

Project	<b>IEEE 802.16 Broadband Wireless Access Working Group</b> < <a href="http://ieee802.org/16">http://ieee802.org/16</a> >
Title	<b>Addition of Classifier Action Rule</b>
Date Submitted	<b>2007-05-09</b>
Source(s)	Richard van Leeuwen <a href="mailto:rvanleeuwen@motorola.com">rvanleeuwen@motorola.com</a> Motorola
Re:	IEEE P802.16i/D2
Abstract	This contribution proposes the addition of the Classifier Action Rule classification parameter.
Purpose	Adopt proposed fix.
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	The contributor is familiar with the IEEE 802.16 Patent Policy and Procedures < <a href="http://ieee802.org/16/ipr/patents/policy.html">http://ieee802.org/16/ipr/patents/policy.html</a> >, including the statement "IEEE standards may include the known use of patent(s), including patent applications, provided the IEEE receives assurance from the patent holder or applicant with respect to patents essential for compliance with both mandatory and optional portions of 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 < <a href="mailto:chair@wirelessman.org">mailto:chair@wirelessman.org</a> > as early as possible, in written or electronic form, if patented technology (or technology under patent application) might be incorporated into a draft standard being developed within the IEEE 802.16 Working Group. The Chair will disclose this notification via the IEEE 802.16 web site < <a href="http://ieee802.org/16/ipr/patents/notices">http://ieee802.org/16/ipr/patents/notices</a> >.

## Addition of Classifier Action Rule

*Richard van Leeuwen*

*Motorola*

### Problem

IEEE 802.16e-2005 amendment adds the following classification parameters:

11.13.19.3.4.16 Large Context ID for ROHC- or ECRTTP-compressed packet or ROHC  
feedback packet

11.13.19.3.4.17 Classifier Action Rule

11.13.19.3.4.18 Short-format Context ID for ROHC- or ECRTTP-compressed packet or ROHC  
feedback packet

There are currently no objects defined for these new classification parameters in IEEE P802.16i/D2.

Note that Corrigendum2 removes subclause 11.13.19.3.4.16 and 11.13.19.3.4.18. It doesn't seem very useful to add objects for these parameters.

### Remedy

In IEEE P802.16i/D2 in 13.2.2, modify the text as follows:

```
WmanIf2ClassifierBitMap ::= TEXTUAL-CONVENTION
    STATUS         current
    DESCRIPTION
        "A bit of of this object is set to 1 if the parameter
        indicated by the comment was present in the classifier
        encoding, and 0 otherwise.
        Note: that BITS are encoded most significant bit first,
        so that if e.g. bits 6 and 7 are set, this object is
        encoded as the octet string '030000'H."
    REFERENCE
        "Subclause 11.13.19.3.4 in IEEE Std 802.16-2004e-2005"
    SYNTAX         BITS {priority(0),
                        ipTos(1),
                        ipProtocol(2),
                        ipMaskedSrcAddr(3),
                        ipMaskedDestAddr(4),
                        srcPort(5),
                        destPort(6),
                        destMacAddr(7),
                        srcMacAddr(8),
                        ethernetProtocol(9),
                        userPriority(10),
                        vlanId(11),
                        ipv6FlowLabel(12),
                        actionRule(13)}
```

WmanIf2ActionRule ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"The value of this field specifies an action associated with the classifier rule

Bit #0:

0 = none.

1 = Discard packet"

REFERENCE

"Subclause 11.13.19.3.4.17 in IEEE Std 802.16e-2005"

