>	<u>0</u>	<u>0</u>	<u>1</u>	<u>5</u>
<u>1</u>	<u>0</u>	<u>1</u>	<u>0</u>	<u>0</u>	<u>5</u>	<u>3</u>
<u>1</u>	<u>1</u>	<u>1</u>	<u>0</u>	<u>0</u>	<u>5</u>	<u>1</u>
<u>0</u>	<u>0</u>	<u>1</u>	<u>0</u>	<u>1</u>	<u>3</u>	<u>3</u>
<u>0</u>	<u>1</u>	<u>1</u>	<u>0</u>	<u>1</u>	<u>1</u>	<u>3</u>
<u>1</u>	<u>0</u>	<u>1</u>	<u>0</u>	<u>1</u>	<u>3</u>	<u>1</u>
<u>1</u>	<u>1</u>	<u>1</u>	<u>0</u>	<u>1</u>	<u>1</u>	<u>1</u>

$q_{2k-1}$	$q_{2k}$	$q_{3k-2}$	$q_{3k-1}$	$q_{3k}$	$\tilde{m}$	
					$2k-1$	$2k$
<u>0</u>	<u>0</u>	<u>1</u>	<u>1</u>	<u>0</u>	<u>3</u>	<u>-5</u>
<u>0</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>0</u>	<u>1</u>	<u>-5</u>
<u>1</u>	<u>0</u>	<u>1</u>	<u>1</u>	<u>0</u>	<u>5</u>	<u>-3</u>
<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>0</u>	<u>5</u>	<u>-1</u>
<u>0</u>	<u>0</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>3</u>	<u>-3</u>
<u>0</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>-3</u>
<u>1</u>	<u>0</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>3</u>	<u>-1</u>
<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>-1</u>

1

2 For ModClass10, the mapping rules are:

3  $q_{7k-6} \quad q_{7k-4} \quad g_{5k-4} \rightarrow X_{12k-11} \quad X_{12k-10} \quad X_{12k-9}$

4  $q_{7k-5} \quad q_{7k-3} \quad g_{5k-3} \rightarrow X_{12k-8} \quad X_{12k-7} \quad X_{12k-6}$

5  $q_{7k-2} \quad q_{7k} \quad g_{5k-1} \rightarrow X_{12k-5} \quad X_{12k-4} \quad X_{12k-3}$

6  $q_{7k-1} \quad g_{5k-2} \quad g_{5k} \rightarrow X_{12k-2} \quad X_{12k-1} \quad X_{12k}$

7

for  $k=1,2,\dots$

8

**Table 131 Symbol mapper for Modulation classes 10**

$X_{3l-2} \quad X_{3l-1} \quad X_{3l}$	$\tilde{m}$
<u>0 0 0</u>	<u>-7</u>
<u>0 0 1</u>	<u>-5</u>
<u>0 1 1</u>	<u>-3</u>
<u>0 1 0</u>	<u>-1</u>
<u>1 1 0</u>	<u>1</u>
<u>1 1 1</u>	<u>3</u>
<u>1 0 1</u>	<u>5</u>
<u>1 0 0</u>	<u>7</u>

9

10 **17.2.10 Interleaving**

11 As defined in 4.2.10 of HC-SDMA [25].

### 17.2.11 Scrambling

As defined in 4.2.11 of HC-SDMA [25] *with the following underlined text included at the end of the section:*

Scrambling for SMB shall be same as that for CM.

### 17.2.12 $\pi/2$ Rotation and Scaling

As defined in 4.2.12 HC-SDMA [25] *with the following underlined text included at the end of the section:*

The modulation scaling for ModClasses 9 and 10 are as follows:

**Table 132 Modulation Scaling**

modclass	$\lambda$
2	$1/\sqrt{2}$
3	$1/\sqrt{2}$
4	1
5	1
6	$1/\sqrt{7}$
7	$1/\sqrt{10}$
8	$1/\sqrt{14}$
<u>9</u>	<u><math>1/\sqrt{20}</math></u>
<u>10</u>	<u><math>1/\sqrt{42}</math></u>

## 17.3 Broadcast Channel Modulation and Coding

The broadcast logical channels shall be as specified in 4.3 of the HC-SDMA [25].

## 1 **18 625k-MC User Terminal Radio Transmission and Reception**

2 The UT shall follow all procedures as specified in Chapter 5 of HC-SDMA [25]. The terminology  
3 used for both BS and UTs radio compliance specifications are defined in 5.1.1 of HC-SDMA [25].

## 1 **19 625k-MC Base Station Radio Transmission and Reception**

2 The BS shall follow all procedures as specified in Chapter 6 of HC-SDMA [25]. The terminology  
3 used for both BS and UTs radio compliance specifications are defined in 5.1.1 of HC-SDMA [25].

## 20 625k-MC L2 MAC Protocol Sublayer Specification

The 625k-MC L2 MAC protocol shall be as specified in Chapter 7 of HC-SDMA [25] *with the supplemental changes outlined herein.*

To support broadcast services, UT and BS L2 MAC protocol defines additional message: *Short Message Broadcast* transported by logical channel CCH.

### 20.1 Logical Channels

The relationship between logical channels, messages, and burst types is described in Table 7.1 of HC-SDMA [25] *with the additional text as underlined in Table 133.*

**Table 133 Logical Channels, Messages, and Burst Types**

Logical Channel	Message	Burst Types
BCH	Frequency synch message timing synch message broadcast message	frequency synchronization timing synchronization broadcast
PCH	page message	page
RACH	request access message access assignment message	standard uplink standard downlink
CCH	configuration request message configuration message <u>short message broadcast</u>	configuration request standard downlink <u>standard downlink</u>
TCH	uplink traffic downlink traffic	standard uplink standard downlink
FACCH	current modulation class (odd RFN's) additional available power (even RFN's) recommended modulation class (odd RFN's) current modulation class (even RFN's)	standard uplink standard uplink standard downlink standard downlink

10



### 20.1.1 Short Message Broadcast (SMB)

*This complete section is added before the end of section 7.3.4 of HC-SDMA [25].*

SMB shall conform to Table 134.

**Table 134 SMB fields**

Field Name	Number of bits
Minimum protocol	4
AFN	10
Network Operator ID	8
Data flag	4
Sequence number	7
Length	13
Null flag	1
Reserve	10
Data	48
Total	105

Short message channel field:

- Minimum protocol: minimal protocol shall be the same as that for CM in 7.3.4. 2 of HC-SDMA [25].
- AFN: The 10 least significant bits of the Absolute Frame Number on which short message channel was transmitted.
- Network Operator ID: Network Operation ID shall be the same as that for CM in 7.3.4. 2 of HC-SDMA [25].
- Data flag: Data flag shall identify the broadcast data among the 16 different sets available
- Sequence number: Sequence number shall specify the sequence number of the broadcast packet where the first packet in the sequence shall be 0.
- Length: Shall be the total length of the broadcast message in bits.
- Null flag: Shows that BS does not have any data to transmit.
- Data: Broadcast Data for transmission.

## 20.2 625k-MC Minimized RMU header

### 20.2.1 Message Format for AA-cts and AA-short

The RMU header defined in HC-SDMA shall be referred to as the *basic RMU header*. For small burst sizes, the *basic RMU header presents significant overhead*. For small burst sizes, UT and BS protocols of the 625k-MC mode shall support *minimized RMU header*. The minimized RMU header removes the ARQ acknowledgement field from bursts in which the successful receipt of the reverse-link burst by the peer can implicitly signal correct reception of forward-link data. In addition, the

1 minimized RMU header incorporates only the 7 LSBs of the ARQ sequence number when it is  
 2 possible to infer the full 13-bit sequence number from the 7 LSBs on the peer. A new ARQ format  
 3 field is added to the minimized RMU header to indicate whether the header carries an explicit or  
 4 implicit acknowledgement, and whether the full sequence number or only the 7 LSBs is present. The  
 5 following sections outline the supplemental changes to the baseline L2 MAC Protocol sublayer  
 6 specifications of HC-SDMA [25] when either UT or BS use *minimized RMU header*.

7 The message format for AA-cts and AA-short shall conform to Table 135 which is an updated version  
 8 of Table 7.14 of HC-SDMA [25] *with the additional text as underlined below*:

9 **Table 135 Message format for subtypes AA-cts and AA-short**

Field	# of Bits	Bit Positions
(AA common fields)	30	0:29
FrameDec	4	30:33
TchIndex	3	34:36
ModClassUp	4	37:40
modClassDown	4	41:44
Resource	6	45:50
<u>rmuHeaderType</u>	<u>1</u>	<u>51</u>
Reserved	<u>53</u>	<u>52:104</u>
<b>Total</b>	105	

10  
 11 AA common fields consist of the first 5 fields on Table 7.11 of HC-SDMA [25].

12 The definitions of frameDec, tchIndex, modClassUp, modClassDown and resource are as defined in  
 13 7.3.5.1.2.3.1 of HC-SDMA [25]. The rmuHeaderType field is defined below.

- 14 ■ rmuHeaderType: Shall indicate whether the basic RMU header or minimized RMU  
 15 header shall be used for the TCH bursts of the stream opened by the AA message. If  
 16 rmuHeaderType = 0, the basic RMU header shall be used. Else, the minimized RMU  
 17 header shall be used. The BS shall set the RMU header type field to 0 unless the BS has  
 18 selected in the Reg Params registration message use of a protocol version that defines the  
 19 minimized RMU header.

## 20 20.2.2 625k-MC RMU Field Definition

21 As specified in Section 7.3.7.2.1 of HC-SDMA [25] *with the additional text as underlined below*:

22 The structure of AM RMU shall conform to Figure 7.9 of HC-SDMA [25]. The gray portion is the  
 23 RMU Header. The basic RMU header is shown in Figure 7.9 of HC-SDMA [25]. The minimized  
 24 RMU header may be used instead, at the discretion of the BS, if both the BS and UT support it. The  
 25 header shall be present in every AM RMU. The white portion is the RMU Payload.

26 Note that the payload size can change when octets are retransmitted. For example, suppose that the  
 27 AM octets of RMU 1 are lost, and suppose further that RMU 2 is to retransmit as many of these  
 28 octets as possible. Since the UM portions of RMU 1 and RMU 2 can be different sizes, it is possible  
 29 that the set of AM octets in RMU 1 will not fit into RMU 2.

### 20.2.3 625k-MC Header Field Insertion

As specified in Section 7.3.7.2.2 of HC-SDMA [25] *with the additional text and table as underlined below*:

The RMU header fields shall conform to either the basic RMU header shown in Table 136 (Table 7.29 of HC-SDMA [25]), or the minimized RMU header shown in Table 137. The pwrCtrl bit in a downlink burst defines the relative adjustment to apply to transmit power uplink. The adjustment is 1dB higher if the bit is 1 and 1dB lower if the bit is 0. On an uplink burst, the pwrCtrl bit indicates the UT's measurement of SINR relative to the target SINR for a received downlink burst (which is a function of the modulation class of the downlink burst). The measured SINR is lower than the target if the bit is a 1 and higher if the bit is a 0. The seqNum and ack fields are used for the retransmission scheme as defined in 8.5.1 and 8.5.2 of HC-SDMA [25]. The AM RMU definition is defined in Figure 7.9 of HC-SDMA. During each frame, L2 MAC shall set the pwrCtrl bit with the value indicated by an L2MacTchTxInfoGet.resp primitive from L3 RRC. The seqNum and ack fields are filled by L2 RLC.

**Table 136 Basic RMU Header Fields (see Table 7.29)**

Field	# of Bits	Bit Positions	Interpretation
pwrCtrl	1	0	Power Control (downlink) / SINR (uplink) bit
Type	2	1:2	00: payload is pure AM data 01: payload is mixture of AM and UM data 10, 11: reserved
Seqnum	13	3:15	sequence number of the first AM octet in the payload. If the AM payload size is 0 octets, this value is ignored.
Ack	12	16:27	RLC acknowledgment

**Table 137 Minimized RMU Header Fields**

Field	# of Bits	Bit Positions	Interpretation
<u>pwrCtrl</u>	<u>1</u>	<u>0</u>	<u>Power Control (downlink) / SINR (uplink) bit</u>
<u>Type</u>	<u>2</u>	<u>1:2</u>	<u>00: payload is pure AM data</u> <u>01: payload is mixture of AM and UM data</u> <u>10, 11: reserved</u>
<u>arqFormat</u>	<u>1 or 3</u>	<u>3:</u>	<u>Coded indication of the format of the ARQ seq num and ack fields.</u> <u>0: 7 bit sequence number, ack field is not present</u> <u>001: 7 bit sequence number, ack field is present</u> <u>111: 13 bit seq num, ack field is not present</u> <u>011: 13 bit seq num, ack field is present</u> <u>101: reserved</u>

<b>Field</b>	<b># of Bits</b>	<b>Bit Positions</b>	<b>Interpretation</b>
<u>seqNum</u>	<u>7 or 13</u>		<u>Sequence number of the first AM octet in the payload. If the AM payload size is 0 octets, this value is ignored. If the arqFormat field is 0 or 001, this field contains the 7 LSBs of the full sequence number.</u>
<u>Ack</u>	<u>0 or 12</u>		<u>RLC acknowledgment. This field is only present when the arqFormat is 001 or 011.</u>
<u>Total</u>	<u>11,19,25, or 31</u>		

1

## 20.2.4 UM Message Insertion

As specified in Section 7.3.7.2.3 of HC-SDMA [25] *with the additional text as underlined below*:

The L2 MAC has two priority levels for AM data (see Chapter 8 of HC-SDMA [25]). The L2 MAC may prioritize UM messages above the high priority AM data, or below the high priority AM data and above the low priority

These respective priority levels for UM messages will be referred to as high, medium and low.

The method that L2 MAC does this prioritization shall be as follows:

1. Let  $N_{tot}$  be the total size of the RMU in bits (which is a function of modulation class) and  $N_{hdr}$  the size of the RLC header in bits (see Figure 7.9 of HC-SDMA [25]). If the basic RMU header is used, then  $N_{hdr} = 28$  (see Table 136). If the minimized RMU header is used, then L2 MAC must ask L2 RLC on each frame whether the RMU header will be 11, 19, 25, or 31 bits (see Table 137). The maximum number of bytes available for UM and AM messages is  $N_{rmu} = \lfloor (N_{tot} - N_{hdr}) / 8 \rfloor$  bytes where the floor function  $\lfloor x \rfloor$  equals the largest integer less than or equal to  $x$ .

The steps 2 to 8 and remaining text shall be as specified in 7.3.7.2.3 of HC-SDMA [25].

## 21 625k-MC L2 RLC Protocol Sublayer Specification

625k-MC L2 RLC protocol shall be as specified in Chapter 8 of HC-SDMA [25] with the supplemental changes outlined herein.

The RMU header defined in HC-SDMA is referred to as the *basic RMU header*. UT and BS of the 625k-MC mode shall support header compression techniques to minimize the *basic RMU header*. The following sections outline the supplemental changes to the baseline L2 RLC Protocol sublayer specifications of HC-SDMA [25] when either UT or BS use the *minimized RMU header*.

## 1 21.1 625k-MC AM RMU

2 As specified in Section 8.2.1.1 of HC-SDMA [25] *with the additional text as underlined below*:

3 The AM RMU shall send L3 SDUs and some control messages generated by L2 MAC using the L2  
4 RLC entity described in 8.1.3 L2 RLC Model of HC-SDMA [25]. Figure 7.9 of HC-SDMA [25]  
5 shows the format of the RMU Pay load.

### 6 21.1.1 RMU Header

7 The RMU header fields shall be as defined in Table 136 and Table 137.

8 For the basic RMU header and for the minimized RMU headers which include an explicit  
9 acknowledgement, the ack field shall contain the 12 LSBs of a sequence number.

## 10 21.2 625k-MC Transmit Procedure

11 As specified in Section 8.5.1 of HC-SDMA [25] *with the additional text as underlined below*:

12 During each frame, L2 MAC shall compute the maximum number of octets  $n_{max}$  available for  
13 transmitting high priority AM octets (see 8.1.3.3 Transmit ARQ of HC-SDMA [25]), by considering  
14 the size of the transmit burst's RMU payload, how many bits are needed for the RMU header, and  
15 how many octets are needed for high priority UM messages. The number of bits needed for the RMU  
16 header is constant (28) if the basic RMU header is used, and variable if the minimized RMU header is  
17 used. Similarly, L2 MAC shall compute  $n_{reserve}$ , the number of octets desired for medium priority UM  
18 messages if L2 RLC has less than  $n_{max}$  high priority AM octets. If the number of high priority AM  
19 octets is less than  $n_{max} - n_{reserve}$ , L2 RLC may use the remaining space to transmit any low priority AM  
20 octets next to the sequence of high priority AM octets. The Transmit ARQ shall also set seqNum and  
21 ack in the RMU header. If the basic RMU header is used, seqNum shall be set to the sequence  
22 number of the first octet written to the payload, ack shall be set to the S – 1 LSBs of the value of  
23 pAck. If the minimized RMU header is used, L2 RLC shall determine whether to put the full  
24 sequence number or only the LSBs into the RMU header, and whether to include or exclude the ack.  
25 The L2 RLC shall set arqFormat to indicate whether the RMU header contains the full sequence  
26 number or the LSBs, and whether the ack is present or implicit (see Table 137).

27 If the minimized RMU header is used, the RMU header shall have only the 7 LSBs of the sequence  
28 number if  $pTransmit -_m pAckPeer < 128$  and  $pSweep = pAckPeer$ . Otherwise the full sequence  
29 number shall be used.

30 If the minimized RMU header is used, then the RMU header shall not include the ack if the following  
31 three conditions are met:

- 32 1.  $pAck =_m pReceive$
- 33 2. the last received RMU was decoded successfully
- 34 3. the sequence number of the last byte of the last received RMU payload =  $pAck -_m 1$

35 Otherwise the ack shall be included in the RMU header.

1 The ack from the peer receive entity shall be processed and the transmit state variable shall be  
 2 updated as described in 8.5.2.1 of HC-SDMA [25] prior to the execution of the transmit states in  
 3 8.5.1.1 of HC-SDMA [25].

#### 4 **21.2.1 625k-MC Transmit State Execution**

5 Transmit state execution shall be as specified in 8.5.1.1 *with the additional text as underlined in the*  
 6 *last bullet point:*

7 The execution of the Transmit ARQ shall follow the following steps:

8 ■ **If  $nRetransmit > 0$  (Retransmit state)**

9 1. If  $nRetransmit > n_{max} - n_{reserve}$ , set  $n := \min(nRetransmit, n_{max})$ . Else, set  
 10  $n := \min(nTransmit + pTransmit -_m pRetransmit, n_{max} - n_{reserve})$ .

11 2. Send the segment of octets in

$$12 \quad [pRetransmit, pRetransmit +_m (n-1)]_m$$

13 3. Update  $mapTx$  as described in 8.5.1.2  $mapTx$  Update Procedure of HC-SDMA [25].

14 4. Set  $pRetransmit :=_m pRetransmit + n$ .

15 5. Set  $nRetransmit := nRetransmit - n$ .

16 ■ **Else if  $nTransmit > 0$  and  $pTransmit -_m pAckPeer < 2^{S-1} - 1$  (Transmit state).**

17 1. Set  $n := \min(nTransmit, n_{max}, 2^{S-1} - (pTransmit -_m pAckPeer) - 1)$ .

18 2. Send the segment of octets in

$$19 \quad [pTransmit, pTransmit +_m (n-1)]_m$$

20 3. Update  $mapTx$  as described in 8.5.1.2  $mapTx$  Update Procedure of HC-SDMA [25].

21 4. Set  $pTransmit :=_m pTransmit + n$ .

22 5. Set  $nTransmit := nTransmit - n$ .

23 Cancel timer  $T_{shutdown}$  if it is running.

24 ■ **Else if  $pAckPeer \neq pTransmit$  (Sweep state)**

25 1. If  $pSweep <_m pAckPeer$ , set  $pSweep := pAckPeer$ .

26 2. If  $pSweep \geq_m pTransmit$ , set  $pSweep := pAckPeer$ .

27 3. Set  $n := \min(pTransmit -_m pSweep, n_{max})$ .

28 4. Send the segment of octets in  $[pSweep, pSweep +_m (n-1)]_m$ .

1           5. If the basic RMU is used, update mapTx as described in 8.5.1.2 mapTx Update  
2           Procedure of HC-SDMA [25].

3           6. Set  $pSweep :=_m pSweep + n$ .

- 4           ■ **Else nothing to transmit(Idle State)**

## 5   21.2.2 mapTx Update Procedure

6   As defined in 8.5.1.2 of HC-SDMA [25].

## 7   21.3 Receive Procedure

8   As specified in Section 8.5.2 of HC-SDMA [25] *with the additional text as underlined below*:

### 9   21.3.1 Receive Task Execution

10   During each frame, the RMU Demux shall deliver to the Receive ARQ the AM payload (possibly 0  
11   octets) as well as the RMU header fields seqNum and ack unless there is a CRC error. (In case of a  
12   CRC error, none of them shall be delivered to the AM unit, and the following tasks are not executed.)

13   If the minimized RMU header is used, then the receiver shall check the argFormat RMU header field  
14   to determine whether the sequence number and ack fields need full length. If argFormat indicates that  
15   the RMU header contains only the 7 LSBs of the sequence number, then the receiver shall recreate  
16   the 13 bit sequence number with the following formulas. In these formulas, the & operator indicates a  
17   bitwise AND operation, and the ~ operator indicates bitwise negation.

18   if seqNum7bits >= (pAck & 0x7f)

19   seqNum13bits = (pAck & ~0x7f) + seqNum7bits

20   else

21   seqNum13bits = (pAck & ~0x7f) + 0x80 + seqNum7bits

22   If argFormat indicates that the RMU header has no ack, then the receiver shall recreate the ack as the  
23   following steps:

24       1. Set AckRFN = RFN -  $N_{turnaround}$

25       2. Here, RFN shows the RFN in which the burst was received.

26       3. Find the range of mapTx elements = AckRFN

27       4. Set ack to the 12 LSBs of the sequence number of the next mapTx element beyond the  
28       range identified in step 2.

29   The receiver shall perform the following three tasks.

1 **Task 1:receive the data in the AM payload**

- 2 1. Set  $first := seqNum$ .
- 3 2. Set  $last := {}_m seqNum + \text{size of payload}$ .
- 4 3. If  $last \geq {}_m pAck + 2^{S-1}$  and  $first \geq {}_m pAck$ .
- 5 i) Recommend forced shut down ( $E_{seqNum\_OB}$  error).
- 6 ii) Go to task 2.
- 7 4. If  $last \leq {}_m pAck$ , ignore the AM octets in this RMU and go to task 2.
- 8 5. Store data in appropriate location.
- 9 6. Cancel timer  $T_{shutdown}$  if it is running.
- 10 7. If  $last > {}_m pReceive$ , set  $pReceive := {}_m last$ .
- 11 8. If  $first \leq {}_m pAck$ ,
- 12 i) Find  $p$  ( $0 \leq p < 2^S$ ), the first sequence number that satisfies  $p \geq \leq {}_m last$  and  $mapRx[p] =$
- 13  $0$  (that is, there is not a  $q < {}_m p$  such that  $q \geq last$  and  $mapRx[q]=0$ ).
- 14 ii) Set  $mapRx[k] = 0$  for  $k \in [pAck, p - {}_m 1]_m$ .
- 15 iii) Deliver and record octets in  $[pAck, p - {}_m 1]_m$  to L3.
- 16 iv) Set  $pAck := p$ .
- 17 v) Cancel timer  $T_{receive}$ .
- 18 vi) If  $pAck \neq pReceive$ , restart timer  $T_{receive}$ .
- 19 Else (that is,  $first > {}_m pAck$ ) Set  $mapRx[k] = 1$  for  $k \in [first, last - {}_m 1]_m$ .

20 **Task 2:Process ack in RMU header (for transmit side)**

- 21 1. ack is  $S-1$  bits only. Convert it to  $S$  bits:
- 22 If  $ack < {}_m pAckPeer$ , set  $ack := {}_m ack + 2^{S-1}$ .
- 23 2. If  $ack > {}_m pTransmit$
- 24 i) Recommend forced shut down ( error  $E_{ack\_OB}$ ).
- 25 ii) Go to task 3.
- 26 3. If  $ack > {}_m pAckPeer$
- 27 i) Set  $mapTx[k] = -1$  for  $k \in [pAckPeer, ack - {}_m 1]_m$ .



1           ii) If  $pSweep <_m ack$  Set  $pSweep :=_m ack$ .

2           iii) Set  $pAckPeer :=_m ack$ .

3           iv) Cancel timer  $T_{transmit}$ .

4           v) If  $pAckPeer \neq pTransmit$ , restart timer  $T_{transmit}$

5           vi) If  $nRetransmit > 0$  and  $pRetransmit <_m ack$

6           Perform a mapTx retransmit update as described in 8.5.2.2 *mapTx Retransmit Update* of  
7           HC-SDMA [25].

8           4. If  $RFN - mapTx[pAckPeer] \geq N_{turnaround}$  (here,  $RFN$  is when the relevant octets were  
9           transmitted on the air interface)

10           Perform a mapTx retransmit update as outlined in 8.5.2.2 *mapTx Retransmit Update* of HC-  
11           SDMA [25].

### 12           **Task 3: check if stream is ready to end**

13           If timer  $T_{shutDown}$  is not running

14           If  $nTransmit = 0$ ,  $pAckPeer = pTransmit$ ,  $pAck = pReceive$ , a complete packet has been  
15           transmitted, a complete packet has been received, and pAck has not changed for at least 1  
16           frame, start timer  $T_{shutDown}$

### 17           **21.3.2 MapTx Retransmit Update**

18           As defined in 8.5.2.2 of HC-SDMA [25].

### 19           **21.3.3 Reset Procedure**

20           As defined in Sect. 8.5.2.3 of HC-SDMA [25].

1 **22 625k-MC L3 Protocol Specification**

2 The L3 protocol shall be as specified in Chapter 9 of HC-SDMA [25].

## 23 625k-MC Protocol Layer Primitives (Informative)

The protocol layer primitive shall be as specified in Chapter 10 of HC-SDMA [25]. To support broadcast and multicast services, following primitives are added to individual interfaces in Chapter 10 of HC-SDMA [25].

### 23.1 Interface list

The interface list shall be as specified in 10.3.1 of HC-SDMA [25].

