

Project	IEEE 802.16 Broadband Wireless Access Working Group < http://ieee802.org/16 >
Title	Proposed text and ASN.1 code for QoS Management
Date Submitted	2007-05-08
Source(s)	Joey Chou Intel Corporation [mailto:joey.chou@intel.com]
Re:	
Abstract	This contribution proposes the text and ASN.1 code in wmanIf2mMib to support QoS management.
Purpose	Adoption
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 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 802.16.
Patent Policy and Procedures	<p>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."</p> <p>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>.</p>

Table of Content

- 1. Introduction..... 3**
- 2. Proposed changes..... 3**
 - 2.1 wmanlf2mMib Change..... 3**
 - 2.2 ASN.1 Code Change..... 4**

1

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
27
28
29
30

1. Introduction

This contribution proposes the text and ASN.1 code in wmanIf2mMib to support QoS management.

2. Proposed changes

2.1 wmanIf2mMib Change

[Add a new subclause as the following:]

13.1.4.3 wmanIf2mCmnObjects

13.1.4.3.1 wmanIf2mCmnCm

Figure 19 shows the structure of wmanIf2mCmnCm subtree that contains common managed objects related to Configuration Management.

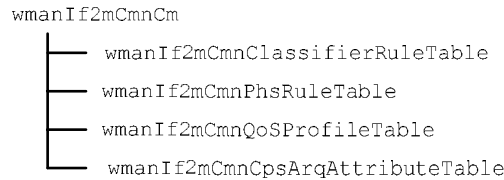


Figure 19—wmanIf2mCmnCm structure

[Change the following subclause as below:]

13.1.4.3.1.1 wmanIf2mCmnClassifierRuleTable

wmanIf2mCmnQoSProfileTable contains packet classifier rules associated with service flows.

13.1.4.3.1.2 wmanIf2mCmnPhsRuleTable

wmanIf2mCmnPhsRuleTable contains PHS rule dictionary entries. Each entry contains the data of the header to be suppressed along with its identification - PHSI.

13.1.4.3.1.3 wmanIf2mCmnQoSProfileTable

wmanIf2mCmnQoSProfileTable contains QoS profiles that are associated with service flows or CIDs via the wmanIf2mCmnQoSProfileIndex.

13.1.4.3.1.4 wmanIf2mCmnArqAttributeTable

wmanIf2mCmnArqAttributeTable contains ARQ parameters that are associated with the Service Flows.

1 2.2 ASN.1 Code Change

2 13.2 ASN.1 Definitions of MIB Modules

3 13.2.3 wmanIf2mMib

4 [Add the following ASN.1 code:]

```

5
6 -- XXX
7 wmanIf2mCmnClassifierRuleTable OBJECT-TYPE
8     SYNTAX      SEQUENCE OF WmanIf2mCmnClassifierRuleEntry
9     MAX-ACCESS  not-accessible
10    STATUS      current
11    DESCRIPTION
12        "This table contains packet classifier rules associated
13         with service flows."
14    ::= { wmanIf2mCmnCm 1 }
15
16 wmanIf2mCmnClassifierRuleEntry OBJECT-TYPE
17     SYNTAX      WmanIf2mCmnClassifierRuleEntry
18     MAX-ACCESS  not-accessible
19     STATUS      current
20     DESCRIPTION
21        "This table provides one row for each packet classifier rule
22         , and is indexed by ifIndex, and
23         wmanIf2mCmnClassifierRuleIndex."
24     INDEX { ifIndex, wmanIf2mCmnClassifierRuleIndex }
25     ::= { wmanIf2mCmnClassifierRuleTable 1 }
26
27 WmanIf2mCmnClassifierRuleEntry ::= SEQUENCE {
28     wmanIf2mCmnClassifierRuleIndex      Unsigned32,
29     wmanIf2mCmnClassifierRulePriority    INTEGER,
30     wmanIf2mCmnClassifierRuleIpTosLow   INTEGER,
31     wmanIf2mCmnClassifierRuleIpTosHigh  INTEGER,
32     wmanIf2mCmnClassifierRuleIpTosMask  INTEGER,
33     wmanIf2mCmnClassifierRuleIpProtocol Integer32,
34     wmanIf2mCmnClassifierRuleIpSrcAddr   InetAddress,
35     wmanIf2mCmnClassifierRuleIpSrcMask   InetAddress,
36     wmanIf2mCmnClassifierRuleIpDestAddr  InetAddress,
37     wmanIf2mCmnClassifierRuleIpDestMask InetAddress,
38     wmanIf2mCmnClassifierRuleSrcPortStart Integer32,
39     wmanIf2mCmnClassifierRuleSrcPortEnd  Integer32,
40     wmanIf2mCmnClassifierRuleDestPortStart Integer32,
41     wmanIf2mCmnClassifierRuleDestPortEnd Integer32,
42     wmanIf2mCmnClassifierRuleDestMacAddr MacAddress,
43     wmanIf2mCmnClassifierRuleDestMacMask MacAddress,
44     wmanIf2mCmnClassifierRuleSrcMacAddr  MacAddress,
45     wmanIf2mCmnClassifierRuleSrcMacMask  MacAddress,
46     wmanIf2mCmnClassifierRuleEnetType    INTEGER,
47     wmanIf2mCmnClassifierRuleEnetProtocol Integer32,
48     wmanIf2mCmnClassifierRuleUserPriLow  Integer32,
49     wmanIf2mCmnClassifierRuleUserPriHigh Integer32,
50     wmanIf2mCmnClassifierRuleVlanId      Integer32,
51     wmanIf2mCmnClassifierRuleIpv6FlowLabel WmanIf2mIpv6FlowLabel,
52     wmanIf2mCmnClassifierRulePkts       Counter64,
53     wmanIf2mCmnClassifierRuleBitMap     WmanIf2mClassifierBitMap
54 }
55
56 wmanIf2mCmnClassifierRuleIndex OBJECT-TYPE
57     SYNTAX      Unsigned32 (1..4294967295)
58     MAX-ACCESS  not-accessible
59     STATUS      current