SYNTAX BITS {discardPacket(0)}

...

```

WmanIf2BsClassifierRuleEntry ::= SEQUENCE {
    wmanIf2BsClassifierRuleIndex           Unsigned32,
    wmanIf2BsClassifierRulePriority        INTEGER,
    wmanIf2BsClassifierRuleIpTosLow       INTEGER,
    wmanIf2BsClassifierRuleIpTosHigh      INTEGER,
    wmanIf2BsClassifierRuleIpTosMask      INTEGER,
    wmanIf2BsClassifierRuleIpProtocol     Integer32,
    wmanIf2BsClassifierRuleIpSourceAddr   InetAddress,
    wmanIf2BsClassifierRuleIpSourceMask   InetAddress,
    wmanIf2BsClassifierRuleIpDestAddr     InetAddress,
    wmanIf2BsClassifierRuleIpDestMask     InetAddress,
    wmanIf2BsClassifierRuleSourcePortStart Integer32,
    wmanIf2BsClassifierRuleSourcePortEnd  Integer32,
    wmanIf2BsClassifierRuleDestPortStart  Integer32,
    wmanIf2BsClassifierRuleDestPortEnd    Integer32,
    wmanIf2BsClassifierRuleDestMacAddr    MacAddress,
    wmanIf2BsClassifierRuleDestMacMask    MacAddress,
    wmanIf2BsClassifierRuleSourceMacAddr  MacAddress,
    wmanIf2BsClassifierRuleSourceMacMask  MacAddress,
    wmanIf2BsClassifierRuleEnetProtocolType INTEGER,
    wmanIf2BsClassifierRuleEnetProtocol   Integer32,
    wmanIf2BsClassifierRuleUserPriLow     Integer32,
    wmanIf2BsClassifierRuleUserPriHigh    Integer32,
    wmanIf2BsClassifierRuleVlanId         Integer32,
    wmanIf2BsClassifierRulePhsSize        Integer32,
    wmanIf2BsClassifierRulePhsMask        OCTET STRING,
    wmanIf2BsClassifierRulePhsVerify      WmanIf2PhsRuleVerify,
    wmanIf2BsClassifierRuleIpv6FlowLabel  WmanIf2Ipv6FlowLabel,
    wmanIf2BsClassifierRuleActionRule    WmanIf2ActionRule,
    wmanIf2BsClassifierRuleBitMap         WmanIf2ClassifierBitMap,
    wmanIf2BsClassifierRuleRowStatus      RowStatus}

```

...

wmanIf2BsClassifierRuleIpv6FlowLabel OBJECT-TYPE

SYNTAX WmanIf2Ipv6FlowLabel

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The value of this field specifies the matching values for

```

        the IPv6 Flow label field."
 ::= { wmanIf2BsClassifierRuleEntry 28 }
wmanIf2BsClassifierRuleActionRule OBJECT-TYPE
SYNTAX      WmanIf2ActionRule
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    "The value of this field specifies an action associated with
    the classifier rule.
    If this classification action rule exists, its action shall
    be applied on the packets that match this classifier rule."
 ::= { wmanIf2BsClassifierRuleEntry 29 }

```

```

wmanIf2BsClassifierRuleBitMap OBJECT-TYPE
SYNTAX      WmanIf2ClassifierBitMap
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    "This object indicates which parameter encodings were
    actually present in the entry. A bit set to '1' indicates
    the corresponding classifier encoding is present, and '0'
    means otherwise"
 ::= { wmanIf2BsClassifierRuleEntry 2930 }

```

```

wmanIf2BsClassifierRuleRowStatus OBJECT-TYPE
SYNTAX      RowStatus
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    "This object is used to create a new row or modify or
    delete an existing row in this table.

    If the implementator of this MIB has chosen not
    to implement 'dynamic assignment' of profiles, this
    object is not useful and should return noSuchName
    upon SNMP request."
 ::= { wmanIf2BsClassifierRuleEntry 3031 }

```

...

```

WmanIf2CmnClassifierRuleEntry ::= SEQUENCE {
    wmanIf2CmnClassifierRuleIndex      Unsigned32,
    wmanIf2CmnClassifierRulePriority    INTEGER,
    wmanIf2CmnClassifierRuleIpTosLow   INTEGER,
    wmanIf2CmnClassifierRuleIpTosHigh  INTEGER,
    wmanIf2CmnClassifierRuleIpTosMask  INTEGER,
    wmanIf2CmnClassifierRuleIpProtocol Integer32,
    wmanIf2CmnClassifierRuleIpSourceAddr InetAddress,
    wmanIf2CmnClassifierRuleIpSourceMask InetAddress,
    wmanIf2CmnClassifierRuleIpDestAddr  InetAddress,
    wmanIf2CmnClassifierRuleIpDestMask  InetAddress,
    wmanIf2CmnClassifierRuleSourcePortStart Integer32,
    wmanIf2CmnClassifierRuleSourcePortEnd Integer32,