### 23.2 Individual Interfaces

The basic individual interface primitives shall be as specified in 10.3.2 of HC-SDMA [25] along with additional primitives as defined in the sections 23.2.1 to 23.2.6 below:

#### 23.2.1 L2 MAC to L3 RM Interface Primitives

The L2 MAC to L3 RM Interface Primitives shall be as specified in 10.3.2.10 of HC-SDMA [25] *with the following additional primitive described herein.*

Primitive list:

- L2MacUtBCData.ind

<b><u>L2MacUtBCData.ind</u></b>	
<b><u>Applies to:</u></b>	<u>UT only</u>
<b><u>Associated primitives:</u></b>	<u>L2MacBsBCData.req (on BS)</u>
<b><u>Description:</u></b> <u>indication from UT L2 MAC to UT L3 RM that incoming broadcast data packets are available</u>	
<b><u>Parameters:</u></b>	
<ul style="list-style-type: none"> <li>• <u>processID -- ID of the L2 MAC process associated with this stream</u></li> <li>• <u>L2MacUtBcDataInfoSort -- the information of the broadcast data (struct)</u></li> <li>• <u>L2MacUtBcDataInfoSort :: MinimumProtocol -- a lowest version of the 625k-MC protocol that the BS can support</u></li> <li>• <u>L2MacUtBcDataInfoSort :: NetworkOperationID -- a identification of the network operator that owns the BS</u></li> <li>• <u>L2MacUtBcDataInfoSort :: DataFlag -- a flag categories of broadcast data</u></li> <li>• <u>L2MacUtBcDataInfoSort :: SequenceNumber -- a sequence number of broadcast data</u></li> <li>• <u>L2MacUtBcDataInfoSort :: Length -- a length of broadcast delivery data</u></li> </ul>	

16

### 23.2.2 L3 RM to L2 MAC Interface Primitives

The L3 RM to L2 MAC Interface Primitives shall be as specified in 10.3.2.11 of HC-SDMA [25] *with the following additional primitive described herein.*

Primitive list:

- L2MacBsBCData.req

<b><u>L2MacBsBCData.req</u></b>	
<b><u>Applies to:</u></b>	BS Only
<b><u>Associated primitives:</u></b>	<u>L2MacUtBCData.ind (on UT) (on peer)</u>
<b><u>Description:</u></b> <u>request from BS L3 RM to BS L2 MAC to deliver broadcast data packets across the air interface to UT L3 RM</u>	
<b><u>Parameters:</u></b> <ul style="list-style-type: none"> <li>• <u>processID -- ID of the L3 RM process requesting the stream</u></li> <li>• <u>L2MacBsBcDataInfoSort -- the information of the broadcast data (struct)</u></li> <li>• <u>L2MacBsBcDataInfoSort :: MinimumProtocol -- a lowest version of the 625k-MC protocol that the BS can support</u></li> <li>• <u>L2MacBsBcDataInfoSort :: NetworkOperationID -- a identification of the network operator that owns the BS</u></li> <li>• <u>L2MacBsBcDataInfoSort :: DataFlag -- a flag categories of broadcast data</u></li> <li>• <u>L2MacBsBcDataInfoSort :: SequenceNumber -- a sequence number of broadcast data</u></li> <li>• <u>L2MacBsBcDataInfoSort :: Length -- a length of broadcast delivery data</u></li> </ul>	

### 23.2.3 L3 RM to L3 CM Interface Primitives

The L3 RM to L3 CM Interface Primitives shall be as specified in 10.3.2.18 of HC-SDMA [25] *with the following additional primitives described herein.*

Primitive list:

- L3RmBsMCStart.ind
- L3RmUtMCDData.ind
- L3RmUtBCData.ind

<b><u>L3RmBsMCStart.ind</u></b>	
<b><u>Applies to:</u></b>	BS only
<b><u>Associated primitives:</u></b>	<u>L3RmBsMcStart.resp/L3RmBsMCSSStart.ind (on peer)</u>
<b><u>Description:</u></b> <u>indication from BS L3 RM to BS L3 CM that a new multicast service has been allowed</u>	
<b><u>Parameters:</u></b> <ul style="list-style-type: none"> <li>• <u>MCstrmID – desired multicast information is requested across the L4.</u></li> </ul>	

1

<b><u>L3RmUtMCData.ind</u></b>	
<b><u>Applies to:</u></b>	<u>UT only</u>
<b><u>Associated primitives:</u></b>	<u>L3RmBsMCDData.req (on BS) (on peer)</u>
<b><u>Description:</u></b> <u>indication from UT L3 RM to UT L3 CM that incoming delivered multicast packets are available</u>	
<b><u>Parameters:</u></b> <ul style="list-style-type: none"> <li>• <u>PktQueue -- a list of multicast data packets received across the air interface from the peer BS L3 CM entity</u></li> </ul>	

2

<b><u>L3RmUtBCData.ind</u></b>	
<b><u>Applies to:</u></b>	<u>UT only</u>
<b><u>Associated primitives:</u></b>	<u>L3RmBsBCData.req (on BS) ( on peer )</u>
<b><u>Description:</u></b> <u>Indication from UT L3 RM to UT L3 CM that incoming broadcast data packets are available</u>	
<b><u>Parameters:</u></b> <ul style="list-style-type: none"> <li>• <u>PktQueue -- a list of broadcast data packets received across the air interface from the peer UT L3 RM entity</u></li> <li>• <u>MinimumProtocol -- a lowest version of the 625k-MC protocol that the BS can support</u></li> <li>• <u>NetworkOperationID -- a identification of the network operator that owns the BS</u></li> <li>• <u>DataFlag -- a flag categories of broadcast data</u></li> <li>• <u>Length -- a length of broadcast delivery data</u></li> </ul>	

## 23.2.4 L3 CM to L3 RM Interface Primitives

The L3 CM to L3 RM Interface Primitives shall be as specified in 10.3.2.19 of HC-SDMA [25] *with the following additional primitives described herein.*

Primitive list:

- L3RMUtMCStart.req
- L3RmBsMCStart.resp
- L3RmBsMCData.req
- L3RmBsBCData.req

<b><u>L3RmUtMCStart.req</u></b>	
<b><u>Applies to:</u></b>	UT only
<b><u>Associated primitives:</u></b>	L3RmBsMCStart.ind (on BS) (on peer)
<b><u>Description:</u></b> request from UT L3 CM to UT L3 RM for a multicast service to allow communication across the air interface	
<b><u>Parameters:</u></b> • MCstrmID – desired multicast information is tunneled across the air interface	

<b><u>L3RmBsMCStart.resp</u></b>	
<b><u>Applies to:</u></b>	BS only
<b><u>Associated primitives:</u></b>	L3RmBsMCStart.ind
<b><u>Description:</u></b> response from BS L3 CM to BS L3 RM allowance whether BS L3 CM was able to accept a multicast service in response to the L3RmStart.ind from BS L3 RM	
<b><u>Parameters:</u></b> • Boolean -- was the connection started	

<b><u>L3RmBsMCData.req</u></b>	
<b><u>Applies to:</u></b>	BS only
<b><u>Associated primitives:</u></b>	L3RmUtMCData.ind (on UT) (on peer)
<b><u>Description:</u></b> request to UT L3 RM to deliver multicast data packets across the air interface to BS L3 CM	
<b><u>Parameters:</u></b> • PktQueue -- a list of multicast data packets to deliver across the air interface to the BS L3 CM entity	

<b><u>L3RmBsBCData.req</u></b>	
<b><u>Applies to:</u></b>	<u>BS only</u>
<b><u>Associated primitives:</u></b>	<u>L3RmUtBCData.ind (on UT) (on peer)</u>
<b><u>Description:</u></b> <u>request from BS L3 CM to BS L3 RM to deliver broadcast data packets across the air interface to UT L3 CM</u>	
<b><u>Parameters:</u></b> <ul style="list-style-type: none"> <li>• <u>PktQueue -- a list of broadcast data packets to send across the air interface to the UT L3 CM entity</u></li> <li>• <u>MinimumProtocol -- a lowest version of the 625k-MC protocol that the BS can support</u></li> <li>• <u>NetworkOperationID -- a identification of the network operator that owns the BS</u></li> <li>• <u>DataFlag -- a flag categories of broadcast data</u></li> <li>• <u>SequenceNumber -- a sequence number of broadcast data</u></li> <li>• <u>Length -- a length of broadcast delivery data</u></li> </ul>	

1

### 2 23.2.5 L3 CM to L4 Interface Primitives

3 The L3 CM to L4 Interface Primitives shall be as specified in 10.3.2.20 of HC-SDMA [25] *with the*  
4 *following additional primitives described herein.*

5 Primitive list:

- 6 ■ L3CmBsMCStart.ind
- 7 ■ L3CmUtMCData.ind
- 8 ■ L3CmUtBCData.ind

9

<b><u>L3CmBsMCStart.ind</u></b>	
<b><u>Applies to:</u></b>	<u>BS only</u>
<b><u>Associated primitives:</u></b>	<u>L3CmBsMCStart.resp</u>
<b><u>Description:</u></b> <u>indication from BS L3 CM to BS L4 to confirm a multicast service</u>	
<b><u>Parameters:</u></b> <ul style="list-style-type: none"> <li>• <u>MCstrmID – desired multicast information is requested.</u></li> </ul>	

10

<b><u>L3CmUtMCData.ind</u></b>	
<b><u>Applies to:</u></b>	<u>UT only</u>
<b><u>Associated primitives:</u></b>	<u>L3CmBsMCData.req (on BS) (on peer)</u>
<b><u>Description:</u></b> <u>indication from UT L3 CM to UT L4 that incoming multicast data packets are available</u>	
<b><u>Parameters:</u></b> <ul style="list-style-type: none"> <li>• <u>PktQueue -- a list of data packets received across the air interface from the peer L4 entity</u></li> </ul>	

11

<b><u>L3CmUtBCData.ind</u></b>	
<b><u>Applies to:</u></b>	<u>UT only</u>
<b><u>Associated primitives:</u></b>	<u>L3CmBsBCData.req (on BS)</u>
<b><u>Description:</u></b> <u>indication from UT L3 CM to UT L4 that incoming broadcast data packets are available</u>	
<b><u>Parameters:</u></b> <ul style="list-style-type: none"> <li>• <u>PktQueue -- a list of broadcast data packets received across the air interface from the peer L3 CM entity</u></li> <li>• <u>DataFlag -- a flag categories of broadcast data</u></li> <li>• <u>Length -- a length of broadcast delivery data</u></li> </ul>	

1

## 2 23.2.6 L4 to L3 CM Interface Primitives

3 The L4 to L3 CM Interface Primitives shall be as specified in 10.3.2.21 of HC-SDMA [25] *with the*  
4 *following additional primitives described herein.*

### 5 Primitive list:

- 6     ▪ L3CmUtMCStart.req
- 7     ▪ L3CmBsMCStart.resp
- 8     ▪ L3CmMCDData.req
- 9     ▪ L3CmBsBCData.req

10

<b><u>L3CmUtMCStart.req</u></b>	
<b><u>Applies to:</u></b>	<u>UT only</u>
<b><u>Associated primitives:</u></b>	<u>None</u>
<b><u>Description:</u></b> <u>request from UT L4 to UT L3 CM to start a multicast streaming across the air interface</u>	
<b><u>Parameters:</u></b> <ul style="list-style-type: none"> <li>• <u>MCstrmID – desired multicast information is tunneled across the air interface</u></li> </ul>	

11

<b><u>L3CmBsMCStart.resp</u></b>	
<b><u>Applies to:</u></b>	<u>BS only</u>
<b><u>Associated primitives:</u></b>	<u>L3CmBsMCStart.ind</u>
<b><u>Description:</u></b> <u>response from BS L4 to BS L3 CM allowance whether BS L4 was able to accept a multicast service in response to the L3CmBsMCStart.ind from BS L3 CM</u>	
<b><u>Parameters:</u></b> <ul style="list-style-type: none"> <li>• <u>Boolean -- was the connection started</u></li> </ul>	

12



<b><u>L3CmBsMCData.req</u></b>	
<b><u>Applies to:</u></b>	<u>BS only</u>
<b><u>Associated primitives:</u></b>	<u>L3CmUtMCData.ind (on peer)</u>
<b><u>Description:</u></b> <u>request from BS L4 to BS L3 CM to deliver multicast data packets across the air interface to UT L4</u>	
<b><u>Parameters:</u></b> <ul style="list-style-type: none"> <li>• <u>PktQueue -- a list of multicast data packets to deliver across the air interface to the UT L4 entity</u></li> </ul>	

1

<b><u>L3CmBsBCData.req</u></b>	
<b><u>Applies to:</u></b>	<u>BS only</u>
<b><u>Associated primitives:</u></b>	<u>L3CmUtBCData.ind (on UT) (on peer)</u>
<b><u>Description:</u></b> <u>request from BS L4 to BS L3 CM to deliver broadcast data packets across the air interface to UT L4</u>	
<b><u>Parameters:</u></b> <ul style="list-style-type: none"> <li>• <u>PktQueue -- a list of broadcast data packets to deliver across the air interface to the UT L4 entity</u></li> <li>• <u>DataFlag -- a flag categories of broadcast data</u></li> <li>• <u>Length -- a length of broadcast delivery data</u></li> </ul>	

2

## 24 625k-MC QoS Enhancements

*This Chapter is in addition to the baseline specification HC-SDMA [25].*

This chapter describes the QoS enhancements of 625k-MC mode.

### 24.1 Classes of Services

The 625k-MC system is optimized for application of Quality of Service (QoS) to connections. QoS L3 AM control message is used between BS and UT to handle the changes in session's QoS requirements. Different sessions (connections) within a UT may have different QoS markings same as DSCP (Differential Service Code Point from RFC2474) markings, which are conveyed by QoS message as specified in Table 24.2. If so, the BS may allot the stream resources available to the UT between its registrations based on the appropriate relative priority. See the description of *ibQoS* (QoS) argument of the *L3CmBsData.req* primitives in 10.3.2.1 and 10.3.2.19 of HC-SDMA [25].

To support the Standard DiffServ QoS Models, UT and BS of 625k-MC shall support the three classes of services as defined in Table 138 that incorporate forwarding Behaviors in the descending order of priority: Priority 1, Priority 2 and Priority 3. These classes are handled by Expedited Forwarding (EF), Assured Forwarding (AF) and Best Effort (BE) forwarding, respectively. In this table, QoS value refers to 6 MSB (most significant bits) of QoS marking in the QoS message of Table 138.

**Table 138 Classes of QoS Service**

	Class of Sessions/Users	Class of QoS	QoS Value (as listed in Table 139 QoS Message)	Example of Service
1)	Priority1	Expedited Forwarding (EF) - PHB	101110	VoIP
2)	Priority2	Assured Forwarding (AF) - PHB	001010 / 001100 / 001110 010010 / 010100 / 010110 011010 / 011100 / 011110 100010 / 100100 / 100110	Video Stream, Bandwidth Priority
3)	Priority3	Best Effort	000000	Data

The QoS value (or DSCP) for Priority 1 is EF Code Point 101110 (Refer RFC 2598), the DSCP for Priority 2 are defined with twelve kinds of AF Code Point (same as defined in RFC 2597) and the DSCP for Priority 3 is default Code Point 000000.

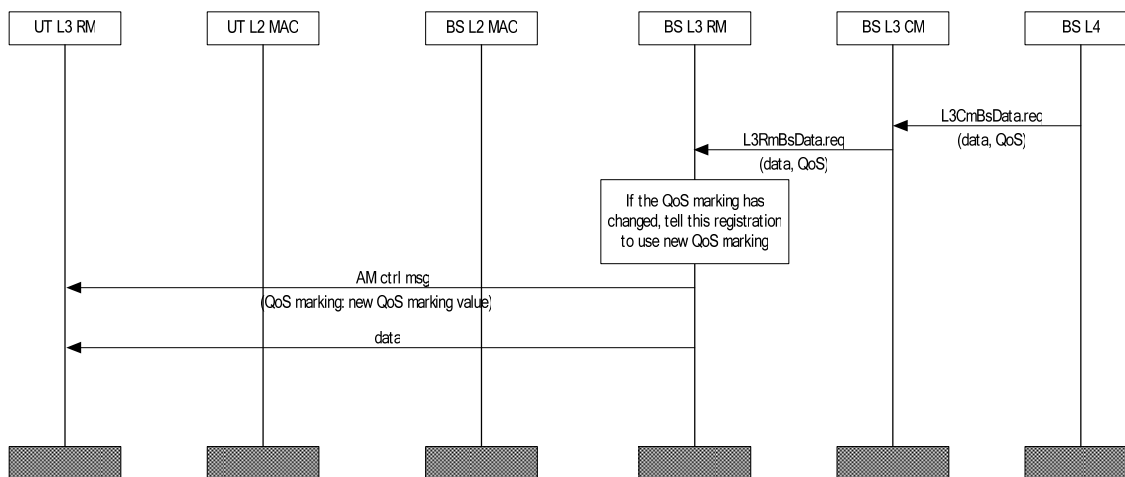
## 24.2 Session QoS Information Exchange Procedures

BS informs any change in the session’s QoS by using ‘AM-QoS Message’. The Formats of "AM-QoS Message" is shown in 9.5.1 in HC-SDMA [25], the sequence is shown in Figure 124 which is inherited from Fig.9.25 of HC-SDMA [25].

“QoS Value” (Table 138) is given by 6 msb of the octet: “QoS Marking” of “AM-QoS Message” in Section 9.5.1.3.12 of HC-SDMA [25]. “QoS marking” which is shown Figure 9.59 of HC-SDMA [25], is defined by 8 bits, therefore “QoS Value” (6bits) is set up as follows.

**Table 139 QoS Message**

msb		bits						lsb	octet
8	7	6	5	4	3	2	1		
0	0	0	0	1	1	0	0	0	
msgID									
0	TLVG length								1
ext bit	(assuming<128bytes)								
0	0	0	0	1	1	0	0	2	
TLVG ID									
QoS marking								3	
<u>QoS Value</u>						⋮	<u>Reserved</u>		



**Figure 124 QoS Message sequence**

1 When the BS is told by network to change the session's quality of service, BS delivers the QoS L3  
2 AM Control (AM-QoS) Message and thereby notifies a new QoS value to UT.

### 3 **24.3 QoS Priority**

4 As listed in Table 138, there are three Classes of services with deferent priorities.

5 For each class, Shutdown Timer (defined in 8.2.2.3 of HC-SDMA [25]), which can specify the time  
6 to hold a stream can be set. Based on the priority, the Shutdown Timer can be set differently for  
7 different streams. For high priority data, a large timeout value should be used for the Shutdown  
8 Timer. These values are different for different streams.

9 It can perform differentiation by reservation of the stream according to priority. Generally, longer  
10 Shutdown Timer is set up for the sessions with in high class of priority. The values of longer  
11 Shutdown Timer may be implementation specific .

#### 12 **24.3.1 QoS Priority 1**

13 Priority 1 of QoS shall be the highest priority of three Classes of Service.

14 It shall support high quality transmission achieving low packet loss, low latency, and low jitter  
15 duration. It is defined as the EF-PHB, which 6bits from MSB of QoS Marking field in the QoS  
16 Message is 101110.

17 The Shutdown Timer shall be specified to a larger value than that of other classes.

#### 18 **24.3.2 QoS Priority 2**

19 Priority 2 of QoS shall be defined to a priority level lower than Priority 1.

20 It shall support the higher quality transmission than Priority 3.

21 It is defined as the AF-PHB, and twelve kinds of QoS Marking field in the QoS Message could be  
22 defined and used flexibly in each implementation.

23 The Shutdown Timer shall be set to a value smaller than Priority 1 and larger than Priority 3.

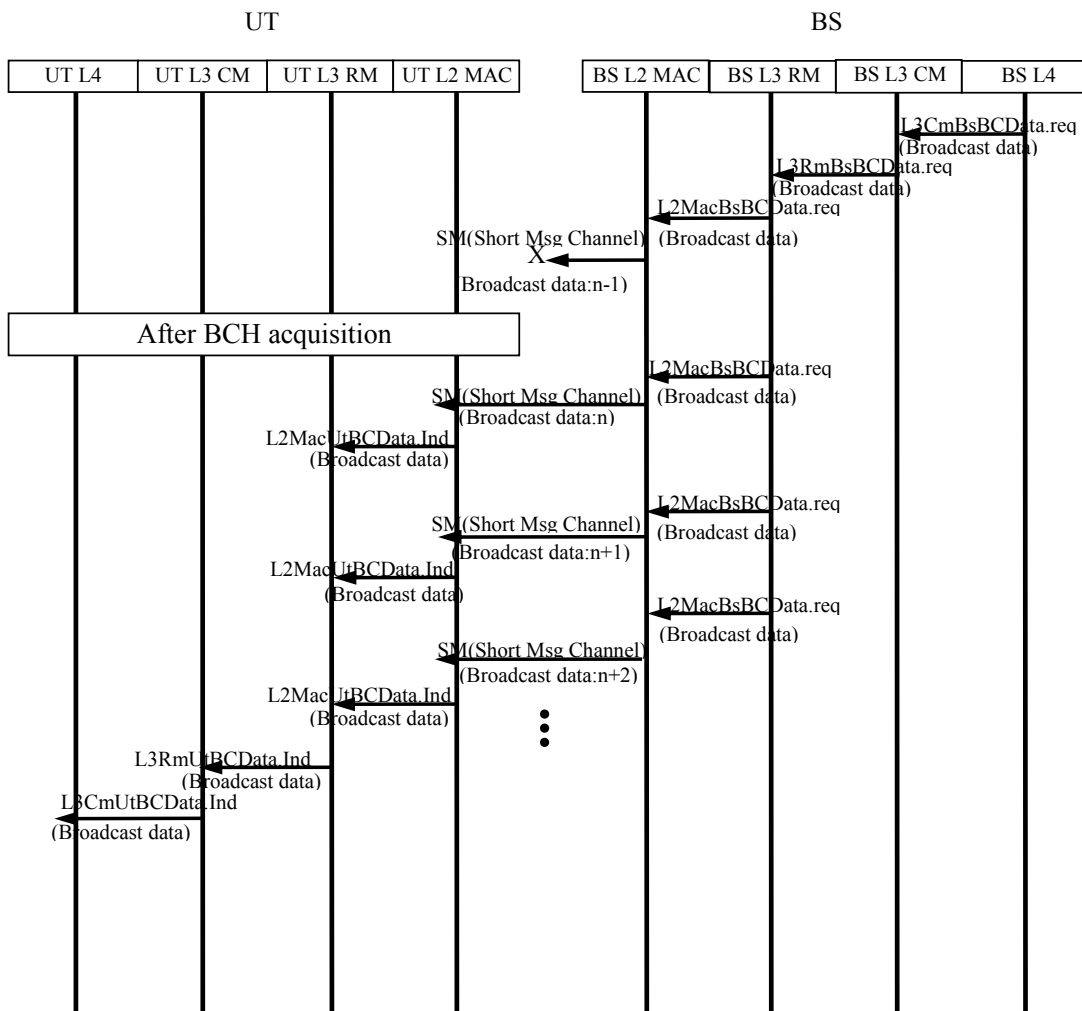
#### 24 **24.3.3 QoS Priority 3**

25 Priority 3 of QoS shall be defined to Best Effort PHB.

26 It is defined as the BE-PHB, which 6bits from MSB of QoS Marking field in the QoS Message is  
27 000000.

28 The Shutdown Timer shall be specified to a smaller value than that of other classes.





1

2

**Figure 126 Message Flow for short message channel**

3

**25.3 Multicast Service**

4

**25.3.1 Overview**

5

Multicast channels are made available to many users, therefore the functions of RRC are necessary for individual TCH of multicast channel.

6

7

**25.3.2 Multicast handshake**

8

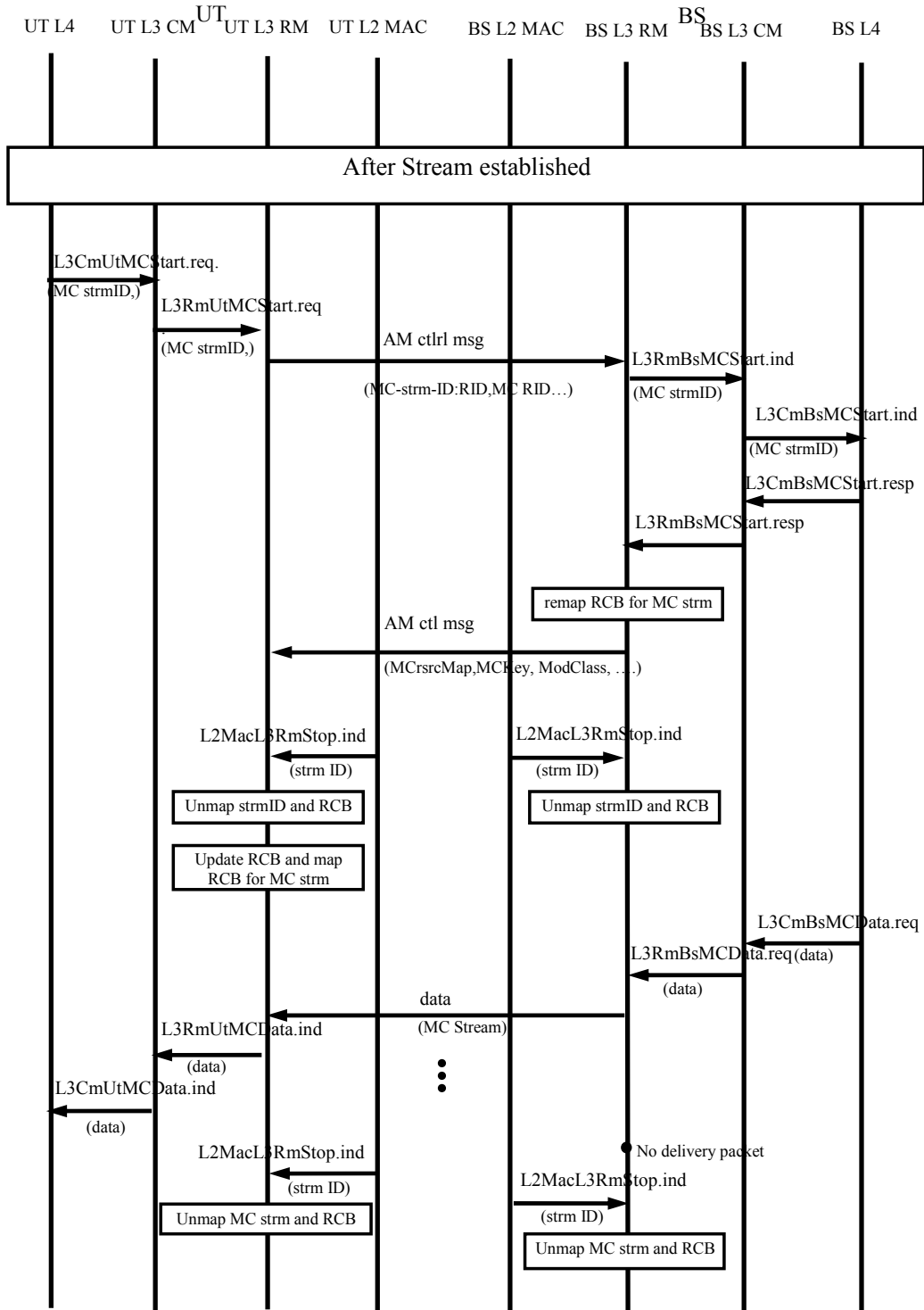
To connect to a multicast channel, UT must send *Multicast stream request* to BS. BS sends *Multicast stream response* which includes *resource map* information. Resource map information tells the UT about the TCH channel over which multicast data is transmitted. This TCH channel shall be called the *Multicast channel*. UT changes the receiving TCH channel to Multicast Channel. The messaging handshake for Multicast stream shall conform to Figure 127.