```

```

1      DESCRIPTION
2          "An index is assigned to each classifier in the classifiers
3          table"
4      REFERENCE
5          "Subclause 11.13.19.3.4.14 in IEEE Std 802.16-2004"
6          ::= { wmanIf2mCmnClassifierRuleEntry 1 }
7
8      wmanIf2mCmnClassifierRulePriority OBJECT-TYPE
9          SYNTAX      INTEGER (0..255)
10         MAX-ACCESS  read-only
11         STATUS      current
12         DESCRIPTION
13             "The value specifies the order of evaluation of the
14             classifiers. The higher the value the higher the priority.
15             The value of 0 is used as default in provisioned service
16             flows classifiers. The default value of 64 is used for
17             dynamic service flow classifiers. If the referenced
18             parameter is not present in a classifier, this object
19             reports the default value as defined above"
20         REFERENCE
21             "Subclause 11.13.19.3.4.1 in IEEE Std 802.16-2004"
22         DEFVAL      { 0 }
23         ::= { wmanIf2mCmnClassifierRuleEntry 2 }
24
25     wmanIf2mCmnClassifierRuleIpTosLow OBJECT-TYPE
26         SYNTAX      INTEGER (0 .. 255)
27         MAX-ACCESS  read-only
28         STATUS      current
29         DESCRIPTION
30             "The low value of a range of TOS byte values. If the
31             referenced parameter is not present in a classifier, this
32             object reports the value of 0."
33         REFERENCE
34             "Subclause 11.13.19.3.4.2 in IEEE Std 802.16-2004"
35         ::= { wmanIf2mCmnClassifierRuleEntry 3 }
36
37     wmanIf2mCmnClassifierRuleIpTosHigh OBJECT-TYPE
38         SYNTAX      INTEGER (0 .. 255)
39         MAX-ACCESS  read-only
40         STATUS      current
41         DESCRIPTION
42             "The 8-bit high value of a range of TOS byte values. If the
43             referenced parameter is not present in a classifier, this
44             object reports the value of 0."
45         REFERENCE
46             "Subclause 11.13.19.3.4.2 in IEEE Std 802.16-2004"
47         ::= { wmanIf2mCmnClassifierRuleEntry 4 }
48
49     wmanIf2mCmnClassifierRuleIpTosMask OBJECT-TYPE
50         SYNTAX      INTEGER (0 .. 255)
51         MAX-ACCESS  read-only
52         STATUS      current
53         DESCRIPTION
54             "The mask value is bitwise ANDed with TOS byte in an IP
55             packet and this value is used for the range checking of
56             TosLow and TosHigh. If the referenced parameter is not
57             present in a classifier, this object reports the value of
58             0."
59         REFERENCE
60             "Subclause 11.13.19.3.4.2 in IEEE Std 802.16-2004"
61         ::= { wmanIf2mCmnClassifierRuleEntry 5 }
62
63     wmanIf2mCmnClassifierRuleIpProtocol OBJECT-TYPE
64         SYNTAX      Integer32 (0..255)

```

```

1      MAX-ACCESS  read-only
2      STATUS      current
3      DESCRIPTION
4          "This object indicates the value of the IP Protocol field
5           required for IP packets to match this rule. If the
6           referenced parameter is not present in a classifier, this
7           object reports the value of 0."
8      REFERENCE
9          "Subclause 11.13.19.3.4.3 in IEEE Std 802.16-2004"
10     ::= { wmanIf2mCmnClassifierRuleEntry 6 }
11
12     wmanIf2mCmnClassifierRuleIpSrcAddr OBJECT-TYPE
13         SYNTAX      InetAddress
14         MAX-ACCESS  read-only
15         STATUS      current
16         DESCRIPTION
17             "This object specifies the value of the IP Source Address
18              required for packets to match this rule. An IP packet
19              matches the rule when the packet ip source address bitwise
20              ANDED with the wmanIf2mCmnClassifierRuleIpSrcMask value
21              equals the wmanIf2mCmnClassifierRuleIpSrcAddr value.
22              If the referenced parameter is not present in a classifier
23              , this object reports the value of 0.0.0.0."
24         REFERENCE
25             "Subclause 11.13.19.3.4.4 in IEEE Std 802.16-2004"
26         ::= { wmanIf2mCmnClassifierRuleEntry 7 }
27
28     wmanIf2mCmnClassifierRuleIpSrcMask OBJECT-TYPE
29         SYNTAX      InetAddress
30         MAX-ACCESS  read-only
31         STATUS      current
32         DESCRIPTION
33             "This object specifies which bits of a packet's IP Source
34              Address that are compared to match this rule. An IP packet
35              matches the rule when the packet source address bitwise
36              ANDED with the
37              wmanIf2mCmnClassifierRuleIpSrcMask value equals the
38              wmanIf2mCmnClassifierRuleIpSrcAddr value.
39              If the referenced parameter is not present in a classifier
40              , this object reports the value of 0.0.0.0."
41         REFERENCE
42             "Subclause 11.13.19.3.4.4 in IEEE Std 802.16-2004"
43         ::= { wmanIf2mCmnClassifierRuleEntry 8 }
44
45     wmanIf2mCmnClassifierRuleIpDestAddr OBJECT-TYPE
46         SYNTAX      InetAddress
47         MAX-ACCESS  read-only
48         STATUS      current
49         DESCRIPTION
50             "This object specifies the value of the IP Destination
51              Address required for packets to match this rule. An IP
52              packet matches the rule when the packet IP destination
53              address bitwise ANDED with the
54              wmanIf2mCmnClassifierRuleIpDestMask value equals the
55              wmanIf2mCmnClassifierRuleIpDestAddr value.
56              If the referenced parameter is not present in a
57              classifier, this object reports the value of 0.0.0.0."
58         REFERENCE
59             "Subclause 11.13.19.3.4.5 in IEEE Std 802.16-2004"
60         ::= { wmanIf2mCmnClassifierRuleEntry 9 }
61
62     wmanIf2mCmnClassifierRuleIpDestMask OBJECT-TYPE
63         SYNTAX      InetAddress
64         MAX-ACCESS  read-only

