Cable Wi-Fi
Wi-Fi Requirements for Cable Modem Gateways
Important notice

This technical specification is the result of a cooperative effort undertaken at the direction of Cable Europe Labs for the benefit of the cable industry and its customers. This document may contain references to other documents not owned or controlled by Cable Europe Labs. Use and understanding of this document may require access to such other documents. Designing, manufacturing, distributing, using, selling, or servicing products, or providing services, based on this document may require intellectual property licenses from third parties for technology referenced in this document.

Neither Cable Europe Labs nor any member company is responsible to any party for any liability of any nature whatsoever resulting from or arising out of use or reliance upon this document, or any document referenced herein. This document is furnished on an "AS IS" basis and neither Cable Europe Labs nor its members provides any representation or warranty, express or implied, regarding the accuracy, completeness, noninfringement, or fitness for a particular purpose of this document, or any document referenced herein.

Individual copies of the present document can be downloaded from:

http://www.cable-europe.eu/labs

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF) as maintained by the secretariat of Cable Europe Labs.

Users of the present document should be aware that the document may be subject to revision or change of status. If you find errors in the present document, please send your comment to the following services:
labs@cable-europe.eu

Copyright Notification

No part may be reproduced except as authorized by written permission. The copyright and the foregoing restriction extend to reproduction in all media.

© Cable Europe Labs 2013. All rights reserved.
Contents
1 Scope ................................................................................................................................. 5
  1.1 Introduction and Purpose .............................................................................................. 5
  1.2 Requirements ................................................................................................................ 5
2 List of References ............................................................................................................. 6
3 Terms and Definitions ...................................................................................................... 7
4 Abbreviations and Acronyms .......................................................................................... 7
5 Overview ............................................................................................................................ 8
6 Requirements .................................................................................................................... 9
  6.1 802.11 Air Interface Requirements .............................................................................. 9
    6.1.1 Requirements for 802.11n Wi-Fi Gateways for use with EuroDOCSIS 3.0 CMs ........ 9
    6.1.2 Interoperability ....................................................................................................... 9
    6.1.3 Channel Selection .................................................................................................... 9
    6.1.4 Antenna Requirements .......................................................................................... 9
    6.1.5 Transmit Power, Range, Receiver Sensitivity ......................................................... 9
    6.1.6 Air interface performance ...................................................................................... 9
    6.1.7 Other Requirements ............................................................................................. 9
    6.1.8 Configurations of SSIDs ...................................................................................... 9
    6.1.9 Security Requirements ........................................................................................ 10
    6.1.10 Other Requirements .......................................................................................... 10
  6.2 Traffic Classification, Forwarding and Encapsulation .................................................. 10
    6.2.1 eRouter Interface ................................................................................................... 10
    6.2.2 Upstream Traffic Forwarding ................................................................................. 10
      6.2.2.1 Classifiers ....................................................................................................... 11
      6.2.2.2 Policies ......................................................................................................... 11
      6.2.2.2.1 GRE Tunnel Endpoint .............................................................................. 12
      6.2.2.2.2 ToS Set ..................................................................................................... 12
    6.2.3 Downstream Traffic Forwarding ............................................................................ 12
    6.2.4 Broadcast/Multicast Traffic Forwarding ............................................................... 12
      6.2.4.1 Traffic from Client Wi-Fi Device ..................................................................... 12
      6.2.4.2 Traffic to Client Wi-Fi Device (GRE Encapsulated) ....................................... 13
      6.2.4.3 Traffic to Client Wi-Fi Device (Bridged) ......................................................... 13
    6.2.5 IP Fragmentation .................................................................................................... 13
    6.2.6 Redundancy .......................................................................................................... 13
      6.2.6.1 FQDN ............................................................................................................ 13
      6.2.6.2 Failover ......................................................................................................... 13
      6.2.7 DHCP Intercept Parameters ............................................................................... 14
    6.3 Resources and Traffic Priority ................................................................................... 14
    6.4 RADIUS Client Interface .......................................................................................... 15
    6.5 Management Interface Requirements ...................................................................... 16
      6.5.1 Status and Performance Reports ........................................................................ 16
    6.6 Configured Admission Control ................................................................................ 16
    6.7 MIB Architecture ...................................................................................................... 16
      6.7.1 Base Objects ...................................................................................................... 16
      6.7.2 802.11 Objects .................................................................................................. 17
      6.7.3 Forwarding and Classification Objects ............................................................... 19
7 Annex A: MIB Definitions ................................................................................................ 20
  7.1 Base Objects .............................................................................................................. 20
  7.2 802.11 Objects .......................................................................................................... 33
  7.3 Forwarding and Classification Objects ......................................................................... 77
**List of figures**

Figure 1: Wi-Fi gateway interfaces........................................................................................................ 8
Figure 2: Wi-Fi gateway upstream traffic classification and forwarding mechanism .............. 11
Figure 3: MIB Base Objects Diagram.................................................................................................. 17
Figure 4: MIB 802.11 Objects Diagram (a) ....................................................................................... 18
Figure 5: MIB 802.11 Objects Diagram (b) ....................................................................................... 18
Figure 6: MIB 802.11 Objects Diagram (c)....................................................................................... 19
Figure 7: MIB Forwarding and Classification Objects Diagram...................................................... 19

**List of tables**

Table 1: DSCP Value to WMM Priorities Mapping ................................................................................. 15
1 Scope

1.1 Introduction and Purpose

In the [CEL-TR-WIFI] document, use cases and an end-to-end architecture are described to support adding new (wireless) services, using existing Wi-Fi gateways in the cable operator’s network.

This document lists the requirements for the Wi-Fi gateway, necessary to support deploying the use cases described in [CEL-TR-WIFI], additional to these mentioned in the [WR-SP-WiFi-GW] document.

Furthermore, this document will clarify and interpret, where necessary, the requirements from the [WR-SP-WiFi-GW] document.

1.2 Requirements

Throughout this document, the words that are used to define the significance of particular requirements are capitalized. These words are:

"MUST" This word means that the item is an absolute requirement of this specification.

"MUST NOT" This phrase means that the item is an absolute prohibition of this specification.

"SHOULD" This word means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications should be understood and the case carefully weighed before choosing a different course. As applied to this specification, requirements signified by "SHOULD" are expected to be mandatory for Wi-Fi Gateways (Wi-Fi GWs) deployed in public settings and in enterprises. On the other hand, particular circumstances driven by lower end residential deployments may prevent the full implementation of requirements signified by "SHOULD" for residential Wi-Fi GWs.

"SHOULD NOT" This phrase means that there may exist valid reasons in particular circumstances when the listed behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.

"MAY" This word means that this item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because it enhances the product, for example; another vendor may omit the same item.
2 List of References

[WR-SP-WiFi-GW] Cable Television Laboratories Inc.: Wi-Fi Requirements for Cable Modem Gateways
[WMM] Wi-Fi Alliance: Wi-Fi Multi-Media QoS based on 802.11e, Version 1.2.0.
[RFC 5580] IETF: Carrying Location Objects in RADIUS and Diameter
[RFC 2866] IETF: RADIUS Accounting

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For a specific reference, subsequent revisions do not apply. For a non-specific reference, the latest version applies.
3 Terms and Definitions

No change required

4 Abbreviations and Acronyms

No change required
5 Overview

The Wi-Fi gateway considered in this specification integrates a [MULPI3.0] modem with an [802.11n] air interface as illustrated in Figure 1. Other functional elements may be integrated with the cable modem as well, but are not illustrated or addressed in this document. The Wi-Fi gateway requirements support residential, enterprise and public deployments. Requirements are placed on the air interface in order to support the uses cases described in [CEL-TR-WIFI]. The cable modem interface to the CMTS is defined in the [MULPI3.0] specifications. Functional requirements are placed on the Wi-Fi gateway management interface.

A Remote Authentication Dial In User Service (RADIUS) client interface is specified to support Authentication, Authorization and Accounting (AAA) functions.

The Wi-Fi gateway requirements apply to cable modem router CPEs. For example, the Wi-Fi requirements can be added to a number of standardized networking functions, such as those defined in [eRouter], or network functions defined in operator specifications.

Figure 1: Wi-Fi gateway interfaces
6 Requirements

This section contains normative requirements on the Wi-Fi gateway air interface. These requirements encourage multi-vendor interoperability on the Wi-Fi air interface. The Wi-Fi gateway MUST NOT use technologies that place non-standard or proprietary requirements on the Wi-Fi subscriber devices.

The requirements apply to Wi-Fi gateways that support [MULPI3.0] cable modems. Please see [MULPI3.0] specifications for cable modem requirements.

6.1 802.11 Air Interface Requirements

No change required.

6.1.1 Requirements for 802.11n Wi-Fi Gateways for use with EuroDOCSIS 3.0 CMs

No change required.

6.1.2 Interoperability

Since the Wi-Fi gateway is required to support multiple SSIDs, the interoperability requirement is extended to the SSID level. The 802.11n capable Wi-Fi gateway MUST support tri-mode (.11b/g/n), dual-mode (e.g., .11g/n) and single mode (e.g., .11n only) operations per SSID as defined in [802.11n].

The Wi-Fi gateway SHOULD support disassociating client Wi-Fi devices as soon as their SNR drops below a predefined threshold.

The Wi-Fi gateway SHOULD support denying connection to a client Wi-Fi device which tries to connect with an SNR below a predefined threshold.

6.1.3 Channel Selection

The Wi-Fi gateway SHOULD allow the subscriber to select between Manual and Auto channel selection mode through the local configuration web page.

6.1.4 Antenna Requirements

No change required.

6.1.5 Transmit Power, Range, Receiver Sensitivity

No change required.

6.1.6 Air interface performance

No change required.

6.1.7 Other Requirements

No change required.

6.1.8 Configurations of SSIDs

The Wi-Fi gateway MUST support independent remote configuration of parameters associated with each SSID. The Wi-Fi gateway MUST support independently configurable parameters on a per SSID basis that includes but is not limited to: the SSID name, security type, bridge mode enable/disable, NAT enable/disable, authentication, encryption and broadcast/multicast behaviour (cfr. section 6.2). This requirement is primarily targeted for the three operator-configured SSIDs.
The Wi-Fi gateway MUST support adding/removing a configuration setting to the local web page to let the subscriber shut down the radio interface, and thus disabling all SSIDs, including the operator-configured ones.

The Wi-Fi gateway MUST provide the ability to read out the state of the radio interface (turned on/shut down) by the operator (cfr. section 6.7).