9

10

11

12



1  
2

**Figure 127 Message Flow for Multicast channel**

- 1           ■ To send the MCStream.req to BS, UT opens a stream after TCH connection following the  
2           *UT Initiated stream Start* sequence specified in Fig. 9.19 of the HC-SDMA [25].
- 3           ■ UT sends information about StrmID, MC, RID together with the MCStream.req over the  
4           connected TCH. If BS allows the stream to open, it shall respond by sending the  
5           MCStream.resp with information necessary for Multicast like Mckey, ModClass, and  
6           rsrcMap.
- 7           ■ UT and BS shall close the stream following the *Graceful Stream Closure* sequence  
8           specified in Fig. 9.27 of HC-SDMA [25].
- 9           ■ BS shall transmit Multicast information over the channel specified using the MCsrc  
10          during MCStream.resp; UT shall receive over the same channel. During Multicast  
11          information transmission and reception, RRC functions are not considered. Data shall be  
12          transmitted from BS at a constant level with the same ModClass specified during  
13          MCStream.resp. The data shall be encrypted at BS and the UT shall decrypt it using the  
14          Mckey. Mckey length shall choose between 56 bit and 280 bit for Multicast registration.

15 In case the UT is unable to receive multicast data or the multicast data finishes, the Multicast session  
16 shall be closed following the graceful stream closure sequence as specified in Fig. 9.27 of HC-SDMA  
17 [25]. The BS shall continue to transmit until the data finishes.

### 18 **25.3.3 Message Format**

19 Before the UT receives the multicast data, UT needs to register with the BS using the AM message  
20 data, viz., MCStream.req and MCStream.resp. BS uses AM mode also to deliver Multicast data to  
21 UT. The RMU header field shall conform to the Table 7.29 in 7.3.7.2.2 of HC-SDMA [25]. BS shall  
22 use RMU header type as "01" which indicates "payload is mixture of AM and UM data". During  
23 multicast session, BS and UT shall ignore "power control", "FACCH" and "ack" for AM in this  
24 BCMCS protocol.

25 The message format for MCStream.req and MCStream.resp are as shown in Figure 128 and  
26 Figure 129, respectively.

#### 27 **25.3.3.1 MCStream.req Message**

28 Message ID: 16

29 TLVG: MCStream.req

30 TLVG ID: 16

31 Information Elements:

- 32           ■ IE: Assigned protocol version, 8 bits

33 Description:

34 The current protocol version field indicates which revision of the protocol the sending  
35 party is currently using. This is used by the UT in the MCStream.req message and by the  
36 BS in the MCStream.resp message to inform each other of which protocol version is  
37 being used during the multicast stream.



- 1 ■ IE: RID, 15 bits

2 Description:

3 The registration ID (RID) is used in various of the RA/AA messages that open streams to  
4 identify which registration the stream is for.

- 5 ■ IE: MCstrmID, 17 bits

6 Description:

7 The MCstrmID indicates the UT's desired multicast information which is depended on  
8 delivered contents. When the BS delivers only one multicast channel, MCstrmID value is  
9 all zero.

msb							lsb
1	2	3	4	5	6	7	8
0	0	0	1	0	0	0	0
msg ID							
0	TLVG length						
ext bit	(assuming < 128 bytes)						
0	0	0	1	0	0	0	0
msg ID							
Assigned 625K-MC protocol version							
RID							
RID							MCstrmID
MCstrmID							
MCstrmID							
Reserved							

10

11

**Figure 128 MCStream.req format**

### 25.3.3.2 MCStream.resp Message

Message ID: 17

TLVG: MCStream.resp

TLVG ID: 17

Information Elements:

- IE: Assigned 625k-MC protocol version, 8 bits

Description:

The assigned protocol version field indicates which revision of the 625k-MC protocol the sending party is requiring the receiving party to use. This is used by the BS in the MCStream.resp message to tell the UT what protocol version to use throughout the multicast registration.

- IE: McsrcMap, 48 bits

Description:

The McsrcMap specifies which conventional channels of the BS the UT is allowed to use for Multicast stream. The LSB of this bit map corresponds to conventional channel ID 0. See Section Conventional Channel ID on HC-SDMA [25] page A-3 for a definition of the conventional channel ID.

- IE: reject presence, 1 bit

Description:

This reject presence indicates whether the BS chose to reject the multicast registration. When this field is 0, the multicast registration is not rejected. When this field is 1, the multicast registration is rejected by the BS.

- IE: MC reject cause, 5 bit

Description:

This MC reject cause is indicated the reason for the multicast rejection. The rejection causes are listed in Table 140 when reject presence field is 1.

**Table 140 Multicast connection rejection values**

Value	Rejection/Cause
0	Unknown/unspecified
1	The UT's protocol version is too old
2	The UT's protocol version is too new
3	The BS's current traffic load is too high
4	The UT is not registered
5	The BS is not delivered for multicast data
6	The BS is preparing multicast service

Value	Rejection/Cause
7	Invalid MCStreamID value
8-31	Reserved

1

2

- IE: MC key size, 5 bits

3

Description:

4

5

6

7

The K size field MCKey is used by the BS to specify how many bytes of the (decrypted) multicast security stream secret will be used as the i-SEC encryption secret. If this field is non-zero, the key size is the value of this field plus four, in bytes. If the field is zero, this indicates that for testing purposes multicast security is disabled for the registration.

8

- IE: MC key, 326-640 bits

9

Description:

10

11

12

The MC key consists of i-SEC key (K), i-SEC key(Kr), counter value and AFN. This K, AFN and counter value changes every frame. In fact, i-SEC key and AFN are one-on-one relationship.

13

14

15

This K is composed in multiples of 8 bits that total size is between 40 bits and 280 bits. The key size depends on the multicast provider. And AFN size is 26 bits, counter value size is 8 bits, and Kr is 326 bits. Total size is 640 bits.

16

17

18

Multicast delivery data is encrypted to use the part of MC key which is shared secret key by the BS. In contrast, the UT identifies the encrypted data to decrypt the same parts of MC key

19

- IE: MC Modclass, 4 bits

20

Description:

21

22

The MC Modclass indicates multicast delivering modClass from the BS. This delivery modclass shall be set by the BS.

msb								lsb	
1	2	3	4	5	6	7	8		
0	0	0	1	0	0	0	0	1	
msg ID									
0	TLVG length								
ext bit	(assuming < 128 bytes)								
0	0	0	1	0	0	0	0	1	
msg ID									
Assigned 625K-MC protocol version									
MCsrcMap									
MCsrcMap									
MCsrcMap									
MCsrcMap									
MCsrcMap									
MCsrcMap									
MCsrcMap									
reject presence	MC reject cause						Reserved		
Reserved				MC key size					
MC key									
MC key									
·									
·									
·									
MC Modclass					Reserved				
Reserved									

1

2

**Figure 129 MCStream.resp format**

## 1 26 625k-MC Privacy and Authentication Enhancement

2 As defined in pages 11-1 and 11-2 of Chapter 11 of HC-SDMA [25] *with the additional text as*  
 3 *underlined below:*

4 The symmetric key encryption engine utilized in the AES for AES algorithm is as described in FIPS  
 5 PUB 197.

### 6 26.1 Overview

7 As defined in 11.1 of HC-SDMA [25].

### 8 26.2 625k-MC Handshake and BS Authentication Protocol, i-HAP

9 As defined in 11.2 of HC-SDMA [25], *with the additional definitions and text for the certificate*  
 10 *message fields in the section of 11.2.2 f HC-SDMA [25] as underlined below:*

11 10. *i-SEC* Bulk Encryption Algorithm Choices: Shall be 4 bits. This shall be a bitmap of bulk  
 12 encryption algorithms supported by the BS. Every supported algorithm shall have a '1' in the relevant  
 13 bit. For example, if the stream cipher method and the AES modes are supported then this field shall  
 14 be set to [1,1,0,0].

15 Bit 0: Stream Cipher (see Section 11.4.2).

16 Bit 1: AES Cipher (see Section 11.4.3).

17 Bit 2: Reserved.

18 Bit 3: Reserved.

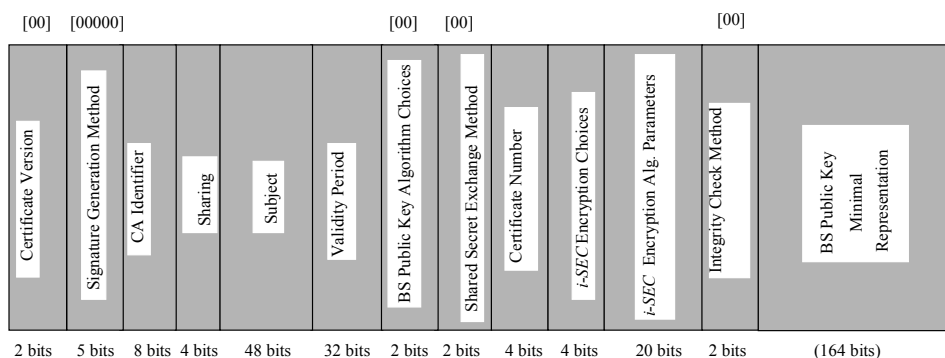
19 11. *i-SEC* Bulk Encryption Parameters: Shall be 20 bits. This field shall contain 4 subfields, each 5  
 20 bits. The  $k$ -th subfield shall contain parameters related to the  $k$ -th bulk encryption algorithm presented  
 21 in Field (10) above. If a bulk encryption algorithm is not supported, then the corresponding subfield  
 22 shall be set to all zeros. The information in each subfield shall depend on the definition and  
 23 parameters of the corresponding bulk encryption algorithm.

24 ■ For the stream cipher:

- 25 □ The first subfield shall provide the secret key size in bytes: if this field is all zeros,  
 26 then the stream cipher is disabled; otherwise, the secret key size is the value  
 27 represented in first subfield (in little endian format) plus four in bytes.
- 28 □ The resulting value (in bytes) shall be defined as the maximum value of encryption  
 29 key size ( $L_{\text{key}}$ ) supported by the BS.
- 30 □ For example, a BS that supports the stream cipher alone, with a maximum encryption  
 31 secret key size of 56 bits (7 bytes) has:  
 32 – [1,0,0,0] as its Algorithm Choices field (Field 9 above), and

- 1                   – [1,1,0,0,0;0,0,0,0,0;0,0,0,0,0;0,0,0,0,0] as its Bulk Encryption Parameters field.  
 2                   Note that each subfield represents an integer as a bit string encoded with LSB  
 3                   first.
- 4                   ■ For the AES cipher:
  - 5                   □ The first two bits of the second subfield shall be used to indicate the maximum length  
 6                   of secret key size to be used in the AES cipher. The value [10] indicates 128-bit  
 7                   secret key, [01] indicates 192-bit secret key, and [11] indicates a 256-bit secret key  
 8                   for the cipher. The value [0,0] indicates that the AES cipher is disabled.
  - 9                   □ The remaining three bits of the second subfield shall be used to indicate the  
 10                   encryption mode of the AES cipher. If the AES cipher is disabled, then this field  
 11                   must be set to [0,0,0]. The value [1,0,0] indicates that the output feedback (OFB)  
 12                   mode is supported. The remaining three-bit combinations are reserved.
  - 13                   □ For example, a BS that supports the AES cipher alone, with a maximum of 192 bit  
 14                   secret key length and supports the OFB mode has:
    - 15                   – [0,1,0,0] as its Algorithm Choices Field, and
    - 16                   – [0,0,0,0,0;**0,1,1,0,0**; 0,0,0,0,0; 0,0,0,0,0; 0,0,0,0,0]. Note that each subfield  
 17                   represents an integer as a bit string encoded with LSB first.
  - 18                   □ A BS station that supports **both** the stream cipher (key size 56 bits) and the AES  
 19                   cipher (key size 256 bits) has:
    - 20                   – [1,1,0,0] as its Algorithm Choices Field, and
    - 21                   – [1,1,0,0,0;1,1,1,0,0; 0,0,0,0,0; 0,0,0,0,0; 0,0,0,0,0] as its Bulk Encryption  
 22                   Parameters field.

23 With the above additional definitions, the BS certificate message fields are illustrated in Figure 130.



24

25 **Figure 130 HC-SDMA BS certificate message fields. Mandatory field values are identi-**  
 26 **fied in square brackets. Not to scale.**

27 **26.3 625k-MC Terminal Authentication Protocol, i-TAP**

28 As defined in 11.3 of HC-SDMA [25].

## 26.4 625k-MC Secure Communications Protocol, i-SEC

The Secure Communications Protocol, i-SEC shall be as specified in 11.4 of HC-SDMA [25] *with additional support for minimized RMU header and AES as specified herein.*

In HC-SDMA [25], the *i-SEC* protocol is responsible for symmetric encryption of data and control messages based on the shared secret. In 625k-MC mode, BS and UT shall support the AES cipher algorithm in addition to the stream cipher algorithm used in HC-SDMA [25].

i-SEC protocol is responsible for the encryption of TCH traffic streams that carry AM data and control messages. The BS certificate provides information about the capabilities of the base station in terms of supported algorithms and secret key sizes. In 625k-MC mode, i-SEC shall support the stream cipher algorithm and the AES algorithm explained in this section and shall use a portion<sup>104</sup> of the shared secret established during i-HAP for encryption.

In addition, the BS shall have two thresholds: regThreshBS that limits the number of TCH streams that use the same shared secret and streamThreshBS that limits the number of bursts per TCH stream. The UT shall have its own thresholds regThreshUT and streamThreshUT for the same purpose. The UT shall be responsible for closing TCH streams or initiating new registrations based on the updated values of regThreshUT and streamThreshUT gracefully. The BS shall be responsible for terminating TCH streams and registrations based on its thresholds streamThreshBS and regThreshBS.

The i-SEC encryption key  $K$  shall consist of the  $L_{key}$  lowest significant bytes of the shared secret (326 bit long shared secret which shall be exchanged with the current BS public key encryption algorithm). The largest value of the  $L_{key}$  parameter supported by a BS shall be presented by field (12) of the BS certificate. For the stream cipher,  $L_{key}$  shall have a maximum value of 35 bytes, allowing for encryption keys of up to 280 bits. The remaining 46 bits of the shared secret shall be reserved. For the AES cipher, shall have a maximum value of 32 bytes, allowing for encryption keys of up to 256 bits. The next 32 bits of the shared secret shall be used for initialization purposes (shared secret bits 0-255 for AES encryption key, and shared secret bits 256-287 are used for initialization). The remaining shared secret bits are reserved. can take on only three values: {16,24,32} bytes. The actual length of the i-SEC encryption key is negotiated between the UT and the BS.

### 26.4.1 TCH Streams

As defined in 11.4.1 of HC-SDMA [25].

### 26.4.2 625k-MC Symmetric Key Stream Cipher Algorithm

As specified in 11.4.2 of HC-SDMA [25] *with the following changes to support minimized RMU header.*

#### 26.4.2.1 Initialization Vector Selection

As specified in 11.4.2.1 of HC-SDMA [25].

---

<sup>104</sup> A disjoint portion of the shared secret shall be used for scrambling in *i-TAP*.

### 26.4.2.2 Stream Counter Test

As specified in 11.4.2.2 of HC-SDMA [25]

### 26.4.2.3 Determination of Encryption Key from Shared Secret

As specified in 11.4.2.3 of HC-SDMA [25]

### 26.4.2.4 Stream Cipher State Initialization

As specified in 11.4.2.4 of HC-SDMA [25]

### 26.4.2.5 Keystream Generation

As specified in 11.4.2.5 of HC-SDMA [25] *with the additional text as underlined below:*

- • The maximum uplink burst is 1000 bits, including 16 bits of CRC. The CRC shall not be encrypted in *i-SEC*.
- • The maximum downlink burst is 2539 bits, including 16 bits of CRC. The CRC shall not be encrypted in *i-SEC*.

### 26.4.2.6 Burst Counter Test

As specified in 11.4.2.6 of HC-SDMA [25]

### 26.4.2.7 Encryption and Decryption Using *i-SEC* Keystream

As specified in Section 11.4.2.7 of HC-SDMA [25] with the additional text as underlined below:

*i-SEC* encryption shall be performed by XORing the RMU header and payload (see Section 7.3.7.2 RMU MuxFunction on page 7-45 and Figure 7.9 on page 7-47 of HC-SDMA [25] for definitions) by the *i-SEC* keystream before transmission. Therefore, *i-SEC* encryption is not a bandwidth expanding transformation. The 16 bit CRC shall be calculated over the encrypted bits. The CRC shall not be encrypted. *i-SEC* decryption shall be performed by XORing the received encrypted RMU header and payload octets by the *i-SEC* keystream. The 16 bit CRC check shall be performed on the received encrypted bits before decryption.

In streams using the basic RMU header, The RMU header and payload shall be aligned such that the last payload falls exactly at the end of the relevant keystream buffer. In streams using the minimized RMU header, the last burst bit before the CRC bits shall be aligned with the end of the relevant keystream buffer. During this process, some of the keystream bytes or bits are possibly left unused and they shall be discarded.

For example, consider an uplink burst payload that contains 28 bits of basic RMU header and 3 payload octets {ABC} with a basic RMU header H represented by a bit vector  $[h_0, h_1, \dots, h_{27}]$  where  $h_0$  is transmitted first and the header is partitioned into four bytes:  $H_0 = [0, 0, 0, 0, h_0, h_1, h_2, h_3]$ ,  $H_1 = [h_4, h_5, h_6, h_7, h_8, h_9, h_{10}, h_{11}]$ ,  $H_2 = [h_{12}, h_{13}, h_{14}, h_{15}, h_{16}, h_{17}, h_{18}, h_{19}]$ , and  $H_3 = [h_{20}, h_{21}, h_{22}, h_{23}, h_{24}, h_{25}, h_{26}, h_{27}]$ .



### 26.4.3 625k-MC AES Cipher Algorithm

*This complete subsection 26.4.3 is added before end of Section 11.4 of HC-SDMA [25]*

*i-SEC* shall employ the symmetric key encryption algorithm for AES. The AES algorithm is defined in FIPS PUB 197. *i-SEC* shall support the AES encryption algorithm. The following sections provide the details. Note that the AES cipher can only utilize up to 256 least significant bits of the shared secret ( $K_r$ ) established during *i-HAP* as the AES cipher key ( $K_{AES}$ ).

#### 26.4.3.1 AES Initialization Vector Selection

The *i-SEC* encryption key  $K_{AES}$  shall be utilized together with a 128-bit AES initialization vector ( $IV_{AES}$ ) to initialize the AES cipher operation at the beginning of a stream.  $IV_{AES}$  shall be different for each TCH stream over the expected lifetime of the shared secret key.

Let  $IV$  denote the 32-bit initialization vector defined in Section 11.3.3 of the HC-SDMA standard for the stream cipher. Let us define the 32 bit vector  $W$ :

$$W(n) = K_r(n+256), \quad n=0,1,\dots,31.$$

Where  $W(n)$  denotes the  $n$ -th bit of  $W$ . The 32-bit vector  $W$  will be modified by a component-wise XOR with the  $IV$  defined in Section 11.3.3 of the HC-SDMA standard.

$$W(n) = W(n) \text{ XOR } IV(n), \quad n=0,1,\dots,31.$$

The initialization vector for the AES algorithm is defined as the concatenation of  $W$  to obtain a length 128-bit vector:

$$IV_{AES} = [W, W, W, W];$$

As a result the true value of  $IV_{AES}$  is unknown to parties other than the BS and the UT.

#### 26.4.3.2 AES Stream Counter Test

As specified in 11.4.2.2 of HC-SDMA [25].

#### 26.4.3.3 Determination of AES Encryption Key from Shared Secret

The length (in bytes) of *i-SEC* encryption key ( $K$ ) shall be defined as  $L_{key}$ . The second subfield of Field (12) of the BS certificate shall specify the maximum value of  $L_{key}$  supported by the BS. The value of  $L_{key}$  shall be in the set {16, 24, 32} bytes for the AES cipher.

The first ( $L_{key} \times 8$ ) least significant bits of the shared secret  $K_r$  shall be defined as the encryption key,  $K_{AES}$ :

$$K_{AES}(n) = K_r(n), \quad n=0,1,\dots,8L_{key}-1.$$

Where  $K_{AES}(n)$  is the  $n$ -th bit of the *i-SEC* encryption key  $K_{AES}$ , and  $K_r(n)$  is the  $n$ -th bit of the *i-SEC* encryption key  $K_r$ . For maximally secure deployments,  $L_{key}$  may be set to  $32^1$ . The actual key size used by *i-SEC* shall be negotiated during registration. The BS certificate shall set the maximum

length on the key size that the BS supports. The UT shall send the key length to be used for the registration in the `UT Params` message. Test UTs may use a method to bypass *i-SEC* encryption.

#### 26.4.3.4 AES Cipher State Initialization

Definition 1: Let  $Y=E(Z;M)$  denote the encryption of a 128 bit message vector  $M$ , with an AES key named as  $Z$  (128, 192 or 256 bits) as defined in the FIPS Standard 197. The cipher output is a 128 bit vector  $Y$ .

Definition 2: Let  $M=D(Z;Y)$  denote the decryption of a 128 bit vector  $Y$ , with an AES key named as  $Z$  (128, 192 or 256 bits) as defined in the FIPS Standard 197.

Each stream shall maintain a single AES encryption engine, and a 128-bit initialization vector ( $IV_{AES}$ ). The computation initialization vector ( $IV_{AES}$ ) is defined in Section 26.4.3.1 above.

At the beginning of each stream, the AES initialization vector is updated ( $IV_{AES}$ ) by two successive encryption operations with the AES encryption key ( $K_{AES}$ ) before encrypting any information bits:

$$IV_{AES} = E(K_{AES}, E(K_{AES}, IV_{AES}));$$

This operation takes place both at the message originator (encryptor) and the message receiver (decryptor).

#### 26.4.3.5 AES Keystream Generation

After the double encryption operation on  $IV_{AES}$  (refer to previous section) that precedes the keystream generation procedure, a block of 128 keystream bits  $U=[U_0, \dots, U_{127}]$  is generated and the initialization vector  $IV_{AES}$  is updated by the following procedure:

$$U = E(K_{AES}, IV_{AES});$$

$$IV_{AES}(n) = U_n \text{ XOR } IV_{AES}(n), n=0,1,\dots,127;$$

Note that each call to the above two-line procedure generates 16 keystream bytes.

On every frame following the RA/AA frame, 448-keystream bytes (28 calls to the above procedure per frame) shall be generated for uplink and downlink encryption. The first 128-keystream bytes (8 calls) shall be allocated for uplink encryption and the remaining 320-keystream bytes (20 calls) are allocated for downlink encryption.

Note that these keystream bytes will be generated regardless of the modulation class employed. The unused keystream bytes will be disregarded. The encryption/decryption procedure using the keystream is explained in Section 26.4.3.7 below.

#### 26.4.3.6 AES Burst Counter Test

The UT shall increment its counter `burstCount` after generating the keystream for a burst. This counter shall be set to zero at every stream start. After incrementing `burstCount`, the UT shall compare the value of the counter to a threshold entitled `streamThreshUT`. The UT application software shall set the value for the stream termination threshold `streamThreshUT`. If `streamThreshUT` is non-zero, and the value of `burstCount` is equal to `streamThreshUT`,

1 then the UT shall close the current stream and shall start a new stream with a new stream secret and  
 2 IV. If the value of `streamThreshUT` is zero, then the UT shall not force a stream closure.

3 The BS shall perform the same operations independently by using its own counter `burstCount`,  
 4 and its own threshold `streamThreshBS`. A nominal value for `streamThreshBS` and  
 5 `streamThreshUT` shall be 20,000 bursts. At a combined uplink/downlink approximate data rate  
 6 of 700 kilobits/sec (single burst/frame, 200 frames/sec), this results in a stream closure approximately  
 7 after 7.5 megabytes of over the air traffic before the initialization vector is changed and the stream  
 8 cipher is reinitialized. This selection also limits the stream duration to 100 seconds regardless of the  
 9 data rate.”

### 10 **26.4.3.7 Encryption and Decryption Using i-SEC AES Keystream**

11 As specified in Section 11.4.2.7 of HC-SDMA [25] *with the additional text as underlined below*:

12 *i-SEC* encryption shall be performed by XORing the RMU header and payload (see Section 7.3.7.2  
 13 RMU MuxFunction on page 7-45 and Figure 7.9 on page 7-47 of HC-SDMA [25] for definitions) by  
 14 the *i-SEC* keystream before transmission. Therefore, *i-SEC* encryption is not a bandwidth expanding  
 15 transformation. The 16 bit CRC shall be calculated over the encrypted bits. The CRC shall not be  
 16 encrypted. *i-SEC* decryption shall be performed by XORing the received encrypted RMU header and  
 17 payload by the *i-SEC* keystream. The 16 bit CRC check shall be performed on the received encrypted  
 18 bits before decryption.

19 In streams using the basic RMU header, The RMU header and payload shall be aligned such that the  
 20 last payload falls exactly at the end of the relevant keystream buffer. In streams using the minimized  
 21 RMU header, the last burst bit before the CRC bits shall be aligned with the end of the relevant  
 22 keystream buffer. During this process, some of the keystream bytes or bits are possibly left unused  
 23 and they shall be discarded.

24 For example, consider an uplink burst payload that contains 28 bits of basic RMU header and 3  
 25 payload octets {ABC} with a basic RMU header H represented by a bit vector  $[h_0, h_1, \dots, h_{27}]$  where  
 26  $h_0$  is transmitted first and the header is partitioned into four bytes:  $H_0 = [0, 0, 0, 0, h_0, h_1, h_2, h_3]$ ,  $H_1 =$   
 27  $[h_4, h_5, h_6, h_7, h_8, h_9, h_{10}, h_{11}]$ ,  $H_2 = [h_{12}, h_{13}, h_{14}, h_{15}, h_{16}, h_{17}, h_{18}, h_{19}]$ , and  $H_3 = [h_{20}, h_{21}, h_{22}, h_{23}, h_{24}, h_{25}, h_{26}, h_{27}]$ .

## 1 **27 625k-MC Sleep Mode Control Protocol**

2 *This Chapter is an added to the baseline specification HC-SDMA [25].*

3 In Idle mode UT periodically monitors own paging PCH (Refer to 7.3.3 of HC-SDMA [25]) for wake  
4 up requirement. When BS needs to deliver data to UT in Idle Mode, BS shall follow the procedure as  
5 specified in 9.3.4.1 of HC-SDMA [25]. The frequency of pages from BS shall conform to five  
6 discrete levels of paging activity levels (Paging States) transitions from high level to low level: Page  
7 every frame, Page every 8th frame, Page every 64th frame, Page every 512th frame and No paging at  
8 all, as specified in 9.3.4.1.2 of HC-SDMA [25]. BS and UT shall agree during the registration phase  
9 on the length of duration of paging activity before changing to next lower paging activity, thereby  
10 allowing the power conservation feature in 625k-MC mode. This feature may be used to plan the  
11 hardware resources at UT at powerdown mode.

## 28 625k-MC OA & M Radio Network Quality Monitor and Control Enhancement

*This Chapter is an added to the baseline specification HC-SDMA [25].*

625k-MC network systems provide radio network quality monitoring and control functionality. The MIB of 625k-MC mode comprises of the managed objects, attributes, actions, and notifications required to manage a BS. The definition of these managed objects, attributes, actions, and notifications, as well as their structure, is presented below.