```

```

1      STATUS      current
2      DESCRIPTION
3          "This object specifies which bits of a packet's IP
4      Destination Address that are compared to match this rule.
5      An IP packet matches the rule when the packet destination
6      address bitwise ANDed with the
7      wmanIf2mCmnClassifierRuleIpDestMask value equals the
8      wmanIf2mCmnClassifierRuleIpDestAddr value.
9      If the referenced parameter is not present in a classifier
10     , this object reports the value of 0.0.0.0."
11     REFERENCE
12         "Subclause 11.13.19.3.4.5 in IEEE Std 802.16-2004"
13     ::= { wmanIf2mCmnClassifierRuleEntry 10 }
14
15     wmanIf2mCmnClassifierRuleSrcPortStart OBJECT-TYPE
16     SYNTAX      Integer32 (0..65535)
17     MAX-ACCESS  read-only
18     STATUS      current
19     DESCRIPTION
20         "This object specifies the low end inclusive range of
21     TCP/UDP source port numbers to which a packet is compared
22     . This object is irrelevant for non-TCP/UDP IP packets.
23     If the referenced parameter is not present in a
24     classifier, this object reports the value of 0."
25     REFERENCE
26         "Subclause 11.13.19.3.4.6 in IEEE Std 802.16-2004"
27     ::= { wmanIf2mCmnClassifierRuleEntry 11 }
28
29     wmanIf2mCmnClassifierRuleSrcPortEnd OBJECT-TYPE
30     SYNTAX      Integer32 (0..65535)
31     MAX-ACCESS  read-only
32     STATUS      current
33     DESCRIPTION
34         "This object specifies the high end inclusive range of
35     TCP/UDP source port numbers to which a packet is compared.
36     This object is irrelevant for non-TCP/UDP IP packets.
37     If the referenced parameter is not present in a classifier,
38     this object reports the value of 65535."
39     REFERENCE
40         "Subclause 11.13.19.3.4.6 in IEEE Std 802.16-2004"
41     ::= { wmanIf2mCmnClassifierRuleEntry 12 }
42
43     wmanIf2mCmnClassifierRuleDestPortStart OBJECT-TYPE
44     SYNTAX      Integer32 (0..65535)
45     MAX-ACCESS  read-only
46     STATUS      current
47     DESCRIPTION
48         "This object specifies the low end inclusive range of
49     TCP/UDP destination port numbers to which a packet is
50     compared. If the referenced parameter is not present in a
51     classifier, this object reports the value of 0."
52     REFERENCE
53         "Subclause 11.13.19.3.4.7 in IEEE Std 802.16-2004"
54     ::= { wmanIf2mCmnClassifierRuleEntry 13 }
55
56     wmanIf2mCmnClassifierRuleDestPortEnd OBJECT-TYPE
57     SYNTAX      Integer32 (0..65535)
58     MAX-ACCESS  read-only
59     STATUS      current
60     DESCRIPTION
61         "This object specifies the high end inclusive range of
62     TCP/UDP destination port numbers to which a packet is
63     compared. If the referenced parameter is not present
64     in a classifier, this object reports the value of

```

```

1           65535."
2     REFERENCE
3       "Subclause 11.13.19.3.4.7 in IEEE Std 802.16-2004"
4     ::= { wmanIf2mCmnClassifierRuleEntry 14 }
5
6     wmanIf2mCmnClassifierRuleDestMacAddr OBJECT-TYPE
7       SYNTAX      MacAddress
8       MAX-ACCESS  read-only
9       STATUS      current
10      DESCRIPTION
11        "An Ethernet packet matches an entry when its destination
12        MAC address bitwise ANDed with
13        wmanIf2mCmnClassifierRuleDestMacMask equals the value of
14        wmanIf2mCmnClassifierRuleDestMacAddr. If the referenced
15        parameter is not present in a classifier, this object
16        reports the value of '000000000000'H."
17      REFERENCE
18        "Subclause 11.13.19.3.4.8 in IEEE Std 802.16-2004"
19      ::= { wmanIf2mCmnClassifierRuleEntry 15 }
20
21     wmanIf2mCmnClassifierRuleDestMacMask OBJECT-TYPE
22       SYNTAX      MacAddress
23       MAX-ACCESS  read-only
24       STATUS      current
25       DESCRIPTION
26        "An Ethernet packet matches an entry when its destination
27        MAC address bitwise ANDed with
28        wmanIf2mCmnClassifierRuleDestMacMask equals the value of
29        wmanIf2mCmnClassifierRuleDestMacAddr. If the referenced
30        parameter is not present in a classifier, this object
31        reports the value of '000000000000'H."
32      REFERENCE
33        "Subclause 11.13.19.3.4.8 in IEEE Std 802.16-2004"
34      ::= { wmanIf2mCmnClassifierRuleEntry 16 }
35
36     wmanIf2mCmnClassifierRuleSrcMacAddr OBJECT-TYPE
37       SYNTAX      MacAddress
38       MAX-ACCESS  read-only
39       STATUS      current
40       DESCRIPTION
41        "An Ethernet packet matches this entry when its source MAC
42        address bitwise ANDed with
43        wmanIf2mCmnClassifierRuleSrcMacMask equals the value of
44        wmanIf2mCmnClassifierRuleSrcMacAddr. If the referenced
45        parameter is not present in a classifier, this object
46        reports the value of '000000000000'H."
47      REFERENCE
48        "Subclause 11.13.19.3.4.9 in IEEE Std 802.16-2004"
49      ::= { wmanIf2mCmnClassifierRuleEntry 17 }
50
51     wmanIf2mCmnClassifierRuleSrcMacMask OBJECT-TYPE
52       SYNTAX      MacAddress
53       MAX-ACCESS  read-only
54       STATUS      current
55       DESCRIPTION
56        "An Ethernet packet matches an entry when its destination
57        MAC address bitwise ANDed with
58        wmanIf2mCmnClassifierRuleSrcMacMask equals the value of
59        wmanIf2mCmnClassifierRuleSrcMacAddr. If the referenced
60        parameter is not present in a classifier, this object
61        reports the value of '000000000000'H."
62      REFERENCE
63        "Subclause 11.13.19.3.4.9 in IEEE Std 802.16-2004"
64      ::= { wmanIf2mCmnClassifierRuleEntry 18 }