The Wi-Fi gateway MUST support configuring a maximum number of connected client Wi-Fi devices on a per SSID base. When a client Wi-Fi device tries to associate on an SSID where the limit reached, the Wi-Fi gateway MUST deny the association.

The Wi-Fi gateway does not have to support the VLAN requirements found in the last paragraphs of [WR-SP-WiFi-GW], section 6.1.8. The requirements, related to VLANs are discussed in section 6.2 of this document.

### 6.1.9 Security Requirements

The Wi-Fi gateway does not have to support the VLAN requirements found in [WR-SP-WiFi-GW], section 6.1.9. The requirements, related to VLANs are discussed in section 6.2 of this document.

The Wi-Fi gateway MUST block all non-authentication traffic prior to successful completion of the authentication phase on a secured SSID.

The Wi-Fi gateway does not have to block any traffic on a clear SSID.

### 6.1.10 Other Requirements

The Wi-Fi gateway does not have to support redirection upon successful authentication.

### 6.2 Traffic Classification, Forwarding and Encapsulation

#### 6.2.1 eRouter Interface

The eRouter must support IP address provisioning for its interface through DHCP. Furthermore, the eRouter must support adding extra options to these DHCP messages, as defined in the MIB tables, described in 6.7.

#### 6.2.2 Upstream Traffic Forwarding

Figure 2 shows the reference architecture for the Wi-Fi gateway with respect to the traffic classification and encapsulation (for traffic in the upstream direction). Traffic is classified and the matched classifier links to a policy, which defines the way the data packets are encapsulated before they are send upstream by the CM.

REMARK: RADIUS traffic to the AAA server to authenticate the client Wi-Fi devices will not go through this traffic classification and encapsulation mechanism.
6.2.2.1 Classifiers

The eRouter MUST support packet classification, based on the following parameters of the incoming data traffic:

- Interface (each SSID has a different interface id)
- Source and/or destination MAC address (with an optional mask)
- Source and/or destination IP address (with an optional mask)
- IP protocol
- Source and/or destination UDP/TCP port number

REMARK: If ethernet ports are available on the eRouter, it SHOULD be possible to also use these as interface in a classifier.

The eRouter MUST support classifier priorities.

6.2.2.2 Policies

The eRouter MUST support policies that define how packets are encapsulated/ altered by the eRouter before forwarding them to the eCM. A policy is defined by following items, which are described in detail in the sections below:

- GRE Tunnel Endpoint
- 802.1Q
- ToS Set

There are 2 policies with a special purpose:

- The default policy, which is the policy that is applied for packets that don’t match any classifier. This is the first policy defined.
- The drop policy, which drops the packets that match a classifier, attached to it. This policy MUST be statically installed on the eRouter and MUST NOT be modifiable.

If no default policy is defined, the drop policy is the default policy.
6.2.2.2.1 GRE Tunnel Endpoint

The eRouter MUST support L2oGRE encapsulation of incoming packets (upstream). In case no tunnel endpoint is defined, the packet is forwarded without GRE encapsulation. With respect to L2oGRE encapsulation, the Wi-Fi gateway MUST support all mandatory features and SHOULD support all optional features described in [WR-SP-WiFi-GW], section 6.2.802.1Q.

The eRouter MUST support adding a 32-bit VLAN tag to the incoming packets (upstream). In case a GRE tunnel endpoint is defined (cfr. section 6.2.2.2.1), the VLAN tag is added before the packet is GRE encapsulated. The policy MUST support configuration of the VLAN id and priority.

The eRouter MUST silently drop incoming packets that are already VLAN tagged.

6.2.2.2.2 ToS Set

The eRouter MUST support overriding the ToS bits in the incoming packets. If no ToS bits are defined, the ToS bits override will default to all zeros. In case a GRE tunnel endpoint is defined (cfr. section 6.2.2.2.1), the ToS bits override is performed after the packet is GRE encapsulated and thus in the outer IP header of the GRE packet.

6.2.3 Downstream Traffic Forwarding

The eRouter MUST support downstream forwarding as listed below:

- If the destination MAC address (of the outer header in case the packet is GRE encapsulated) equals the eRouter's MAC address and the IP protocol (also in the outer header) equals 47 (GRE), the eRouter MUST decapsulate GRE.
- If the protocol type in the GRE header equals 0x6558 (Transparent LAN Bridging), the eRouter MUST process the 802.3 frame following the GRE header.
- If the protocol type in the GRE header does not equal 0x6558 (Transparent LAN Bridging), the eRouter MUST silently drop the frame.
- If the destination MAC address equals the eRouter's MAC address and the IP protocol does not equal 47 (GRE), the eRouter MUST check it's NAT table if there is a valid NAT mapping available for this packet. In this case, the eRouter MUST map the packet and forward it onto the matching CPE interface. If there is no valid NAT mapping available, the eRouter MUST silently drop the frame.
- The eRouter MUST use the destination MAC address (of the incoming 802.3 frame or the decapsulated 802.3 frame in case the incoming packet was GRE encapsulated) to locate the SSID.
- The eRouter MUST strip off any VLAN tags (of the incoming 802.3 frame or the decapsulated 802.3 frame in case the incoming packet was GRE encapsulated) before forwarding it to the SSID.

6.2.4 Broadcast/Multicast Traffic Forwarding

Broadcast/multicast block behaviours (upstream/downstream) must be configurable per SSID.

The eRouter MUST support preventing user-to-user switching of frames within the same SSID or across SSID.

6.2.4.1 Traffic from Client Wi-Fi Device

The eRouter MUST only forward broadcast/multicast traffic from client Wi-Fi devices in the upstream direction (interface to eCM).
6.2.4.2 Traffic to Client Wi-Fi Device (GRE Encapsulated)

Broadcast/multicast traffic, within the GRE-tunnel that is, destined for client Wi-Fi devices, MUST be blocked at the eRouter.

There is one exception from this rule and that is for DHCP Offer/Ack messages as these can be broadcast on the MAC layer and are needed for a successful DHCP sequence. In this case the eRouter MUST forward the DHCP Offer/Ack messages as listed below:

- The broadcast destination MAC address MUST be converted to unicast, by overriding it with the value from the *Client MAC address* field which is present in these messages.
- The broadcast destination IP address MUST not be altered.

6.2.4.3 Traffic to Client Wi-Fi Device (Bridged)

The eRouter MUST NOT block broadcast/multicast traffic associated with a policy that does not include a tunnel endpoint.

6.2.5 IP Fragmentation

If IP fragmentation is needed in the upstream direction, the eRouter MUST fragment the packets after GRE encapsulation (in case the policy defines a GRE tunnel endpoint).

The eRouter MUST support reassembly of fragmented GRE encapsulated packets in the downstream direction with a minimum of 2 fragments per configured GRE tunnel endpoint.

The eRouter MUST rewrite MSS option, like described in [WR-SP-WiFi-GW], section 6.2.1.

The Wi-Fi gateway must support disabling IP fragmentation. In case IP fragmentation is disabled, upstream packets that exceed the MTU of the EuroDOCSIS interface of the Wi-Fi gateway, will be dropped by the Wi-Fi gateway.

6.2.6 Redundancy

6.2.6.1 FQDN

Provisioning of a GRE tunnel endpoint (as part of the policy definition) can be through either an IP address or an FQDN. In case an FQDN is used, the eRouter MUST query the DNS for the provisioned FQDN.

If the eRouter has no valid endpoint (e.g. DNS-server not available), it MUST mark the GRE tunnel endpoint as unreachable (cfr. section 6.2.6.2).

The eRouter MUST use an exponential backoff algorithm for timeout value in case the DNS is unavailable and the DNS query needs to be retried.

6.2.6.2 Failover

When defining a GRE tunnel endpoint as part of a policy, the eRouter MUST support configuration of both a primary and secondary GRE tunnel endpoint.

Detecting an unreachable GRE tunnel endpoint MUST be supported through the following mechanism:

The eRouter MUST support sending ICMP PING packets to the GRE tunnel endpoint at regular, configurable intervals.

The PING packets must be sourced from the eRouter’s IP address.

The eRouter MUST only send PING packets when there was no traffic received from the GRE tunnel endpoint for a configurable amount of time.
If there is no ICMP PING reply for a configurable number of ICMP PING packets, the eRouter MUST deem the GRE tunnel endpoint as unreachable.

If a failover GRE tunnel endpoint is defined in the policy, the eRouter MUST start tunneling traffic to the other GRE tunnel endpoint and this endpoint is promoted to primary tunnel endpoint (while the primary GRE tunnel endpoint is renamed to secondary GRE tunnel endpoint).

If both GRE tunnel endpoints are unreachable (or only one is defined and unreachable), the eRouter MUST disable all SSIDs that link to it, even if there are other Wi-Fi services deployed via those SSIDs that do not link to this policy.

While both GRE tunnel endpoints are unreachable (or only one is defined and unreachable), the eRouter MUST continue sending ICMP PING packets to the unreachable GRE tunnel endpoint at configurable intervals (this interval is independent to the one that is used when the GRE tunnel endpoint is reachable). When ICMP PING replies are received, the eRouter MUST set the state of the GRE tunnel endpoint to reachable again and enable the SSIDs that were disabled.

The eRouter MUST support enabling/disabling this failover mechanism on a per policy base. In case the failover mechanism is disabled, no detection of unreachable GRE tunnel endpoints is to be performed.

6.2.7 DHCP Intercept Parameters

The eRouter MUST support all mandatory features and SHOULD support all optional features described in [WR-SP-WiFi-GW], section 6.2.2.

DHCP intercept parameters MUST be configurable per SSID.

6.3 Resources and Traffic Priority

The Wi-Fi gateways does not have to support [WR-SP-WiFi-GW], section 6.3.

The Wi-Fi gateway MUST support traffic prioritization procedures and capabilities called out in [MULPI3.0].

The Wi-Fi gateway MUST support traffic prioritization procedures and capabilities called out in [WMM].

The Wi-Fi gateway MUST support DSCP to WMM Traffic Priority mapping like listed in Table 1.

<table>
<thead>
<tr>
<th>DSCP Traffic Type</th>
<th>DSCP Value</th>
<th>WMM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control</td>
<td>56 (0x38)</td>
<td>VO (Voice priority)</td>
</tr>
<tr>
<td>Audio</td>
<td>56 (0x38)</td>
<td>VO (Voice priority)</td>
</tr>
<tr>
<td>Video</td>
<td>40 (0x28)</td>
<td>VI (Video priority)</td>
</tr>
<tr>
<td>Best Effort</td>
<td>0 (0x00)</td>
<td>BE (Best effort priority)</td>
</tr>
<tr>
<td>Excellent Effort</td>
<td>24 (0x18)</td>
<td>BE (Best effort priority)</td>
</tr>
</tbody>
</table>
### 6.4 RADIUS Client Interface

The eRouter MUST support a RADIUS signalling client.