### 28.1 625k-MC Mode MIB

#### 28.1.1 Overview

This chapter defines a Management Information Base (MIB) module for managing the 625k-MC mode. Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). The objects in this MIB are defined using the mechanisms specified in the Structure of Management Information (SMI). The MIB module specified is compliant to SMIV2 which is described in RFC 2578 [18], RFC 2579 [19], and RFC 2580 [20].

#### 28.1.2 Definition

```

17
18 IEEE802dot20-625k-MC-MIB DEFINITIONS ::= BEGIN
19
20 IMPORTS
21     OBJECT-TYPE
22     FROM RFC-1212
23     enterprises, Gauge, Counter
24     FROM RFC1155-SMI
25     transmission FROM RFC1213-MIB
26 ;
27
28 IEEE802dot20-625k-MC-MIB MODULE-IDENTITY
29     LAST-UPDATED      " "
30     ORGANIZATION      "IEEE 802.20"
31     CONTACT-INFO      " "
32     DESCRIPTION
33     "The MIB private module for IEEE 802.20 entities"
34     ::= { enterprises 9999 }
35
36 _625k-MCSystem          OBJECT IDENTIFIER
37     -- DESCRIPTION      "System Elements"
38     ::= { IEEE802dot20-625k-MC-MIB 1 }
39
40
41 _625k-MCSysAlarms      OBJECT IDENTIFIER
42     -- DESCRIPTION      "Alarms"
43     ::= { _625k-MCSystem 1 }
44
45
46
47 _625k-MCAlarmScalars   OBJECT IDENTIFIER
48     -- DESCRIPTION      "Alarm Scalars"
49     ::= { _625k-MCSysAlarms 1 }

```

```

1
2
3
4  _625k-MCCommonAlarmStatus          OBJECT-TYPE
5      SYNTAX          INTEGER -- Unsigned32Type
6      ACCESS          read-only
7      STATUS          mandatory
8      DESCRIPTION
9          "Common alarm atatus.
10
11
12
13          (From mibCtl ElementType 16 CommonAlarmStatus)
14          Description for mibCtl Type 14 Unsigned32Type :
15              32 bit unsigned integer.
16          Type derived from mibCtl Type 11 Word32Type :
17              32 bits of raw opaque data.
18          Derived from basic 32 bit word type.
19          "
20      ::= { _625k-MCAAlarmScalars 1 }
21
22
23
24  _625k-MCFailReasonForAlarm          OBJECT-TYPE
25      SYNTAX          INTEGER -- Unsigned32Type
26      ACCESS          read-only
27      STATUS          mandatory
28      DESCRIPTION
29          "Fail reason for alarm.
30
31
32
33          (From mibCtl ElementType 15 FailReasonForAlarm)
34          Description for mibCtl Type 14 Unsigned32Type :
35              32 bit unsigned integer.
36          Type derived from mibCtl Type 11 Word32Type :
37              32 bits of raw opaque data.
38          Derived from basic 32 bit word type.
39          "
40      ::= { _625k-MCAAlarmScalars 2 }
41
42
43
44
45  _625k-MCAAlarmSummaryTable          OBJECT-TYPE
46      SYNTAX SEQUENCE OF _625k-MCAAlarmSummaryTableEntry
47      ACCESS          not-accessible
48      STATUS          mandatory
49      DESCRIPTION    "Alarm Summary Table"
50      ::= { _625k-MCSysAlarms 2 }
51
52
53
54  _625k-MCAAlarmSummaryTableEntry      OBJECT-TYPE
55      SYNTAX          _625k-MCAAlarmSummaryTableEntry
56      ACCESS          not-accessible
57      STATUS          mandatory
58      DESCRIPTION    ""
59      INDEX { _625k-MCAAlarmSummaryTableIndex }
60      ::= { _625k-MCAAlarmSummaryTable 1 }
61
62  _625k-MCAAlarmSummaryTableEntry ::= SEQUENCE {
63      _625k-MCAAlarmSummaryTableIndex    INTEGER, -- AlarmEventType
64      _625k-MCAAlarmSummary              INTEGER -- AlarmStateType
65      }
66
67
68

```

```

1  _625k-MCAlarmSummaryTableIndex          OBJECT-TYPE
2      SYNTAX          INTEGER -- AlarmEventType
3      ACCESS          read-only
4      STATUS          mandatory
5      DESCRIPTION    "
6          Description for mibCtl Type 85 AlarmEventType :
7              Enumeration of alarm event types.
8
9              Defines semantics of events that are also alarms.
10             All alarm events are enumerated first in the list of event types.
11             The highest alarm event index will never be more than 255.
12             [Limits: 0 255 ]
13             Type derived from mibCtl Type 3 EventType :
14                 Enumeration of event types.
15
16                 Defines semantics of events.
17                 An event is re. an event log message.
18                 [Limits: 0 255 ]
19             Type derived from mibCtl Type 14 Unsigned32Type :
20                 32 bit unsigned integer.
21             Type derived from mibCtl Type 11 Word32Type :
22                 32 bits of raw opaque data.
23             Derived from basic 32 bit word type.
24         "
25     ::= { _625k-MCAlarmSummaryTableEntry 1 }
26
27
28
29  _625k-MCAlarmSummary                    OBJECT-TYPE
30      SYNTAX          INTEGER -- AlarmStateType
31      ACCESS          read-only
32      STATUS          mandatory
33      DESCRIPTION    "Summary of all alarms generated by the base station.
34
35
36      Each element contains the summary of a type of alarm (e.g. module
37      over temperature).  When the management station sees that alarm
38      summary is SET, it can, for example, query AlarmModuleOverTemp
39      to see which module(s) is over temperature.
40
41      (From mibCtl ElementType 5210 AlarmSummary)
42      Description for mibCtl Type 80 AlarmStateType :
43          Current state of an alarm.
44
45          This value is CLEARED when
46          the conditions which caused the alarm to occur are taken care of
47          and no longer exist.
48          The value is SET when due to some conditions, the Base Station
49          software decides that an alarm is necessary.
50          Typically (though this may not be true for all alarms
51          or if the alarm changes state too frequently)
52          an event is logged when an alarm is SET and then again when it
53          is CLEARED.
54          [Limits: 0 1 ]
55      Description for mibCtl AlarmStateType 0 CLEARED :
56          No alarm.
57      Description for mibCtl AlarmStateType 1 SET :
58          Alarm is set.
59         "
60     ::= { _625k-MCAlarmSummaryTableEntry 2 }
61
62
63
64  _625k-MCSysFiles                        OBJECT IDENTIFIER
65      -- DESCRIPTION    "Files"
66      ::= { _625k-MCSystem 2}
67
68

```

```

1
2  _625k-MCStatsFiles                OBJECT IDENTIFIER
3  -- DESCRIPTION                    "Statistics file"
4  ::= { _625k-MCSysFiles 1 }
5
6
7
8  _625k-MCStatsUploadURL            OBJECT-TYPE
9  SYNTAX                            OCTET STRING (SIZE(0..64)) -- URLType
10 ACCESS                             read-write
11 STATUS                             mandatory
12 DESCRIPTION
13     "EMS location to upload BS statistics file.
14
15
16
17     (From mibCtl ElementType 2831 StatsUploadURL)
18     Description for mibCtl Type 401 URLType :
19     Universal Resource Locator (URL).
20
21     A Universal Resource Locator (URL) is a text string
22     that specifies a network location for a file.
23     The general format for a URL consists of 2 parts:
24
25     1. Protocol name: lower case letters, followed by a colon.
26     See below for supported protocols.
27     This field may be omitted, to default to the file: protocol.
28
29     2. Additional information, depending on the protocol.
30     For many protocols, a host name is required,
31     which consists of a dotted numerical Internet Protocol (IP)
32     address,
33     or a dotted symbolic name with alphanumerical components,
34     where supported.
35
36     Supported protocols are:
37
38     tftp: is the Trivial File Transfer Protocol.
39     The additional information should begin with two slashes (//)
40     followed by a host name, a slash (/) and a file path.
41     The file path is interpreted by the host system,
42     frequently relative to a special directory set up for this
43     purpose.
44
45     file: is the plain old file protocol.
46     The additional information consists of a file path, which
47     should begin with a slash (/).
48     This is only useful if Base Station has been configured
49     to be an Network File System (NFS) client of the host.
50     The filepath is interpreted on the Base Station, so it
51     must begin with the mount name specified in the NFS configuration.
52     Type derived from mibCtl Type 15 TextType :
53     ASCII or compatible text.
54     Type derived from mibCtl Type 12 OctetType :
55     8 bits of raw opaque data.
56     Derived from basic 8 bit word type.
57     "
58     ::= { _625k-MCStatsFiles 1 }
59
60
61
62  _625k-MCStatsUploadStatus          OBJECT-TYPE
63  SYNTAX                            INTEGER -- FileUploadStatusType
64  ACCESS                             read-only
65  STATUS                             mandatory
66  DESCRIPTION
67     "Stats file upload status.
68

```



```

1
2
3      (From mibCtl ElementType 2832 StatsUploadStatus)
4      Description for mibCtl Type 403 FileUploadStatusType :
5          File upload status.
6
7
8      Description for mibCtl FileUploadStatusType 0 Unknown :
9          File upload status is unknown.
10     Description for mibCtl FileUploadStatusType 1 Missing :
11         File is missing or invalid.
12     Description for mibCtl FileUploadStatusType 2 PartialUpload :
13         File is in the process of being upload to EMS.
14     Description for mibCtl FileUploadStatusType 4 Complete :
15         File is completely uploaded to EMS.
16     Description for mibCtl FileUploadStatusType 5 Failure :
17         Upload process is failure.
18     Description for mibCtl FileUploadStatusType 6 NotManaged :
19         File upload is not being managed.
20     "
21     ::= { _625k-MCStatsFiles 2 }
22
23
24
25     _625k-MCStatsUploadFailReason          OBJECT-TYPE
26     SYNTAX          INTEGER -- FileUploadFailReasonType
27     ACCESS          read-only
28     STATUS          mandatory
29     DESCRIPTION
30         "Reason for last stats file upload failure.
31
32
33
34     (From mibCtl ElementType 2833 StatsUploadFailReason)
35     Description for mibCtl Type 406 FileUploadFailReasonType :
36         Reason for failure to upload a file..
37
38
39     Description for mibCtl FileUploadFailReasonType 0 NoFailure :
40         File upload in progress or completed without problem.
41     Description for mibCtl FileUploadFailReasonType 1 BadPathSpecified :
42         File upload failed because network path not found.
43     Description for mibCtl FileUploadFailReasonType 2 FlashDiskReadError :
44         File upload failed because of flash disk read error.
45     Description for mibCtl FileUploadFailReasonType 3 Aborted :
46         File upload aborted due to change of specification.
47     Description for mibCtl FileUploadFailReasonType 4 WriteError :
48         Error in putting a file.
49     "
50     ::= { _625k-MCStatsFiles 3 }
51
52
53
54     _625k-MCStatsUploadBytes              OBJECT-TYPE
55     SYNTAX          INTEGER -- Unsigned32Type
56     ACCESS          read-only
57     STATUS          mandatory
58     DESCRIPTION
59         "Upload size of BS stats file in bytes.
60
61
62
63     (From mibCtl ElementType 2834 StatsUploadBytes)
64     Description for mibCtl Type 14 Unsigned32Type :
65         32 bit unsigned integer.
66     Type derived from mibCtl Type 11 Word32Type :
67         32 bits of raw opaque data.
68     Derived from basic 32 bit word type.

```

```

1      "
2      ::= { _625k-MCStatsFiles 4 }
3
4
5
6      _625k-MCStatsUploadDate          OBJECT-TYPE
7      SYNTAX          Gauge -- AbsoluteTimeType
8      ACCESS          read-only
9      STATUS          mandatory
10     DESCRIPTION
11         "BS Stats File upload complete time.
12
13
14
15         (From mibCtl ElementType 2835 StatsUploadDate)
16         Description for mibCtl Type 801 AbsoluteTimeType :
17         Absolute time in GPS seconds.
18
19         GPS (Global Positioning System) time in seconds since Jan. 6,
20         1980.
21
22         Note that this differs from UTC (in addition to a possible
23         offset due to starting time) due to leap seconds; see
24         the GpsLeapSecond element.
25         Type derived from mibCtl Type 18 Gauge32Type :
26         32 bits of Gauge data.
27         Derived from basic 32 bit word type.
28     "
29     ::= { _625k-MCStatsFiles 5 }
30
31
32     _625k-MCUploadStatsFile          OBJECT-TYPE
33     SYNTAX          INTEGER -- Unsigned32Type
34     ACCESS          read-write -- REALLY: write-only
35     STATUS          mandatory
36     DESCRIPTION
37         "Upload Stats file.
38
39
40
41         (From mibCtl ElementType 2836 UploadStatsFile)
42         Description for mibCtl Type 14 Unsigned32Type :
43         32 bit unsigned integer.
44         Type derived from mibCtl Type 11 Word32Type :
45         32 bits of raw opaque data.
46         Derived from basic 32 bit word type.
47     "
48     ::= { _625k-MCStatsFiles 6 }
49
50
51     _625k-MCSysInterfaces          OBJECT IDENTIFIER
52     -- DESCRIPTION          "System Interfaces"
53     ::= { _625k-MCSystem 3 }
54
55
56
57     _625k-MCInterfaceNetwork          OBJECT IDENTIFIER
58     -- DESCRIPTION          "Network Interfaces"
59     ::= { _625k-MCSysInterfaces 1 }
60
61
62
63     _625k-MCTypeOfNetworkProtocol          OBJECT-TYPE
64     SYNTAX          INTEGER -- NetworkProtocolType
65     ACCESS          read-only
66     STATUS          mandatory
67     DESCRIPTION
68         "Type of Network Protocol used with the Network.

```

```

1
2     Type of Network Protocol is Ethernet or ATM.
3
4     (From mibCtl ElementType 3002 TypeOfNetworkProtocol)
5     Description for mibCtl Type 214 NetworkProtocolType :
6         Network Protocol type.
7     Description for mibCtl NetworkProtocolType 0 Unknown :
8         Network protocol type is unknown.
9     Description for mibCtl NetworkProtocolType 1 Ethernet :
10        Ethernet interface.
11    Description for mibCtl NetworkProtocolType 2 ATM :
12        ATM interface.
13    "
14    ::= { _625k-MCInterfaceNetwork 1 }
15
16
17
18
19    _625k-MCMgmtNetConfigTable          OBJECT-TYPE
20    SYNTAX SEQUENCE OF _625k-MCMgmtNetConfigTableEntry
21    ACCESS not-accessible
22    STATUS mandatory
23    DESCRIPTION "Mgmt Network Configuration"
24    ::= { _625k-MCInterfaceNetwork 2 }
25
26
27
28    _625k-MCMgmtNetConfigTableEntry      OBJECT-TYPE
29    SYNTAX _625k-MCMgmtNetConfigTableEntry
30    ACCESS not-accessible
31    STATUS mandatory
32    DESCRIPTION ""
33    INDEX { _625k-MCMgmtNetConfigTableIndex }
34    ::= { _625k-MCMgmtNetConfigTable 1 }
35
36    _625k-MCMgmtNetConfigTableEntry ::= SEQUENCE {
37        _625k-MCMgmtNetConfigTableIndex      INTEGER, -- MoNerdAddressType
38        _625k-MCEthernetIPAddress           OCTET STRING (SIZE(0..15)), --
39        IPAddressTextType
40        _625k-MCEthernetIPLocalBits         OCTET STRING (SIZE(0..15)), --
41        IPAddressTextType
42        _625k-MCEthernetHostName           OCTET STRING (SIZE(0..20)) --
43        TextType X 20
44    }
45
46
47
48    _625k-MCMgmtNetConfigTableIndex      OBJECT-TYPE
49    SYNTAX INTEGER -- MoNerdAddressType
50    ACCESS read-only
51    STATUS mandatory
52    DESCRIPTION "
53        Description for mibCtl Type 204 MoNerdAddressType :
54        Base station network component address.
55
56        A network address is a subset of Base Station component addresses,
57        restricted to network components only.
58        Network components interface with a telephony switch or similar.
59        [Limits: 0 1 ]
60        Type derived from mibCtl Type 14 Unsigned32Type :
61        32 bit unsigned integer.
62        Type derived from mibCtl Type 11 Word32Type :
63        32 bits of raw opaque data.
64        Derived from basic 32 bit word type.
65    "
66    ::= { _625k-MCMgmtNetConfigTableEntry 1 }
67
68

```

```

1
2  _625k-MCEthernetIPAddress          OBJECT-TYPE
3      SYNTAX          OCTET STRING (SIZE(0..15)) -- IPAddressTextType
4      ACCESS          read-write
5      STATUS          mandatory
6      DESCRIPTION
7          "Internet Protocol (IP) address for ethernet port of Module.
8
9          This is the actual IP address in use for the ethernet port
10         of a given Module.
11         If IP is not being used on the ethernet port, or there is
12         no ethernet port, then an empty string is provided for this element.
13
14         (From mibCtl ElementType 2811 EthernetIPAddress)
15         Description for mibCtl Type 420 IPAddressTextType :
16             Internet Protocol Address (Text).
17
18             This text must currently be in the dotted abc.def.ghi.jkl format.
19             In the future, hostnames might be allowed.
20         Type derived from mibCtl Type 15 TextType :
21             ASCII or compatible text.
22         Type derived from mibCtl Type 12 OctetType :
23             8 bits of raw opaque data.
24         Derived from basic 8 bit word type.
25         "
26     ::= { _625k-MCMgmtNetConfigTableEntry 2 }
27
28
29
30  _625k-MCEthernetIPLocalBits        OBJECT-TYPE
31      SYNTAX          OCTET STRING (SIZE(0..15)) -- IPAddressTextType
32      ACCESS          read-write
33      STATUS          mandatory
34      DESCRIPTION
35          "Ethernet IP (Internet Protocol) local routing bit count.
36
37          This indicates how many of the low-order bits of
38          the IP address of the ethernet connection are used
39          within the local network.
40          The remaining (high-order) bits are the same for all
41          hosts on the local network.
42          This is used as the first part of the routing algorithm.
43          IP addresses that do not share the upper bits of the ethernet
44          IP address and which are not otherwise resolved will be sent
45          through the gateway, if defined.
46
47          For example, 255.255.255.0
48
49          (From mibCtl ElementType 2812 EthernetIPLocalBits)
50          Description for mibCtl Type 420 IPAddressTextType :
51             Internet Protocol Address (Text).
52
53             This text must currently be in the dotted abc.def.ghi.jkl format.
54             In the future, hostnames might be allowed.
55         Type derived from mibCtl Type 15 TextType :
56             ASCII or compatible text.
57         Type derived from mibCtl Type 12 OctetType :
58             8 bits of raw opaque data.
59         Derived from basic 8 bit word type.
60         "
61     ::= { _625k-MCMgmtNetConfigTableEntry 3 }
62
63
64
65  _625k-MCEthernetHostName            OBJECT-TYPE
66      SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
67      ACCESS          read-write
68      STATUS          mandatory

```

```

1      DESCRIPTION
2          "Ethernet IP host name for module.
3
4          (From mibCtl ElementType 2813 EthernetHostName)
5          Description for mibCtl Type 15 TextType :
6              ASCII or compatible text.
7          Type derived from mibCtl Type 12 OctetType :
8              8 bits of raw opaque data.
9          Derived from basic 8 bit word type.
10         "
11         ::= { _625k-MCMgmtNetConfigTableEntry 4 }
12
13
14
15
16     _625k-MCUserNetConfigTable                OBJECT-TYPE
17         SYNTAX SEQUENCE OF _625k-MCUserNetConfigTableEntry
18         ACCESS not-accessible
19         STATUS mandatory
20         DESCRIPTION "User Network Configuration"
21         ::= { _625k-MCInterfaceNetwork 3 }
22
23
24
25     _625k-MCUserNetConfigTableEntry          OBJECT-TYPE
26         SYNTAX _625k-MCUserNetConfigTableEntry
27         ACCESS not-accessible
28         STATUS mandatory
29         DESCRIPTION ""
30         INDEX { _625k-MCUserNetConfigTableIndex }
31         ::= { _625k-MCUserNetConfigTable 1 }
32
33     _625k-MCUserNetConfigTableEntry ::= SEQUENCE {
34         _625k-MCUserNetConfigTableIndex      INTEGER, -- MoNerdAddressType
35         _625k-MCUserEthernetIPAddress        OCTET STRING (SIZE(0..15)), --
36         IPAddressTextType
37         _625k-MCUserEthernetIPLocalBits      OCTET STRING (SIZE(0..15)), --
38         IPAddressTextType
39         _625k-MCUserEthernetHostName         OCTET STRING (SIZE(0..20)) --
40         TextType X 20
41     }
42
43
44
45     _625k-MCUserNetConfigTableIndex          OBJECT-TYPE
46         SYNTAX INTEGER -- MoNerdAddressType
47         ACCESS read-only
48         STATUS mandatory
49         DESCRIPTION "
50             Description for mibCtl Type 204 MoNerdAddressType :
51             Base station network component address.
52
53             A network address is a subset of Base Station component addresses,
54             restricted to network components only.
55             Network components interface with a telephony switch or similar.
56             [Limits: 0 1 ]
57             Type derived from mibCtl Type 14 Unsigned32Type :
58             32 bit unsigned integer.
59             Type derived from mibCtl Type 11 Word32Type :
60             32 bits of raw opaque data.
61             Derived from basic 32 bit word type.
62         "
63         ::= { _625k-MCUserNetConfigTableEntry 1 }
64
65
66
67     _625k-MCUserEthernetIPAddress            OBJECT-TYPE
68         SYNTAX OCTET STRING (SIZE(0..15)) -- IPAddressTextType

```

```

1     ACCESS          read-write
2     STATUS          mandatory
3     DESCRIPTION
4         "Internet Protocol (IP) address for user ethernet port of Module.
5
6         This is the actual IP address in use for the ethernet port
7         of a given Module.
8         If IP is not being used on the ethernet port, or there is
9         no ethernet port, then an empty string is provided for this element.
10
11        (From mibCtl ElementType 2817 UserEthernetIPAddress)
12        Description for mibCtl Type 420 IPAddressTextType :
13            Internet Protocol Address (Text).
14
15            This text must currently be in the dotted abc.def.ghi.jkl format.
16            In the future, hostnames might be allowed.
17        Type derived from mibCtl Type 15 TextType :
18            ASCII or compatible text.
19        Type derived from mibCtl Type 12 OctetType :
20            8 bits of raw opaque data.
21        Derived from basic 8 bit word type.
22        "
23    ::= { _625k-MCUserNetConfigTableEntry 2 }
24
25
26
27    _625k-MCUserEthernetIPLocalBits          OBJECT-TYPE
28        SYNTAX          OCTET STRING (SIZE(0..15)) -- IPAddressTextType
29        ACCESS          read-write
30        STATUS          mandatory
31        DESCRIPTION
32            "Ethernet IP subnet mask for user network.
33
34            Ethernet IP subnet mask for user network
35
36            (From mibCtl ElementType 2818 UserEthernetIPLocalBits)
37            Description for mibCtl Type 420 IPAddressTextType :
38                Internet Protocol Address (Text).
39
40                This text must currently be in the dotted abc.def.ghi.jkl format.
41                In the future, hostnames might be allowed.
42            Type derived from mibCtl Type 15 TextType :
43                ASCII or compatible text.
44            Type derived from mibCtl Type 12 OctetType :
45                8 bits of raw opaque data.
46            Derived from basic 8 bit word type.
47            "
48    ::= { _625k-MCUserNetConfigTableEntry 3 }
49
50
51
52    _625k-MCUserEthernetHostName          OBJECT-TYPE
53        SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
54        ACCESS          read-write
55        STATUS          mandatory
56        DESCRIPTION
57            "User ethernet IP host name for module.
58
59            (From mibCtl ElementType 2819 UserEthernetHostName)
60            Description for mibCtl Type 15 TextType :
61                ASCII or compatible text.
62            Type derived from mibCtl Type 12 OctetType :
63                8 bits of raw opaque data.
64            Derived from basic 8 bit word type.
65            "
66    ::= { _625k-MCUserNetConfigTableEntry 4 }
67
68

```

```

1
2
3  _625k-MCUserNetStatusTable          OBJECT-TYPE
4      SYNTAX SEQUENCE OF _625k-MCUserNetStatusTableEntry
5      ACCESS          not-accessible
6      STATUS          mandatory
7      DESCRIPTION    "Network Status"
8      ::= { _625K-MCInterfaceNetwork 4 }
9
10
11
12 _625k-MCUserNetStatusTableEntry      OBJECT-TYPE
13     SYNTAX          _625k-MCUserNetStatusTableEntry
14     ACCESS          not-accessible
15     STATUS          mandatory
16     DESCRIPTION    ""
17     INDEX { _625k-MCUserNetStatusTableIndex }
18     ::= { _625k-MCUserNetStatusTable 1 }
19
20 _625k-MCUserNetStatusTableEntry ::= SEQUENCE {
21     _625k-MCUserNetStatusTableIndex      INTEGER, -- MoNerdAddressType
22     _625k-MCNetworkInOctets              Counter, -- Counter32Type
23     _625k-MCNetworkOutOctets             Counter -- Counter32Type
24 }
25
26
27
28 _625k-MCUserNetStatusTableIndex        OBJECT-TYPE
29     SYNTAX          INTEGER -- MoNerdAddressType
30     ACCESS          read-only
31     STATUS          mandatory
32     DESCRIPTION    "
33     Description for mibCtl Type 204 MoNerdAddressType :
34     Base station network component address.
35
36     A network address is a subset of Base Station component addresses,
37     restricted to network components only.
38     Network components interface with a telephony switch or similar.
39     [Limits: 0 1 ]
40     Type derived from mibCtl Type 14 Unsigned32Type :
41     32 bit unsigned integer.
42     Type derived from mibCtl Type 11 Word32Type :
43     32 bits of raw opaque data.
44     Derived from basic 32 bit word type.
45     "
46     ::= { _625k-MCUserNetStatusTableEntry 1 }
47
48
49
50 _625k-MCNetworkInOctets                OBJECT-TYPE
51     SYNTAX          Counter -- Counter32Type
52     ACCESS          read-only
53     STATUS          mandatory
54     DESCRIPTION    "In octets user data of network.
55
56     (From mibCtl ElementType 1000 NetworkInOctets)
57     Description for mibCtl Type 19 Counter32Type :
58     32 bits of Counter data.
59     Derived from basic 32 bit word type.
60     "
61     ::= { _625k-MCUserNetStatusTableEntry 2 }
62
63
64
65
66
67
68 _625k-MCNetworkOutOctets                OBJECT-TYPE

