Project	**IEEE 802.16 Broadband Wireless Access Working Group <http://ieee802.org/16>**
Title	**Report of the TG3/4 Ad Hoc Committee**
Date Submitted	**2001-09-05**
Source(s)	Huanchun Ye (Chair, TG3/4 Ad Hoc ARQ Committee) BeamReach Networks 755 North Mathilda Avenue Sunnyvale, CA 94086 Voice: 408-869-8748 Fax: 408-869-8701 mailto:hye@beamreachnetworks.com
Re:	Invitation to join TG3/4 MAC Ad Hoc Group on ARQ, July 26, 2001
Abstract	This proposal is a revised version of the proposal that came into the ARQ ad hoc group, incorporating the results of all the straw polls to date.
Purpose	
Notice	This document has been prepared to assist IEEE 802.16. 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 text 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 802.16.
Patent Policy and Procedures	The contributor is familiar with the IEEE 802.16 Patent Policy and Procedures (Version 1.0) <http://ieee802.org/16/ipr/patents/policy.html>, including the statement "IEEE standards may include the known use of patent(s), including patent applications, if there is technical justification in the opinion of the standards-developing committee and provided the IEEE receives assurance from the patent holder that it will license applicants under reasonable terms and conditions for the purpose of implementing the standard." Early disclosure to the Working Group of patent information that might be relevant to the standard is essential to reduce the possibility for delays in the development process and increase the likelihood that the draft publication will be approved for publication. Please notify the Chair <mailto:r.b.marks@ieee.org> as early as possible, in written or electronic form, of any patents (granted or under application) that may cover technology that is under consideration by or has been approved by IEEE 802.16. The Chair will disclose this notification via the IEEE 802.16 web site <http://ieee802.org/16/ipr/patents/notices>.

Report of the TG3/4 Ad Hoc ARQ Committee

Huanchun Ye (Chair)
BeamReach Networks

1. ARQ proposal accepted by the ad hoc committee

The following text is the output of the ad hoc committee intended to be inserted into the ARQ section in the IEEE 802.16ab-01/01r1.

Introduction

This contribution is loosely based on GPRS specification for ARQ. For the purpose of simplicity, we used GPRS only as a guide for the ARQ operational procedures and kept only those elements that are relevant to our own purpose. We also try to incorporate as many elements of possible from IEEE 802.16abc-01/01 and contributions and comments from session #14.

In the following, we shall use LLE (Logical Link Entity) to denote an ARQ state machine. The term MPDU (MAC Protocol Data Unit), used throughout this document, refers to a MAC PDU with a single MAC header, zero or more optional main or packing sub-headers and a payload, where the presence of the sub-headers are indicated by the TYPE field in the generic MAC header. An MPDU may carry one or more whole fragmented MSDUs.

ARQ Blocks

An **ARQ Block** is employed as an identifiable logical unit. The transmitted MSDUs and the MSDU fragments are logically divided into blocks that never change but MAY be assembled differently when retransmitting the data.

The parameter **ARQ_BLK_SIZE** should be of the form 2^N. It defines the block size in bytes. It is negotiated between the peers during the connection creation/change. **ARQ_BLK_SIZE** may vary from 1 to TBD bytes.

The block size MAY be more than the maximum MAC Message size. Then the only incomplete blocks appear.

Another parameter is acknowledgment window size **ACK_WIN_SIZE** that limits the amount of the blocks, transmitted but not acknowledged.

Transmitter Operations: MAC Message Creation and Numbering

The following is the sequence of MAC operations at the transmitting side with ARQ enabled

The complete MSDUs and fragments are logically divided into portions (***ARQ blocks***) of the given size **ARQ_BLK_SIZE**. The last block in the MSDU MAY be smaller than **ARQ_BLK_SIZE,** such a block is called “incomplete block”. Once defined as a piece of data, block never changes (split or recombined)

A set of blocks is selected for the transmission and aggregated into MAC Messages. This set may include also the blocks selected for the retransmission. At this step fragmentation of the MAC messages MAY be performed so that the fragment boundary MUST be aligned to a block boundary. A Sequential Number should be assigned to any block not having yet such a number. Sequential Numbers, taken in the order of their assignment, form a sequence of numbers 0 .. 2^N-1 where N is the number bits (with wrap-around at 2^N). The following are the restrictions:

Only contiguous Block Sequential Numbers may appear within a single MAC Message

An incomplete block may be placed only at the end of a MAC Message or at the end of partial payload in the case of packed MAC Message

Each MAC Message gets a Sequential Number, which is the Sequential Number of the FIRST, block in the MAC Message. This number is encoded in the ARQ sub-header (see 0). Note that according to MAC rules, if a payload (partial payload) of a MAC message contains a MSDU fragment, it should be described correspondently in the Fragmentation Sub-header or Packing Sub-header.

It is a matter of transmitter's policy whether the set of blocks once transmitted as a single MAC Message, will be retransmitted also as a single MAC Message.