The eRouter does not have to support reported operator and location attributes defined in Error! Reference source not found..

The eRouter MUST send all RADIUS messages with source IP address equal to the eRouter WAN IP address.

The eRouter MUST support provisioning of the RADIUS server by IP address or FQDN.

The Wi-Fi gateway must include following information in each RADIUS message to the AAA server:

- **SSID**
- **Calling-Station-Id:** Client Wi-Fi device MAC address
- **Called-Station-Id:** eRouter MAC address (WAN interface)

The eRouter MUST support RADIUS VSAs to define classifiers (cfr. section 6.2.2.1) at the moment the first client Wi-Fi devices authenticates for a specific wireless service.

The eRouter MUST support the RADIUS Session-Timeout attribute and MUST disassociate the client Wi-Fi device if the session is expired.

The eRouter MUST support RADIUS Accounting messages/attributes according to [RFC 2866]. The eRouter MUST be able to detect when a new client Wi-Fi device connects and MUST send an Accounting-Request packet with Acct-Status-Type=Start to the RADIUS server. The eRouter MUST be able to detect when a client Wi-Fi device is disconnected (through a Disassociation packet or inactivity of the client) and MUST send an Accounting-Request packet with Acct-Status-Type=Stop to the RADIUS server.

### 6.5 Management Interface Requirements

#### 6.5.1 Status and Performance Reports

The Wi-Fi gateway SHOULD support a mechanism to trigger a site survey. When the site survey is triggered on a Wi-Fi gateway that supports it, either by a timing constraint or by a manual trigger, the Wi-Fi gateway MUST perform a site survey and list the SSIDs, BSSIDs (MAC address), SNRs in a MIB table (cfr. section 6.7).

### 6.6 Configured Admission Control

The Wi-Fi gateway SHOULD support MAC address black/white listing.

### 6.7 MIB Architecture

The MIBs are split up in three major parts:

1. Base Objects
2. 802.11 Objects
3. Forwarding and Classification Objects
Each of these parts are described in detail in the following paragraphs.

6.7.1 Base Objects

The base objects are the objects used to configure the eRouter interface and SSID interfaces. The configuration of the SSID interfaces includes bridging and broadcast behavior, DHCP intercept and RADIUS attributes. The eRouter interface configuration includes DNS settings and DHCP attributes.

Furthermore, these objects can be used to configure the attributes that need to be included by the eRouter in RADIUS messages (both for authenticating and accounting purposes) to the AAA server.

Figure 3 shows the links between the different base object tables.

The DHCP and RADIUS attributes tables are linked to an interface (SSID or eRouter) using a link table which is used to configure pointers to the attribute tables. This gives the user the option to link a single interface to multiple (different) attribute tables, including vendor-specific attribute tables.

![MIB Base Objects Diagram](image)

Figure 3: MIB Base Objects Diagram

6.7.2 802.11 Objects

The 802.11 objects are used to configure the 802.11-specific settings on the SSID interfaces.

These settings include both physical layer parameters (channel, channel-width, protection mechanisms, ...) as the mac layer parameters, like security, WMM, WPS and RADIUS parameters.

Figure 4 and Figure 5 show the links between the different 802.11 object tables.
Figure 4: MIB 802.11 Objects Diagram (a)

Figure 5: MIB 802.11 Objects Diagram (b)
The 802.11 MIB objects also include objects for monitoring purposes. These are shown in Figure 6 and include the Wi-Fi survey parameters and a table which shows the currently connected Wi-Fi client devices per SSID.

Figure 6: MIB 802.11 Objects Diagram (c)

6.7.3 Forwarding and Classification Objects

The forwarding and classification objects are used to configure the classifiers and link them to a policy.

Again, link tables are used to link a classifier to its attributes and a policy to its attributes, which, again, allows vendor-specific extensions.

Figure 7: MIB Forwarding and Classification Objects Diagram
7 Annex A: MIB Definitions

7.1 Base Objects

WIFI-GW-BASE-MIB DEFINITIONS ::= BEGIN
 IMPORTS
   NOTIFICATION-TYPE,
   MODULE-IDENTITY,
   OBJECT-TYPE,
   Integer32,
   Unsigned32,
   Counter32,
   enterprises
   FROM SNMPv2-SMI -- RFC 2578
   TEXTUAL-CONVENTION,
   RowStatus,
   TruthValue,
   MacAddress,
   RowStatus,
   RowPointer,
   DateAndTime,
   DisplayString
   FROM SNMPv2-TC -- RFC 2579
   OBJECT-GROUP,
   NOTIFICATION-GROUP,
   MODULE-COMPLIANCE
   FROM SNMPv2-CONF -- RFC 2580
   InetAddressType,
   InetAddress,
   InetAddressDNS,
   InetPortNumber,
   InetAddressPrefixLength
   FROM INET-ADDRESS-MIB -- RFC 4001
   ifIndex
   FROM IF-MIB; -- RFC 2863

--
-- Path to root
--

euroCableLabs OBJECT IDENTIFIER ::= { enterprises 24624 }
eclProject OBJECT IDENTIFIER ::= { euroCableLabs 2 }
eclProjWifiGateway OBJECT IDENTIFIER ::= { eclProject 3 }

--
-- Wi-Fi Gateway Module
--

wifiGwBase MODULE-IDENTITY
   LAST-UPDATED "201211140000Z" -- November 14, 2012
This MIB module contains the management objects for the base interfaces of Wi-Fi Gateway devices.

Copyright 2012 EuroCableLabs
All rights reserved.

::= { eclProjWifiGateway 1 }
DESCRIPTION
"The list contains the parameters, associated with the interface of the Wi-Fi Gateway.

These interfaces include the Ethernet interfaces and all SSIDs (wireless)"

 ::= { wifiGwBaseMibObjects 1 }

wifiGwIfConfigurationEntry OBJECT-TYPE
SYNTAX     WifiGwIfConfigurationEntry
MAX-ACCESS not-accessible
STATUS      current
DESCRIPTION
"Objects to configure an interface (Ethernet/SSID) of the Wi-Fi Gateway"
INDEX { ifIndex }
 ::= { wifiGwIfConfigurationTable 1 }

WifiGwIfConfigurationEntry ::= SEQUENCE {
  wifiGwIfRoutingMode               IfRoutingMode,
  wifiGwIfLocalSwitchingEnabled     TruthValue,
  wifiGwIfBroadcastMulticastBlockedEnabled TruthValue,
  wifiGwIfDhcpBroadcastToUnicastEnabled TruthValue,
  wifiGwIfAllowConfigFromLan        TruthValue
}

wifiGwIfRoutingMode OBJECT-TYPE
SYNTAX     IfRoutingMode
MAX-ACCESS read-write
STATUS      current
DESCRIPTION
"The routing mode for this interface.

The enumerated values associated with the IfRoutingMode are:

'broaded' : indicates that the traffic is not routed by the Wi-Fi Gateway, but
  bridged through the eCM.
'routed' : indicates that the traffic is forwarded through the NAT/Routing
  module of the Wi-Fi Gateway."

REFERENCE
"Wi-Fi Requirements for Cable Modem Gateways CEL-WR-SP-WIFI-GW-V0.3,
  section ?.
DEFVAL { routed }
 ::= { wifiGwIfConfigurationEntry 1 }

wifiGwIfLocalSwitchingEnabled OBJECT-TYPE
SYNTAX     TruthValue
MAX-ACCESS read-write
STATUS      current
DESCRIPTION
"Enables/Disables the local (within the Wi-Fi Gateway) switching of packets
  from connected CPE devices on this interface."

REFERENCE
"Wi-Fi Requirements for Cable Modem Gateways CEL-WR-SP-WIFI-GW-V0.3,
  section Broadcast/Multicast Traffic Forwarding."
DEFVAL { false } ::= { wifiGwIfConfigurationEntry 2 }

wifiGwIfBroadcastMulticastBlockedEnabled OBJECT TYPE
  SYNTAX     TruthValue
  MAX-ACCESS read-write
  STATUS     current
  DESCRIPTION "Enables/Disables the forwarding of broadcast/multicast to connected CPE devices."
  REFERENCE "Wi-Fi Requirements for Cable Modem Gateways CEL-WR-SP-WIFI-GW-V0.3, section Broadcast/Multicast Traffic Forwarding."
DEFVAL { false } ::= { wifiGwIfConfigurationEntry 3 }

wifiGwIfDhcpBroadcastToUnicastEnabled OBJECT TYPE
  SYNTAX     TruthValue
  MAX-ACCESS read-write
  STATUS     current
  DESCRIPTION "Enables/Disables the conversion (at layer 2) from broadcast to unicast DHCPv4 packets,
          destined for connected CPE devices.
          The destination MAC address is converted, by the Wi-Fi Gateway, to unicast
          by replacing it with the value of the 'Client MAC address' field inside the
          DHCPv4 packet"
  REFERENCE "Wi-Fi Requirements for Cable Modem Gateways CEL-WR-SP-WIFI-GW-V0.3, section Broadcast/Multicast Traffic Forwarding."
DEFVAL { false } ::= { wifiGwIfConfigurationEntry 4 }

wifiGwIfAllowConfigFromLan OBJECT TYPE
  SYNTAX     TruthValue
  MAX-ACCESS read-write
  STATUS     current
  DESCRIPTION "Enables/Disables the configuration of this interface from a connected CPE devices.
          The configuration interface (HTTP, SNMP, ...) is defined by the vendor."
DEFVAL { true } ::= { wifiGwIfConfigurationEntry 5 }

-- Wi-Fi Gateway eRouter Interface Configuration Table
wifiGwERouterIfConfigurationTable OBJECT TYPE
  SYNTAX     SEQUENCE OF WifiGwERouterIfConfigurationEntry
  MAX-ACCESS not-accessible
  STATUS     current
  DESCRIPTION "The list contains the parameters, associated with an eRouter interface of the Wi-Fi Gateway.
          These eRouter interfaces are used for GRE encapsulation and NAT purposes."