```

```

1      SYNTAX          Counter -- Counter32Type
2      ACCESS          read-only
3      STATUS          mandatory
4      DESCRIPTION
5          "Out octets user data of network.
6
7
8
9          (From mibCtl ElementType 1001 NetworkOutOctets)
10         Description for mibCtl Type 19 Counter32Type :
11             32 bits of Counter data.
12         Derived from basic 32 bit word type.
13         "
14     ::= { _625k-MCUserNetStatusTableEntry 3 }
15
16
17
18
19     _625k-MCL2TPConfigTable          OBJECT-TYPE
20     SYNTAX SEQUENCE OF _625k-MCL2TPConfigTableEntry
21     ACCESS          not-accessible
22     STATUS          mandatory
23     DESCRIPTION    "L2TP Configuration Table"
24     ::= { _625k-MCInterfaceNetwork 5 }
25
26
27
28     _625k-MCL2TPConfigTableEntry      OBJECT-TYPE
29     SYNTAX          _625k-MCL2TPConfigTableEntry
30     ACCESS          not-accessible
31     STATUS          mandatory
32     DESCRIPTION    ""
33     INDEX { _625k-MCL2TPConfigTableIndex }
34     ::= { _625k-MCL2TPConfigTable 1 }
35
36     _625k-MCL2TPConfigTableEntry ::= SEQUENCE {
37         _625k-MCL2TPConfigTableIndex      INTEGER, -- MoNerdAddressType
38         _625k-MCL2TPPeerName              OCTET STRING (SIZE(0..20)), --
39         TextType X 20
40         _625k-MCL2TPPeerIPAddress         OCTET STRING (SIZE(0..15)), --
41         IPAddressTextType
42         _625k-MCL2TPAVPHostName           OCTET STRING (SIZE(0..20)), --
43         TextType X 20
44         _625k-MCL2TPAVPChallAndRes       OCTET STRING (SIZE(0..20)) --
45         TextType X 20
46     }
47
48
49
50     _625k-MCL2TPConfigTableIndex      OBJECT-TYPE
51     SYNTAX          INTEGER -- MoNerdAddressType
52     ACCESS          read-only
53     STATUS          mandatory
54     DESCRIPTION    "
55         Description for mibCtl Type 204 MoNerdAddressType :
56         Base station network component address.
57
58         A network address is a subset of Base Station component addresses,
59         restricted to network components only.
60         Network components interface with a telephony switch or similar.
61         [Limits: 0 1 ]
62         Type derived from mibCtl Type 14 Unsigned32Type :
63             32 bit unsigned integer.
64         Type derived from mibCtl Type 11 Word32Type :
65             32 bits of raw opaque data.
66         Derived from basic 32 bit word type.
67         "
68     ::= { _625k-MCL2TPConfigTableEntry 1 }

```



```

1
2
3
4  _625k-MCL2TPPeerName                OBJECT-TYPE
5      SYNTAX                OCTET STRING (SIZE(0..20)) -- TextType X 20
6      ACCESS                 read-write
7      STATUS                 mandatory
8      DESCRIPTION
9          "L2TP peer name.
10
11         Tunnel switch host name
12
13         (From mibCtl ElementType 2000 L2TPPeerName)
14         Description for mibCtl Type 15 TextType :
15             ASCII or compatible text.
16         Type derived from mibCtl Type 12 OctetType :
17             8 bits of raw opaque data.
18         Derived from basic 8 bit word type.
19         "
20     ::= { _625k-MCL2TPConfigTableEntry 2 }
21
22
23
24  _625k-MCL2TPPeerIPAddress            OBJECT-TYPE
25      SYNTAX                OCTET STRING (SIZE(0..15)) -- IPAddressTextType
26      ACCESS                 read-write
27      STATUS                 mandatory
28      DESCRIPTION
29          "L2TP peer IP Address.
30
31         Tunnel switch IP Address
32
33         (From mibCtl ElementType 2001 L2TPPeerIPAddress)
34         Description for mibCtl Type 420 IPAddressTextType :
35             Internet Protocol Address (Text).
36
37             This text must currently be in the dotted abc.def.ghi.jkl format.
38             In the future, hostnames might be allowed.
39         Type derived from mibCtl Type 15 TextType :
40             ASCII or compatible text.
41         Type derived from mibCtl Type 12 OctetType :
42             8 bits of raw opaque data.
43         Derived from basic 8 bit word type.
44         "
45     ::= { _625k-MCL2TPConfigTableEntry 3 }
46
47
48
49  _625k-MCL2TPAVPHostName              OBJECT-TYPE
50      SYNTAX                OCTET STRING (SIZE(0..20)) -- TextType X 20
51      ACCESS                 read-write
52      STATUS                 mandatory
53      DESCRIPTION
54          "BS host name using L2TP.
55
56
57         (From mibCtl ElementType 2008 L2TPAVPHostName)
58         Description for mibCtl Type 15 TextType :
59             ASCII or compatible text.
60         Type derived from mibCtl Type 12 OctetType :
61             8 bits of raw opaque data.
62         Derived from basic 8 bit word type.
63         "
64     ::= { _625k-MCL2TPConfigTableEntry 4 }
65
66
67
68

```

```

1  _625k-MCL2TPAVPChallAndRes          OBJECT-TYPE
2      SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
3      ACCESS          read-write
4      STATUS          mandatory
5      DESCRIPTION
6          "AVP challenge and response name.
7
8
9
10         (From mibCtl ElementType 2012 L2TPAVPChallAndRes)
11         Description for mibCtl Type 15 TextType :
12             ASCII or compatible text.
13         Type derived from mibCtl Type 12 OctetType :
14             8 bits of raw opaque data.
15         Derived from basic 8 bit word type.
16         "
17         ::= { _625k-MCL2TPConfigTableEntry 5 }
18
19
20
21
22  _625k-MCL2TPStatusTable              OBJECT-TYPE
23      SYNTAX SEQUENCE OF _625k-MCL2TPStatusTableEntry
24      ACCESS          not-accessible
25      STATUS          mandatory
26      DESCRIPTION    "L2TP Status Table"
27      ::= { _625k-MCInterfaceNetwork 6 }
28
29
30
31  _625k-MCL2TPStatusTableEntry         OBJECT-TYPE
32      SYNTAX          _625k-MCL2TPStatusTableEntry
33      ACCESS          not-accessible
34      STATUS          mandatory
35      DESCRIPTION    ""
36      INDEX          { _625k-MCL2TPStatusTableIndex }
37      ::= { _625k-MCL2TPStatusTable 1 }
38
39  _625k-MCL2TPStatusTableEntry ::= SEQUENCE {
40      _625k-MCL2TPStatusTableIndex      INTEGER, -- MoNerdAddressType
41      _625k-MCL2TPActiveSession          INTEGER, -- Unsigned32Type
42      _625k-MCL2TPActiveTunnel           INTEGER -- Unsigned32Type
43  }
44
45
46
47  _625k-MCL2TPStatusTableIndex         OBJECT-TYPE
48      SYNTAX          INTEGER -- MoNerdAddressType
49      ACCESS          read-only
50      STATUS          mandatory
51      DESCRIPTION    "
52          Description for mibCtl Type 204 MoNerdAddressType :
53              Base station network component address.
54
55              A network address is a subset of Base Station component addresses,
56              restricted to network components only.
57              Network components interface with a telephony switch or similar.
58              [Limits: 0 1 ]
59              Type derived from mibCtl Type 14 Unsigned32Type :
60                  32 bit unsigned integer.
61              Type derived from mibCtl Type 11 Word32Type :
62                  32 bits of raw opaque data.
63              Derived from basic 32 bit word type.
64              "
65      ::= { _625k-MCL2TPStatusTableEntry 1 }
66
67
68

```

```

1  _625k-MCL2TPActiveSession          OBJECT-TYPE
2      SYNTAX          INTEGER -- Unsigned32Type
3      ACCESS          read-only
4      STATUS          mandatory
5      DESCRIPTION
6          "L2TP active session.
7
8
9
10         (From mibCtl ElementType 2013 L2TPActiveSession)
11         Description for mibCtl Type 14 Unsigned32Type :
12             32 bit unsigned integer.
13         Type derived from mibCtl Type 11 Word32Type :
14             32 bits of raw opaque data.
15         Derived from basic 32 bit word type.
16         "
17     ::= { _625k-MCL2TPStatusTableEntry 2 }
18
19
20
21  _625k-MCL2TPActiveTunnel           OBJECT-TYPE
22      SYNTAX          INTEGER -- Unsigned32Type
23      ACCESS          read-only
24      STATUS          mandatory
25      DESCRIPTION
26          "L2TP active tunnel.
27
28
29
30         (From mibCtl ElementType 2014 L2TPActiveTunnel)
31         Description for mibCtl Type 14 Unsigned32Type :
32             32 bit unsigned integer.
33         Type derived from mibCtl Type 11 Word32Type :
34             32 bits of raw opaque data.
35         Derived from basic 32 bit word type.
36         "
37     ::= { _625k-MCL2TPStatusTableEntry 3 }
38
39
40
41
42  _625k-MCATMConfigTable             OBJECT-TYPE
43      SYNTAX SEQUENCE OF _625k-MCATMConfigTableEntry
44      ACCESS          not-accessible
45      STATUS          mandatory
46      DESCRIPTION    "ATM Configuration Table"
47     ::= { _625k-MCInterfaceNetwork 7 }
48
49
50
51  _625k-MCATMConfigTableEntry        OBJECT-TYPE
52      SYNTAX          _625k-MCATMConfigTableEntry
53      ACCESS          not-accessible
54      STATUS          mandatory
55      DESCRIPTION    ""
56      INDEX          { _625k-MCATMConfigTableIndex }
57     ::= { _625k-MCATMConfigTable 1 }
58
59  _625k-MCATMConfigTableEntry ::= SEQUENCE {
60      _625k-MCATMConfigTableIndex    INTEGER, -- MoNerdAddressType
61      _625k-MCATmAddress              OCTET STRING (SIZE(0..40)), --
62      TextType X 40
63      _625k-MCATmVCTypes              INTEGER, -- AtmVCType
64      _625k-MCATmFrameTypes          INTEGER, -- AtmFrameType
65      _625k-MCATmUNIVersion          INTEGER, -- AtmUNIVersionType
66      _625k-MCATmLineStatus          INTEGER, -- LineStatusType
67      _625k-MCATmParameterFailReason INTEGER, -- Unsigned32Type
68      _625k-MCATmOpenChannelFailReason INTEGER, -- Unsigned32Type

```

```

1      _625k-MCAtmChannelNumber          INTEGER, -- Unsigned32Type
2      _625k-MCAtmAlarmCauseRegister    INTEGER, -- Unsigned32Type
3      _625k-MCAtmPHYIntrCauseRegister  INTEGER -- Unsigned32Type
4      }
5
6
7
8      _625k-MCATMConfigTableIndex        OBJECT-TYPE
9      SYNTAX          INTEGER -- MoNerdAddressType
10     ACCESS          read-only
11     STATUS          mandatory
12     DESCRIPTION    "
13         Description for mibCtl Type 204 MoNerdAddressType :
14         Base station network component address.
15
16         A network address is a subset of Base Station component addresses,
17         restricted to network components only.
18         Network components interface with a telephony switch or similar.
19         [Limits: 0 1 ]
20         Type derived from mibCtl Type 14 Unsigned32Type :
21         32 bit unsigned integer.
22         Type derived from mibCtl Type 11 Word32Type :
23         32 bits of raw opaque data.
24         Derived from basic 32 bit word type.
25     "
26     ::= { _625k-MCATMConfigTableEntry 1 }
27
28
29
30     _625k-MCAtmAddress                  OBJECT-TYPE
31     SYNTAX          OCTET STRING (SIZE(0..40)) -- TextType X 40
32     ACCESS          read-write
33     STATUS          mandatory
34     DESCRIPTION    "
35         "Atm Address.
36
37
38
39         (From mibCtl ElementType 1950 AtmAddress)
40         Description for mibCtl Type 15 TextType :
41         ASCII or compatible text.
42         Type derived from mibCtl Type 12 OctetType :
43         8 bits of raw opaque data.
44         Derived from basic 8 bit word type.
45     "
46     ::= { _625k-MCATMConfigTableEntry 2 }
47
48
49
50     _625k-MCAtmVCTypes                  OBJECT-TYPE
51     SYNTAX          INTEGER -- AtmVCType
52     ACCESS          read-write
53     STATUS          mandatory
54     DESCRIPTION    "
55         "Atm VC Type.
56
57
58
59         (From mibCtl ElementType 1951 AtmVCTypes)
60         Description for mibCtl Type 300 AtmVCType :
61         ATM VC Type.
62         Description for mibCtl AtmVCType 0 Unknown :
63         ATM VC Type is Unknown.
64         Description for mibCtl AtmVCType 1 PVC :
65         ATM VC Type is PVC.
66         Description for mibCtl AtmVCType 2 SVC :
67         ATM VC Type is SVC.
68         Description for mibCtl AtmVCType 3 PVC SVC :

```

```

1         ATM VC Type is PVC & SVC.
2     "
3     ::= { _625k-MCATMConfigTableEntry 3 }
4
5
6
7     _625k-MCAtmFrameTypes                                OBJECT-TYPE
8     SYNTAX          INTEGER -- AtmFrameType
9     ACCESS          read-write
10    STATUS          mandatory
11    DESCRIPTION
12        "Atm Frame Type.
13
14
15
16        (From mibCtl ElementType 1952 AtmFrameTypes)
17    Description for mibCtl Type 301 AtmFrameType :
18        ATM Frame Type.
19    Description for mibCtl AtmFrameType 0 Unknown :
20        ATM Frame Type is Unknown.
21    Description for mibCtl AtmFrameType 1 OC48 :
22        ATM Frame Type is OC48.
23    Description for mibCtl AtmFrameType 2 OC36 :
24        ATM Frame Type is OC36.
25    Description for mibCtl AtmFrameType 3 OC24 :
26        ATM Frame Type is OC24.
27    Description for mibCtl AtmFrameType 4 OC18 :
28        ATM Frame Type is OC18.
29    Description for mibCtl AtmFrameType 5 OC12 :
30        ATM Frame Type is OC12.
31    Description for mibCtl AtmFrameType 6 OC9 :
32        ATM Frame Type is OC9.
33    Description for mibCtl AtmFrameType 7 OC3 :
34        ATM Frame Type is OC3.
35    Description for mibCtl AtmFrameType 8 OC1 :
36        ATM Frame Type is OC1.
37    Description for mibCtl AtmFrameType 9 STM16 :
38        ATM Frame Type is STM16.
39    Description for mibCtl AtmFrameType 10 STM4 :
40        ATM Frame Type is STM4.
41    Description for mibCtl AtmFrameType 11 STM1 :
42        ATM Frame Type is STM1.
43    Description for mibCtl AtmFrameType 12 DS3 :
44        ATM Frame Type is DS3.
45    Description for mibCtl AtmFrameType 13 DS2 :
46        ATM Frame Type is DS2.
47    Description for mibCtl AtmFrameType 14 DS1 :
48        ATM Frame Type is DS1.
49    Description for mibCtl AtmFrameType 15 DS0 :
50        ATM Frame Type is DS0.
51    Description for mibCtl AtmFrameType 16 E3 :
52        ATM Frame Type is E3.
53    Description for mibCtl AtmFrameType 17 E2 :
54        ATM Frame Type is E2.
55    Description for mibCtl AtmFrameType 18 E1 :
56        ATM Frame Type is E1.
57    Description for mibCtl AtmFrameType 19 E0 :
58        ATM Frame Type is E0.
59    "
60    ::= { _625k-MCATMConfigTableEntry 4 }
61
62
63
64    _625k-MCAtmUNIVersion                                OBJECT-TYPE
65    SYNTAX          INTEGER -- AtmUNIVersionType
66    ACCESS          read-write
67    STATUS          mandatory
68    DESCRIPTION

```

```

1      "Atm UNI version.
2
3
4
5      (From mibCtl ElementType 1953 AtmUNIVersion)
6      Description for mibCtl Type 302 AtmUNIVersionType :
7          ATM UNI Version Type.
8      Description for mibCtl AtmUNIVersionType 0 Unknown :
9          ATM UNI Version Type is Unknown.
10     Description for mibCtl AtmUNIVersionType 1 V30 :
11         ATM UNI Version Type is 3.0.
12     Description for mibCtl AtmUNIVersionType 2 V31 :
13         ATM UNI Version Type is 3.1.
14     Description for mibCtl AtmUNIVersionType 3 V40 :
15         ATM UNI Version Type is 4.0.
16     "
17     ::= { _625k-MCATMConfigTableEntry 5 }
18
19
20
21     _625k-MCAtmLineStatus          OBJECT-TYPE
22     SYNTAX          INTEGER -- LineStatusType
23     ACCESS          read-only
24     STATUS          mandatory
25     DESCRIPTION
26         "Atm line status.
27
28         ATM line status
29
30         (From mibCtl ElementType 1956 AtmLineStatus)
31     Description for mibCtl Type 72 LineStatusType :
32         Line status type.
33     Description for mibCtl LineStatusType 0 LinkUp :
34         Line status is link up.
35     Description for mibCtl LineStatusType 1 LinkDown :
36         Line status is link down.
37     "
38     ::= { _625k-MCATMConfigTableEntry 6 }
39
40
41
42     _625k-MCAtmParameterFailReason  OBJECT-TYPE
43     SYNTAX          INTEGER -- Unsigned32Type
44     ACCESS          read-only
45     STATUS          mandatory
46     DESCRIPTION
47         "Atm parameter fail reason.
48
49         ATM parameter fail reason
50
51         (From mibCtl ElementType 1957 AtmParameterFailReason)
52     Description for mibCtl Type 14 Unsigned32Type :
53         32 bit unsigned integer.
54     Type derived from mibCtl Type 11 Word32Type :
55         32 bits of raw opaque data.
56     Derived from basic 32 bit word type.
57     "
58     ::= { _625k-MCATMConfigTableEntry 7 }
59
60
61
62     _625k-MCAtmOpenChannelFailReason OBJECT-TYPE
63     SYNTAX          INTEGER -- Unsigned32Type
64     ACCESS          read-only
65     STATUS          mandatory
66     DESCRIPTION
67         "Atm open channel fail reason.
68

```

```

1       ATM open channel fail reason
2
3       (From mibCtl ElementType 1958 AtmOpenChannelFailReason)
4       Description for mibCtl Type 14 Unsigned32Type :
5           32 bit unsigned integer.
6       Type derived from mibCtl Type 11 Word32Type :
7           32 bits of raw opaque data.
8       Derived from basic 32 bit word type.
9       "
10      ::= { _625k-MCATMConfigTableEntry 8 }
11
12
13
14      _625k-MCATmChannelNumber          OBJECT-TYPE
15      SYNTAX          INTEGER -- Unsigned32Type
16      ACCESS          read-only
17      STATUS          mandatory
18      DESCRIPTION
19          "Atm open channel fail reason.
20
21      ATM open channel fail reason
22
23      (From mibCtl ElementType 1959 AtmChannelNumber)
24      Description for mibCtl Type 14 Unsigned32Type :
25          32 bit unsigned integer.
26      Type derived from mibCtl Type 11 Word32Type :
27          32 bits of raw opaque data.
28      Derived from basic 32 bit word type.
29      "
30      ::= { _625k-MCATMConfigTableEntry 9 }
31
32
33
34      _625k-MCATmAlarmCauseRegister      OBJECT-TYPE
35      SYNTAX          INTEGER -- Unsigned32Type
36      ACCESS          read-only
37      STATUS          mandatory
38      DESCRIPTION
39          "Atm alarm cause register.
40
41      ATM alarm cause register
42
43      (From mibCtl ElementType 1960 AtmAlarmCauseRegister)
44      Description for mibCtl Type 14 Unsigned32Type :
45          32 bit unsigned integer.
46      Type derived from mibCtl Type 11 Word32Type :
47          32 bits of raw opaque data.
48      Derived from basic 32 bit word type.
49      "
50      ::= { _625k-MCATMConfigTableEntry 10 }
51
52
53
54      _625k-MCATmPHYIntrCauseRegister     OBJECT-TYPE
55      SYNTAX          INTEGER -- Unsigned32Type
56      ACCESS          read-only
57      STATUS          mandatory
58      DESCRIPTION
59          "Atm PHY Interrupt cause register.
60
61      ATM PHY interrupt cause register
62
63      (From mibCtl ElementType 1961 AtmPHYIntrCauseRegister)
64      Description for mibCtl Type 14 Unsigned32Type :
65          32 bit unsigned integer.
66      Type derived from mibCtl Type 11 Word32Type :
67          32 bits of raw opaque data.
68      Derived from basic 32 bit word type.

```

```

1      "
2      ::= { _625k-MCATMConfigTableEntry 11 }
3
4
5
6
7      _625k-MCA10ConfigTable                                OBJECT-TYPE
8          SYNTAX SEQUENCE OF _625k-MCA10ConfigTableEntry
9          ACCESS not-accessible
10         STATUS mandatory
11         DESCRIPTION "A10 Configuration Table"
12         ::= { _625k-MCInterfaceNetwork 9 }
13
14
15
16     _625k-MCA10ConfigTableEntry                            OBJECT-TYPE
17         SYNTAX _625k-MCA10ConfigTableEntry
18         ACCESS not-accessible
19         STATUS mandatory
20         DESCRIPTION ""
21         INDEX { _625k-MCA10ConfigTableIndex }
22         ::= { _625k-MCA10ConfigTable 1 }
23
24     _625k-MCA10ConfigTableEntry ::= SEQUENCE {
25         _625k-MCA10ConfigTableIndex                      INTEGER, -- MoNerdAddressType
26         _625k-MC3GPP2PriPDSNIPAddress                    OCTET STRING (SIZE(0..15)), --
27         IPAddressTextType
28         _625k-MC3GPP2PriPDSNSharedSecret                OCTET STRING (SIZE(0..64)), --
29         TextType X 64
30         _625k-MC3GPP2PriPDSNSPI                          INTEGER, -- Unsigned32Type
31         _625k-MC3GPP2SecPDSNIPAddress                    OCTET STRING (SIZE(0..15)), --
32         IPAddressTextType
33         _625k-MC3GPP2SecPDSNSharedSecret                OCTET STRING (SIZE(0..64)), --
34         TextType X 64
35         _625k-MC3GPP2SecPDSNSPI                          INTEGER -- Unsigned32Type
36     }
37
38
39
40     _625k-MCA10ConfigTableIndex                            OBJECT-TYPE
41         SYNTAX INTEGER -- MoNerdAddressType
42         ACCESS read-only
43         STATUS mandatory
44         DESCRIPTION "
45             Description for mibCtl Type 204 MoNerdAddressType :
46             Base station network component address.
47
48             A network address is a subset of Base Station component addresses,
49             restricted to network components only.
50             Network components interface with a telephony switch or similar.
51             [Limits: 0 1 ]
52             Type derived from mibCtl Type 14 Unsigned32Type :
53             32 bit unsigned integer.
54             Type derived from mibCtl Type 11 Word32Type :
55             32 bits of raw opaque data.
56             Derived from basic 32 bit word type.
57         "
58         ::= { _625k-MCA10ConfigTableEntry 1 }
59
60
61
62     _625k-MC3GPP2PriPDSNIPAddress                            OBJECT-TYPE
63         SYNTAX OCTET STRING (SIZE(0..15)) -- IPAddressTextType
64         ACCESS read-write
65         STATUS mandatory
66         DESCRIPTION
67             "3GPP2 primary PDSN IP address.
68

```



```

1      3GPP2 Primary PDSN IP Address
2
3      (From mibCtl ElementType 2100 3GPP2PriPDSNIPAddress)
4      Description for mibCtl Type 420 IPAddressTextType :
5          Internet Protocol Address (Text).
6
7          This text must currently be in the dotted abc.def.ghi.jkl format.
8          In the future, hostnames might be allowed.
9      Type derived from mibCtl Type 15 TextType :
10         ASCII or compatible text.
11     Type derived from mibCtl Type 12 OctetType :
12         8 bits of raw opaque data.
13     Derived from basic 8 bit word type.
14     "
15     ::= { _625k-MCA10ConfigTableEntry 2 }
16
17
18
19     _625k-MC3GPP2PriPDSNSharedSecret          OBJECT-TYPE
20     SYNTAX          OCTET STRING (SIZE(0..64)) -- TextType X 64
21     ACCESS          read-write
22     STATUS          mandatory
23     DESCRIPTION
24         "3GPP2 primary PDSN shared secret.
25
26         3GPP2 primary PDSN shared secret
27
28         (From mibCtl ElementType 2101 3GPP2PriPDSNSharedSecret)
29     Description for mibCtl Type 15 TextType :
30         ASCII or compatible text.
31     Type derived from mibCtl Type 12 OctetType :
32         8 bits of raw opaque data.
33     Derived from basic 8 bit word type.
34     "
35     ::= { _625k-MCA10ConfigTableEntry 3 }
36
37
38
39     _625k-MC3GPP2PriPDSNSPI                    OBJECT-TYPE
40     SYNTAX          INTEGER -- Unsigned32Type
41     ACCESS          read-write
42     STATUS          mandatory
43     DESCRIPTION
44         "3GPP2 primary PDSN SPI.
45
46         3GPP2 primary PDSN SPI
47
48         (From mibCtl ElementType 2102 3GPP2PriPDSNSPI)
49     Description for mibCtl Type 14 Unsigned32Type :
50         32 bit unsigned integer.
51     Type derived from mibCtl Type 11 Word32Type :
52         32 bits of raw opaque data.
53     Derived from basic 32 bit word type.
54     "
55     ::= { _625k-MCA10ConfigTableEntry 4 }
56
57
58
59     _625k-MC3GPP2SecPDSNIPAddress              OBJECT-TYPE
60     SYNTAX          OCTET STRING (SIZE(0..15)) -- IPAddressTextType
61     ACCESS          read-write
62     STATUS          mandatory
63     DESCRIPTION
64         "3GPP2 secondary PDSN IP address.
65
66         3GPP2 secondary PDSN IP address
67
68         (From mibCtl ElementType 2103 3GPP2SecPDSNIPAddress)

```

```

1      Description for mibCtl Type 420 IPAddressTextType :
2          Internet Protocol Address (Text).
3
4          This text must currently be in the dotted abc.def.ghi.jkl format.
5          In the future, hostnames might be allowed.
6      Type derived from mibCtl Type 15 TextType :
7          ASCII or compatible text.
8      Type derived from mibCtl Type 12 OctetType :
9          8 bits of raw opaque data.
10     Derived from basic 8 bit word type.
11     "
12     ::= { _625k-MCA10ConfigTableEntry 5 }
13
14
15
16     _625k-MC3GPP2SecPDSNSharedSecret          OBJECT-TYPE
17     SYNTAX          OCTET STRING (SIZE(0..64)) -- TextType X 64
18     ACCESS          read-write
19     STATUS          mandatory
20     DESCRIPTION
21         "3GPP2 secondary PDSN shared secret.
22
23         3GPP2 secondary PDSN shared secret
24
25         (From mibCtl ElementType 2104 3GPP2SecPDSNSharedSecret)
26     Description for mibCtl Type 15 TextType :
27         ASCII or compatible text.
28     Type derived from mibCtl Type 12 OctetType :
29         8 bits of raw opaque data.
30     Derived from basic 8 bit word type.
31     "
32     ::= { _625k-MCA10ConfigTableEntry 6 }
33
34
35
36     _625k-MC3GPP2SecPDSNSPI                    OBJECT-TYPE
37     SYNTAX          INTEGER -- Unsigned32Type
38     ACCESS          read-write
39     STATUS          mandatory
40     DESCRIPTION
41         "3GPP2 secondary PDSN SPI.
42
43         3GPP2 secondary PDSN SPI
44
45         (From mibCtl ElementType 2105 3GPP2SecPDSNSPI)
46     Description for mibCtl Type 14 Unsigned32Type :
47         32 bit unsigned integer.
48     Type derived from mibCtl Type 11 Word32Type :
49         32 bits of raw opaque data.
50     Derived from basic 32 bit word type.
51     "
52     ::= { _625k-MCA10ConfigTableEntry 7 }
53
54
55
56
57     _625k-MCA10StatusTable                    OBJECT-TYPE
58     SYNTAX          SEQUENCE OF _625k-MCA10StatusTableEntry
59     ACCESS          not-accessible
60     STATUS          mandatory
61     DESCRIPTION          "A10 Status Table"
62     ::= { _625k-MCInterfaceNetwork 10 }
63
64
65
66     _625k-MCA10StatusTableEntry                OBJECT-TYPE
67     SYNTAX          _625k-MCA10StatusTableEntry
68     ACCESS          not-accessible