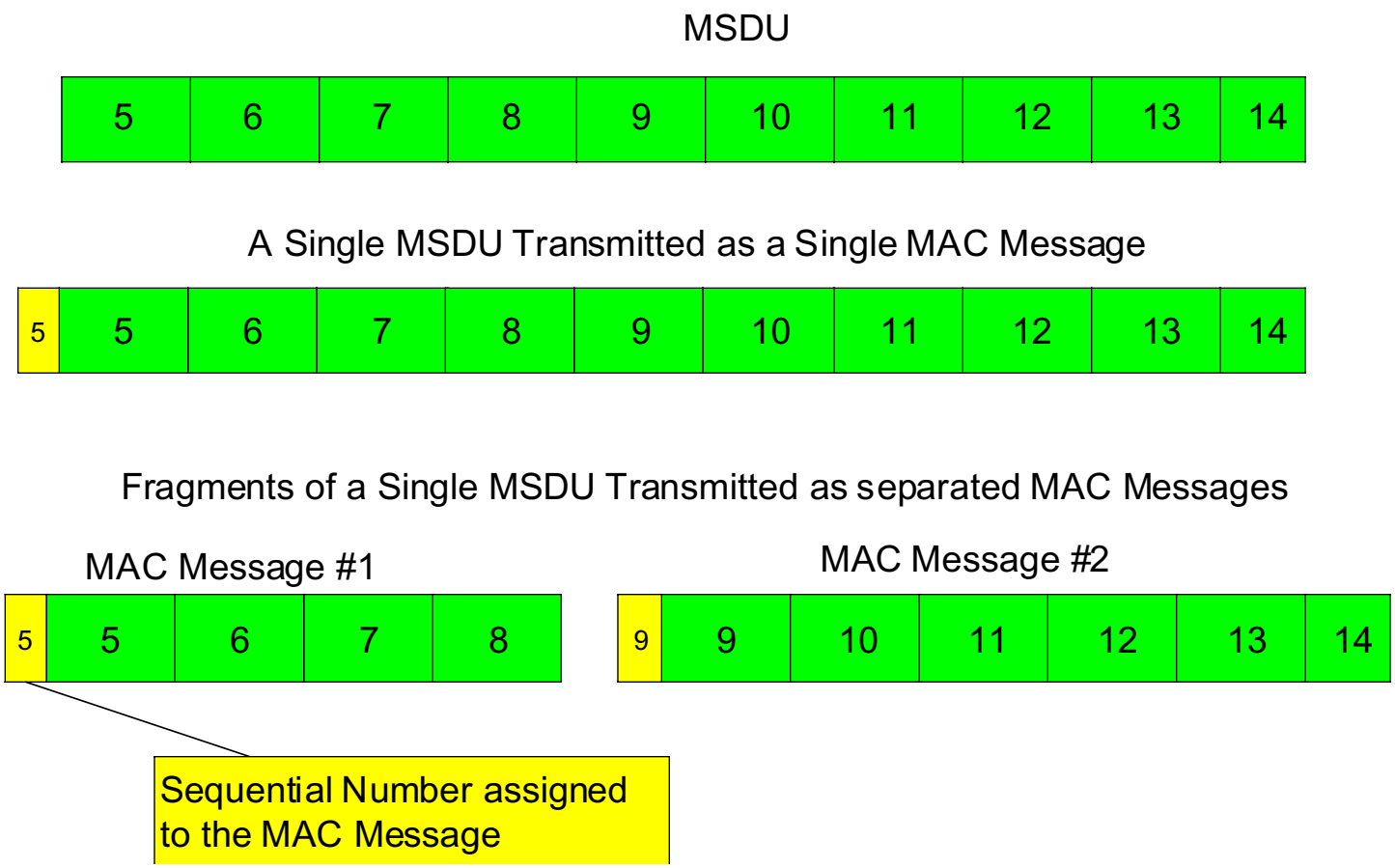

MSDU
5
6
7
8
9
10
11
12
13
14
A Single MSDU Transmitted as a Single MAC Message
5
5
6
7
8
9
10
11
12
13
14
Fragments of a Single MSDU Transmitted as separated MAC Messages
MAC Message #1
5
5
6
7
8
MAC Message #2
9
9
10
11
12
13
14
Sequential Number assigned to the MAC Message