::= { wifiGwBaseMibObjects 2 }

wifiGwERouterIfConfigurationEntry OBJECT-TYPE
SYNTAX     WifiGwERouterIfConfigurationEntry
MAX-ACCESS not-accessible
STATUS      current
DESCRIPTION "Objects to configure an eRouter interface of the Wi-Fi Gateway"
INDEX { ifIndex }
::= { wifiGwERouterIfConfigurationTable 1 }

WifiGwERouterIfConfigurationEntry ::= SEQUENCE {
    wifiGwERouterIfDnsServerAddressType InetAddressType,
    wifiGwERouterIfDnsServerAddress1 InetAddress,
    wifiGwERouterIfDnsServerAddress2 InetAddress,
    wifiGwERouterIfDhcpAttributesLinkId Unsigned32,
}

wifiGwERouterIfDnsServerAddressType OBJECT-TYPE
SYNTAX     InetAddressType
MAX-ACCESS read-write
STATUS      current
DESCRIPTION "The address type for the DNS server(s), configured for this eRouter interface. The allowed enumerated values associated with the InetAddressType are:

'ipv4' (1) indicates that the wifiGwERouterIfConfDnsServerAddress1 and wifiGwERouterIfConfDnsServerAddress2 entries will be of type InetAddressIPv4.
'ipv6' (2) indicates that the wifiGwERouterIfConfDnsServerAddress1 and wifiGwERouterIfConfDnsServerAddress2 entries will be of type InetAddressIPv6."
DEFVAL { ipv4 }
::= { wifiGwERouterIfConfigurationEntry 1 }

wifiGwERouterIfDnsServerAddress1 OBJECT-TYPE
SYNTAX     InetAddress
MAX-ACCESS read-write
STATUS      current
DESCRIPTION "The primary DNS server, configured for this eRouter interface. This entry needs to be consistent with the wifiGwERouterIfConfDnsServerAddressType value:
'ipv4' (1) indicates that this entry needs to be of type InetAddressIPv4.
'ipv6' (2) indicates that this entry needs to be of type InetAddressIPv6."
::= { wifiGwERouterIfConfigurationEntry 2 }

wifiGwERouterIfDnsServerAddress2 OBJECT-TYPE
SYNTAX     InetAddress
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"The secondary DNS server, configured for this eRouter interface.

This entry needs to be consistent with the wifiGwERouterIfConfDnsServerAddressType value:
 'ipv4' (1) indicates that this entry needs to be of type InetAddressIPv4. A value of 0.0.0.0 indicates that there is no secondary DNS
 'ipv6' (2) indicates that this entry needs to be of type InetAddressIPv6. A value of 0000:0000:0000:0000:0000:0000:0000:0000 indicates that there is no secondary DNS
"
 ::= { wifiGwERouterIfConfigurationEntry 3 }

wifiGwERouterIfDhcpAttributesLinkId OBJECT-TYPE
SYNTAX     Unsigned32
MAX-ACCESS read-write
STATUS     current
DESCRIPTION
"The DHCP Attributes Link ID, associated with this eRouter interface. This links to the wifiGwDhcpAttributesLinkTable where the attributes can be configured that the Wi-Fi Gateway must include when performing DHCP on this eRouter interface.
"
 ::= { wifiGwERouterIfConfigurationEntry 4 }

-- Wi-Fi Gateway DHCP Intercept Configuration Table
wifiGwDhcpInterceptConfigurationTable OBJECT-TYPE
SYNTAX      SEQUENCE OF WifiGwDhcpInterceptConfigurationEntry
MAX-ACCESS not-accessible
STATUS     current
DESCRIPTION
"This table configures the DHCP interception parameters (i.e. DHCP attributes) for a Wi-Fi Gateway's interface (Ethernet or SSIDs).

These parameters will be used when DHCP traffic, from/to connected client devices, is intercepted by the Wi-Fi Gateway.
"
 ::= { wifiGwBaseMibObjects 3 }

WifiGwDhcpInterceptConfigurationEntry OBJECT-TYPE
SYNTAX     WifiGwDhcpInterceptConfigurationEntry
MAX-ACCESS not-accessible
STATUS     current
DESCRIPTION
"Objects to configure DHCP intercept parameters for a Wi-Fi Gateway interface (Ethernet/SSID)"
INDEX { ifIndex } ::= { wifiGwDhcpInterceptConfigurationTable 1 }

WifiGwDhcpInterceptConfigurationEntry ::= SEQUENCE {
  wifiGwDhcpInterceptAttributesLinkId     Unsigned32,  
  wifiGwDhcpInterceptEnabled             TruthValue 
}
wifiGwDhcpInterceptAttributesLinkId OBJECT-TYPE
   SYNTAX     Unsigned32 (0..65535)
   MAX-ACCESS read-write
   STATUS     current
   DESCRIPTION
      "This key represents a link to the wifiGwDhcpAttributesLinkTable"
   ::= { wifiGwDhcpInterceptConfigurationEntry 1 }

wifiGwDhcpInterceptEnabled OBJECT-TYPE
   SYNTAX     TruthValue
   MAX-ACCESS read-write
   STATUS     current
   DESCRIPTION
      "Enables/disables DHCP interception on this interface (Ethernet/SSID)"
   DEFVAL { false }
   ::= { wifiGwDhcpInterceptConfigurationEntry 2 }

-- Wi-Fi Gateway DHCP Attributes Link Table
wifiGwDhcpAttributes OBJECT IDENTIFIER ::= { wifiGwBaseMibObjects 4 }

wifiGwDhcpAttributesLinkTable OBJECT-TYPE
   SYNTAX      SEQUENCE OF WifiGwDhcpAttributesLinkEntry
   MAX-ACCESS not-accessible
   STATUS     current
   DESCRIPTION
      "This table links to specific DHCP (IPv4/IPv6) attribute tables and is referenced
by other tables
   (i.e. wifiGwERouterIfConfigurationTable and wifiGwDhcpInterceptConfigurationTable)
to link to these DHCP attributes."
   ::= { wifiGwDhcpAttributes 1 }

WifiGwDhcpAttributesLinkEntry OBJECT-TYPE
   SYNTAX      WifiGwDhcpAttributesLinkEntry
   MAX-ACCESS not-accessible
   STATUS     current
   DESCRIPTION
      "Objects to configure links to DHCP attributes"
   INDEX { wifiGwDhcpAttributesLinkId,
            wifiGwDhcpAttributesPointerId }
   ::= { wifiGwDhcpAttributesLinkTable 1 }

WifiGwDhcpAttributesLinkEntry ::= SEQUENCE {
   wifiGwDhcpAttributesLinkId           Unsigned32,
   wifiGwDhcpAttributesPointerId        Unsigned32,
   wifiGwDhcpAttributesPointer          RowPointer,
   wifiGwDhcpAttributesRowStatus        RowStatus
}

wifiGwDhcpAttributesLinkId OBJECT-TYPE
   SYNTAX     Unsigned32 (0..65535)
   MAX-ACCESS not-accessible
   STATUS     current
DESCRIPTION

"This key represents the identifier for a link to this table (from the
wifiGwERouterIfConfigurationTable or wifiGwDhcpInterceptConfigurationTable)
It is used to group multiple wifiGwDhcpAttributesPointer entries together."
::= { wifiGwDhcpAttributesLinkEntry 1 }

wifiGwDhcpAttributesPointerId OBJECT-TYPE
SYNTAX Unsigned32 (0..65535)
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"This key represents the identifier for a wifiGwDhcpAttributesPointer entry, which points
to another table (i.e. wifiGwDhcpv4AttributesTable or
wifiGwDhcpv6AttributesTable)"
::= { wifiGwDhcpAttributesLinkEntry 2 }

wifiGwDhcpAttributesPointer OBJECT-TYPE
SYNTAX RowPointer
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"A pointer to an entry in a DHCP attribute configuration table. e.g
wifiGwDhcpv4AttributesRowStatus in wifiGwDhcpv4AttributesEntry or
wifiGwDhcpv6AttributesRowStatus in wifiGwDhcpv6AttributesEntry.
A value pointing to zeroDotZero, an inactive Row or a
non-existing entry is treated as no attributes defined for this
entry."
DEFVAL {zeroDotZero }
::= { wifiGwDhcpAttributesLinkEntry 3 }

wifiGwDhcpAttributesRowStatus OBJECT-TYPE
SYNTAX RowStatus
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"This object is to create or delete rows in this table."
::= { wifiGwDhcpAttributesLinkEntry 4 }

-- Wi-Fi Gateway DHCPv4 Attributes Table
wifiGwDhcpv4AttributesTable OBJECT-TYPE
SYNTAX SEQUENCE OF WifiGwDhcpv4AttributesEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"This table configures the DHCPv4 attributes."
::= { wifiGwDhcpAttributes 2 }

wifiGwDhcpv4AttributesEntry OBJECT-TYPE
SYNTAX WifiGwDhcpv4AttributesEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"Objects to configure the DHCPv4 attributes."
INDEX { wifiGwDhcpv4AttributesId }
 ::= { wifiGwDhcpv4AttributesTable 1 }

WifiGwDhcpv4AttributesEntry ::= SEQUENCE {
  wifiGwDhcpv4AttributesId Unsigned32,
  wifiGwDhcpv4AttributesRowStatus RowStatus,
  wifiGwDhcpv4CmMacOption82p2Enabled TruthValue,
  wifiGwDhcpv4SsidOption60Enabled TruthValue,
  wifiGwDhcpv4SecurityKeyOption82Enabled TruthValue
}

wifiGwDhcpv4AttributesId OBJECT-TYPE
SYNTAX Unsigned32 (0..65535)
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "This key represents the identifier for this entry in the DHCPv4 Attributes table."
 ::= { wifiGwDhcpv4AttributesEntry 1 }

wifiGwDhcpv4AttributesRowStatus OBJECT-TYPE
SYNTAX RowStatus
MAX-ACCESS read-create
STATUS current
DESCRIPTION "This object is to create or delete rows in this table."
 ::= { wifiGwDhcpv4AttributesEntry 2 }

wifiGwDhcpv4CmMacOption82p2Enabled OBJECT-TYPE
SYNTAX TruthValue
MAX-ACCESS read-create
STATUS current
DESCRIPTION "Enables/Disables CM MAC address included in DHCP option 82.2."
 ::= { wifiGwDhcpv4AttributesEntry 3 }

wifiGwDhcpv4SsidOption60Enabled OBJECT-TYPE
SYNTAX TruthValue
MAX-ACCESS read-create
STATUS current
DESCRIPTION "Enables/Disables SSID included in DHCP option 60."
 ::= { wifiGwDhcpv4AttributesEntry 4 }

wifiGwDhcpv4SecurityKeyOption82Enabled OBJECT-TYPE
SYNTAX TruthValue
MAX-ACCESS read-create
STATUS current
DESCRIPTION "Enables/Disables Security Key (wifiGwGreTunnelSecurityKey) included in DHCP option 82."
 ::= { wifiGwDhcpv4AttributesEntry 5 }
-- Wi-Fi Gateway DHCPv6 Attributes Table

wifiGwDhcpv6AttributesTable OBJECT-TYPE
   SYNTAX     SEQUENCE OF WifiGwDhcpv6AttributesEntry
   MAX-ACCESS not-accessible
   STATUS     current
   DESCRIPTION
      "This table configures the DHCPv6 attributes."
   ::= { wifiGwDhcpAttributes 3 }

wifiGwDhcpv6AttributesEntry OBJECT-TYPE
   SYNTAX      WifiGwDhcpv6AttributesEntry
   MAX-ACCESS not-accessible
   STATUS     current
   DESCRIPTION
      "Objects to configure the DHCPv6 attributes."
   INDEX { wifiGwDhcpv6AttributesId }
   ::= { wifiGwDhcpv6AttributesTable 1 }