```

```

1      STATUS          mandatory
2      DESCRIPTION     ""
3      INDEX { _625k-MCA10StatusTableIndex }
4      ::= { _625k-MCA10StatusTable 1 }
5
6      _625k-MCA10StatusTableEntry ::= SEQUENCE {
7          _625k-MCA10StatusTableIndex      INTEGER, -- MoNerdAddressType
8          _625k-MC3GPP2PDSNIPAddress      OCTET STRING (SIZE(0..15)) --
9      IPAddressTextType
10     }
11
12
13
14     _625k-MCA10StatusTableIndex          OBJECT-TYPE
15     SYNTAX          INTEGER -- MoNerdAddressType
16     ACCESS          read-only
17     STATUS          mandatory
18     DESCRIPTION     "
19         Description for mibCtl Type 204 MoNerdAddressType :
20         Base station network component address.
21
22         A network address is a subset of Base Station component addresses,
23         restricted to network components only.
24         Network components interface with a telephony switch or similar.
25         [Limits: 0 1 ]
26         Type derived from mibCtl Type 14 Unsigned32Type :
27         32 bit unsigned integer.
28         Type derived from mibCtl Type 11 Word32Type :
29         32 bits of raw opaque data.
30         Derived from basic 32 bit word type.
31     "
32     ::= { _625k-MCA10StatusTableEntry 1 }
33
34
35
36     _625k-MC3GPP2PDSNIPAddress          OBJECT-TYPE
37     SYNTAX          OCTET STRING (SIZE(0..15)) -- IPAddressTextType
38     ACCESS          read-only
39     STATUS          mandatory
40     DESCRIPTION     "
41         "3GPP2 PDSN IP Address.
42
43         Current main using PDSN IP address
44
45         (From mibCtl ElementType 2113 3GPP2PDSNIPAddress)
46         Description for mibCtl Type 420 IPAddressTextType :
47         Internet Protocol Address (Text).
48
49         This text must currently be in the dotted abc.def.ghi.jkl format.
50         In the future, hostnames might be allowed.
51         Type derived from mibCtl Type 15 TextType :
52         ASCII or compatible text.
53         Type derived from mibCtl Type 12 OctetType :
54         8 bits of raw opaque data.
55         Derived from basic 8 bit word type.
56     "
57     ::= { _625k-MCA10StatusTableEntry 2 }
58
59
60
61     _625k-MCInterfaceRF          OBJECT IDENTIFIER
62     -- DESCRIPTION     "Radio Frequency Interfaces"
63     ::= { _625k-MCSysInterfaces 2 }
64
65
66
67     _625k-MCCarrierTable          OBJECT-TYPE
68     SYNTAX SEQUENCE OF _625k-MCCarrierTableEntry

```

```

1     ACCESS          not-accessible
2     STATUS          mandatory
3     DESCRIPTION    "Carrier Frequency Table"
4     ::= { _625k-MCInterfaceRF 1 }
5
6
7
8     _625k-MCCarrierTableEntry          OBJECT-TYPE
9     SYNTAX          _625k-MCCarrierTableEntry
10    ACCESS          not-accessible
11    STATUS          mandatory
12    DESCRIPTION    ""
13    INDEX          { _625k-MCCarrierTableIndex }
14    ::= { _625k-MCCarrierTable 1 }
15
16    _625k-MCCarrierTableEntry ::= SEQUENCE {
17        _625k-MCCarrierTableIndex          INTEGER, -- BaseStationCarrierType
18        _625k-MCCarrierUsage              INTEGER -- CarrierUsageType
19    }
20
21
22
23    _625k-MCCarrierTableIndex          OBJECT-TYPE
24    SYNTAX          INTEGER -- BaseStationCarrierType
25    ACCESS          read-only
26    STATUS          mandatory
27    DESCRIPTION    "
28        Description for mibCtl Type 219 BaseStationCarrierType :
29        Base station carrier number.
30
31        Base station carriers are a contiguous set of carriers
32        that are used by the Base Station;
33        they are numbered from 0 to a current maximum of 32-1.
34        [Limits: 0 15 ]
35        Type derived from mibCtl Type 14 Unsigned32Type :
36        32 bit unsigned integer.
37        Type derived from mibCtl Type 11 Word32Type :
38        32 bits of raw opaque data.
39        Derived from basic 32 bit word type.
40    "
41    ::= { _625k-MCCarrierTableEntry 1 }
42
43
44
45    _625k-MCCarrierUsage              OBJECT-TYPE
46    SYNTAX          INTEGER -- CarrierUsageType
47    ACCESS          read-only
48    STATUS          mandatory
49    DESCRIPTION    "Current assigned usage per base station carrier.
50
51
52
53
54        (From mibCtl ElementType 54 CarrierUsage)
55    Description for mibCtl Type 220 CarrierUsageType :
56        The assigned use of a radio carrier.
57
58        A radio carrier is a frequency band.
59        The assigned use of a carrier can be Reserved, Control
60        or Traffic.
61        Base station transmits control information on one of the time
62    slots
63        of given Control carrier.
64        Base station does not transmit anything on Reserved carriers.
65    Description for mibCtl CarrierUsageType 0 NotUse :
66        Not Use for this carrier.
67    Description for mibCtl CarrierUsageType 1 TCH :
68        All timeslots in this carrier are for traffic only.

```

```

1      Description for mibCtl CarrierUsageType 2 TCHBCH :
2          One timeslot in this carrier is for BCH, others for TCH.
3      "
4      ::= { _625k-MCCarrierTableEntry 2 }
5
6
7
8      _625k-MCBSCC                                OBJECT-TYPE
9          SYNTAX          INTEGER -- Unsigned32Type
10         ACCESS         read-write
11         STATUS         mandatory
12         DESCRIPTION
13             "Base Station Color Code.
14
15             (From mibCtl ElementType 61 BSCC)
16             Description for mibCtl Type 14 Unsigned32Type :
17                 32 bit unsigned integer.
18             Type derived from mibCtl Type 11 Word32Type :
19                 32 bits of raw opaque data.
20             Derived from basic 32 bit word type.
21             "
22         ::= { _625k-MCInterfaceRF 2 }
23
24
25
26         _625k-MCBSLowestCarrier                    OBJECT-TYPE
27             SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
28             ACCESS         read-write
29             STATUS         mandatory
30             DESCRIPTION
31                 "The lowest carrier of the base station operating band.
32
33                 This is an extended carrier number
34                 that identifies the lowest carrier of the bandwidth
35                 to which the base station is tuned.
36                 This value cannot be changed while the Base Station state
37                 is Operating.
38
39                 (From mibCtl ElementType 52 BSLowestCarrier)
40                 Description for mibCtl Type 15 TextType :
41                     ASCII or compatible text.
42                 Type derived from mibCtl Type 12 OctetType :
43                     8 bits of raw opaque data.
44                 Derived from basic 8 bit word type.
45                 "
46             ::= { _625k-MCInterfaceRF 3 }
47
48
49
50         _625k-MCBCHModuleAddress                    OBJECT-TYPE
51             SYNTAX          INTEGER -- ModuleAddressType
52             ACCESS         read-only
53             STATUS         mandatory
54             DESCRIPTION
55                 "Which module is handling the broadcast channel.
56
57                 (From mibCtl ElementType 57 BCHModuleAddress)
58                 Description for mibCtl Type 202 ModuleAddressType :
59                     Base station bus slot address.
60
61                     Most components of the Base Station for which data can
62                     be obtained are identified by a ModuleAddressType address
63                     and possibly a subsidiary address.
64                     [Limits: 0 7 ]
65                 Type derived from mibCtl Type 14 Unsigned32Type :
66                     32 bit unsigned integer.
67                 Type derived from mibCtl Type 11 Word32Type :
68                     32 bits of raw opaque data.

```

```

1         Derived from basic 32 bit word type.
2         "
3         ::= { _625k-MCInterfaceRF 4 }
4
5
6
7         _625k-MCBCHCarrierNumber                OBJECT-TYPE
8         SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
9         ACCESS          read-write
10        STATUS          mandatory
11        DESCRIPTION
12        "Number Of BCH Carrier.
13
14        (From mibCtl ElementType 58 BCHCarrierNumber)
15        Description for mibCtl Type 15 TextType :
16        ASCII or compatible text.
17        Type derived from mibCtl Type 12 OctetType :
18        8 bits of raw opaque data.
19        Derived from basic 8 bit word type.
20        "
21        ::= { _625k-MCInterfaceRF 5 }
22
23
24
25        _625k-MCRACHCarrierMask                OBJECT-TYPE
26        SYNTAX          INTEGER -- Unsigned32Type
27        ACCESS          read-write
28        STATUS          mandatory
29        DESCRIPTION
30        "RACH carrier mask.
31
32
33
34        (From mibCtl ElementType 73 RACHCarrierMask)
35        Description for mibCtl Type 14 Unsigned32Type :
36        32 bit unsigned integer.
37        Type derived from mibCtl Type 11 Word32Type :
38        32 bits of raw opaque data.
39        Derived from basic 32 bit word type.
40        "
41        ::= { _625k-MCInterfaceRF 6 }
42
43
44
45        _625k-MCRACHSlotMask                OBJECT-TYPE
46        SYNTAX          INTEGER -- Unsigned32Type
47        ACCESS          read-write
48        STATUS          mandatory
49        DESCRIPTION
50        "RACH slot mask.
51
52
53
54        (From mibCtl ElementType 72 RACHSlotMask)
55        Description for mibCtl Type 14 Unsigned32Type :
56        32 bit unsigned integer.
57        Type derived from mibCtl Type 11 Word32Type :
58        32 bits of raw opaque data.
59        Derived from basic 32 bit word type.
60        "
61        ::= { _625k-MCInterfaceRF 7 }
62
63
64
65        _625k-MCCalibrationInterval                OBJECT-TYPE
66        SYNTAX          INTEGER -- Unsigned32Type
67        ACCESS          read-write
68        STATUS          mandatory

```

```

1 DESCRIPTION
2     "Calibration interval time.
3
4
5
6     (From mibCtl ElementType 75 CalibrationInterval)
7     Description for mibCtl Type 14 Unsigned32Type :
8         32 bit unsigned integer.
9     Type derived from mibCtl Type 11 Word32Type :
10        32 bits of raw opaque data.
11    Derived from basic 32 bit word type.
12    "
13    ::= { _625k-MCInterfaceRF 8 }
14
15
16
17    _625k-MCSpatialParameter                OBJECT-TYPE
18        SYNTAX          INTEGER -- Unsigned32Type
19        ACCESS          read-write
20        STATUS          mandatory
21        DESCRIPTION
22            "Spatial parameter.
23
24
25
26        (From mibCtl ElementType 78 SpatialParameter)
27        Description for mibCtl Type 14 Unsigned32Type :
28            32 bit unsigned integer.
29        Type derived from mibCtl Type 11 Word32Type :
30            32 bits of raw opaque data.
31        Derived from basic 32 bit word type.
32        "
33        ::= { _625k-MCInterfaceRF 9 }
34
35
36
37    _625k-MCCostCalcParameter                OBJECT-TYPE
38        SYNTAX          INTEGER -- Unsigned32Type
39        ACCESS          read-write
40        STATUS          mandatory
41        DESCRIPTION
42            "Cost calculation parameter.
43
44
45
46        (From mibCtl ElementType 79 CostCalcParameter)
47        Description for mibCtl Type 14 Unsigned32Type :
48            32 bit unsigned integer.
49        Type derived from mibCtl Type 11 Word32Type :
50            32 bits of raw opaque data.
51        Derived from basic 32 bit word type.
52        "
53        ::= { _625k-MCInterfaceRF 10 }
54
55
56
57    _625k-MCBSRegistrationCapacity            OBJECT-TYPE
58        SYNTAX          INTEGER -- Unsigned32Type
59        ACCESS          read-write
60        STATUS          mandatory
61        DESCRIPTION
62            "Capacity of UT registration on BS.
63
64
65
66        (From mibCtl ElementType 76 BSRegistrationCapacity)
67        Description for mibCtl Type 14 Unsigned32Type :
68            32 bit unsigned integer.

```

```

1      Type derived from mibCtl Type 11 Word32Type :
2      32 bits of raw opaque data.
3      Derived from basic 32 bit word type.
4      "
5      ::= { _625k-MCInterfaceRF 11 }
6
7
8
9      _625k-MCBSRegistrationTimer          OBJECT-TYPE
10     SYNTAX          INTEGER -- Unsigned32Type
11     ACCESS          read-write
12     STATUS          mandatory
13     DESCRIPTION
14     "Timer of keeping UT registration on BS.
15
16
17     (From mibCtl ElementType 77 BSRegistrationTimer)
18     Description for mibCtl Type 14 Unsigned32Type :
19     32 bit unsigned integer.
20     Type derived from mibCtl Type 11 Word32Type :
21     32 bits of raw opaque data.
22     Derived from basic 32 bit word type.
23     "
24     ::= { _625k-MCInterfaceRF 12 }
25
26
27
28
29     _625k-MCPCHFrequencyHopping          OBJECT-TYPE
30     SYNTAX          INTEGER -- BooleanType
31     ACCESS          read-write
32     STATUS          mandatory
33     DESCRIPTION
34     "Propriety of frequency hopping (PCH).
35
36
37     (From mibCtl ElementType 70 PCHFrequencyHopping)
38     Description for mibCtl Type 16 BooleanType :
39     Truth value, 0=FALSE, 1=TRUE.
40
41     This is a subset of TriStateType; no UNDEFINED value is provided.
42     [Limits: 0 1 ]
43     Description for mibCtl BooleanType 0 FALSE :
44     False.
45     Description for mibCtl BooleanType 1 TRUE :
46     True.
47     "
48     ::= { _625k-MCInterfaceRF 13 }
49
50
51
52
53     _625k-MCTCHFrequencyHopping          OBJECT-TYPE
54     SYNTAX          INTEGER -- BooleanType
55     ACCESS          read-write
56     STATUS          mandatory
57     DESCRIPTION
58     "Propriety of frequency hopping (TCH).
59
60
61     (From mibCtl ElementType 71 TCHFrequencyHopping)
62     Description for mibCtl Type 16 BooleanType :
63     Truth value, 0=FALSE, 1=TRUE.
64
65     This is a subset of TriStateType; no UNDEFINED value is provided.
66     [Limits: 0 1 ]
67     Description for mibCtl BooleanType 0 FALSE :
68

```



```

1         False.
2         Description for mibCtl BooleanType 1 TRUE :
3         True.
4         "
5         ::= { _625k-MCInterfaceRF 14 }
6
7
8
9
10        _625k-MCRFStatusTable                OBJECT-TYPE
11        SYNTAX SEQUENCE OF _625k-MCRFStatusTableEntry
12        ACCESS                not-accessible
13        STATUS                mandatory
14        DESCRIPTION           "RF Status Table"
15        ::= { _625k-MCInterfaceRF 15 }
16
17
18
19        _625k-MCRFStatusTableEntry            OBJECT-TYPE
20        SYNTAX                _625k-MCRFStatusTableEntry
21        ACCESS                not-accessible
22        STATUS                mandatory
23        DESCRIPTION           ""
24        INDEX { _625k-MCRFStatusTableIndex }
25        ::= { _625k-MCRFStatusTable 1 }
26
27        _625k-MCRFStatusTableEntry ::= SEQUENCE {
28            _625k-MCRFStatusTableIndex        INTEGER, -- MoNerdAddressType
29            _625k-MCBSAirBitRateUpLink        INTEGER, -- Unsigned32Type
30            _625k-MCBSAirBitRateDownLink      INTEGER, -- Unsigned32Type
31            _625k-MCBSActiveStream            INTEGER, -- Unsigned32Type
32            _625k-MCBSActiveRegistration      INTEGER -- Unsigned32Type
33        }
34
35
36
37        _625k-MCRFStatusTableIndex            OBJECT-TYPE
38        SYNTAX                INTEGER -- MoNerdAddressType
39        ACCESS                read-only
40        STATUS                mandatory
41        DESCRIPTION           "
42            Description for mibCtl Type 204 MoNerdAddressType :
43            Base station network component address.
44
45            A network address is a subset of Base Station component addresses,
46            restricted to network components only.
47            Network components interface with a telephony switch or similar.
48            [Limits: 0 1 ]
49            Type derived from mibCtl Type 14 Unsigned32Type :
50            32 bit unsigned integer.
51            Type derived from mibCtl Type 11 Word32Type :
52            32 bits of raw opaque data.
53            Derived from basic 32 bit word type.
54            "
55        ::= { _625k-MCRFStatusTableEntry 1 }
56
57
58
59        _625k-MCBSAirBitRateUpLink            OBJECT-TYPE
60        SYNTAX                INTEGER -- Unsigned32Type
61        ACCESS                read-only
62        STATUS                mandatory
63        DESCRIPTION           "Radio bit rate of up link per Modem control board.
64
65
66
67
68        (From mibCtl ElementType 4022 BSAirBitRateUpLink)

```

```

1      Description for mibCtl Type 14 Unsigned32Type :
2          32 bit unsigned integer.
3      Type derived from mibCtl Type 11 Word32Type :
4          32 bits of raw opaque data.
5      Derived from basic 32 bit word type.
6      "
7      ::= { _625k-MCRFStatusTableEntry 2 }
8
9
10
11     _625k-MCBSAirBitRateDownLink          OBJECT-TYPE
12     SYNTAX          INTEGER -- Unsigned32Type
13     ACCESS          read-only
14     STATUS          mandatory
15     DESCRIPTION
16         "Radio bit rate of down link per Modem control board.
17
18
19
20         (From mibCtl ElementType 4023 BS AirBitRateDownLink)
21     Description for mibCtl Type 14 Unsigned32Type :
22         32 bit unsigned integer.
23     Type derived from mibCtl Type 11 Word32Type :
24         32 bits of raw opaque data.
25     Derived from basic 32 bit word type.
26     "
27     ::= { _625k-MCRFStatusTableEntry 3 }
28
29
30
31     _625k-MCBSActiveStream                OBJECT-TYPE
32     SYNTAX          INTEGER -- Unsigned32Type
33     ACCESS          read-only
34     STATUS          mandatory
35     DESCRIPTION
36         "Number of streams currently connected in a base station.
37
38         Number of active streams.
39
40         (From mibCtl ElementType 4020 BSActiveStream)
41     Description for mibCtl Type 14 Unsigned32Type :
42         32 bit unsigned integer.
43     Type derived from mibCtl Type 11 Word32Type :
44         32 bits of raw opaque data.
45     Derived from basic 32 bit word type.
46     "
47     ::= { _625k-MCRFStatusTableEntry 4 }
48
49
50
51     _625k-MCBSActiveRegistration          OBJECT-TYPE
52     SYNTAX          INTEGER -- Unsigned32Type
53     ACCESS          read-only
54     STATUS          mandatory
55     DESCRIPTION
56         "Number of registrations currently existed in a base station.
57
58         The call capacity is determined by the available resources in a
59         base station.
60
61         (From mibCtl ElementType 4021 BSActiveRegistration)
62     Description for mibCtl Type 14 Unsigned32Type :
63         32 bit unsigned integer.
64     Type derived from mibCtl Type 11 Word32Type :
65         32 bits of raw opaque data.
66     Derived from basic 32 bit word type.
67     "
68     ::= { _625k-MCRFStatusTableEntry 5 }

```

```

1
2  _625k-MCSysScalars                OBJECT IDENTIFIER
3  -- DESCRIPTION                    "System Scalars"
4  ::= { _625k-MCSystem 4 }
5
6
7
8  _625k-MCBaseStationID              OBJECT-TYPE
9  SYNTAX                            OCTET STRING (SIZE(0..18)) -- TextType X 18
10 ACCESS                            read-write
11 STATUS                            mandatory
12 DESCRIPTION
13     "Base Station Identification Code.
14
15     This text string must represent in hexadecimal a 42 bit number
16     to be used as the Base Station Identification Code (BSID).
17     The BSID is used by the base station to identify itself to
18     subscriber units.
19     The BSID of a base station must at a minimum
20     differ from that of any other base station
21     where both would be within radio reception distance
22     of any subscriber unit.
23
24     This cannot be changed while the Base Station state is Operating.
25
26     (From mibCtl ElementType 60 BaseStationID)
27     Description for mibCtl Type 15 TextType :
28     ASCII or compatible text.
29     Type derived from mibCtl Type 12 OctetType :
30     8 bits of raw opaque data.
31     Derived from basic 8 bit word type.
32     "
33 ::= { _625k-MCSysScalars 1 }
34
35
36
37 _625k-MCBaseStationTypeID            OBJECT-TYPE
38 SYNTAX                            OCTET STRING (SIZE(0..20)) -- TextType X 20
39 ACCESS                            read-write
40 STATUS                            mandatory
41 DESCRIPTION
42     "Type ID of base station.
43
44
45     (From mibCtl ElementType 66 BaseStationTypeID)
46     Description for mibCtl Type 15 TextType :
47     ASCII or compatible text.
48     Type derived from mibCtl Type 12 OctetType :
49     8 bits of raw opaque data.
50     Derived from basic 8 bit word type.
51     "
52 ::= { _625k-MCSysScalars 2 }
53
54
55
56
57 _625k-MCBaseStationGroupID          OBJECT-TYPE
58 SYNTAX                            OCTET STRING (SIZE(0..20)) -- TextType X 20
59 ACCESS                            read-write
60 STATUS                            mandatory
61 DESCRIPTION
62     "Group ID of base station.
63
64
65     (From mibCtl ElementType 67 BaseStationGroupID)
66     Description for mibCtl Type 15 TextType :
67     ASCII or compatible text.
68

```

```

1      Type derived from mibCtl Type 12 OctetType :
2          8 bits of raw opaque data.
3      Derived from basic 8 bit word type.
4      "
5      ::= { _625k-MCSysScalars 3 }
6
7
8
9      _625k-MCBaseStationSubGroupID          OBJECT-TYPE
10     SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
11     ACCESS          read-write
12     STATUS          mandatory
13     DESCRIPTION
14         "Sub group ID of base station.
15
16
17         (From mibCtl ElementType 68 BaseStationSubGroupID)
18     Description for mibCtl Type 15 TextType :
19         ASCII or compatible text.
20     Type derived from mibCtl Type 12 OctetType :
21         8 bits of raw opaque data.
22     Derived from basic 8 bit word type.
23     "
24     ::= { _625k-MCSysScalars 4 }
25
26
27
28
29     _625k-MCDesiredStateOfBaseStation      OBJECT-TYPE
30     SYNTAX          INTEGER -- ComponentStateType
31     ACCESS          read-write
32     STATUS          mandatory
33     DESCRIPTION
34         "Desired state of base station as a whole.
35
36         This indicates the Base Station state desired by the operator.
37         These desired states are currently supported:
38
39         Operating - for normal operation.
40
41         Ready - to avoid taking any new calls.
42         Existing calls will not be terminated except normally or by
43         command from the operator.
44         While existing calls remain, the base station state will
45         remain as Operating.
46
47         This information is permanently stored on the base station.
48
49         (From mibCtl ElementType 42 DesiredStateOfBaseStation)
50     Description for mibCtl Type 71 ComponentStateType :
51         Component operational state.
52
53         A component begins in the Unknown state.
54         If not detected, it enters and remains in the NotPresent state.
55         If detected, it enters the Uninitialized state, from where it
56         may go to the Testing and Initializing states and then to the
57         Standby or Operating state depending upon permissions.
58         Due to loss of permissions or resources, it may revert from
59         the Operating state to the Standby state.
60         Due to failure or loss of permission, it may revert to the
61         Uninitialized state, perhaps by way of the ShuttingDown state
62         depending on the device.
63         From the Uninitialized state it may return to more advanced
64         states depending upon permissions.
65         In case of a waiting period before (again) initializing,
66         the component is considered to be Initializing.
67
68         Permissions include administrative permissions (from the

```

```

1         operator); excessive failure restrictions; etc.
2     Description for mibCtl ComponentStateType 0 Unknown :
3         Component state not known.
4     Description for mibCtl ComponentStateType 1 NotPresent :
5         Component is not present.
6     Description for mibCtl ComponentStateType 2 PowerOff :
7         Component is present but powered off.
8     Description for mibCtl ComponentStateType 3 Uninitialized :
9         Component is present but not in use.
10
11         The power on/off state of the component is not specified in
12         this case.
13     Description for mibCtl ComponentStateType 4 Testing :
14         Component is being tested.
15     Description for mibCtl ComponentStateType 5 Initializing :
16         Component is being initialized.
17     Description for mibCtl ComponentStateType 6 Ready :
18         Component is ready but not operating.
19     Description for mibCtl ComponentStateType 7 Operating :
20         Component is operating for normal use without restriction.
21
22         The component is either in actual use or may be used at any time,
23         without restriction.
24     Description for mibCtl ComponentStateType 8 Abandoned :
25         Component state is not the desired state due to excessive errors.
26
27         The component state is not that desired, and the Base Station
28         software has abandoned attempts to place the component in
29         the desired state.
30         The actual state of the component is undefined.
31         The Base Station software will resume attempting to place the
32         component in the desired state if the appropriate Reinitialize
33         action element is written with the correct value.
34         Also, the software may resume attempts under other conditions,
35         not all of which may be documented.
36     Description for mibCtl ComponentStateType 9 InitialSetUp :
37         Component is initial set up..
38
39         Initial set up state.
40     Description for mibCtl ComponentStateType 10 Degrading :
41         Component is degrading..
42
43         Degrading state.
44     Description for mibCtl ComponentStateType 11 Restriction :
45         Component is restriction..
46
47         Restriction state.
48     "
49     ::= { _625k-MCSysScalars 5 }
50
51
52
53     _625k-MCTypeOfReboot                OBJECT-TYPE
54         SYNTAX          INTEGER -- RebootType
55         ACCESS          read-write
56         STATUS          mandatory
57         DESCRIPTION
58             "Type of reboot for base station.
59
60
61             (From mibCtl ElementType 63 TypeOfReboot)
62     Description for mibCtl Type 250 RebootType :
63         Reboot Type.
64     Description for mibCtl RebootType 0 Force :
65         Force mode.
66     Description for mibCtl RebootType 1 Graceful :
67         Graceful mode.
68