Figure 1

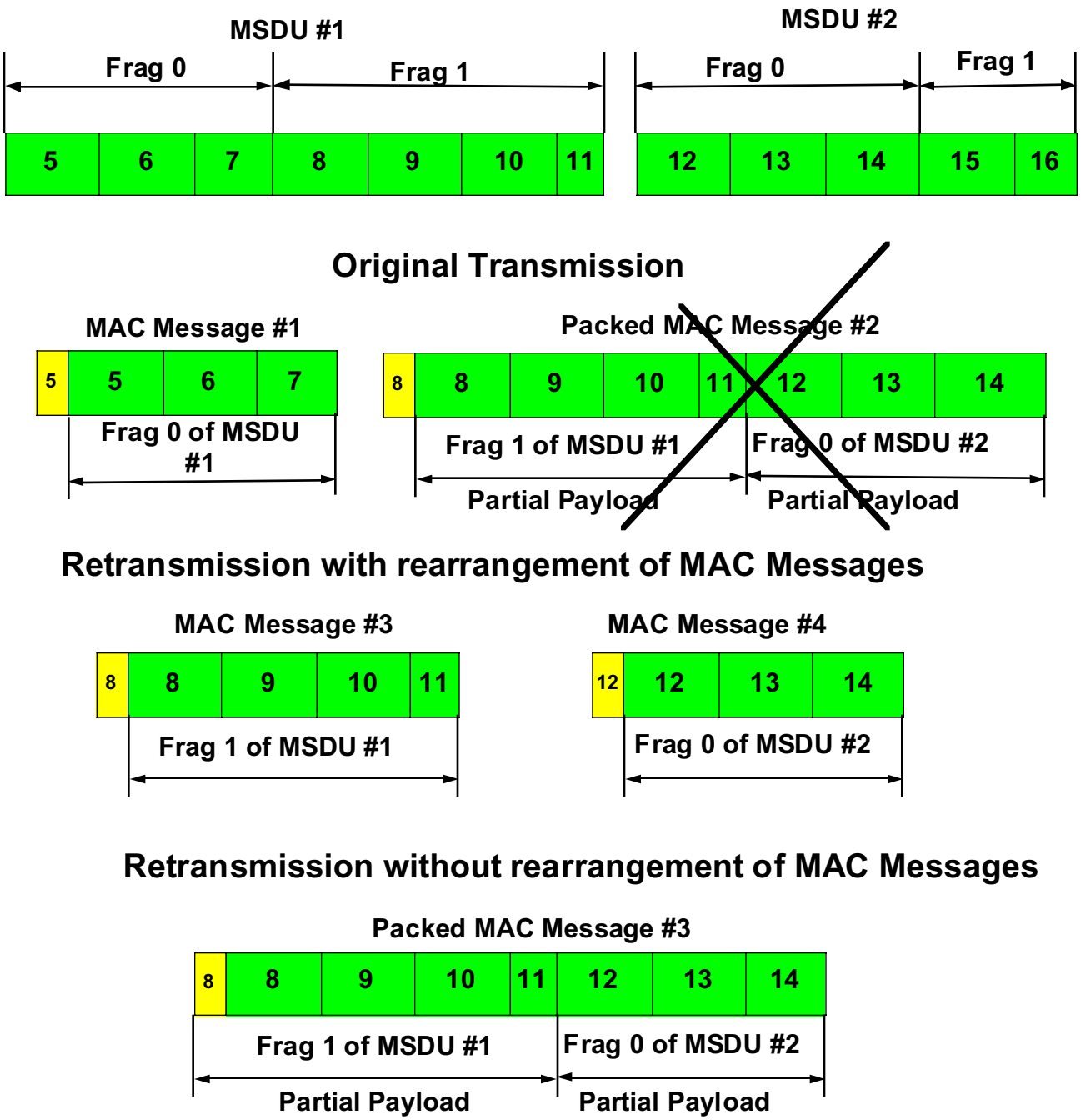

MSDU #1
Frag 0
Frag 1
5
6
7
8
9
10
11
MSDU #2
Frag 0
Frag 1
12
13
14
15
16
Original Transmission
MAC Message #1
5
5
6
7
Frag 0 of MSDU #1
Packed MAC Message #2
8
8
9
10
11
12
13
14
Frag 1 of MSDU #1
Partial Payload
Frag 0 of MSDU #2
Partial Payload
Retransmission with rearrangement of MAC Messages
MAC Message #3
8
8
9
10
11
Frag 1 of MSDU #1
MAC Message #4
12
12
13
14
Frag 0 of MSDU #2
Retransmission without rearrangement of MAC Messages
Packed MAC Message #3
8
8
9
10
11
12
13
14
Frag 1 of MSDU #1
Partial Payload
Frag 0 of MSDU #2
Partial Payload