```



```

1
2 wmanIf2mCmnClassifierRuleEnetType OBJECT-TYPE
3     SYNTAX      INTEGER {none(0),
4                   ethertype(1),
5                   dsap(2)}
6     MAX-ACCESS  read-only
7     STATUS      current
8     DESCRIPTION
9         "This object indicates the format of the layer 3 protocol
10        id in the Ethernet packet. A value of none(0) means that
11        the rule does not use the layer 3 protocol type as a
12        matching criteria. A value of ethertype(1) means that the
13        rule applies only to frames which contains an EtherType
14        value. Ethertype values are contained in packets using
15        the Dec-Intel-Xerox (DIX) encapsulation or the RFC1042
16        Sub-Network Access Protocol (SNAP) encapsulation formats.
17        A value of dsap(2) means that the rule applies only to
18        frames using the IEEE802.3 encapsulation format with a
19        Destination Service Access Point (DSAP) other than 0xAA
20        (which is reserved for SNAP). If the Ethernet frame
21        contains an 802.1P/Q Tag header (i.e. EtherType 0x8100),
22        this object applies to the embedded EtherType field within
23        the 802.1P/Q header. If the referenced parameter is not
24        present in a classifier, this object reports the value of
25        0."
26     REFERENCE
27         "Subclause 11.13.19.3.4.10 in IEEE Std 802.16-2004"
28     ::= { wmanIf2mCmnClassifierRuleEntry 19 }
29
30 wmanIf2mCmnClassifierRuleEnetProtocol OBJECT-TYPE
31     SYNTAX      Integer32 (0..65535)
32     MAX-ACCESS  read-only
33     STATUS      current
34     DESCRIPTION
35         "If wmanIf2mCmnClassifierRuleEnetType is none(0), this
36        object is ignored when considering whether a packet matches
37        the current rule. If wmanIf2mCmnClassifierRuleEnetType is
38        ethertype(1), this object gives the 16-bit value of the
39        EtherType that the packet must match in order to match the
40        rule. If wmanIf2mCmnClassifierRuleEnetType is dsap(2), the
41        lower 8 bits of this object's value must match the DSAP
42        byte of the packet in order to match the rule.
43        If the Ethernet frame contains an 802.1P/Q Tag header
44        (i.e. EtherType 0x8100), this object applies to the
45        embedded EtherType field within the 802.1P/Q header.
46        If the referenced parameter is not present in the
47        classifier, the value of this object is reported as 0."
48     REFERENCE
49         "Subclause 11.13.19.3.4.10 in IEEE Std 802.16-2004"
50     ::= { wmanIf2mCmnClassifierRuleEntry 20 }
51
52 wmanIf2mCmnClassifierRuleUserPriLow OBJECT-TYPE
53     SYNTAX      Integer32 (0..7)
54     MAX-ACCESS  read-only
55     STATUS      current
56     DESCRIPTION
57         "This object applies only to Ethernet frames using the
58        802.1P/Q tag header (indicated with EtherType 0x8100).
59        Such frames include a 16-bit Tag that contains a 3 bit
60        Priority field and a 12 bit VLAN number.
61        Tagged Ethernet packets must have a 3-bit Priority field
62        within the range of wmanIf2mCmnClassifierRulePriLow and
63        wmanIf2mCmnClassifierRulePriHigh in order to match this
64        rule."

```

```

1           If the referenced parameter is not present in the
2           classifier, the value of this object is reported as 0."
3 REFERENCE
4           "Subclause 11.13.19.3.4.11 in IEEE Std 802.16-2004"
5 ::= { wmanIf2mCmnClassifierRuleEntry 21 }
6
7 wmanIf2mCmnClassifierRuleUserPriHigh OBJECT-TYPE
8     SYNTAX      Integer32 (0..7)
9     MAX-ACCESS  read-only
10    STATUS      current
11    DESCRIPTION
12       "This object applies only to Ethernet frames using the
13       802.1P/Q tag header (indicated with EtherType 0x8100).
14       Such frames include a 16-bit Tag that contains a 3 bit
15       Priority field and a 12 bit VLAN number.
16       Tagged Ethernet packets must have a 3-bit Priority
17       field within the range of wmanIf2mCmnClassifierRulePriLow
18       and wmanIf2mCmnClassifierRulePriHigh in order to match
19       this rule.
20       If the referenced parameter is not present in the
21       classifier, the value of this object is reported as 7."
22    REFERENCE
23       "Subclause 11.13.19.3.4.11 in IEEE Std 802.16-2004"
24    ::= { wmanIf2mCmnClassifierRuleEntry 22 }
25
26 wmanIf2mCmnClassifierRuleVlanId OBJECT-TYPE
27     SYNTAX      Integer32 (0..4095)
28     MAX-ACCESS  read-only
29     STATUS      current
30     DESCRIPTION
31       "This object applies only to Ethernet frames using the
32       802.1P/Q tag header.
33       If this object's value is nonzero, tagged packets must
34       have a VLAN Identifier that matches the value in order
35       to match the rule.
36       Only the least significant 12 bits of this object's
37       value are valid.
38       If the referenced parameter is not present in the
39       classifier, the value of this object is reported as 0."
40    REFERENCE
41       "Subclause 11.13.19.3.4.12 in IEEE Std 802.16-2004"
42    ::= { wmanIf2mCmnClassifierRuleEntry 23 }
43
44 wmanIf2mCmnClassifierRuleIpv6FlowLabel OBJECT-TYPE
45     SYNTAX      WmanIf2mIpv6FlowLabel
46     MAX-ACCESS  read-only
47     STATUS      current
48     DESCRIPTION
49       "The value of this field specifies the matching values for
50       the IPv6 Flow label field."
51    ::= { wmanIf2mCmnClassifierRuleEntry 24 }
52
53 wmanIf2mCmnClassifierRulePkts OBJECT-TYPE
54     SYNTAX      Counter64
55     MAX-ACCESS  read-only
56     STATUS      current
57     DESCRIPTION
58       "This object counts the number of packets that have
59       been classified using this entry."
60    ::= { wmanIf2mCmnClassifierRuleEntry 25 }
61
62 wmanIf2mCmnClassifierRuleBitMap OBJECT-TYPE
63     SYNTAX      WmanIf2mClassifierBitMap
64     MAX-ACCESS  read-only