```

wmanIf2CmnClassifierRuleDestPortStart	Integer32,
wmanIf2CmnClassifierRuleDestPortEnd	Integer32,
wmanIf2CmnClassifierRuleDestMacAddr	MacAddress,
wmanIf2CmnClassifierRuleDestMacMask	MacAddress,
wmanIf2CmnClassifierRuleSourceMacAddr	MacAddress,
wmanIf2CmnClassifierRuleSourceMacMask	MacAddress,
wmanIf2CmnClassifierRuleEnetProtocolTyp	INTEGER,
wmanIf2CmnClassifierRuleEnetProtocol	Integer32,
wmanIf2CmnClassifierRuleUserPriLow	Integer32,
wmanIf2CmnClassifierRuleUserPriHigh	Integer32,
wmanIf2CmnClassifierRuleVlanId	Integer32,
wmanIf2CmnClassifierRulePkts	Counter64,
wmanIf2CmnClassifierRuleIpv6FlowLabel	WmanIf2Ipv6FlowLabel,
<u>wmanIf2CmnClassifierRuleActionRule</u>	<u>WmanIf2ActionRule,</u>
wmanIf2CmnClassifierRuleBitMap	WmanIf2ClassifierBitMap}

...

wmanIf2CmnClassifierRuleIpv6FlowLabel OBJECT-TYPE

SYNTAX WmanIf2Ipv6FlowLabel

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this field specifies the matching values for the IPv6 Flow label field."

::= { wmanIf2CmnClassifierRuleEntry 26 }

wmanIf2CmnClassifierRuleActionRule OBJECT-TYPE

SYNTAX WmanIf2ActionRule

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this field specifies an action associated with the classifier rule.

If this classification action rule exists, its action shall be applied on the packets that match this classifier rule."

::= { wmanIf2CmnClassifierRuleEntry 27 }

wmanIf2CmnClassifierRuleBitMap OBJECT-TYPE

SYNTAX WmanIf2ClassifierBitMap

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object indicates which parameter encodings were actually present in the entry. A bit set to '1' indicates the corresponding classifier encoding is present, and '0' means otherwise"

::= { wmanIf2CmnClassifierRuleEntry 27~~8~~ }

...

wmanIf2MibCommonGroup OBJECT-GROUP

```

OBJECTS {-- Classification
wmanIf2CmnClassifierRulePriority,
wmanIf2CmnClassifierRuleIpTosLow,
wmanIf2CmnClassifierRuleIpTosHigh,
wmanIf2CmnClassifierRuleIpTosMask,
wmanIf2CmnClassifierRuleIpProtocol,
wmanIf2CmnClassifierRuleIpSourceAddr,
wmanIf2CmnClassifierRuleIpSourceMask,
wmanIf2CmnClassifierRuleIpDestAddr,
wmanIf2CmnClassifierRuleIpDestMask,
wmanIf2CmnClassifierRuleSourcePortStart,
wmanIf2CmnClassifierRuleSourcePortEnd,
wmanIf2CmnClassifierRuleDestPortStart,
wmanIf2CmnClassifierRuleDestPortEnd,
wmanIf2CmnClassifierRuleDestMacAddr,
wmanIf2CmnClassifierRuleDestMacMask,
wmanIf2CmnClassifierRuleSourceMacAddr,
wmanIf2CmnClassifierRuleSourceMacMask,
wmanIf2CmnClassifierRuleEnetProtocolTyp,
wmanIf2CmnClassifierRuleEnetProtocol,
wmanIf2CmnClassifierRuleUserPriLow,
wmanIf2CmnClassifierRuleUserPriHigh,
wmanIf2CmnClassifierRuleVlanId,
wmanIf2CmnClassifierRulePkts,
wmanIf2CmnClassifierRuleIpv6FlowLabel,
wmanIf2CmnClassifierRuleActionRule,
wmanIf2CmnClassifierRuleBitMap,

-- Configuration parameters
wmanIf2CmnCpsTargetSaid,
wmanIf2CmnInvitedRangRetries,
wmanIf2CmnDSxReqRetries,
wmanIf2CmnDSxRespRetries,
wmanIf2CmnT7Timeout,
wmanIf2CmnT8Timeout,
wmanIf2CmnT10Timeout,
wmanIf2CmnT22Timeout}

STATUS      current
DESCRIPTION
    "This group contains objects for both BS and SS,
    and are independent of PHY."
 ::= { wmanIf2MibGroups 1 }