Figure 2

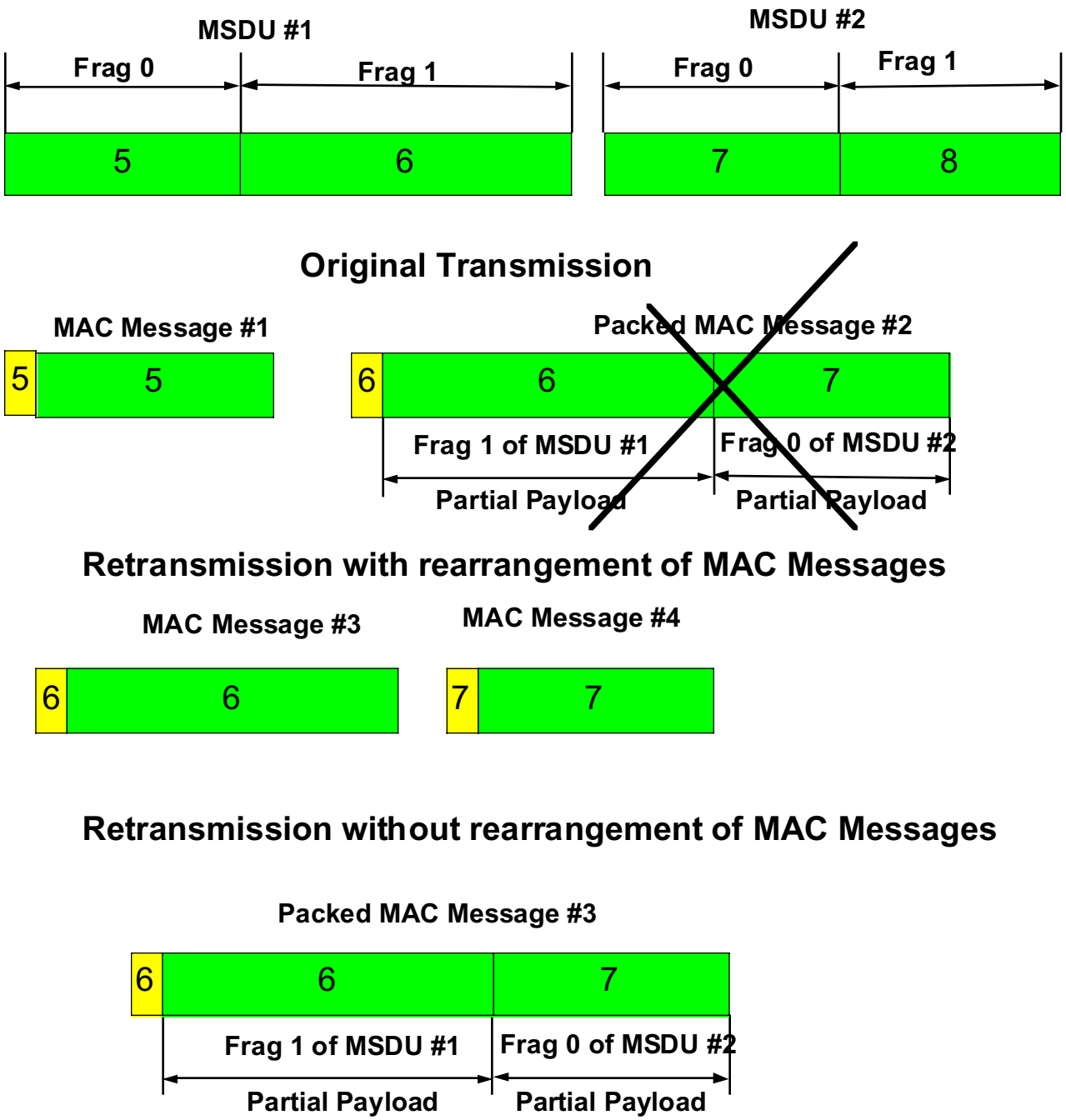
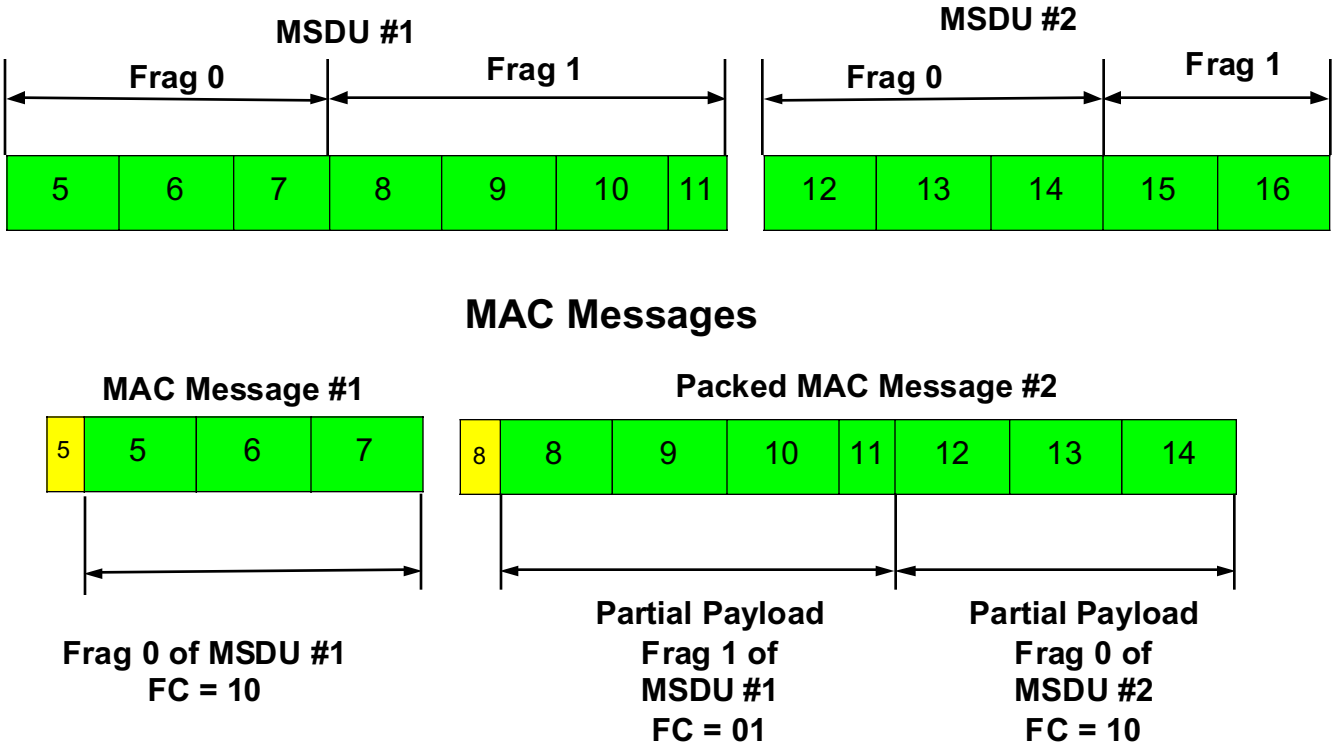