```

```

1      "
2      ::= { _625k-MCSysScalars 6 }
3
4
5
6      _625k-MCBaseStationRebootTime          OBJECT-TYPE
7          SYNTAX          Gauge -- AbsoluteTimeType
8          ACCESS          read-write
9          STATUS          mandatory
10         DESCRIPTION
11             "Time of base station reboot.
12
13             This is the base station reboot time (GPS time).
14
15             (From mibCtl ElementType 62 BaseStationRebootTime)
16             Description for mibCtl Type 801 AbsoluteTimeType :
17                 Absolute time in GPS seconds.
18
19                 GPS (Global Positioning System) time in seconds since Jan. 6,
20 1980.
21
22                 Note that this differs from UTC (in addition to a possible
23                 offset due to starting time) due to leap seconds; see
24                 the GpsLeapSecond element.
25             Type derived from mibCtl Type 18 Gauge32Type :
26                 32 bits of Gauge data.
27             Derived from basic 32 bit word type.
28             "
29         ::= { _625k-MCSysScalars 7 }
30
31
32         _625k-MCTypeOfBSDiagnosis          OBJECT-TYPE
33             SYNTAX          INTEGER -- DiagnosisType
34             ACCESS          read-write
35             STATUS          mandatory
36             DESCRIPTION
37                 "Type of diagnosis for base station.
38
39
40
41             (From mibCtl ElementType 64 TypeOfBSDiagnosis)
42             Description for mibCtl Type 251 DiagnosisType :
43                 Diagnosis Type.
44             Type derived from mibCtl Type 14 Unsigned32Type :
45                 32 bit unsigned integer.
46             Type derived from mibCtl Type 11 Word32Type :
47                 32 bits of raw opaque data.
48             Derived from basic 32 bit word type.
49             "
50         ::= { _625k-MCSysScalars 8 }
51
52
53
54         _625k-MCBSdiagnosisStatus          OBJECT-TYPE
55             SYNTAX          INTEGER -- DiagnosisStatusType
56             ACCESS          read-only
57             STATUS          mandatory
58             DESCRIPTION
59                 "Diagnosis status for base station.
60
61
62
63             (From mibCtl ElementType 370 BSDiagnosisStatus)
64             Description for mibCtl Type 252 DiagnosisStatusType :
65                 Diagnosis status Type.
66             Type derived from mibCtl Type 14 Unsigned32Type :
67                 32 bit unsigned integer.
68             Type derived from mibCtl Type 11 Word32Type :

```

```

1         32 bits of raw opaque data.
2         Derived from basic 32 bit word type.
3         "
4         ::= { _625k-MCSysScalars 9 }
5
6
7
8         _625k-MCBSDiagnosisFailReason          OBJECT-TYPE
9         SYNTAX          INTEGER -- DiagFailReasonType
10        ACCESS          read-only
11        STATUS          mandatory
12        DESCRIPTION
13        "Diagnosis fail reason for base station.
14
15
16
17        (From mibCtl ElementType 371 BSDiagnosisFailReason)
18        Description for mibCtl Type 253 DiagFailReasonType :
19        Diagnosis fail reason Type.
20        Description for mibCtl DiagFailReasonType 1 PDSNPing :
21        Diagnosis fail reason is PDSN Ping.
22        Description for mibCtl DiagFailReasonType 2 Calibration :
23        Diagnosis fail reason is Calibration.
24        Description for mibCtl DiagFailReasonType 3 AntPath :
25        Diagnosis fail reason is TRx Antenna Path.
26        Description for mibCtl DiagFailReasonType 4 LOalive :
27        Diagnosis fail reason is Local Oscillator DSP Alive.
28        Description for mibCtl DiagFailReasonType 5 GCLoopBack :
29        Diagnosis fail reason is GCLoopBack.
30        Description for mibCtl DiagFailReasonType 6 SlaveNM :
31        Diagnosis fail reason is Slave Modem control board.
32        Description for mibCtl DiagFailReasonType 7 GPSAnt :
33        Diagnosis fail reason is GPS Antenna.
34        Description for mibCtl DiagFailReasonType 8 SlotDSP :
35        Diagnosis fail reason is Modem control board DSP Alive.
36        Description for mibCtl DiagFailReasonType 9 ATMAlive :
37        Diagnosis fail reason is ATM Alive.
38        Description for mibCtl DiagFailReasonType 96 UndefinedName :
39        Diagnosis fail reason is Undefined Diag Name.
40        Description for mibCtl DiagFailReasonType 97 TimeOut :
41        Diagnosis fail reason is Time Out.
42        Description for mibCtl DiagFailReasonType 98 InvalidStateExec :
43        Diagnosis fail reason is Invalid State Execute.
44        Description for mibCtl DiagFailReasonType 99 ExecFail :
45        Diagnosis fail reason is Execute Fail.
46        "
47        ::= { _625k-MCSysScalars 10 }
48
49
50
51        _625k-MCDiskDbUpdateSequence          OBJECT-TYPE
52        SYNTAX          Gauge -- Gauge32Type
53        ACCESS          read-only
54        STATUS          mandatory
55        DESCRIPTION
56        "Base station Flach update sequence number.
57
58        This number is incremented on disk every time any other
59        database element is actually changed on flash.
60        It is not incremented on redundant sets.
61        This number may also be set to a desired value.
62
63        (From mibCtl ElementType 6 DiskDbUpdateSequence)
64        Description for mibCtl Type 18 Gauge32Type :
65        32 bits of Gauge data.
66        Derived from basic 32 bit word type.
67        "
68        ::= { _625k-MCSysScalars 11 }

```

```

1
2
3
4  _625k-MCStateOfBaseStation          OBJECT-TYPE
5      SYNTAX          INTEGER -- ComponentStateType
6      ACCESS          read-only
7      STATUS          mandatory
8      DESCRIPTION
9          "State of base station as a whole.
10
11         This will not have values of Unknown or PowerOff since the
12         base station would be unable to report such values.
13
14         When sufficiently initialized, the state will be Operating if
15         accepting new calls (according to the desired state of the
16         base station) or continuing ongoing calls;
17         or Ready if the desired state is Ready and there are no
18         ongoing calls.
19
20         (From mibCtl ElementType 41 StateOfBaseStation)
21         Description for mibCtl Type 71 ComponentStateType :
22             Component operational state.
23
24             A component begins in the Unknown state.
25             If not detected, it enters and remains in the NotPresent state.
26             If detected, it enters the Uninitialized state, from where it
27             may go to the Testing and Initializing states and then to the
28             Standby or Operating state depending upon permissions.
29             Due to loss of permissions or resources, it may revert from
30             the Operating state to the Standby state.
31             Due to failure or loss of permission, it may revert to the
32             Uninitialized state, perhaps by way of the ShuttingDown state
33             depending on the device.
34             From the Uninitialized state it may return to more advanced
35             states depending upon permissions.
36             In case of a waiting period before (again) initializing,
37             the component is considered to be Initializing.
38
39             Permissions include administrative permissions (from the
40             operator); excessive failure restrictions; etc.
41         Description for mibCtl ComponentStateType 0 Unknown :
42             Component state not known.
43         Description for mibCtl ComponentStateType 1 NotPresent :
44             Component is not present.
45         Description for mibCtl ComponentStateType 2 PowerOff :
46             Component is present but powered off.
47         Description for mibCtl ComponentStateType 3 Uninitialized :
48             Component is present but not in use.
49
50             The power on/off state of the component is not specified in
51             this case.
52         Description for mibCtl ComponentStateType 4 Testing :
53             Component is being tested.
54         Description for mibCtl ComponentStateType 5 Initializing :
55             Component is being initialized.
56         Description for mibCtl ComponentStateType 6 Ready :
57             Component is ready but not operating.
58         Description for mibCtl ComponentStateType 7 Operating :
59             Component is operating for normal use without restriction.
60
61             The component is either in actual use or may be used at any time,
62             without restriction.
63         Description for mibCtl ComponentStateType 8 Abandoned :
64             Component state is not the desired state due to excessive errors.
65
66             The component state is not that desired, and the Base Station
67             software has abandoned attempts to place the component in
68             the desired state.

```



```

1         The actual state of the component is undefined.
2         The Base Station software will resume attempting to place the
3         component in the desired state if the appropriate Reinitialize
4         action element is written with the correct value.
5         Also, the software may resume attempts under other conditions,
6         not all of which may be documented.
7     Description for mibCtl ComponentStateType 9 InitialSetUp :
8         Component is initial set up..
9
10        Initial set up state.
11    Description for mibCtl ComponentStateType 10 Degrading :
12        Component is degrading..
13
14        Degrading state.
15    Description for mibCtl ComponentStateType 11 Restriction :
16        Component is restriction..
17
18        Restriction state.
19    "
20 ::= { _625k-MCSysScalars 12 }
21
22
23
24 _625k-MCBSTotalIndication          OBJECT-TYPE
25     SYNTAX          INTEGER -- IndicationType
26     ACCESS          read-only
27     STATUS          mandatory
28     DESCRIPTION
29         "Status of BS total indicator.
30
31         In the current implementation,
32         this indication is set to the value On by BS.
33
34         (From mibCtl ElementType 525 BSTotalIndication)
35     Description for mibCtl Type 230 IndicationType :
36         Hardware indication status (LEDs).
37
38
39     Description for mibCtl IndicationType 0 Off :
40         Off.
41     Description for mibCtl IndicationType 1 Amber :
42         Amber.
43     Description for mibCtl IndicationType 2 Red :
44         Red.
45     Description for mibCtl IndicationType 3 Green :
46         Green.
47     Description for mibCtl IndicationType 4 NotPresent :
48         Not present.
49     "
50 ::= { _625k-MCSysScalars 13 }
51
52
53
54 _625k-MCMasterAddress              OBJECT-TYPE
55     SYNTAX          INTEGER -- ModuleAddressType
56     ACCESS          read-only
57     STATUS          mandatory
58     DESCRIPTION
59         "Bus slot address of master Modem control board.
60
61         This indicates which Modem control board is master
62         of the base station.
63
64         (From mibCtl ElementType 32 MasterAddress)
65     Description for mibCtl Type 202 ModuleAddressType :
66         Base station bus slot address.
67
68         Most components of the Base Station for which data can

```

```

1         be obtained are identified by a ModuleAddressType address
2         and possibly a subsidiary address.
3         [Limits: 0 7 ]
4         Type derived from mibCtl Type 14 Unsigned32Type :
5         32 bit unsigned integer.
6         Type derived from mibCtl Type 11 Word32Type :
7         32 bits of raw opaque data.
8         Derived from basic 32 bit word type.
9         "
10        ::= { _625k-MCSysScalars 14 }
11
12
13
14        _625k-MCBSManufactureID                OBJECT-TYPE
15        SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
16        ACCESS          read-only
17        STATUS          mandatory
18        DESCRIPTION
19        "Base Station manufacture identification number.
20
21        The manufacture identification assigned by Vendor
22
23        (From mibCtl ElementType 201 BSManufactureID)
24        Description for mibCtl Type 15 TextType :
25        ASCII or compatible text.
26        Type derived from mibCtl Type 12 OctetType :
27        8 bits of raw opaque data.
28        Derived from basic 8 bit word type.
29        "
30        ::= { _625k-MCSysScalars 15 }
31
32
33
34        _625k-MCBSSerialNumber                OBJECT-TYPE
35        SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
36        ACCESS          read-only
37        STATUS          mandatory
38        DESCRIPTION
39        "Base Station serial number.
40
41        This is the character serial number of the base station.
42        This serial number will be unique among all base stations
43        of this type regardless of manufacturer.
44
45        (From mibCtl ElementType 203 BSSerialNumber)
46        Description for mibCtl Type 15 TextType :
47        ASCII or compatible text.
48        Type derived from mibCtl Type 12 OctetType :
49        8 bits of raw opaque data.
50        Derived from basic 8 bit word type.
51        "
52        ::= { _625k-MCSysScalars 16 }
53
54
55
56        _625k-MCDiagnosisBaseStation          OBJECT-TYPE
57        SYNTAX          INTEGER -- BooleanType
58        ACCESS          read-write -- REALLY: write-only
59        STATUS          mandatory
60        DESCRIPTION
61        "Diagnosis base station.
62
63        This is a write-only element; only a value of TRUE is valid.
64
65        (From mibCtl ElementType 47 DiagnosisBaseStation)
66        Description for mibCtl Type 16 BooleanType :
67        Truth value, 0=FALSE, 1=TRUE.
68

```

```

1         This is a subset of TriStateType; no UNDEFINED value is provided.
2         [Limits: 0 1 ]
3         Description for mibCtl BooleanType 0 FALSE :
4         False.
5         Description for mibCtl BooleanType 1 TRUE :
6         True.
7         "
8 ::= { _625k-MCSysScalars 17 }
9
10
11
12 _625k-MCRebootBaseStation          OBJECT-TYPE
13     SYNTAX          INTEGER -- BooleanType
14     ACCESS          read-write -- REALLY: write-only
15     STATUS          mandatory
16     DESCRIPTION
17         "Reboot base station.
18
19         This is a write-only element; only a value of TRUE is valid.
20         All existing calls will be terminated abruptly.
21         All components of the base station will be reinitialized
22         according to the permanent contents of the Base Station database.
23         The base station may be incommunicado for a period of time.
24
25         The reinitialization may be delayed by a few seconds to allow
26         for a clean shutdown.
27
28         (From mibCtl ElementType 44 RebootBaseStation)
29         Description for mibCtl Type 16 BooleanType :
30         Truth value, 0=FALSE, 1=TRUE.
31
32         This is a subset of TriStateType; no UNDEFINED value is provided.
33         [Limits: 0 1 ]
34         Description for mibCtl BooleanType 0 FALSE :
35         False.
36         Description for mibCtl BooleanType 1 TRUE :
37         True.
38         "
39 ::= { _625k-MCSysScalars 18 }
40
41
42
43 _625k-MCBSModelNumber              OBJECT-TYPE
44     SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
45     ACCESS          read-only
46     STATUS          mandatory
47     DESCRIPTION
48         "Base Station model number.
49
50         Base Station model number
51
52         (From mibCtl ElementType 204 BSModelNumber)
53         Description for mibCtl Type 15 TextType :
54         ASCII or compatible text.
55         Type derived from mibCtl Type 12 OctetType :
56         8 bits of raw opaque data.
57         Derived from basic 8 bit word type.
58         "
59 ::= { _625k-MCSysScalars 19 }
60
61
62
63 _625k-MCBSManufactureDate          OBJECT-TYPE
64     SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
65     ACCESS          read-only
66     STATUS          mandatory
67     DESCRIPTION
68         "Base Station manufacture date.

```

```

1
2     Base Station manufacture date
3
4     (From mibCtl ElementType 205 BSManufactureDate)
5     Description for mibCtl Type 15 TextType :
6         ASCII or compatible text.
7     Type derived from mibCtl Type 12 OctetType :
8         8 bits of raw opaque data.
9     Derived from basic 8 bit word type.
10    "
11    ::= { _625k-MCSysScalars 20 }
12
13
14
15    _625k-MCBSHardwareRevision          OBJECT-TYPE
16        SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
17        ACCESS          read-only
18        STATUS          mandatory
19        DESCRIPTION
20            "Base Station hardware revision.
21
22            Base Station hardware revision
23
24            (From mibCtl ElementType 206 BSHardwareRevision)
25            Description for mibCtl Type 15 TextType :
26                ASCII or compatible text.
27            Type derived from mibCtl Type 12 OctetType :
28                8 bits of raw opaque data.
29            Derived from basic 8 bit word type.
30            "
31    ::= { _625k-MCSysScalars 21 }
32
33
34
35    _625k-MCMiscComponents              OBJECT IDENTIFIER
36        -- DESCRIPTION          "Miscellaneous Component"
37    ::= { IEEE802dot20-625k-MC-MIB 2 }
38
39
40
41    _625k-MCAntenna                    OBJECT IDENTIFIER
42        -- DESCRIPTION          "Antenna"
43    ::= { _625k-MCMiscComponents 1 }
44
45
46
47    _625k-MCAntennaTable                OBJECT-TYPE
48        SYNTAX SEQUENCE OF _625k-MCAntennaTableEntry
49        ACCESS          not-accessible
50        STATUS          mandatory
51        DESCRIPTION    "Antenna Table"
52    ::= { _625k-MCAntenna 1 }
53
54
55
56    _625k-MCAntennaTableEntry          OBJECT-TYPE
57        SYNTAX          _625k-MCAntennaTableEntry
58        ACCESS          not-accessible
59        STATUS          mandatory
60        DESCRIPTION    ""
61        INDEX          { _625k-MCAntennaTableIndex }
62    ::= { _625k-MCAntennaTable 1 }
63
64    _625k-MCAntennaTableEntry ::= SEQUENCE {
65        _625k-MCAntennaTableIndex      INTEGER, -- AntennaAddressType
66        _625k-MCStateOfAntenna        INTEGER -- ComponentStateType
67    }
68

```

```

1
2
3  _625k-MCAntennaTableIndex          OBJECT-TYPE
4      SYNTAX          INTEGER -- AntennaAddressType
5      ACCESS          read-only
6      STATUS          mandatory
7      DESCRIPTION    "
8          Description for mibCtl Type 210 AntennaAddressType :
9              Component antenna address.
10             [Limits: 0 11 ]
11             Type derived from mibCtl Type 14 Unsigned32Type :
12                 32 bit unsigned integer.
13             Type derived from mibCtl Type 11 Word32Type :
14                 32 bits of raw opaque data.
15             Derived from basic 32 bit word type.
16         "
17     ::= { _625k-MCAntennaTableEntry 1 }
18
19
20
21  _625k-MCStateOfAntenna              OBJECT-TYPE
22      SYNTAX          INTEGER -- ComponentStateType
23      ACCESS          read-only
24      STATUS          mandatory
25      DESCRIPTION    "
26          "State of Antenna as a whole.
27
28
29          (From mibCtl ElementType 211 StateOfAntenna)
30      Description for mibCtl Type 71 ComponentStateType :
31          Component operational state.
32
33          A component begins in the Unknown state.
34          If not detected, it enters and remains in the NotPresent state.
35          If detected, it enters the Uninitialized state, from where it
36          may go to the Testing and Initializing states and then to the
37          Standby or Operating state depending upon permissions.
38          Due to loss of permissions or resources, it may revert from
39          the Operating state to the Standby state.
40          Due to failure or loss of permission, it may revert to the
41          Uninitialized state, perhaps by way of the ShuttingDown state
42          depending on the device.
43          From the Uninitialized state it may return to more advanced
44          states depending upon permissions.
45          In case of a waiting period before (again) initializing,
46          the component is considered to be Initializing.
47
48          Permissions include administrative permissions (from the
49          operator); excessive failure restrictions; etc.
50      Description for mibCtl ComponentStateType 0 Unknown :
51          Component state not known.
52      Description for mibCtl ComponentStateType 1 NotPresent :
53          Component is not present.
54      Description for mibCtl ComponentStateType 2 PowerOff :
55          Component is present but powered off.
56      Description for mibCtl ComponentStateType 3 Uninitialized :
57          Component is present but not in use.
58
59          The power on/off state of the component is not specified in
60          this case.
61      Description for mibCtl ComponentStateType 4 Testing :
62          Component is being tested.
63      Description for mibCtl ComponentStateType 5 Initializing :
64          Component is being initialized.
65      Description for mibCtl ComponentStateType 6 Ready :
66          Component is ready but not operating.
67      Description for mibCtl ComponentStateType 7 Operating :
68          Component is operating for normal use without restriction.

```

```

1
2     The component is either in actual use or may be used at any time,
3     without restriction.
4 Description for mibCtl ComponentStateType 8 Abandoned :
5     Component state is not the desired state due to excessive errors.
6
7     The component state is not that desired, and the Base Station
8     software has abandoned attempts to place the component in
9     the desired state.
10    The actual state of the component is undefined.
11    The Base Station software will resume attempting to place the
12    component in the desired state if the appropriate Reinitialize
13    action element is written with the correct value.
14    Also, the software may resume attempts under other conditions,
15    not all of which may be documented.
16 Description for mibCtl ComponentStateType 9 InitialSetUp :
17     Component is initial set up..
18
19     Initial set up state.
20 Description for mibCtl ComponentStateType 10 Degrading :
21     Component is degrading..
22
23     Degrading state.
24 Description for mibCtl ComponentStateType 11 Restriction :
25     Component is restriction..
26
27     Restriction state.
28 "
29 ::= { _625k-MCAntennaTableEntry 2 }
30
31
32
33 _625k-MCBSTemperatures          OBJECT IDENTIFIER
34 -- DESCRIPTION                  "BS Temperature"
35 ::= { _625k-MCMiscComponents 3 }
36
37
38
39 _625k-MCBSTemperature          OBJECT-TYPE
40 SYNTAX                          OCTET STRING (SIZE(0..4)) -- DegreesCelsiusType
41 ACCESS                          read-only
42 STATUS                          mandatory
43 DESCRIPTION
44     "The temperature of Base station (degrees Celsius).
45
46     The latest recorded temperature of a given BS.
47
48     (From mibCtl ElementType 536 BSTemperature)
49 Description for mibCtl Type 807 DegreesCelsiusType :
50     Temperature in degrees Celsius.
51     Type derived from mibCtl Type 15 TextType :
52     ASCII or compatible text.
53     Type derived from mibCtl Type 12 OctetType :
54     8 bits of raw opaque data.
55     Derived from basic 8 bit word type.
56 "
57 ::= { _625k-MCBSTemperatures 1 }
58
59
60
61 _625k-MCCableInfo              OBJECT IDENTIFIER
62 -- DESCRIPTION                  "Cable Info"
63 ::= { _625k-MCMiscComponents 4 }
64
65
66
67 _625k-MCCableLossValueForLoCal OBJECT-TYPE
68 SYNTAX                          OCTET STRING (SIZE(0..20)) -- TextType X 20

```

```

1     ACCESS          read-only
2     STATUS          mandatory
3     DESCRIPTION
4         "Value of cable loss for Local Oscilator.
5
6         Value of cable loss.
7
8         (From mibCtl ElementType 801 CableLossValueForLoCal)
9     Description for mibCtl Type 15 TextType :
10        ASCII or compatible text.
11    Type derived from mibCtl Type 12 OctetType :
12        8 bits of raw opaque data.
13    Derived from basic 8 bit word type.
14    "
15    ::= { _625k-MCCableInfo 1 }
16
17
18
19
20    _625k-MCAntCableTable          OBJECT-TYPE
21    SYNTAX SEQUENCE OF _625k-MCAntCableTableEntry
22    ACCESS          not-accessible
23    STATUS          mandatory
24    DESCRIPTION    "Antenna Cable"
25    ::= { _625k-MCCableInfo 2 }
26
27
28
29    _625k-MCAntCableTableEntry     OBJECT-TYPE
30    SYNTAX          _625k-MCAntCableTableEntry
31    ACCESS          not-accessible
32    STATUS          mandatory
33    DESCRIPTION    ""
34    INDEX { _625k-MCAntCableTableIndex }
35    ::= { _625k-MCAntCableTable 1 }
36
37    _625k-MCAntCableTableEntry ::= SEQUENCE {
38        _625k-MCAntCableTableIndex    INTEGER, -- AntennaAddressType
39        _625k-MCCableLossValueForAntenna    OCTET STRING (SIZE(0..20)) --
40    TextType X 20
41    }
42
43
44
45    _625k-MCAntCableTableIndex     OBJECT-TYPE
46    SYNTAX          INTEGER -- AntennaAddressType
47    ACCESS          read-only
48    STATUS          mandatory
49    DESCRIPTION    "
50        Description for mibCtl Type 210 AntennaAddressType :
51        Component antenna address.
52        [Limits: 0 11 ]
53        Type derived from mibCtl Type 14 Unsigned32Type :
54        32 bit unsigned integer.
55        Type derived from mibCtl Type 11 Word32Type :
56        32 bits of raw opaque data.
57        Derived from basic 32 bit word type.
58    "
59    ::= { _625k-MCAntCableTableEntry 1 }
60
61
62
63    _625k-MCCableLossValueForAntenna    OBJECT-TYPE
64    SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
65    ACCESS          read-only
66    STATUS          mandatory
67    DESCRIPTION
68        "Value of cable loss for Antenna.

```

```

1
2     Value of cable loss.
3
4     (From mibCtl ElementType 802 CableLossValueForAntenna)
5     Description for mibCtl Type 15 TextType :
6         ASCII or compatible text.
7     Type derived from mibCtl Type 12 OctetType :
8         8 bits of raw opaque data.
9     Derived from basic 8 bit word type.
10    "
11    ::= { _625k-MCAntCableTableEntry 2 }
12
13
14
15
16    _625k-MCGPSCableTable                OBJECT-TYPE
17        SYNTAX SEQUENCE OF _625k-MCGPSCableTableEntry
18        ACCESS not-accessible
19        STATUS mandatory
20        DESCRIPTION "GPS Cable"
21        ::= { _625k-MCCableInfo 10 }
22
23
24
25    _625k-MCGPSCableTableEntry            OBJECT-TYPE
26        SYNTAX _625k-MCGPSCableTableEntry
27        ACCESS not-accessible
28        STATUS mandatory
29        DESCRIPTION ""
30        INDEX { _625k-MCGPSCableTableIndex }
31        ::= { _625k-MCGPSCableTable 1 }
32
33    _625k-MCGPSCableTableEntry ::= SEQUENCE {
34        _625k-MCGPSCableTableIndex        INTEGER, -- GpsAddressType
35        _625k-MCCableLengthForGps        OCTET STRING (SIZE(0..20)) --
36        TextType X 20
37    }
38
39
40
41    _625k-MCGPSCableTableIndex            OBJECT-TYPE
42        SYNTAX INTEGER -- GpsAddressType
43        ACCESS read-only
44        STATUS mandatory
45        DESCRIPTION "
46            Description for mibCtl Type 209 GpsAddressType :
47            Base station GPS component address.
48
49
50            [Limits: 0 1 ]
51            Type derived from mibCtl Type 14 Unsigned32Type :
52            32 bit unsigned integer.
53            Type derived from mibCtl Type 11 Word32Type :
54            32 bits of raw opaque data.
55            Derived from basic 32 bit word type.
56        "
57        ::= { _625k-MCGPSCableTableEntry 1 }
58
59
60
61    _625k-MCCableLengthForGps              OBJECT-TYPE
62        SYNTAX OCTET STRING (SIZE(0..20)) -- TextType X 20
63        ACCESS read-only
64        STATUS mandatory
65        DESCRIPTION
66            "Cable length for Gps.
67
68            This cable is used for calibration.