```

...

```

wmanIf2MibBsGroup      OBJECT-GROUP
OBJECTS {-- Service classes
wmanIf2BsSfDirection,
wmanIf2BsServiceClassIndex,
wmanIf2BsSfState,
wmanIf2BsSfProvisionedTime,
wmanIf2BsProvisionedSfRowStatus,
wmanIf2BsSsProvisionedForSfRowStatus,
wmanIf2BsSfCsSpecification,

```

```
wmanIf2BsQoSServiceClassName,  
wmanIf2BsQoSSTrafficPriority,  
wmanIf2BsQoSMaxSustainedRate,  
wmanIf2BsQoSMaxTrafficBurst,  
wmanIf2BsQoSMinReservedRate,  
wmanIf2BsQOSToleratedJitter,  
wmanIf2BsQoSMaxLatency,  
wmanIf2BsQoSFixedVsVariableSduInd,  
wmanIf2BsQoSsduSize,  
wmanIf2BsQoSschedulingType,  
wmanIf2BsQoSsArqEnable,  
wmanIf2BsQoSsArqWindowSize,  
wmanIf2BsQoSsArqBlockLifetime,  
wmanIf2BsQoSsArqSyncLossTimeout,  
wmanIf2BsQoSsArqDeliverInOrder,  
wmanIf2BsQoSsArqRxPurgeTimeout,  
wmanIf2BsQoSsArqBlockSize,  
wmanIf2BsQoSsCMinRsvdTolerableRate,  
wmanIf2BsQoSsReqTxPolicy,  
wmanIf2BsQoSServiceClassRowStatus,  
  
-- Classification  
wmanIf2BsClassifierRulePriority,  
wmanIf2BsClassifierRuleIpTosLow,  
wmanIf2BsClassifierRuleIpTosHigh,  
wmanIf2BsClassifierRuleIpTosMask,  
wmanIf2BsClassifierRuleIpProtocol,  
wmanIf2BsClassifierRuleIpSourceAddr,  
wmanIf2BsClassifierRuleIpSourceMask,  
wmanIf2BsClassifierRuleIpDestAddr,  
wmanIf2BsClassifierRuleIpDestMask,  
wmanIf2BsClassifierRuleSourcePortStart,  
wmanIf2BsClassifierRuleSourcePortEnd,  
wmanIf2BsClassifierRuleDestPortStart,  
wmanIf2BsClassifierRuleDestPortEnd,  
wmanIf2BsClassifierRuleDestMacAddr,  
wmanIf2BsClassifierRuleDestMacMask,  
wmanIf2BsClassifierRuleSourceMacAddr,  
wmanIf2BsClassifierRuleSourceMacMask,  
wmanIf2BsClassifierRuleEnetProtocolType,  
wmanIf2BsClassifierRuleEnetProtocol,  
wmanIf2BsClassifierRuleUserPriLow,  
wmanIf2BsClassifierRuleUserPriHigh,  
wmanIf2BsClassifierRuleVlanId,  
wmanIf2BsClassifierRulePhsSize,  
wmanIf2BsClassifierRulePhsMask,  
wmanIf2BsClassifierRulePhsVerify,  
wmanIf2BsClassifierRuleIpv6FlowLabel,  
wmanIf2BsClassifierRuleActionRule,  
wmanIf2BsClassifierRuleBitMap,  
wmanIf2BsClassifierRuleRowStatus,
```

...