MSDU #1
Frag 0
Frag 1
5
6
MSDU #2
Frag 0
Frag 1
7
8
Original Transmission
MAC Message #1
5
5
Packed MAC Message #2
6
6
7
Frag 1 of MSDU #1
Frag 0 of MSDU #2
Partial Payload
Partial Payload
Retransmission with rearrangement of MAC Messages
MAC Message #3
MAC Message #4
6
6
7
7
Retransmission without rearrangement of MAC Messages
Packed MAC Message #3
6
6
7
Frag 1 of MSDU #1
Frag 0 of MSDU #2
Partial Payload
Partial Payload

Figure 3


MSDU #1
Frag 0
Frag 1
5
6
7
8
9
10
11
MSDU #2
Frag 0
Frag 1
12
13
14
15
16
MAC Messages
MAC Message #1
5
5
6
7
Frag 0 of MSDU #1
FC = 10
Packed MAC Message #2
8
8
9
10
11
12
13
14
Partial Payload
Frag 1 of
MSDU #1
FC = 01
Partial Payload
Frag 0 of
MSDU #2
FC = 10

Figure

New Sub-headers for Block-based Fragmentation, Packing, and ARQ

The block-based fragmentation, packing, and ARQ require various sub-headers to be defined. The presence of these sub-headers is indicated by the value of the TYPE field in the generic MAC header. The table below lists the encoding of the TYPE field.

TYPE Field Encoding (binary)	Sub-headers
000000	No sub-header
xxxxx1	Grant management sub-header
00001x	TG1 packing sub-header
00010x	TG1 fragmentation sub-header
00011x	TG3/4 fragmentation sub-header
00100x	TG3/4 packing sub-header

Table 1. TYPE field encoding

Grant management sub-header and ARQ feedback sub-header can co-exist with other sub-headers.

1. TG 3/4 Fragmentation Sub-header

This sub-header can be used on any connection (with or without ARQ). Its format is given in Table 2 below. For a non ARQ connection, this sub-header is used for fragmentation and reassembly. For an ARQ connection, the BSN in this sub-header is used for managing ARQ retransmissions, in addition to fragmentation and reassembly. The section on Block Numbering scheme describes the use of BSN for ARQ in detail.

2. TG 3/4 Packing Sub-header

Similar to the fragmentation sub-header, this sub-header can be used on any connection. It is used when the MPDU contains multiple MSDU or fragments. Its format is shown in Table 3.

Syntax	Size (bits)	Notes
Fragmentation Control	2	00 = unfragmented, 01 = last fragment, 10 = first fragment, 11 = middle fragment
Reserved	2	
Ack Request (“A-bit”)	1	Receiver must send an ack when this bit is set
Block Sequence Number	11	

Table 2 Format of TG3/4 fragmentation sub-header

Interpretation of A-bit in the fragmentation sub-header for ARQ

The A-bit is used by ARQ sender to request an ack from ARQ receiver. Implementations have the freedom to choose when to set the A-bit, depending on the system characteristics. For example, a system with low frame error rate can set A-bit less frequently than one with high frame rate, so as to reduce the overhead. The standard only requires that acks be requested or sent in a few special circumstances, such as when ARQ window limit is reached. The receiver may send an ack at any time, in addition to responding to A-bit.