WifiGwDhcpv6AttributesEntry ::= SEQUENCE {
   wifiGwDhcpv6AttributesId     Unsigned32,
   wifiGwDhcpv6AttributesRowStatus RowStatus
}

wifiGwDhcpv6AttributesId OBJECT-TYPE
   SYNTAX     Unsigned32 (0..65535)
   MAX-ACCESS not-accessible
   STATUS     current
   DESCRIPTION
      "This key represents the identifier for this entry in the DHCPv6 Attributes table."
   ::= { wifiGwDhcpv6AttributesEntry 1 }

wifiGwDhcpv6AttributesRowStatus OBJECT-TYPE
   SYNTAX      RowStatus
   MAX-ACCESS read-create
   STATUS     current
   DESCRIPTION
      "This object is to create or delete rows in this table."
   ::= { wifiGwDhcpv6AttributesEntry 2 }

-- Wi-Fi Gateway RADIUS Authentication Attributes Link Table

wifiGwRadiusAttributes OBJECT IDENTIFIER ::= { wifiGwBaseMibObjects 5 }

wifiGwRadiusAuthenticationAttributesLinkTable OBJECT-TYPE
   SYNTAX     SEQUENCE OF WifiGwRadiusAuthenticationAttributesLinkEntry
   MAX-ACCESS not-accessible
   STATUS     current
   DESCRIPTION
      "This table links to specific RADIUS attribute tables to enable/disable RADIUS attributes that must be used for RADIUS authentication traffic on this interface (SSID)."
   ::= { wifiGwRadiusAttributes 1 }
CELE: Wi-Fi Requirements for Cable Modem Gateways

Wi-Fi Gateway RADIUS Account Attributes Link Table

\\*wifiGwRadiusAccountingAttributesLinkTable* OBJECT-TYPE
SYNTAX \ SEQUENCE OF WifiGwRadiusAccountingAttributesLinkEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"Attributes representing other RADIUS accounting attributes for a Wi-Fi Gateway interface (SSID)"
INDEX { ifIndex, 
wifiGwRadiusAuthAttributesPointerId } ::= { wifiGwRadiusAuthenticationAttributesLinkTable 1 }

**WifiGwRadiusAccountingAttributesLinkEntry** ::= SEQUENCE {
 wifiGwRadiusAuthAttributesPointerId Unsigned32,
 wifiGwRadiusAuthAttributesPointer RowPointer,
 wifiGwRadiusAuthAttributesRowStatus RowStatus
}

---

© Cable Europe Labs
This table links to specific RADIUS attribute tables to enable/disable RADIUS attributes that must be used for RADIUS Accounting traffic on this interface (SSID).

```
::= { wifiGwRadiusAttributes 2 }
```

```
WifiGwRadiusAccountingAttributesLinkEntry ::= SEQUENCE {
    wifiGwRadiusAcctAttributesPointerId     Unsigned32,
    wifiGwRadiusAcctAttributesPointer       RowPointer,
    wifiGwRadiusAcctAttributesRowStatus     RowStatus
}
```

```
wifiGwRadiusAcctAttributesPointerId OBJECT-TYPE
SYNTAX     Unsigned32 (0..65535)
MAX-ACCESS not-accessible
STATUS     current
DESCRIPTION
   "This key represents the identifier for a wifiGwRadiusAcctAttributesPointer entry, which points to another table (i.e. wifiGwRadiusAttributesTable)"
::= { wifiGwRadiusAccountingAttributesLinkEntry 1 }
```

```
wifiGwRadiusAcctAttributesPointer OBJECT-TYPE
SYNTAX     RowPointer
MAX-ACCESS read-create
STATUS     current
DESCRIPTION
   "A pointer to an entry in a RADIUS attribute configuration table. e.g
    wifiGwRadiusAttributesRowStatus in wifiGwRadiusAttributesEntry

    A value pointing to zeroDotZero, an inactive Row or a non-existing entry is treated as no attributes defined for this entry."
DEFVAL {zeroDotZero }
::= { wifiGwRadiusAccountingAttributesLinkEntry 2 }
```

```
wifiGwRadiusAcctAttributesRowStatus OBJECT-TYPE
SYNTAX     RowStatus
MAX-ACCESS read-create
STATUS     current
DESCRIPTION
   "This object is to create or delete rows in this table."
```
::= { wifiGwRadiusAccountingAttributesLinkEntry 3 }

-- Wi-Fi Gateway RADIUS Attributes Table

wifiGwRadiusAttributesTable OBJECT-TYPE
  SYNTAX      SEQUENCE OF WifiGwRadiusAttributesEntry
  MAX-ACCESS not-accessible
  STATUS      current
  DESCRIPTION
    "This table configures the RADIUS attributes."
  ::= { wifiGwRadiusAttributes 3 }

wifiGwRadiusAttributesEntry OBJECT-TYPE
  SYNTAX      WifiGwRadiusAttributesEntry
  MAX-ACCESS not-accessible
  STATUS      current
  DESCRIPTION
    "Objects to configure the RADIUS attributes."
  INDEX { wifiGwRadiusAttributesId }
  ::= { wifiGwRadiusAttributesTable 1 }

WifiGwRadiusAttributesEntry ::= SEQUENCE {
  wifiGwRadiusAttributesId       Unsigned32,
  wifiGwRadiusAttributesRowStatus RowStatus,
  wifiGwRadiusAttributesSsidEnabled TruthValue
}

wifiGwRadiusAttributesId OBJECT-TYPE
  SYNTAX       Unsigned32 (0..65535)
  MAX-ACCESS  not-accessible
  STATUS      current
  DESCRIPTION
    "This key represents the identifier for this entry in the RADIUS Attributes
    table."
  ::= { wifiGwRadiusAttributesEntry 1 }

wifiGwRadiusAttributesRowStatus OBJECT-TYPE
  SYNTAX       RowStatus
  MAX-ACCESS  read-create
  STATUS      current
  DESCRIPTION
    "This object is to create or delete rows in this table."
  ::= { wifiGwRadiusAttributesEntry 2 }

wifiGwRadiusAttributesSsidEnabled OBJECT-TYPE
  SYNTAX       TruthValue
  MAX-ACCESS  read-create
  STATUS      current
  DESCRIPTION
    "Enable/Disable the SSID attribute in RADIUS messages"
  DEFVAL { false }
  ::= { wifiGwRadiusAttributesEntry 3 }
7.2 802.11 Objects

WIFI-GW-802dot11-MIB DEFINITIONS ::= BEGIN
IMPORTS
  NOTIFICATION-TYPE,
  MODULE-IDENTITY,
  OBJECT-TYPE,
  Integer32,
  Unsigned32,
  Counter32,
  enterprises
   FROM SNMPv2-SMI -- RFC 2578
  TEXTUAL-CONVENTION,
  RowStatus,
  TruthValue,
  MacAddress,
  RowStatus,
  RowPointer,
  DateAndTime,
  DisplayString
   FROM SNMPv2-TC -- RFC 2579
  OBJECT-GROUP,
  NOTIFICATION-GROUP,
  MODULE-COMPLIANCE
   FROM SNMPv2-CONF -- RFC 2580
SnmpAdminString
   FROM SNMP-FRAMEWORK-MIB
TenthdB,
   FROM DOCS-IF-MIB -- RFC 4546
InetAddressType,
  InetAddress,
  InetAddressDNS,
  InetPortNumber,
  InetAddressPrefixLength
   FROM INET-ADDRESS-MIB -- RFC 4001
ifIndex
   FROM IF-MIB; -- RFC 2863

--
-- Path to root
--

euroCableLabs OBJECT IDENTIFIER ::= { enterprises 24624 }
eclProject OBJECT IDENTIFIER ::= { euroCableLabs 2 }
eclProjWifiGateway OBJECT IDENTIFIER ::= { eclProject 3 }

--
-- Wi-Fi Gateway Module
--
DESCRIPTION

"This MIB module contains the management objects for the
802.11 radio interface(s) and SSID interfaces of Wi-Fi Gateway devices.

Copyright 2012 EuroCableLabs
All rights reserved."

::= { eclProjWifiGateway 2 }

--

-- Textual Conventions

--

RadioWifiVersion ::= TEXTUAL-CONVENTION
  STATUS current
  DESCRIPTION "This data type defines the Wi-Fi version.

The enumerated values associated with the RadioWifiVersion are:
'dot11b'  : 802.11b
'dot11a'  : 802.11a
'dot11g'  : 802.11g
'dot11n'  : 802.11n
"
  SYNTAX INTEGER
  
  { dot11b(1),
    dot11a(2),
    dot11g(3),
    dot11n(4) }

RadioOperMode ::= TEXTUAL-CONVENTION
  STATUS current
  DESCRIPTION "This data type defines the operation mode of a Wi-Fi Gateway's radio interface.