```



```

1
2     (From mibCtl ElementType 803 CableLengthForGps)
3     Description for mibCtl Type 15 TextType :
4         ASCII or compatible text.
5     Type derived from mibCtl Type 12 OctetType :
6         8 bits of raw opaque data.
7     Derived from basic 8 bit word type.
8     "
9     ::= { _625k-MCGPSCableTableEntry 2 }
10
11
12
13 _625k-MCGPS                                OBJECT IDENTIFIER
14     -- DESCRIPTION                          "GPS"
15     ::= { _625k-MCMiscComponents 6 }
16
17
18
19 _625k-MCGPSTable                            OBJECT-TYPE
20     SYNTAX SEQUENCE OF _625k-MCGPSTableEntry
21     ACCESS not-accessible
22     STATUS mandatory
23     DESCRIPTION "GPS Table"
24     ::= { _625k-MCGPS 1 }
25
26
27
28 _625k-MCGPSTableEntry                       OBJECT-TYPE
29     SYNTAX _625k-MCGPSTableEntry
30     ACCESS not-accessible
31     STATUS mandatory
32     DESCRIPTION ""
33     INDEX { _625k-MCGPSTableIndex }
34     ::= { _625k-MCGPSTable 1 }
35
36 _625k-MCGPSTableEntry ::= SEQUENCE {
37     _625k-MCGPSTableIndex                    INTEGER, -- GpsAddressType
38     _625k-MCStateOfGps                      INTEGER, -- ComponentStateType
39     _625k-MCGpsNumberOfSatelliteSeen        INTEGER, -- Unsigned32Type
40     _625k-MCGpsIndication                   INTEGER, -- IndicationType
41     _625k-MCGpsSerialNumber                 OCTET STRING (SIZE(0..20)) --
42     TextType X 20
43     }
44
45
46
47 _625k-MCGPSTableIndex                       OBJECT-TYPE
48     SYNTAX INTEGER -- GpsAddressType
49     ACCESS read-only
50     STATUS mandatory
51     DESCRIPTION "
52     Description for mibCtl Type 209 GpsAddressType :
53     Base station GPS component address.
54
55     [Limits: 0 1 ]
56     Type derived from mibCtl Type 14 Unsigned32Type :
57     32 bit unsigned integer.
58     Type derived from mibCtl Type 11 Word32Type :
59     32 bits of raw opaque data.
60     Derived from basic 32 bit word type.
61     "
62     ::= { _625k-MCGPSTableEntry 1 }
63
64
65
66
67 _625k-MCStateOfGps                          OBJECT-TYPE
68     SYNTAX INTEGER -- ComponentStateType

```

```

1     ACCESS          read-only
2     STATUS          mandatory
3     DESCRIPTION
4         "GPS state.
5
6         The state of the GPS (Global Positioning System)
7         on the active local oscillator unit
8
9         (From mibCtl ElementType 420 StateOfGps)
10        Description for mibCtl Type 71 ComponentStateType :
11            Component operational state.
12
13            A component begins in the Unknown state.
14            If not detected, it enters and remains in the NotPresent state.
15            If detected, it enters the Uninitialized state, from where it
16            may go to the Testing and Initializing states and then to the
17            Standby or Operating state depending upon permissions.
18            Due to loss of permissions or resources, it may revert from
19            the Operating state to the Standby state.
20            Due to failure or loss of permission, it may revert to the
21            Uninitialized state, perhaps by way of the ShuttingDown state
22            depending on the device.
23            From the Uninitialized state it may return to more advanced
24            states depending upon permissions.
25            In case of a waiting period before (again) initializing,
26            the component is considered to be Initializing.
27
28            Permissions include administrative permissions (from the
29            operator); excessive failure restrictions; etc.
30        Description for mibCtl ComponentStateType 0 Unknown :
31            Component state not known.
32        Description for mibCtl ComponentStateType 1 NotPresent :
33            Component is not present.
34        Description for mibCtl ComponentStateType 2 PowerOff :
35            Component is present but powered off.
36        Description for mibCtl ComponentStateType 3 Uninitialized :
37            Component is present but not in use.
38
39            The power on/off state of the component is not specified in
40            this case.
41        Description for mibCtl ComponentStateType 4 Testing :
42            Component is being tested.
43        Description for mibCtl ComponentStateType 5 Initializing :
44            Component is being initialized.
45        Description for mibCtl ComponentStateType 6 Ready :
46            Component is ready but not operating.
47        Description for mibCtl ComponentStateType 7 Operating :
48            Component is operating for normal use without restriction.
49
50            The component is either in actual use or may be used at any time,
51            without restriction.
52        Description for mibCtl ComponentStateType 8 Abandoned :
53            Component state is not the desired state due to excessive errors.
54
55            The component state is not that desired, and the Base Station
56            software has abandoned attempts to place the component in
57            the desired state.
58            The actual state of the component is undefined.
59            The Base Station software will resume attempting to place the
60            component in the desired state if the appropriate Reinitialize
61            action element is written with the correct value.
62            Also, the software may resume attempts under other conditions,
63            not all of which may be documented.
64        Description for mibCtl ComponentStateType 9 InitialSetUp :
65            Component is initial set up..
66
67            Initial set up state.
68        Description for mibCtl ComponentStateType 10 Degrading :

```

```

1         Component is degrading..
2
3         Degrading state.
4         Description for mibCtl ComponentStateType 11 Restriction :
5         Component is restriction..
6
7         Restriction state.
8         "
9         ::= { _625k-MCGPSTableEntry 2 }
10
11
12
13 _625k-MCGpsNumberOfSatelliteSeen          OBJECT-TYPE
14     SYNTAX          INTEGER -- Unsigned32Type
15     ACCESS          read-only
16     STATUS          mandatory
17     DESCRIPTION
18         "Number of satellites seen by GPS.
19
20         The number of satellites seen by the GPS (Global Positioning System)
21         on the active local oscillator unit
22
23         (From mibCtl ElementType 421 GpsNumberOfSatelliteSeen)
24         Description for mibCtl Type 14 Unsigned32Type :
25         32 bit unsigned integer.
26         Type derived from mibCtl Type 11 Word32Type :
27         32 bits of raw opaque data.
28         Derived from basic 32 bit word type.
29         "
30     ::= { _625k-MCGPSTableEntry 3 }
31
32
33
34 _625k-MCGpsIndication                      OBJECT-TYPE
35     SYNTAX          INTEGER -- IndicationType
36     ACCESS          read-only
37     STATUS          mandatory
38     DESCRIPTION
39         "Status of GPS indicator.
40
41
42
43         (From mibCtl ElementType 530 GpsIndication)
44         Description for mibCtl Type 230 IndicationType :
45         Hardware indication status (LEDs).
46
47
48         Description for mibCtl IndicationType 0 Off :
49         Off.
50         Description for mibCtl IndicationType 1 Amber :
51         Amber.
52         Description for mibCtl IndicationType 2 Red :
53         Red.
54         Description for mibCtl IndicationType 3 Green :
55         Green.
56         Description for mibCtl IndicationType 4 NotPresent :
57         Not present.
58         "
59     ::= { _625k-MCGPSTableEntry 4 }
60
61
62
63 _625k-MCGpsSerialNumber                    OBJECT-TYPE
64     SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
65     ACCESS          read-only
66     STATUS          mandatory
67     DESCRIPTION
68         "GPS serial number text.

```

```

1
2     Factory set uniquely for each component.
3
4     (From mibCtl ElementType 570 GpsSerialNumber)
5     Description for mibCtl Type 15 TextType :
6         ASCII or compatible text.
7     Type derived from mibCtl Type 12 OctetType :
8         8 bits of raw opaque data.
9     Derived from basic 8 bit word type.
10    "
11    ::= { _625k-MCGPSTableEntry 5 }
12
13
14
15    _625k-MCPowerAmplifier                OBJECT IDENTIFIER
16    -- DESCRIPTION          "Power Amplifier"
17    ::= { _625k-MCMiscComponents 8 }
18
19
20
21
22    _625k-MCPAUnitTable                  OBJECT-TYPE
23    SYNTAX SEQUENCE OF _625k-MCPAUnitTableEntry
24    ACCESS not-accessible
25    STATUS mandatory
26    DESCRIPTION "PA Table"
27    ::= { _625k-MCPowerAmplifier 1 }
28
29
30
31    _625k-MCPAUnitTableEntry             OBJECT-TYPE
32    SYNTAX _625k-MCPAUnitTableEntry
33    ACCESS not-accessible
34    STATUS mandatory
35    DESCRIPTION ""
36    INDEX { _625k-MCPAUnitTableIndex }
37    ::= { _625k-MCPAUnitTable 1 }
38
39    _625k-MCPAUnitTableEntry ::= SEQUENCE {
40        _625k-MCPAUnitTableIndex         INTEGER, -- PAUnitAddressType
41        _625k-MCRebootPAUnit             INTEGER, -- BooleanType
42        _625k-MCStateOfPAUnit            INTEGER, -- ComponentStateType
43        _625k-MCPAUnitIndication          INTEGER, -- IndicationType
44        _625k-MCPAUnitSerialNumber        OCTET STRING (SIZE(0..20)), --
45        TextType X 20
46        _625k-MCPAUnitModelNumber         OCTET STRING (SIZE(0..20)), --
47        TextType X 20
48        _625k-MCPAUnitManufactureDate     OCTET STRING (SIZE(0..20)), --
49        TextType X 20
50        _625k-MCPAUnitHardwareRevision    OCTET STRING (SIZE(0..20)), --
51        TextType X 20
52        _625k-MCPAUnitManufactureID       OCTET STRING (SIZE(0..20)), --
53        TextType X 20
54        _625k-MCPAUnitTemperature         OCTET STRING (SIZE(0..4)) --
55        DegreesCelsiusType
56    }
57
58
59
60    _625k-MCPAUnitTableIndex             OBJECT-TYPE
61    SYNTAX INTEGER -- PAUnitAddressType
62    ACCESS read-only
63    STATUS mandatory
64    DESCRIPTION ""
65        Description for mibCtl Type 207 PAUnitAddressType :
66        Base station power amplifier component unit address.
67
68        A power amplifier unit address is a subset of Base Station

```

```

1         component addresses,
2         restricted to power amplifier components only.
3         Power amplifiers boost radio frequency
4         signal levels.
5         [Limits: 0 3 ]
6         Type derived from mibCtl Type 14 Unsigned32Type :
7         32 bit unsigned integer.
8         Type derived from mibCtl Type 11 Word32Type :
9         32 bits of raw opaque data.
10        Derived from basic 32 bit word type.
11    "
12    ::= { _625k-MCPAUnitTableEntry 1 }
13
14
15
16    _625k-MCRebootPAUnit                OBJECT-TYPE
17        SYNTAX                INTEGER -- BooleanType
18        ACCESS                read-write -- REALLY: write-only
19        STATUS                mandatory
20        DESCRIPTION
21            "Action to reboot a PA unit.
22
23            This is a write-only element; only a value of TRUE is valid.
24
25            (From mibCtl ElementType 506 RebootPAUnit)
26            Description for mibCtl Type 16 BooleanType :
27            Truth value, 0=FALSE, 1=TRUE.
28
29            This is a subset of TriStateType; no UNDEFINED value is provided.
30            [Limits: 0 1 ]
31            Description for mibCtl BooleanType 0 FALSE :
32            False.
33            Description for mibCtl BooleanType 1 TRUE :
34            True.
35    "
36    ::= { _625k-MCPAUnitTableEntry 2 }
37
38
39
40    _625k-MCStateOfPAUnit                OBJECT-TYPE
41        SYNTAX                INTEGER -- ComponentStateType
42        ACCESS                read-only
43        STATUS                mandatory
44        DESCRIPTION
45            "State of PA as a whole.
46
47
48            (From mibCtl ElementType 212 StateOfPAUnit)
49            Description for mibCtl Type 71 ComponentStateType :
50            Component operational state.
51
52            A component begins in the Unknown state.
53            If not detected, it enters and remains in the NotPresent state.
54            If detected, it enters the Uninitialized state, from where it
55            may go to the Testing and Initializing states and then to the
56            Standby or Operating state depending upon permissions.
57            Due to loss of permissions or resources, it may revert from
58            the Operating state to the Standby state.
59            Due to failure or loss of permission, it may revert to the
60            Uninitialized state, perhaps by way of the ShuttingDown state
61            depending on the device.
62            From the Uninitialized state it may return to more advanced
63            states depending upon permissions.
64            In case of a waiting period before (again) initializing,
65            the component is considered to be Initializing.
66
67            Permissions include administrative permissions (from the
68            operator); excessive failure restrictions; etc.

```

```

1      Description for mibCtl ComponentStateType 0 Unknown :
2          Component state not known.
3      Description for mibCtl ComponentStateType 1 NotPresent :
4          Component is not present.
5      Description for mibCtl ComponentStateType 2 PowerOff :
6          Component is present but powered off.
7      Description for mibCtl ComponentStateType 3 Uninitialized :
8          Component is present but not in use.
9
10         The power on/off state of the component is not specified in
11         this case.
12      Description for mibCtl ComponentStateType 4 Testing :
13          Component is being tested.
14      Description for mibCtl ComponentStateType 5 Initializing :
15          Component is being initialized.
16      Description for mibCtl ComponentStateType 6 Ready :
17          Component is ready but not operating.
18      Description for mibCtl ComponentStateType 7 Operating :
19          Component is operating for normal use without restriction.
20
21         The component is either in actual use or may be used at any time,
22         without restriction.
23      Description for mibCtl ComponentStateType 8 Abandoned :
24          Component state is not the desired state due to excessive errors.
25
26         The component state is not that desired, and the Base Station
27         software has abandoned attempts to place the component in
28         the desired state.
29         The actual state of the component is undefined.
30         The Base Station software will resume attempting to place the
31         component in the desired state if the appropriate Reinitialize
32         action element is written with the correct value.
33         Also, the software may resume attempts under other conditions,
34         not all of which may be documented.
35      Description for mibCtl ComponentStateType 9 InitialSetUp :
36          Component is initial set up..
37
38         Initial set up state.
39      Description for mibCtl ComponentStateType 10 Degrading :
40          Component is degrading..
41
42         Degrading state.
43      Description for mibCtl ComponentStateType 11 Restriction :
44          Component is restriction..
45
46         Restriction state.
47      "
48      ::= { _625k-MCPAUnitTableEntry 3 }
49
50
51
52      _625k-MCPAUnitIndication          OBJECT-TYPE
53          SYNTAX          INTEGER -- IndicationType
54          ACCESS          read-only
55          STATUS          mandatory
56          DESCRIPTION
57              "Status of PA Unit indicator.
58
59
60              (From mibCtl ElementType 526 PAUnitIndication)
61      Description for mibCtl Type 230 IndicationType :
62          Hardware indication status (LEDs).
63
64
65
66      Description for mibCtl IndicationType 0 Off :
67          Off.
68      Description for mibCtl IndicationType 1 Amber :
```

```

1         Amber.
2         Description for mibCtl IndicationType 2 Red :
3         Red.
4         Description for mibCtl IndicationType 3 Green :
5         Green.
6         Description for mibCtl IndicationType 4 NotPresent :
7         Not present.
8         "
9         ::= { _625k-MCPAUnitTableEntry 4 }
10
11
12
13 _625k-MCPAUnitSerialNumber          OBJECT-TYPE
14     SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
15     ACCESS          read-only
16     STATUS          mandatory
17     DESCRIPTION
18         "PA unit serial number text.
19
20         Factory set uniquely for each component.
21
22         (From mibCtl ElementType 560 PAUnitSerialNumber)
23         Description for mibCtl Type 15 TextType :
24             ASCII or compatible text.
25         Type derived from mibCtl Type 12 OctetType :
26             8 bits of raw opaque data.
27         Derived from basic 8 bit word type.
28         "
29         ::= { _625k-MCPAUnitTableEntry 5 }
30
31
32
33 _625k-MCPAUnitModelNumber           OBJECT-TYPE
34     SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
35     ACCESS          read-only
36     STATUS          mandatory
37     DESCRIPTION
38         "PA unit model number.
39
40         Factory set with description of component type, including
41         the major revision level.
42
43         (From mibCtl ElementType 561 PAUnitModelNumber)
44         Description for mibCtl Type 15 TextType :
45             ASCII or compatible text.
46         Type derived from mibCtl Type 12 OctetType :
47             8 bits of raw opaque data.
48         Derived from basic 8 bit word type.
49         "
50         ::= { _625k-MCPAUnitTableEntry 6 }
51
52
53
54 _625k-MCPAUnitManufactureDate       OBJECT-TYPE
55     SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
56     ACCESS          read-only
57     STATUS          mandatory
58     DESCRIPTION
59         "PA unit manufacture date.
60
61         Factory set to month and date of manufacture date of the module.
62
63         (From mibCtl ElementType 562 PAUnitManufactureDate)
64         Description for mibCtl Type 15 TextType :
65             ASCII or compatible text.
66         Type derived from mibCtl Type 12 OctetType :
67             8 bits of raw opaque data.
68         Derived from basic 8 bit word type.

```

```

1      "
2      ::= { _625k-MCPAUnitTableEntry 7 }
3
4
5
6      _625k-MCPAUnitHardwareRevision          OBJECT-TYPE
7      SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
8      ACCESS          read-only
9      STATUS          mandatory
10     DESCRIPTION
11         "PA unit hardware revision name.
12
13         Set at the factory to indicate the minor hardware revision
14         level of the module.
15
16         (From mibCtl ElementType 563 PAUnitHardwareRevision)
17         Description for mibCtl Type 15 TextType :
18             ASCII or compatible text.
19         Type derived from mibCtl Type 12 OctetType :
20             8 bits of raw opaque data.
21         Derived from basic 8 bit word type.
22     "
23     ::= { _625k-MCPAUnitTableEntry 8 }
24
25
26
27     _625k-MCPAUnitManufactureID              OBJECT-TYPE
28     SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
29     ACCESS          read-only
30     STATUS          mandatory
31     DESCRIPTION
32         "PA unit manufacture ID.
33
34
35
36         (From mibCtl ElementType 564 PAUnitManufactureID)
37         Description for mibCtl Type 15 TextType :
38             ASCII or compatible text.
39         Type derived from mibCtl Type 12 OctetType :
40             8 bits of raw opaque data.
41         Derived from basic 8 bit word type.
42     "
43     ::= { _625k-MCPAUnitTableEntry 9 }
44
45
46
47     _625k-MCPAUnitTemperature                OBJECT-TYPE
48     SYNTAX          OCTET STRING (SIZE(0..4)) -- DegreesCelsiusType
49     ACCESS          read-only
50     STATUS          mandatory
51     DESCRIPTION
52         "The temperature of PAUnit (degrees Celsius).
53
54         The latest recorded temperature of a given PAUnit.
55
56         (From mibCtl ElementType 539 PAUnitTemperature)
57         Description for mibCtl Type 807 DegreesCelsiusType :
58             Temperature in degrees Celsius.
59         Type derived from mibCtl Type 15 TextType :
60             ASCII or compatible text.
61         Type derived from mibCtl Type 12 OctetType :
62             8 bits of raw opaque data.
63         Derived from basic 8 bit word type.
64     "
65     ::= { _625k-MCPAUnitTableEntry 10 }
66
67
68

```



```

1
2  _625k-MCPAModuleTable                OBJECT-TYPE
3    SYNTAX SEQUENCE OF _625k-MCPAModuleTableEntry
4    ACCESS      not-accessible
5    STATUS      mandatory
6    DESCRIPTION "PA Module Table"
7    ::= { _625k-MCPowerAmplifier 2 }
8
9
10
11 _625k-MCPAModuleTableEntry            OBJECT-TYPE
12   SYNTAX      _625k-MCPAModuleTableEntry
13   ACCESS      not-accessible
14   STATUS      mandatory
15   DESCRIPTION ""
16   INDEX { _625k-MCPAModuleTableIndex1, _625k-MCPAModuleTableIndex2 }
17   ::= { _625k-MCPAModuleTable 1 }
18
19 _625k-MCPAModuleTableEntry ::= SEQUENCE {
20   _625k-MCPAModuleTableIndex1          INTEGER, -- PAUnitAddressType
21   _625k-MCPAModuleTableIndex2          INTEGER -- PAModuleAddressType
22   }
23
24
25
26 _625k-MCPAModuleTableIndex1            OBJECT-TYPE
27   SYNTAX      INTEGER -- PAUnitAddressType
28   ACCESS      read-only
29   STATUS      mandatory
30   DESCRIPTION "
31     Description for mibCtl Type 207 PAUnitAddressType :
32     Base station power amplifier component unit address.
33
34     A power amplifier unit address is a subset of Base Station
35     component addresses,
36     restricted to power amplifier components only.
37     Power amplifiers boost radio frequency
38     signal levels.
39     [Limits: 0 3 ]
40     Type derived from mibCtl Type 14 Unsigned32Type :
41     32 bit unsigned integer.
42     Type derived from mibCtl Type 11 Word32Type :
43     32 bits of raw opaque data.
44     Derived from basic 32 bit word type.
45   "
46   ::= { _625k-MCPAModuleTableEntry 1 }
47
48
49
50 _625k-MCPAModuleTableIndex2            OBJECT-TYPE
51   SYNTAX      INTEGER -- PAModuleAddressType
52   ACCESS      read-only
53   STATUS      mandatory
54   DESCRIPTION "
55     Description for mibCtl Type 208 PAModuleAddressType :
56     Base station power amplifier component module address.
57
58     A power amplifier module address is a subset of Base Station
59     component addresses,
60     [Limits: 0 2 ]
61     Type derived from mibCtl Type 14 Unsigned32Type :
62     32 bit unsigned integer.
63     Type derived from mibCtl Type 11 Word32Type :
64     32 bits of raw opaque data.
65     Derived from basic 32 bit word type.
66   "
67   ::= { _625k-MCPAModuleTableEntry 2 }
68

```

```

1
2
3  _625k-MCPowerSupply          OBJECT IDENTIFIER
4  -- DESCRIPTION              "Power supply"
5  ::= { _625k-MCMiscComponents 9 }
6
7
8
9
10 _625k-MCPowerSupplyTable     OBJECT-TYPE
11   SYNTAX SEQUENCE OF _625k-MCPowerSupplyTableEntry
12   ACCESS          not-accessible
13   STATUS           mandatory
14   DESCRIPTION     "Power supply Table"
15   ::= { _625k-MCPowerSupply 1 }
16
17
18
19 _625k-MCPowerSupplyTableEntry OBJECT-TYPE
20   SYNTAX          _625k-MCPowerSupplyTableEntry
21   ACCESS          not-accessible
22   STATUS           mandatory
23   DESCRIPTION     ""
24   INDEX { _625k-MCPowerSupplyTableIndex }
25   ::= { _625k-MCPowerSupplyTable 1 }
26
27 _625k-MCPowerSupplyTableEntry ::= SEQUENCE {
28   _625k-MCPowerSupplyTableIndex    INTEGER, -- PowerAddressType
29   _625k-MCStateOfPowerSupply       INTEGER, -- ComponentStateType
30   _625k-MCPowerSupplyIndication    INTEGER -- IndicationType
31   }
32
33
34
35 _625k-MCPowerSupplyTableIndex OBJECT-TYPE
36   SYNTAX          INTEGER -- PowerAddressType
37   ACCESS          read-only
38   STATUS           mandatory
39   DESCRIPTION     "
40     Description for mibCtl Type 211 PowerAddressType :
41     Component power supply address.
42     [Limits: 0 2 ]
43     Type derived from mibCtl Type 14 Unsigned32Type :
44     32 bit unsigned integer.
45     Type derived from mibCtl Type 11 Word32Type :
46     32 bits of raw opaque data.
47     Derived from basic 32 bit word type.
48     "
49   ::= { _625k-MCPowerSupplyTableEntry 1 }
50
51
52
53 _625k-MCStateOfPowerSupply     OBJECT-TYPE
54   SYNTAX          INTEGER -- ComponentStateType
55   ACCESS          read-only
56   STATUS           mandatory
57   DESCRIPTION     "
58     "State of Power supply as a whole.
59
60     (From mibCtl ElementType 213 StateOfPowerSupply)
61     Description for mibCtl Type 71 ComponentStateType :
62     Component operational state.
63
64     A component begins in the Unknown state.
65     If not detected, it enters and remains in the NotPresent state.
66     If detected, it enters the Uninitialized state, from where it
67     may go to the Testing and Initializing states and then to the
68

```

```

1      Standby or Operating state depending upon permissions.
2      Due to loss of permissions or resources, it may revert from
3      the Operating state to the Standby state.
4      Due to failure or loss of permission, it may revert to the
5      Uninitialized state, perhaps by way of the ShuttingDown state
6      depending on the device.
7      From the Uninitialized state it may return to more advanced
8      states depending upon permissions.
9      In case of a waiting period before (again) initializing,
10     the component is considered to be Initializing.
11
12     Permissions include administrative permissions (from the
13     operator); excessive failure restrictions; etc.
14     Description for mibCtl ComponentStateType 0 Unknown :
15     Component state not known.
16     Description for mibCtl ComponentStateType 1 NotPresent :
17     Component is not present.
18     Description for mibCtl ComponentStateType 2 PowerOff :
19     Component is present but powered off.
20     Description for mibCtl ComponentStateType 3 Uninitialized :
21     Component is present but not in use.
22
23     The power on/off state of the component is not specified in
24     this case.
25     Description for mibCtl ComponentStateType 4 Testing :
26     Component is being tested.
27     Description for mibCtl ComponentStateType 5 Initializing :
28     Component is being initialized.
29     Description for mibCtl ComponentStateType 6 Ready :
30     Component is ready but not operating.
31     Description for mibCtl ComponentStateType 7 Operating :
32     Component is operating for normal use without restriction.
33
34     The component is either in actual use or may be used at any time,
35     without restriction.
36     Description for mibCtl ComponentStateType 8 Abandoned :
37     Component state is not the desired state due to excessive errors.
38
39     The component state is not that desired, and the Base Station
40     software has abandoned attempts to place the component in
41     the desired state.
42     The actual state of the component is undefined.
43     The Base Station software will resume attempting to place the
44     component in the desired state if the appropriate Reinitialize
45     action element is written with the correct value.
46     Also, the software may resume attempts under other conditions,
47     not all of which may be documented.
48     Description for mibCtl ComponentStateType 9 InitialSetUp :
49     Component is initial set up..
50
51     Initial set up state.
52     Description for mibCtl ComponentStateType 10 Degrading :
53     Component is degrading..
54
55     Degrading state.
56     Description for mibCtl ComponentStateType 11 Restriction :
57     Component is restriction..
58
59     Restriction state.
60     "
61     ::= { _625k-MCPowerSupplyTableEntry 2 }
62
63
64
65     _625k-MCPowerSupplyIndication          OBJECT-TYPE
66     SYNTAX          INTEGER -- IndicationType
67     ACCESS          read-only
68     STATUS          mandatory

```

```
1 DESCRIPTION
2     "Status of Power Supply indicator.
3
4
5
6     (From mibCtl ElementType 527 PowerSupplyIndication)
7     Description for mibCtl Type 230 IndicationType :
8         Hardware indication status (LEDs).
9
10
11     Description for mibCtl IndicationType 0 Off :
12         Off.
13     Description for mibCtl IndicationType 1 Amber :
14         Amber.
15     Description for mibCtl IndicationType 2 Red :
16         Red.
17     Description for mibCtl IndicationType 3 Green :
18         Green.
19     Description for mibCtl IndicationType 4 NotPresent :
20         Not present.
21     "
22 ::= { _625k-MCPowerSupplyTableEntry 3 }
23
24
25
26 END
27
```

1 **625K-MC Appendix – A (Informative)**

2 As specified in the Appendix Section-A of baseline specification HC-SDMA [25].

3