Syntax	Size (bits)	Notes
Fragmentation Control	2	00 = unfragmented, 01 = last fragment, 10 = first fragment, 11 = middle fragment
Packed Payload Type	3	000 = Regular payload for this connection, 001 for ARQ control information. When Payload Type = 001, the FC bits must be ignored.
Length	11	

Table 3 Format of TG3/4 packing sub-header

5. ARQ Acknowledgements

The Payload Type field in the packing sub-header (Table 3) indicates if the information followed by the packing sub-header is regular payload or ARQ feedback message. Multiple acks can be packed inside the same ARQ feedback message. ARQ feedback message cannot be fragmented. Table 4 below shows the message format.

RR is used for cumulative acknowledgement. It is used when no out-of-sequence blocks are outstanding. The block sequence number (BSN) is the “next block expected”, indicating all blocks up to BSN–1 have been received.

RNR is used for flow control. The receiver sends RNR when it is in local busy condition. The BSN has the same meaning as in RR.

ACK is a special case of SACK, to indicate loss of one block without the overhead of R(n). It is used when all blocks up to BSN-1 have been received, and together with block BSN+1.

SACK is used when multiple blocks are missing. The length of R(n) is variable, up to ARQ window size. At the time when SACK is designated for transmission, the value of each bit R(n) shall be set to 0 or 1 depending on whether BSN+n has been received or not. R(n) = 1 indicates that BSN+n has been received, R(n) = 0 indicates otherwise. R(n) is truncated so that only bitmap octets up to and including the last bitmap octet containing at least one bit set to 1 are transmitted. All 0’s following the last 1 are insignificant.

Syntax	Size (bits)	Notes
While (Last != 0) {		
CID	16	
Last	1	= 0 if last ack, 1 if not
Ack Type	2	= 0 for RR, 1 for RNR, 2 for ACK, 3 for SACK
Reserved	2	
Block sequence number	11	
SACK Bitmap length k	8	Only present in SACK, unit = bytes
SACK Bitmap R(n)	variable	Only present in SACK, up to ARQ window size
}		

Table 4 Format of piggybacked acks

New DSx Information Elements to Support Block Numbering and ARQ

The block numbering scheme requires a new information element in the MAC messages DSA and DSC to indicate the block size. Additional information elements are required for ARQ negotiation. These extended TLV encoded information elements are listed below.

TLV Encoded IE	Size (bits)	Notes
Type = "BLOCK_SIZE"	8	This is used for both ARQ and non ARQ block numbering
Length	8	
Value	8	Unit = bytes. Must be powers of 2
Type = "ARQ_Command"	8	
Length	8	
Value	8	= 1 for SABM, = 2 for DISC
Type = "ARQ_Timeout"	8	Minimum time interval a transmitter shall wait before retransmitting a unacked ARQ block. This IE must be present with SABM
Length	8	
Value	8	

TLV Encoded IE	Size (bits)	Notes
Type = "ARQ_Lifetime"	8	Maximum time interval beyond which a transmitter shall discard unacked ARQ blocks. This IE must be present with SABM
Length	8	
Value	8	

New information elements for DSx-REQ

TLV encoded IE	Size (bits)	Notes
Type = "ARQ_Response"	8	This is required even if ARQ is not implemented, in which case a DM must be sent
Length	8	
Value	8	= 1 for UA, = 2 for DM
Type = "ARQ_WIN_SIZE"	8	This IE must be present with UA.
Length	8	
Value	8	Must not exceed 2^(n-1) where n = 10 is the number of bits in BSN field

New information elements for DSx-RSP

The above encoding scheme let the sender determine most of the ARQ parameters, except for the ARQ_WIND_SIZE, which is determined by the receiver. There is no negotiation. If the receiver cannot support the requested parameters, it must respond with a DM.

The procedures for using these information elements are described in the next section.

ARQ Procedures

LLE States and State Variables

Upon completion of DSA procedure, a connection is in one of the two modes:

ADM (Asynchronous Disconnected Mode): In this mode only non-ARQ data transfer is permitted.

ABM (Asynchronous Balanced Mode): In this mode only ARQ data transfer is permitted.

Sending LLE maintains the following state variables:

V(A): next ack expected, i.e., all BSN up to V(A)-1 have been acknowledged

V(S): next block to send

Retransmission Timer: retransmission timer that expires after ARQ_Timeout

Recovery Timer: recovery timer that expires after ARQ_Lifetime

Receiving LLE maintains the following state variables:

V(R): next block expected, i.e., all BSN up to V(R) have been received

V(H): next highest BSN expected, i.e., V(H)-1 is the highest BSN received so far

Establishment of ABM Mode

To establish a connection in the ABM mode, the following actions shall be taken in addition to the normal DSA procedure. The originating LLE shall send a DSA-REQ message with SABM and other ARQ parameters. The receiving LLE, if it is able to enter the ABM state, shall send back a DSA-RSP message with UA and other ARQ parameters. It then set V(R) and V(H) to 0, enters ABM state, and clear all existing exception conditions. Upon receiving DSA-RSP with UA, the originating LLE shall set V(A) and V(S) to 0, and enter ABM state.