```

```

1      STATUS      current
2      DESCRIPTION
3          "This object indicates which parameter encodings were
4          actually present in the entry. A bit set to '1' indicates
5          the corresponding classifier encoding is present, and '0'
6          means otherwise"
7      ::= { wmanIf2mCmnClassifierRuleEntry 26 }
8
9      -- XXX
10     wmanIf2mCmnPhsRuleTable OBJECT-TYPE
11         SYNTAX      SEQUENCE OF WmanIf2mCmnPhsRuleEntry
12         MAX-ACCESS  not-accessible
13         STATUS      current
14         DESCRIPTION
15             "This table contains PHS rule dictionary entries. Each
16             entry contains the data of the header to be suppressed
17             along with its identification - PHSI. The classifier
18             uniquely maps packets to its associated PHS Rule. The
19             receiving entity uses the CID and the PHSI to restore the
20             PHSF. Once a PHSF has been assigned to a PHSI, it shall
21             not be changed. To change the value of a PHSF on a
22             service flow, a new PHS rule shall be defined, the old
23             rule is removed from the service flow, and the new rule
24             is added. When a classifier is deleted, any associated
25             PHS rule shall also be deleted."
26         REFERENCE
27             "Subclause 5.2.3 in IEEE Std 802.16-2004"
28         ::= { wmanIf2mCmnCm 2 }
29
30     wmanIf2mCmnPhsRuleEntry OBJECT-TYPE
31         SYNTAX      WmanIf2mCmnPhsRuleEntry
32         MAX-ACCESS  not-accessible
33         STATUS      current
34         DESCRIPTION
35             "This table provides one row for each PHS rule created
36             dynamically by the BS and SS on a given service flow. The
37             PHS rule is defined by the pair (PHSS, PHSM) for each
38             distinct header data. It is indexed by IfIndex, and
39             wmanIf2mCmnPhsIndex."
40         INDEX      { ifIndex, wmanIf2mCmnPhsRulePhsIndex }
41         ::= { wmanIf2mCmnPhsRuleTable 1 }
42
43     WmanIf2mCmnPhsRuleEntry ::= SEQUENCE {
44         wmanIf2mCmnPhsRulePhsIndex      INTEGER,
45         wmanIf2mCmnPhsRulePhsField      OCTET STRING,
46         wmanIf2mCmnPhsRulePhsMask      OCTET STRING,
47         wmanIf2mCmnPhsRulePhsSize      Integer32,
48         wmanIf2mCmnPhsRulePhsVerify    WmanIf2mPhsRuleVerify}
49
50     wmanIf2mCmnPhsRulePhsIndex OBJECT-TYPE
51         SYNTAX      INTEGER (1 .. 255)
52         MAX-ACCESS  not-accessible
53         STATUS      current
54         DESCRIPTION
55             "The PHSI (PHS Index) has a value between 1 and 255, which
56             uniquely references the suppressed byte string. The index
57             is unique per service flow. The uplink and downlink PHSI
58             values are independent of each other."
59         REFERENCE
60             "Subclause 11.13.19.3.7.1 in IEEE Std 802.16-2004"
61         ::= { wmanIf2mCmnPhsRuleEntry 1 }
62
63     wmanIf2mCmnPhsRulePhsField OBJECT-TYPE
64         SYNTAX      OCTET STRING (SIZE(0..65535))