The enumerated values associated with the RadioOperMode are:

'enable'   : indicates that the interface is enabled.
'disable'  : indicates that the interface is disabled."
  SYNTAX INTEGER
RadioBand ::= TEXTUAL-CONVENTION
STATUS  current
DESCRIPTION
  "This data type defines the frequency band of a Wi-Fi Gateway's radio interface.
  The enumerated values associated with the RadioBand are:
  'band-2-4G' : indicates that the interface is situated at frequency band 2.4 GHz.
  'band-5G' : indicates that the interface is situated at frequency band 5 GHz."
SYNTAX INTEGER {
  band-2-4G(1),
  band-5G(2)
}

RadioBandWidth ::= TEXTUAL-CONVENTION
STATUS  current
DESCRIPTION
  "This data type defines the bandwidth of a Wi-Fi Gateway's radio interface.
  The enumerated values associated with the RadioBandWidth are:
  'width-20MHz' : indicates that the interface is 20 MHz.
  'width-20-40MHz' : indicates that the interface is 40 MHz (fallback to 20 MHz if required)."
SYNTAX INTEGER {
  width-20MHz(1),
  width-20-40MHz(2)
}

RadioSideBand ::= TEXTUAL-CONVENTION
STATUS  current
DESCRIPTION
  "This data type defines the location of the sideband of a Wi-Fi Gateway's radio interface in case it has a bandwidth of 40 MHz.
  The enumerated values associated with the RadioSideBand are:
  'upper' : indicates that the sideband is on the right side of the primary channel.
  'lower' : indicates that the sideband is on the left side of the primary channel."
SYNTAX INTEGER {
  upper(1),
  lower(2)
}

RadioProtection ::= TEXTUAL-CONVENTION
STATUS       current
DESCRIPTION

"This data type defines the protection mechanism that is used on a Wi-Fi Gateway's radio interface.

The enumerated values associated with the RadioProtection are:

'rts_cts' : indicates that RTS/CTS will be used for protection.
'cts_to_self' : indicates that CTS-to-self will be used for protection.
'off' : indicates that no protection will be used."

SYNTAX INTEGER {
    rts_cts(1),
    cts_to_self(2),
    off(3),
}

BssSecurityMode ::= TEXTUAL-CONVENTION
STATUS       current
DESCRIPTION

"Security mode may be chosen from Disable, WEP, WPA-PSK, WPA2-PSK,
WPA-Enterprise, WPA2-Enterprise, WPA-PSK_WPA2-PSK or
WPA-Enterprise_WPA2-Enterprise.

If set to disabled(0):
    Open system. No authentication or encryption.

If set to wep(1) then the following parameters must be set:
    Default Transmit Key (wifiGw802dot11BssWepDefaultKey):
        Choose which Key to use.
    WEP Key Bit (wifiGw802dot11BssWepEncryptionMode):
        May select from 64-Bit or 128-Bit encryption.
    Passphrase (wifiGw802dot11BssWepPassPhrase):
        Enter a passphrase consisting of any keyboard character to be used to generate a hex WEP key.
    Key 1-4 (wifiGw802dot11BssWep64BitKeyTable or wifiGw802dot11BssWep128BitKeyTable):
        Enter a WEP key manually. Must use only hex character (0-9 and A-F).
        64-bit WEP requires 10 hex characters, 128-bit WEP requires the use of 26 hex characters.

If set to wpaPsk(2) then the following parameters must be set:
    WPA Algorithm (wifiGw802dot11BssWpaPskAlgorithm):
        TKIP or AES.
    WPA Pre-Shared Key (wifiGw802dot11BssWpaPskPreSharedKey):
        Choose a unique key to authenticate with other devices on the network.
        The Pre-Shared Key must be between 8 and 63 characters in length.
    Group Key Renewal (wifiGw802dot11BssWpaPskGroupRekeyInterval):
        This settings determines how often the group key changes.

If set to wpa2Psk(3) then the following parameters must be set:
    WPA2 Algorithm (wifiGw802dot11BssWpaPskAlgorithm):
        AES or TKIP+AES.
    WPA2 Pre-Shared Key (wifiGw802dot11BssWpaPskPreSharedKey):
Choose a unique key to authenticate with other devices on the network. The Pre-Shared Key must be between 8 and 63 characters in length.

Group Key Renewal (wifiGw802dot11BssWpaPskGroupRekeyInterval):
This settings determines how often the group key changes.

If set to wpaEnterprise(4) then the following parameters must be set:
WPA Algorithm (wifiGw802dot11BssWpaEnterpriseAlgorithm):
   TKIP or AES.
RADIUS Configuration (wifiGw802dot11BssWpaEnterpriseRadiusId)
   This settings links to the wifiGw802dot11MgmtRadiusTable.
Group Key Renewal (wifiGw802dot11BssWpaEnterpriseGroupRekeyInterval):
   This settings determines how often the group key changes.

If set to wpa2Enterprise(5) then the following parameters must be set:
WPA2 Algorithm (cmdot11BssWpaEnterpriseAlgorithm):
   AES or TKIP+AES.
RADIUS Configuration (wifiGw802dot11BssWpaEnterpriseRadiusId)
   This settings links to the wifiGw802dot11MgmtRadiusTable.
Group Key Renewal (wifiGw802dot11BssWpaEnterpriseGroupRekeyInterval):
   This settings determines how often the group key changes.

If set to wpaPskwpa2Psk(6) then the following parameters must be set:
WPA or WPA2 Algorithm (wifiGw802dot11BssWpaPskAlgorithm):
   AES or TKIP+AES.
WPA or WPA2 Pre-Shared Key (wifiGw802dot11BssWpaPskPreSharedKey):
   Choose a unique key to authenticate with other devices on the network. The Pre-Shared Key must be between 8 and 63 characters in length.
Group Key Renewal (wifiGw802dot11BssWpaPskGroupRekeyInterval):
   This settings determines how often the group key changes.

If set to wpaEnterpriseWpa2Enterprise(7) then the following parameters must be set:
WPA2 Algorithm (wifiGw802dot11BssWpaPskAlgorithm):
   AES or TKIP+AES.
RADIUS Configuration (wifiGw802dot11BssWpaEnterpriseRadiusId)
   This settings links to the wifiGw802dot11MgmtRadiusTable.
Group Key Renewal (wifiGw802dot11BssWpaEnterpriseGroupRekeyInterval):
   This settings determines how often the group key changes.

"'
SYNTAX INTEGER {
   disabled(0),
   wep(1),
   wpaPsk(2),
   wpa2Psk(3),
   wpaEnterprise(4),
   wpa2Enterprise(5),
   wpaPskwpa2Psk(6),
   wpaEnterpriseWpa2Enterprise(7)
}

BssAccessMode ::= TEXTUAL-CONVENTION
   STATUS current
   DESCRIPTION

"Controls what stations will be given access to the device.

'allowAny' : any station will be allowed to connect.
'restrictToList' : only stations whose MAC address appears in the wifiGw802dot11BssAccessFilterTable will be allowed to connect.
'denyList' : any station will be allowed to connect except those in the wifiGw802dot11anBssAccessFilterTable.

SYNTAX INTEGER {
   allowAny(1),
   restrictToList(2),
   denyList(3)
}

WepEncryptionMode ::= TEXTUAL-CONVENTION
STATUS current
DESCRIPTION "
Selects the WEP encryption method used by this service set.

'wep64' : 64 bit WEP encryption will be used with the keys from the wifiGw802dot11BssWep64BitKeyTable.
'wep128' : 128 bit WEP encryption will be used with keys from the wifiGw802dot11BssWep128BitKeyTable.

SYNTAX INTEGER {
   wep64(1),
   wep128(2)
}

WepAuthenticationMode ::= TEXTUAL-CONVENTION
STATUS current
DESCRIPTION "
This controls the use of Shared Key authentication in WEP protocol.

'optional(1)' : Shared Key authentication is optional.
'required(2)' : Shared Key authentication is required for WEP.

SYNTAX INTEGER {
   optional(1),
   required(2)
}

WpaAlgorithm ::= TEXTUAL-CONVENTION
STATUS current
DESCRIPTION "
Encryption method, in case the security mode of the BSS is set to WPA/WPA2

'tkip' : TKIP
'AES' : AES
'tkipPlusAes' : Support both TKIP and AES (client device choses encryption method)

SYNTAX INTEGER {
    tkip(1),
    aes(2),
    tkipPlusAes(3)
}

WmmAccessCategory ::= TEXTUAL-CONVENTION
STATUS current
DESCRIPTION "Access Category, used for WMM QoS on a BSS"
The enumerated values for the WMM Access Categories are:
- background(1)  : AC(background)
- besteffort(2)  : AC(best effort)
- video(3)      : AC(video)
- voice(4)      : AC(voice)

SYNTAX INTEGER {
    background(1),
    besteffort(2),
    video(3),
    voice(4)
}

WpsMethod ::= TEXTUAL-CONVENTION
STATUS current
DESCRIPTION "WPS method may be choosen from either Push Button Configuration(1) or Personal Information Number(2)."
- wpsPBC(1)      : user has to push a button, either an actual or virtual one, on both WPS devices.
- wpsPIN(2)      : user has to input a wireless client's PIN on the WPS AP.

SYNTAX INTEGER {
    wpsPBC(1),
    wpsPIN(2)
}

RadiusConnectivityStatus ::= TEXTUAL-CONVENTION
STATUS current
DESCRIPTION "Connectivity status of the configured RADIUS server"
Possible states are:
- 'radiusPrimaryInUse' : primary RADIUS address is in use and has no connectivity issues
'radiusSecondaryInUse' : secondary RADIUS address is in use, because there were connectivity issues with the primary RADIUS address

'radiusUnreachable' : all configured RADIUS addresses are unreachable

SYNTAX INTEGER {
  radiusPrimaryInUse(1),
  radiusSecondaryInUse(2),
  radiusUnreachable(3)
}

CompatibilityNetMode ::= TEXTUAL-CONVENTION
STATUS current
DESCRIPTION "The backwards compatibility netmode that the Wi-Fi Gateway supports

  modeB(1) : The AP behaves as an 802.11b (2.4GHz only) AP and only supports 802.11b features.
  modeBG(2) : The AP behaves as an 802.11g (2.4GHz only) AP and is backwards compatible with 802.11b clients.
  modeG(3) : The AP behaves as an 802.11g (2.4GHz only) AP and does not allow legacy 802.11b clients.
  modeGN(4) : The AP behaves as an 802.11n (2.4GHz only) AP and is backwards compatible with 802.11b/g clients
  modeA(6) : The AP behaves as an 802.11a (5GHz only) AP and only supports 802.11a features
  modeAN(7) : The AP behaves as an 802.11n (5GHz only) AP and is backwards compatible with 802.11a clients
  modeAG(9) : The AP behaves as an 802.11a/802.11g (2.4GHz or 5GHz) AP and does not allow legacy 802.11b clients
  modeAGN(10) : The AP behaves as an 802.11n (2.4GHz or 5GHz) AP and is backwards compatible with 802.11a/g clients, but does not allow legacy 802.11b clients
  modeABGN(11) : The AP behaves as an 802.11n (2.4GHz or 5GHz) AP and is backwards compatible with all legacy clients
  modeN(12) : The AP behaves as an 802.11n (2.4GHz or 5GHz) AP and does not allow any legacy client

  "
SYNTAX INTEGER {
  modeB(1),
  modeBG(2),
  modeG(3),
  modeGN(4),
  modeBGN(5),
  modeA(6),
  modeAN(7),
  modeAG(9),
  modeAGN(10),
  modeABGN(11),
  modeN(12)
}

SnrAction ::= TEXTUAL-CONVENTION
STATUS current
DESCRIPTION "This defines the action that MUST be performed when the SNR
of a client device drops below the defined threshold or is below the defined threshold during an association attempt.