If the receiving LLE is unable to enter the ABM state, it shall send back a DSA-RSP with a DM. Upon receiving DM, the originating LLE shall enter the ADM state.

Re-establishment of ABM Mode

The following event shall trigger the re-establishment of ABM operation:

Receiving connection setup request from layer management

Receiving an ack whose BSN is not in the ARQ send window

The Recovery Timer expires

The procedure for re-establishing ABM operation is the same as described before, except that DSC-REQ and DSC-RSP are used.

Termination of ABM Mode

Termination of ABM mode is done using DSC messages. The following actions shall be taken in addition to normal DSC procedure. The originating LLE shall send a DSC-REQ with DISC. The receiving LLE, if it is in the ABM mode, shall send DSC-RSP with UA, indicating a “normal” release, and enter ADM mode. Otherwise, it shall send DSC-RSP with DM, indicating it is already in ADM mode. Upon completion of DSC procedure, the originating LLE shall enter ADM mode.

The ARQ logical link automatically goes away if the MAC connection is terminated.

Procedures for Information Transfer in ABM Mode

ARQ blocks shall be transmitted in ascending BSN order. The sending LLE shall store the history of transmitted blocks, which is used to decide which blocks to retransmit. Due to retransmission, the history is not necessarily an in-order sequence. When blocks are retransmitted, the block with the lowest BSN shall be retransmitted first.

The sending LLE can send ARQ data with ARQ sub-header if it is not in the remote busy condition. It first checks V(A)+ARQ_Win_Size-V(S) to see how many blocks can be transmitted, and creates a MPDU that does not exceed this value. The state variable V(S) is copied into the BSN field before transmission, and V(S) is

incremented after transmission by the number of blocks in the MPDU. If the ARQ_Win_Size limit is reached due to the transmission of MPDU, then the A-bit must be set to 1. Otherwise, the choice of setting A-bit or not is implementation dependent. Timer T202 shall be started if not already running.

When a LLE receives a valid MPDU with BSN equal to V(R), and it is not in local busy condition, it shall

Pass the data to upper layer

Increment V(R) by the number of blocks in the received MPDU

If the A-bit field in the received ARQ sub-header is equal to 1, then respond with a RR, RNR, ACK, or SACK according to the criteria defined in the next section.

If the LLE is not in local busy condition and receives blocks that are not in the range [V(R), V(R)+ARQ_Win_Size), then the LLE shall discard those blocks that lay outside this range as duplicates.

If the LLE is not in local busy condition and receives blocks that are in the range (V(R), V(R)+ARQ_Win_Size), it shall update V(H) to 1 plus the highest received BSN, and store them until all blocks from V(R) to V(H)-1 inclusive are received. The LLE shall use the A-bit field to decide if an ack needs to be sent. It shall then pass all blocks up to V(H)-1 to the upper layer, and set V(R) to V(H).

When the LLE is in local busy condition and receives a valid MPDU, then the acceptance of the data is implementation dependent.

Sending Acknowledgements

Whenever a LLE receives an ARQ sub-header with A-bit set to 1, it shall transmit an acknowledgement. When it is not in local busy condition, it shall select either RR, ACK, or SACK according to the following criteria: If there are no missing blocks, it shall send a RR. If there is one missing block (i.e., V(H) = V(R)+1), it shall send an ACK. If there are multiple missing blocks, it shall send a SACK. V(R) is copied to BSN field before transmission.

If it is in local busy condition, the appropriate ack type is RNR.

Receiving Acknowledgements

When an ack is received, the LLE shall check the validity of BSN. A valid BSN is one in the range [V(A), V(S)]. If BSN is not valid, the LLE shall initiate ABM re-establishment procedure.

If BSN is valid, then the LLE shall consider all blocks in the range [V(A), BSN) as acked, set V(A) to BSN, stop timer T202, and clear remote busy condition. If there are unacked blocks, the LLE shall restart timer T202.

On receiving RNR, the LLE shall set remote busy condition.

On receiving a valid ACK message, the LLE shall consider block BSN+1 as acked, and clears remote busy condition.

On receiving a valid SACK message, the LLE shall consider all blocks with the corresponding bit set to 1 in the SACK Bitmap as acked, and clears remote busy condition.

If Retransmission Timer is active and is associated with an acked block, then it shall be stopped.

The LLE shall determine which blocks to retransmit by analyzing its transmission sequence history and the acks received. A unacked block transmitted earlier than an acked block shall be considered lost and shall be marked for retransmission. Acked blocks are removed from the transmission sequence history.

For the same connection, retransmission shall take priority over new blocks.