```

```

1      MAX-ACCESS read-only
2      STATUS current
3      DESCRIPTION
4          "The PHSF (PHS Field) is a string of bytes containing the
5          header information to be suppressed by the sending CS and
6          reconstructed by the receiving CS. The most significant
7          byte of the string corresponds to the first byte of the
8          CS-SDU."
9      REFERENCE
10         "Subclause 11.13.19.3.7.2 in IEEE Std 802.16-2004"
11     ::= { wmanIf2mCmnPhsRuleEntry 2 }
12
13 wmanIf2mCmnPhsRulePhsMask OBJECT-TYPE
14     SYNTAX OCTET STRING (SIZE(0..65535))
15     MAX-ACCESS read-only
16     STATUS current
17     DESCRIPTION
18         "The PHSM An 8-bit mask that indicates which bytes in the
19         PHS Field (PHSF) to suppress and which bytes to not
20         suppress. The PHSM allows fields, such as sequence numbers
21         or checksums (which vary in value), to be excluded from
22         suppression with the constant bytes around them suppressed.
23         It is encoded as follows:
24         bit 0:
25             0 = don't suppress the 1st byte of the suppression field
26             1 = suppress first byte of the suppression field
27         bit 1:
28             0 = don't suppress the 2nd byte of the suppression field
29             1 = suppress second byte of the suppression field
30         bit x:
31             0 = don't suppress the (x+1) byte of the suppression
32             field
33             1 = suppress (x+1) byte of the suppression field
34         where the length of the octet string is ceiling
35         (wmanIf2mCmnPhsRulePhsSize/8)."
```

```

36     REFERENCE
37         "Subclause 11.13.19.3.7.3 in IEEE Std 802.16-2004"
38     ::= { wmanIf2mCmnPhsRuleEntry 3 }
39
40 wmanIf2mCmnPhsRulePhsSize OBJECT-TYPE
41     SYNTAX Integer32 (0..255)
42     UNITS "byte"
43     MAX-ACCESS read-only
44     STATUS current
45     DESCRIPTION
46         "The value of this field - PHSS is the total number of bytes
47         in the header to be suppressed and then restored in a
48         service flow that uses PHS."
49     REFERENCE
50         "Subclause 11.13.19.3.7.4 in IEEE Std 802.16-2004"
51     DEFVAL { 0 }
52     ::= { wmanIf2mCmnPhsRuleEntry 4 }
53
54 wmanIf2mCmnPhsRulePhsVerify OBJECT-TYPE
55     SYNTAX WmanIf2mPhsRuleVerify
56     MAX-ACCESS read-only
57     STATUS current
58     DESCRIPTION
59         "The value of this field indicates to the sending entity
60         whether or not the packet header contents are to be
61         verified prior to performing suppression."
62     DEFVAL { phsVerifyEnable }
63     ::= { wmanIf2mCmnPhsRuleEntry 5 }
64

```

```

1  -- XXX
2  wmanIf2mCmnQoSProfileTable OBJECT-TYPE
3      SYNTAX      SEQUENCE OF WmanIf2mCmnQoSProfileEntry
4      MAX-ACCESS  not-accessible
5      STATUS      current
6      DESCRIPTION
7          "This table contains QoS profiles that are associated with
8          service flows or CIDs via the wmanIf2mCmnQoSProfileIndex.
9
10         The following table shows the required parameters for
11         different UL grant scheduling type.
12             0 - not required
13             1 - required
14             0-1 - optional
15
16         QoS Parameters                BE  ertPS  UGS  rtPS  nrtPS
17         -----
18         Traffic priority                0-1  0-1   0    0-1  0-1
19         Max sustained traffic rate       0-1  0-1   1    0-1  0-1
20         Min reserved traffic rate        0    1     0-1  1     1
21         Minimum traffic burst            0    0-1   0    0-1  0-1
22         Tolerated jitter                  0    0-1   1     0     0
23         Maximum latency                   0    1     1     1     0
24         Unsolicited Grant Interval       0    1     1     0     0
25         SDU size                          0    0     0-1   0     0
26         Unsolicited Polling Interval     0    0     0     1     0"
27     REFERENCE
28         "Subclause 6.3.14.4 in IEEE Std 802.16-2004"
29     ::= { wmanIf2mCmnCm 3 }
30
31  wmanIf2mCmnQoSProfileEntry OBJECT-TYPE
32      SYNTAX      WmanIf2mCmnQoSProfileEntry
33      MAX-ACCESS  not-accessible
34      STATUS      current
35      DESCRIPTION
36          "This table provides one row for each QoS parameter Set."
37      INDEX { ifIndex, wmanIf2mCmnQoSProfileIndex }
38      ::= { wmanIf2mCmnQoSProfileTable 1 }
39
40  WmanIf2mCmnQoSProfileEntry ::= SEQUENCE {
41      wmanIf2mCmnQoSProfileIndex          INTEGER,
42      wmanIf2mCmnQoSServiceClassName      OCTET STRING,
43      wmanIf2mCmnQoSULGrantScheduleType   WmanIf2mSchedulingType,
44      wmanIf2mCmnQoSTrafficPriority        INTEGER,
45      wmanIf2mCmnQoSMaximumSustainedRate  Unsigned32,
46      wmanIf2mCmnQoSMinimumReservedRate   Unsigned32,
47      wmanIf2mCmnQoSMaximumTrafficBurst   Unsigned32,
48      wmanIf2mCmnQoS_ToleratedJitter      Unsigned32,
49      wmanIf2mCmnQoS_MaxLatency            Unsigned32,
50      wmanIf2mCmnQoSUnsolicitedGrantInterval Unsigned32,
51      wmanIf2mCmnQoS_SduSize               Unsigned32,
52      wmanIf2mCmnQoSUnsolicitedPollInterval Unsigned32 }
53
54  wmanIf2mCmnQoSProfileIndex OBJECT-TYPE
55      SYNTAX      INTEGER (1 .. 65535)
56      MAX-ACCESS  not-accessible
57      STATUS      current
58      DESCRIPTION
59          "The index value which uniquely identifies an entry in the
60          wmanIf2mCmnQoSProfileTable"
61      ::= { wmanIf2mCmnQoSProfileEntry 1 }
62
63  wmanIf2mCmnQoSServiceClassName OBJECT-TYPE
64      SYNTAX      OCTET STRING (SIZE(2..128))