Possible actions are:
- noaction(1) : don't take any action (this is the only mandatory supported action)
- denyassociation(2) : don't allow the client device to associate
- disassociate(3) : send a disassociate message to the client device

```
SYNTAX INTEGER {
    noaction(1),
    denyassociation(2),
    disassociate(3)
}
```

ChannelSurveyMode ::= TEXTUAL-CONVENTION

**STATUS** current
**DESCRIPTION**

The 802.11 channel survey mode

Possible modes are:
- manual(1) : the user has to manual start the channel survey by setting the correct MIB object
- periodic(2) : the Wi-Fi Gateway periodically (based on the configured interval) schedules a channel survey run.

```
SYNTAX INTEGER {
    manual(1),
    periodic(2)
}
```

ClientConnectionStatus ::= TEXTUAL-CONVENTION

**STATUS** current
**DESCRIPTION**

The current status of an client device that is/was associated with the Wi-Fi Gateway

Possible states are:
- connected(1) : client device is connected
- clientDisassociated(2) : client device has sent Disassociate Message
- forcedDisassociatedAuth(3) : client device is disassociated because something went wrong during the security initialization
- forcedDisassociatedTimeout(4) : client device is disassociated because session timed out
- forcedDisassociatedNetMode(5) : client device is disassociated because it was a legacy device which was not supported
- forcedDisassociatedSnr(6) : client device is disassociated because SNR was below threshold
- other(7) : client device is an unspecified other state

```
SYNTAX INTEGER {
    connected(1),
    clientDisassociated(2),
    forcedDisassociatedAuth(3),
```
forcedDisassociatedTimeout(4),
forcedDisassociatedNetMode(5),
forcedDisassociatedSnr(6),
other(7)
}

-- Wi-Fi Gateway MIB Objects

--

wifiGw802dot11MibObjects
OBJECT IDENTIFIER ::= { wifiGw802dot11 1}
wifiGw802dot11Notification
OBJECT IDENTIFIER ::= { wifiGw802dot11 2}
wifiGw802dot11Conformance
OBJECT IDENTIFIER ::= { wifiGw802dot11 3}
wifiGw802dot11Compliances
OBJECT IDENTIFIER ::= {wifiGw802dot11Conformance 1}
wifiGw802dot11Groups
OBJECT IDENTIFIER ::= { wifiGw802dot11Conformance 2}

wifiGw802dot11MgmtRadio
OBJECT IDENTIFIER ::= { wifiGw802dot11MibObjects 1}
wifiGw802dot11MgmtMbss
OBJECT IDENTIFIER ::= { wifiGw802dot11MibObjects 2}
wifiGw802dot11RadioChannelMonitorObjects
OBJECT IDENTIFIER ::= { wifiGw802dot11MibObjects 3}
wifiGw802dot11BssClientObjects
OBJECT IDENTIFIER ::= { wifiGw802dot11MibObjects 4}

wifiGw802dot11MgmtRadioTable
OBJECT-TYPE
SYNTAX SEQUENCE OF WifiGw802dot11MgmtRadioEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "This table defines the objects for the management of the 802.11 radio interface(s) of the Wi-Fi Gateway.
"
::= { wifiGw802dot11MgmtRadio 1 }

WifiGw802dot11MgmtRadioEntry
OBJECT-TYPE
SYNTAX WifiGw802dot11MgmtRadioEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "Objects to configure an 802.11 radio interface on the Wi-Fi Gateway"
INDEX { ifIndex }
::= { wifiGw802dot11MgmtRadioTable 1 }

WifiGw802dot11MgmtRadioEntry ::= SEQUENCE {
  wifiGw802dot11RadioOperMode RadioOperMode, 
  wifiGw802dot11RadioCurrentChannel Unsigned32, 
  wifiGw802dot11RadioChannelSetting Unsigned32, 
  wifiGw802dot11RadioBand RadioBand, 
  wifiGw802dot11RadioBandWidth RadioBandWidth, 
  wifiGw802dot11RadioSideBand RadioSideBand, 
  wifiGw802dot11RadioProtection RadioProtection, 
  wifiGw802dot11RadioMaxConnections Unsigned32 
}
wifiGw802dot11RadioOperMode OBJECT-TYPE
   SYNTAX RadioOperMode
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "Setting this object to enable(1) enables the Wireless interface. Setting this
      object to
disable(2) disables the Wireless interface."
   ::= { wifiGw802dot11MgmtRadioEntry 1 }

wifiGw802dot11RadioCurrentChannel OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "Reading this object displays the current channel number the access point is
      operating in.
      It displays 0 if the Wireless interface is disabled."
   ::= { wifiGw802dot11MgmtRadioEntry 2 }

wifiGw802dot11RadioChannelSetting OBJECT-TYPE
   SYNTAX Unsigned32 (0..165)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This object sets the current channel number (802.11a) or control
      channel (802.11n).

      If set to (0), the AP will be put in auto-channel mode where
      it automatically scans for the least-crowded channel.

      Available channels are 1-11 (2.4GHz) and 36-165 (5GHz).

      Channel selection is also subject to restrictions based on the selected
country code."
   ::= { wifiGw802dot11MgmtRadioEntry 3 }

wifiGw802dot11RadioBand OBJECT-TYPE
   SYNTAX RadioBand
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "Displays the wireless frequency band (2.4 GHz or 5 GHz) the Access Point
      operate in."
   ::= { wifiGw802dot11MgmtRadioEntry 6 }

wifiGw802dot11RadioBandWidth OBJECT-TYPE
   SYNTAX RadioBandWidth
   MAX-ACCESS read-write
   STATUS current

© Cable Europe Labs - CONFIDENTIAL
CEL: Wi-Fi Requirements for Cable Modem Gateways  43

DESCRIPTION
"Specifies the channel width (20 MHz or 20/40 MHz) to be used by the Access Point.

When 20/40MHz is selected 802.11n clients experience improved throughput using 40 MHz, while legacy clients(either 802.11a or 802.11b/g) can still be serviced without interruption using 20MHz.

This MIB object defaults to width-20MHz(1) for radio interfaces where wifiGw802dot11RadioBand equals band-2-4G(1) and to width-20-40MHz(2) for radio interfaces where wifiGw802dot11RadioBand equals band-5G(2)."
::= { wifiGw802dot11MgmtRadioEntry 7 }

wifiGw802dot11RadioSideBand OBJECT-TYPE
SYNTAX RadioSideBand
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"When bonded channels (40 MHz) are used this object specifies if the control (primary) channel is in the lower or upper 20 MHz band of the bonded 40 MHz channel.

Note: This MIB object only applies when wifiGw802dot11RadioBandWidth is set to 20/40MHz."
::= { wifiGw802dot11MgmtRadioEntry 8 }

wifiGw802dot11RadioProtection OBJECT-TYPE
SYNTAX RadioProtection
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"Determines which protection mechanism will be used to protect transmissions for 802.11 legacy devices."
DEFVAL { rts_cts }
::= { wifiGw802dot11MgmtRadioEntry 9 }

wifiGw802dot11RadioMaxConnections OBJECT-TYPE
SYNTAX Unsigned32 (0..1024)
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"Determines how many client devices can be connected simultaneous on this radio interface. When the threshold is exceeded, new client devices will be denied association.

Setting this MIB object to value 0 means no limitation on the maximum number of connected clients on this radio interface."
DEFVAL { 0 }
::= { wifiGw802dot11MgmtRadioEntry 10 }

--
-- wifiGw802dot11MgmtBssBaseTable contains objects used for managing configuration of all basic service sets in an access point which may have multiple BSS's.
wifiGw802dot11MgmtBssBaseTable contains objects used to configure basic operation of each service set.

**Summary**

- **wifiGw802dot11MgmtBssBaseTable**
  - **OBJECT-TYPE**
  - **SEQUENCE OF** WifiGw802dot11MgmtBssBaseEntry
  - **MAX-ACCESS** not-accessible
  - **STATUS** current
  - **DESCRIPTION**
    - "A table of entries in an multiple BSS system."
  - **INDEX** { ifIndex }
  - ** ::= { wifiGw802dot11MgmtMbss 1 }**

- **wifiGw802dot11MgmtBssBaseEntry**
  - **OBJECT-TYPE**
  - **WifiGw802dot11MgmtBssBaseEntry**
  - **MAX-ACCESS** not-accessible
  - **STATUS** current
  - **DESCRIPTION**
    - "An entry describing the characteristics of an individual BSS. An entry exists in this table for each entry of type ieee80211 (71) in the ifTable."
  - **INDEX** { ifIndex }
  - ** ::= { wifiGw802dot11MgmtBssBaseTable 1 }**

**Detailed Description**

- **wifiGw802dot11MgmtBssBaseTable**
  - **SEQUENCE**
    - **{ wifiGw802dot11BssRadioId Unsigned32, wifiGw802dot11BssId MacAddress, wifiGw802dot11BssSsid OCTET STRING, wifiGw802dot11BssMaxConnections Unsigned32, wifiGw802dot11BssWmmEnable TruthValue, wifiGw802dot11BssApsdEnable TruthValue, wifiGw802dot11BssEnable TruthValue }**

- **wifiGw802dot11BssRadioId**
  - **OBJECT-TYPE**
  - **Unsigned32 (0..65535)**
  - **MAX-ACCESS** read-only
  - **STATUS** current
  - **DESCRIPTION**
    - "This MIB object indicates the link (ifIndex) to the 802.11 radio interface on which this BSS resides."
  - ** ::= { wifiGw802dot11MgmtBssBaseEntry 1 }**

- **wifiGw802dot11BssId**
  - **OBJECT-TYPE**
  - **MacAddress**
  - **MAX-ACCESS** read-only
  - **STATUS** current
  - **DESCRIPTION**
"The physical address associated with this service set."

::= { wifiGw802dot11MgmtBssBaseEntry 2 }

wifiGw802dot11BssSsid OBJECT-TYPE
SYNTAX OCTET STRING (SIZE(0..32))
MAX-ACCESS read-write
STATUS current
DESCRIPTION "Controls and reflects the service set identifier for this BSS."