Requesting Acknowledgements

The LLE may request an ack at any time by setting the A-bit field to 1. An ack shall be requested if the ARQ window limit is reached as the result of current transmission. Other than this, an implementation can choose its own policy when to request acks.

When requesting an ack, the LLE shall set Retransmission Timer and associate the timer with the first block currently being transmitted.

Waiting for Acknowledgements

When the Retransmission Timer expires, the LLE looks at the block associated with the timer, and set Retransmission Timer and retransmit the unacked blocks starting with the one associated with this timer, with A-bit set to 1.

When Recovery Timer expires, the LLE shall discard all unacked blocks and initiate ABM re-establishment procedure.

Using UIUC for ARQ Information Exchange

The ARQ IE in the uplink map gives a base station the ability to simultaneously query a subscriber for ARQ information and provide the bandwidth necessary for the subscriber to provide that information. The CID supplied in the ID determines how the subscriber is expected to respond. If a specific transport CID is provided, the subscriber shall respond with the ARQ information it has available for that connection. If any space remains unused after having fulfilled the specific request, additional ARQ information for other CIDs may be included. The information selected for transmission shall be determined with the same algorithm as that described in the next paragraph for queries to the Basic CID.

When the subscriber's Basic CID is specified, the subscriber shall include information for as many ARQ managed connections as will fit in the bandwidth provided. If necessary, the subscriber shall limit the number of map elements included for each connection so that information on the maximum number of connections is provided. If after limiting map elements, there still is insufficient bandwidth available to carry information for all connections, connection information shall be included in the message based on the time the ARQ data has been awaiting transmission. The connections with data which has been waiting the longest shall be included in the transmission first.

The following texts are suggested modifications to other parts of the document:

Page 127: Following section 6.2.7.5.3.6 Empty IE **insert the following:**

6.2.7.5.3.6 ARQ Burst Type IEs (2-11GHz only)

With the ARQ Burst Type IE, a base station is able to simultaneously query a subscriber for ARQ information and grant bandwidth to carry the transmission. Based on the (transport or basic) CID included in uplink map IE, the query may be limited to a single connection or generalized allowing the subscriber to send any ARQ information it may have pending for transmission. See section x.x.x.x "ARQ -----" for additional information.

Page xxx: In the UIUC table/s in section 8.3.xxxx (corresponding to Table 108 in section 8.2.5.3) include an entry with the following info:

IE Name: ARQ

Uplink Interval Usage Code (UIUC): TBD (possibly 12)

Connection Id: Basic or Transport

Description: Starting offset of region to be used for transmission of pending ARQ information.

2. ARQ ad hoc committee activity report

Date	Event
July 17	Invitation to join TG3/4 ad hoc group on ARQ was issued, which defined the rules and schedules of the committee. Eventually, 29 people (listed below) signed up for the ARQ ad hoc committee.
August 2	Jin-Meng Ho submitted Proposal #1.
August 6	Huanchun Ye submitted Proposal #2.
August 16	After extensive discussions, a straw poll was called on 10 major suggestions by the group. They covered elements of both Proposals #1 and #2.
August 20	Straw poll results published.
August 22	Yigal Leiba submitted Proposal #3.
August 27	Rev.1 of Proposal #2 was published incorporating all the results of the straw poll of 8/16. Some elements of Proposal #1 were included. Later the author withdrew #1 from further consideration.
August 29	Two more straw polls were called on further comments made by members on Proposal #2 Rev.1
August 29	A vote was called on choosing Proposal #2 or #3 as the ad hoc output. The voting deadline was later extended to 9/4 to allow more time to work on merging them.
August 31	Results of straw polls of 8/29 published.
September 4	Proposal #2 rev 2 was published incorporating the results of the straw polls.
September 4	Another straw poll was called on the main points in Proposal #3 for the purpose of including them in Proposal #2.
September 4	Voting results was published. Five people actually submitted their votes and by a 3 to 2 margin Proposal #2 Rev.2 was selected as the ad hoc output.
September 5	Straw poll of 9/4 will be concluded by the end of this day. Any points accepted will be incorporated into the ad hoc output.

Ad hoc group member list:

1	Andrew Garrett
2	Arthur Wang
3	Baruch Buskila
4	Bob Nelson

5	Chin-Chen Lee
6	Fabrizio Talucci
7	Heinz Lycklama
8	Huanchun Ye
9	Itzik Kitroser
10	Jack Fijolek
11	Jacob W. Jorgensen
12	Jin-Meng Ho
13	Juha Salokannel
14	Jung Je Son
15	Ken Peirce
16	Kyunghun Jang
17	Lei Wang
18	Malik Audeh
19	Ofer Kelman
20	Paul Truong
21	Radu Selea
22	Subbu Ponnuswamy
23	Subir Varma
24	Todd Taylor
25	Vladimir Yanover
26	Walt Roehr
27	Won-Hyoung Park
28	Xiaolin Lu
29	Yigal Leiba