```

```

1      MAX-ACCESS  read-only
2      STATUS      current
3      DESCRIPTION
4          "This object is the Null-terminated string of ASCII
5           characters. It refers to a predefined BS service
6           configuration to be used for a service flow."
7      REFERENCE
8          "Subclause 11.13.3 in IEEE Std 802.16-2004"
9      ::= { wmanIf2mCmnQoSProfileEntry 2 }
10
11     wmanIf2mCmnQoSULGrantScheduleType OBJECT-TYPE
12         SYNTAX      WmanIf2mSchedulingType
13         MAX-ACCESS  read-only
14         STATUS      current
15         DESCRIPTION
16             "This parameter specifies the Uplink grant scheduling type
17              that shall be enabled for the associated uplink service
18              flow upstream service flow. If the parameter is not
19              present in the corresponding 802.16 QoS Parameter Set of
20              an upstream service flow, the default value is assumed."
21         REFERENCE
22             "Subclause 11.13.11 in IEEE Std 802.16e-2004"
23         DEFVAL      { bestEffort }
24         ::= { wmanIf2mCmnQoSProfileEntry 3 }
25
26     wmanIf2mCmnQoSTrafficPriority OBJECT-TYPE
27         SYNTAX      INTEGER (0..7)
28         MAX-ACCESS  read-only
29         STATUS      current
30         DESCRIPTION
31             "The value of this parameter specifies the priority assigned
32              to a service flow. For uplink service flows, the BS should
33              use this parameter when determining precedence in request
34              service and grant generation, Higher numbers indicate
35              higher priority"
36         REFERENCE
37             "Subclause 11.13.5 in IEEE Std 802.16e-2005"
38         ::= { wmanIf2mCmnQoSProfileEntry 4 }
39
40     wmanIf2mCmnQoSMaximumSustainedRate OBJECT-TYPE
41         SYNTAX      Unsigned32
42         UNITS        "bps"
43         MAX-ACCESS  read-only
44         STATUS      current
45         DESCRIPTION
46             "This parameter defines the peak information rate of the
47              service. The rate is expressed in bits per second and
48              pertains to the SDUs at the input to the Convergence
49              Sublayer."
50         REFERENCE
51             "Subclause 11.13.6 in IEEE Std 802.16e-2005"
52         ::= { wmanIf2mCmnQoSProfileEntry 5 }
53
54     wmanIf2mCmnQoSMinimumReservedRate OBJECT-TYPE
55         SYNTAX      Unsigned32
56         UNITS        "bps"
57         MAX-ACCESS  read-only
58         STATUS      current
59         DESCRIPTION
60             "This parameter specifies the minimum rate reserved for this
61              service flow. It specifies the minimum amount of data to be
62              transported on behalf of the service flow when averaged
63              over time."
64         REFERENCE

```

```

1         "Subclause 11.13.8 in IEEE Std 802.16e-2004"
2         ::= { wmanIf2mCmnQoSProfileEntry 6 }
3
4 wmanIf2mCmnQoSMaximumTrafficBurst OBJECT-TYPE
5     SYNTAX      Unsigned32
6     UNITS       "byte"
7     MAX-ACCESS  read-only
8     STATUS      current
9     DESCRIPTION
10        "This parameter defines the maximum burst size that must be
11        accommodated for the service. It defines the maximum
12        continuous burst the system should accommodate for the
13        service assuming the service is not currently using any of
14        its available resources."
15     REFERENCE
16        "Subclause 11.13.7 in IEEE Std 802.16-2004"
17     ::= { wmanIf2mCmnQoSProfileEntry 7 }
18
19 wmanIf2mCmnQoSoleratedJitter OBJECT-TYPE
20     SYNTAX      Unsigned32
21     UNITS       "millisecond"
22     MAX-ACCESS  read-only
23     STATUS      current
24     DESCRIPTION
25        "This parameter defines the Maximum delay variation (jitter)
26        for the connection."
27     REFERENCE
28        "Subclause 11.13.13 in IEEE Std 802.16-2004"
29     ::= { wmanIf2mCmnQoSProfileEntry 8 }
30
31 wmanIf2mCmnQoSMaxLatency OBJECT-TYPE
32     SYNTAX      Unsigned32
33     UNITS       "millisecond"
34     MAX-ACCESS  read-only
35     STATUS      current
36     DESCRIPTION
37        "This parameter specifies the maximum latency between the
38        ingress of a packet to the Convergence Sublayer and the
39        forwarding of the SDU to its Air Interface."
40     REFERENCE
41        "Subclause 11.13.14 in IEEE Std 802.16-2004"
42     ::= { wmanIf2mCmnQoSProfileEntry 9 }
43
44 wmanIf2mCmnQoSUnsolicitedGrantInterval OBJECT-TYPE
45     SYNTAX      Unsigned32
46     UNITS       "millisecond"
47     MAX-ACCESS  read-only
48     STATUS      current
49     DESCRIPTION
50        "This object specifies the nominal interval between
51        successive data grant opportunities for a service flow."
52     REFERENCE
53        "Subclause 11.13.20 in IEEE Std 802.16e-2004"
54     ::= { wmanIf2mCmnQoSProfileEntry 10 }
55
56 wmanIf2mCmnQoSsduSize OBJECT-TYPE
57     SYNTAX      Unsigned32
58     UNITS       "byte"
59     MAX-ACCESS  read-only
60     STATUS      current
61     DESCRIPTION
62        "This parameter specifies the length of the SDU for a
63        fixed-length SDU service flow. It is used only if packing
64        is on and the service flow is indicated as carrying

```

```

1         fixed-length SDUs. If this object is omitted in the QoS
2         parameter set, it should return 0 that means the
3         variable-length service flow."
4     REFERENCE
5         "Subclause 11.13.16 in IEEE Std 802.16-2004"
6     ::= { wmanIf2mCmnQoSProfileEntry 11 }
7
8     wmanIf2mCmnQoSUnsolicitedPollInterval OBJECT-TYPE
9         SYNTAX      Unsigned32
10        UNITS       "millisecond"
11        MAX-ACCESS  read-only
12        STATUS      current
13        DESCRIPTION
14            "This object specifies the maximal nominal interval between
15            successive polling grants opportunities for this Service
16            Flow."
17        REFERENCE
18            "Subclause 11.13.21 in IEEE Std 802.16e-2004"
19        ::= { wmanIf2mCmnQoSProfileEntry 12 }
20
21    -- XXX
22    wmanIf2mCmnArqAttributeTable OBJECT-TYPE
23        SYNTAX      SEQUENCE OF WmanIf2mCmnArqAttributeEntry
24        MAX-ACCESS  not-accessible
25        STATUS      current
26        DESCRIPTION
27            "This table contains ARQ parameters that are associated
28            with the Service Flows."
29        ::= { wmanIf2mCmnCm 4 }
30
31    wmanIf2mCmnArqAttributeEntry OBJECT-TYPE
32        SYNTAX      WmanIf2mCmnArqAttributeEntry
33        MAX-ACCESS  not-accessible
34        STATUS      current
35        DESCRIPTION
36            "This table provides one row for each created service flow
37            for a given MacAddress, and is indexed by ifIndex, and
38            wmanIf2mCmnArqIndex. IfIndex is associated with the BS
39            sector."
40        INDEX      { ifIndex, wmanIf2mCmnArqIndex }
41        ::= { wmanIf2mCmnArqAttributeTable 1 }
42
43    WmanIf2mCmnArqAttributeEntry ::= SEQUENCE {
44        wmanIf2mCmnArqIndex          INTEGER,
45        wmanIf2mCmnArqEnable        TruthValue,
46        wmanIf2mCmnArqWindowSize    INTEGER,
47        wmanIf2mCmnArqBlockLifetime INTEGER,
48        wmanIf2mCmnArqSyncLossTimeout INTEGER,
49        wmanIf2mCmnArqDeliverInOrder TruthValue,
50        wmanIf2mCmnArqRxPurgeTimeout INTEGER,
51        wmanIf2mCmnArqBlockSize     INTEGER,
52        wmanIf2mCmnArqAckProcessingTime INTEGER}
53
54    wmanIf2mCmnArqIndex OBJECT-TYPE
55        SYNTAX      INTEGER ( 1 .. 65535)
56        MAX-ACCESS  not-accessible
57        STATUS      current
58        DESCRIPTION
59            "The index value which uniquely identifies an entry in the
60            in the wmanIf2mCmnArqAttributeTable."
61        ::= { wmanIf2mCmnArqAttributeEntry 1 }
62
63    wmanIf2mCmnArqEnable OBJECT-TYPE
64        SYNTAX      TruthValue

```



```

1         MAX-ACCESS read-only
2         STATUS current
3         DESCRIPTION
4             "True(1) ARQ enabling is requested for the connection."
5         ::= { wmanIf2mCmnArqAttributeEntry 2 }
6
7     wmanIf2mCmnArqWindowSize OBJECT-TYPE
8         SYNTAX INTEGER (1..1024)
9         MAX-ACCESS read-only
10        STATUS current
11        DESCRIPTION
12            "Indicates the maximum number of unacknowledged fragments
13             at any time."
14        ::= { wmanIf2mCmnArqAttributeEntry 3 }
15
16    wmanIf2mCmnArqBlockLifetime OBJECT-TYPE
17        SYNTAX INTEGER (0 .. 65535)
18        UNITS "10 us"
19        MAX-ACCESS read-only
20        STATUS current
21        DESCRIPTION
22            "The maximum time interval an ARQ fragment will be managed
23             by the transmitter ARQ machine, once initial transmission
24             of the fragment has occurred. If transmission or
25             retransmission of the fragment is not acknowledged by the
26             receiver before the time limit is reached, the fragment is
27             discarded. A value of 0 means Infinite."
28        ::= { wmanIf2mCmnArqAttributeEntry 4 }
29
30    wmanIf2mCmnArqSyncLossTimeout OBJECT-TYPE
31        SYNTAX INTEGER (0 .. 65535 )
32        UNITS "10 us"
33        MAX-ACCESS read-only
34        STATUS current
35        DESCRIPTION
36            "The maximum interval before declaring a loss of
37             synchronization of the sender and receiver state machines.
38             A value of 0 means Infinite."
39        ::= { wmanIf2mCmnArqAttributeEntry 5 }
40
41    wmanIf2mCmnArqDeliverInOrder OBJECT-TYPE
42        SYNTAX TruthValue
43        MAX-ACCESS read-only
44        STATUS current
45        DESCRIPTION
46            "Indicates whether or not data is to be delivered by the
47             receiving MAC to its client application in the order in
48             which data was handed off to the originating MAC."
49        ::= { wmanIf2mCmnArqAttributeEntry 6 }
50
51    wmanIf2mCmnArqRxPurgeTimeout OBJECT-TYPE
52        SYNTAX INTEGER (0 .. 65535)
53        UNITS "10 us"
54        MAX-ACCESS read-only
55        STATUS current
56        DESCRIPTION
57            "Indicates the time interval the ARQ window is advanced
58             after a fragment is received. A value of 0 means
59             Infinite."
60        ::= { wmanIf2mCmnArqAttributeEntry 7 }
61
62    wmanIf2mCmnArqBlockSize OBJECT-TYPE
63        SYNTAX INTEGER (1..2040)
64        UNITS "byte"

```

```
1          MAX-ACCESS read-only
2          STATUS current
3          DESCRIPTION
4              "This value of this parameter specifies the size of an ARQ
5              block. This parameter shall be established by negotiation
6              during the connection creation dialog."
7          REFERENCE
8              "Subclause 11.13.18.8 in IEEE Std 802.16-2004"
9          ::= { wmanIf2mCmnArqAttributeEntry 8 }
10
11 wmanIf2mCmnArqAckProcessingTime OBJECT-TYPE
12     SYNTAX INTEGER (0 .. 255)
13     UNITS "millisecond"
14     MAX-ACCESS read-only
15     STATUS current
16     DESCRIPTION
17         "This parameter indicates the number of ms required by the
18         ARQ receiver to process the received ARQ blocks and provide
19         a valid ACK or NAK."
20     REFERENCE
21         "Subclause 11.13.18.9 in IEEE Std 802.16e-2005"
22     ::= { wmanIf2mCmnArqAttributeEntry 9 }
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
```