::= { wifiGw802dot11MgmtBssBaseEntry 3 }

wifiGw802dot11BssMaxConnections OBJECT-TYPE
SYNTAX Unsigned32 (0..1024)
MAX-ACCESS read-write
STATUS current
DESCRIPTION "Determines how many client devices can be connected simultaneous on this BSS. When the threshold is exceeded, new client devices will be denied association.

Setting this MIB object to value 0 means no limitation on the maximum number of connected clients on this radio interface."
DEFVAL { 0 }
::= { wifiGw802dot11MgmtBssBaseEntry 4 }

wifiGw802dot11BssWmmEnable OBJECT-TYPE
SYNTAX TruthValue
MAX-ACCESS read-write
STATUS current
DESCRIPTION "Setting this object to true(1) enables WMM functionality on this SSID."
::= { wifiGw802dot11MgmtBssBaseEntry 5 }

wifiGw802dot11BssApsdEnable OBJECT-TYPE
SYNTAX TruthValue
MAX-ACCESS read-write
STATUS current
DESCRIPTION "Setting this object to true(1) enables WMM power saving feature on this SSID."
::= { wifiGw802dot11MgmtBssBaseEntry 6 }

wifiGw802dot11BssEnable OBJECT-TYPE
SYNTAX TruthValue
MAX-ACCESS read-write
STATUS current
DESCRIPTION "Setting this object to true(1) enables the BSS, false(2) disables it."
::= { wifiGw802dot11MgmtBssBaseEntry 7 }

--
-- BSS Security Parameters


--

wifiGw802dot11MgmtBssSecurity OBJECT IDENTIFIER ::= {
    wifiGw802dot11MgmtMbss 2}

--

-- wifiGw802dot11MgmtBssSecurityTable contains objects used to configure security
parameters
-- of each service set.
--

wifiGw802dot11MgmtBssSecurityTable OBJECT-TYPE
SYNTAX SEQUENCE OF WifiGw802dot11MgmtBssSecurityEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "A table of entries in an multiple BSS system."
::= { wifiGw802dot11MgmtBssSecurity 1 }

WifiGw802dot11MgmtBssSecurityEntry OBJECT-TYPE
SYNTAX   WifiGw802dot11MgmtBssSecurityEntry
MAX-ACCESS not-accessible
STATUS   current
DESCRIPTION "An entry describing the security parameters of an individual BSS. An
entry exists in this table for each entry of type ieee80211 (71) in
the ifTable."
INDEX   { ifIndex }
::= { wifiGw802dot11MgmtBssSecurityTable 1 }

WifiGw802dot11MgmtBssSecurityEntry ::= 
SEQUENCE    {
    wifiGw802dot11BssSecurityMode            BssSecurityMode,
    wifiGw802dot11BssClosedNetwork           TruthValue,
    wifiGw802dot11BssAccessMode              BssAccessMode
}

wifiGw802dot11BssSecurityMode OBJECT-TYPE
SYNTAX   BssSecurityMode
MAX-ACCESS read-write
STATUS   current
DESCRIPTION "Security mode may be chosen from Disable, WEP, WPA-PSK, WPA2-PSK,
WPA-Enterprise, WPA2-Enterprise, WPA-PSK_WPA2-PSK or
WPA-Enterprise_WPA2-Enterprise.

If set to disabled(0):
    Open system. No authentication or encryption.

If set to wep(1) then the following parameters must be set:
Default Transmit Key (wifiGw802dot11BssWepDefaultKey):
  Choose which Key to use.
WEP Key Bit (wifiGw802dot11BssWepEncryptionMode):
  May select from 64-Bit or 128-Bit encryption.
Passphrase (wifiGw802dot11BssWepPassPhrase):
  Enter a passphrase consisting of any keyboard character
to be used to generate a hex WEP key.
Key 1-4 (wifiGw802dot11BssWep64BitKeyTable or wifiGw802dot11BssWep128BitKeyTable):
  Enter a WEP key manually. Must use only hex character (0-9 and A-F).
  64-bit WEP requires 10 hex characters, 128-Bit WEP requires the use of
  26 hex characters.

If set to wpaPsk(2) then the following parameters must be set:
  WPA Algorithm (wifiGw802dot11BssWpaPskAlgorithm):
    TKIP or AES.
  WPA Pre-Shared Key (wifiGw802dot11BssWpaPskPreSharedKey):
    Choose a unique key to authenticate with other devices on the network.
    The Pre-Shared Key must be between 8 and 63 characters in length.
  Group Key Renewal (wifiGw802dot11BssWpaPskGroupRekeyInterval):
    This settings determines how often the group key changes.

If set to wpa2Psk(3) then the following parameters must be set:
  WPA2 Algorithm (wifiGw802dot11BssWpaPskAlgorithm):
    AES or TKIP+AES.
  WPA2 Pre-Shared Key (wifiGw802dot11BssWpaPskPreSharedKey):
    Choose a unique key to authenticate with other devices on the network.
    The Pre-Shared Key must be between 8 and 63 characters in length.
  Group Key Renewal (wifiGw802dot11BssWpaPskGroupRekeyInterval):
    This settings determines how often the group key changes.

If set to wpaEnterprise(4) then the following parameters must be set:
  WPA Algorithm (wifiGw802dot11BssWpaEnterpriseAlgorithm):
    TKIP or AES.
  RADIUS Configuration (wifiGw802dot11BssWpaEnterpriseRadiusId)
    This settings links to the wifiGw802dot11MgmtRadiusTable.
  Group Key Renewal (wifiGw802dot11BssWpaEnterpriseGroupRekeyInterval):
    This settings determines how often the group key changes.

If set to wpa2Enterprise(5) then the following parameters must be set:
  WPA2 Algorithm (cmdot11BssWpaEnterpriseAlgorithm):
    AES or TKIP+AES.
  RADIUS Configuration (wifiGw802dot11BssWpaEnterpriseRadiusId)
    This settings links to the wifiGw802dot11MgmtRadiusTable.
  Group Key Renewal (wifiGw802dot11BssWpaEnterpriseGroupRekeyInterval):
    This settings determines how often the group key changes.

If set to wpaPskwpa2Psk(6) then the following parameters must be set:
  WPA or WPA2 Algorithm (wifiGw802dot11BssWpaPskAlgorithm):
    AES or TKIP+AES.
  WPA or WPA2 Pre-Shared Key (wifiGw802dot11BssWpaPskPreSharedKey):
    Choose a unique key to authenticate with other devices on the network.
    The Pre-Shared Key must be between 8 and 63 characters in length.
  Group Key Renewal (wifiGw802dot11BssWpaPskGroupRekeyInterval):

© Cable Europe Labs
This settings determines how often the group key changes.

If set to wpaEnterpriseWpa2Enterprise(7) then the following parameters must be set:

- **WPA2 Algorithm (wifiGw802dot11BssWpaPskAlgorithm):**
  - AES or TKIP+AES.
- **RADIUS Configuration (wifiGw802dot11BssWpaEnterpriseRadiusId):**
  This settings links to the wifiGw802dot11MgmtRadiusTable.
- **Group Key Renewal (wifiGw802dot11BssWpaEnterpriseGroupRekeyInterval):**
  This settings determines how often the group key changes.

```
::= { wifiGw802dot11MgmtBssSecurityEntry 1 }
```

**wifiGw802dot11BssClosedNetwork OBJECT-TYPE**
SYNTAX      TruthValue
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
  "Controls whether the device will operate in closed network mode. If set to true(1), then the network mode is closed and the SSID will not be broadcast in beacon packets. If set to false(2), then the network mode is open and the SSID will be broadcast in beacon packets."

```
::= { wifiGw802dot11MgmtBssSecurityEntry 2 }
```

**wifiGw802dot11BssAccessMode OBJECT-TYPE**
SYNTAX      BssAccessMode
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
  "Controls what stations will be given access to the device.

  'allowAny' : any station will be allowed to connect.
  'restrictToList' : only stations whose MAC address appears in the wifiGw802dot11BssAccessTable will be allowed to connect.
  'denyList' : any station will be allowed to connect except those in the wifiGw802dot11anBssAccessTable."

```
::= { wifiGw802dot11MgmtBssSecurityEntry 3 }
```

-- Objects used to control WEP based security modes

**wifiGw802dot11BssWepTable OBJECT-TYPE**
SYNTAX      SEQUENCE OF WifiGw802dot11BssWepEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
  "A table used to configure settings related to the WEP security modes."

```
::= { wifiGw802dot11MgmtBssSecurityEntry 2 }
```

© Cable Europe Labs - CONFIDENTIAL
wifiGw802dot11BssWepEntry OBJECT-TYPE
SYNTAXWifiGw802dot11BssWepEntry
MAX-ACCESSnot-accessible
STATUScurrent
DESCRIPTION
"An entry defining the WEP characteristics of an individual service set.
A row exists in this table for each row in the
wifiGw802dot11MgmtBssSecurityTable for which
wifiGw802dot11BssSecurityMode is set to wep(1).
"
INDEX{ifIndex}
::={ wifiGw802dot11BssWepTable 1 }

WifiGw802dot11BssWepEntry ::= 
SEQUENCE
{
  wifiGw802dot11BssWepDefaultKey Unsigned32,
  wifiGw802dot11BssWepEncryptionMode WepEncryptionMode,
  wifiGw802dot11BssWepPassPhrase DisplayString,
  wifiGw802dot11BssWepSharedKeyAuthentication WepAuthenticationMode
}

wifiGw802dot11BssWepDefaultKey OBJECT-TYPE
SYNTAXUnsigned32
MAX-ACCESSread-write
STATUScurrent
DESCRIPTION
"Controls and reflects the default key which will be used when 64 or
128 bit encryption is enabled. Indicates the entry from the
wifiGw802dot11BssWep64BitKeyTable if wifiGw802dot11EncryptionMode is set to
wep64(1), or
the entry from the wifiGw802dot11BssWep128BitKeyTable if
wifiGw802dot11BssSecurityMode
is set to wep128(2).

This object may only be set to indicate an active row in the
wifiGw802dot11BssWep64BitKeyTable or wifiGw802dot11BssWep128BitKeyTable. If the
value
corresponds to a row which does not exist or a row which is not
active, the set will be rejected with an inconsistentValue error.
"
::={ wifiGw802dot11BssWepEntry 1 }

wifiGw802dot11BssWepEncryptionMode OBJECT-TYPE
SYNTAXWepEncryptionMode
MAX-ACCESSread-write
STATUScurrent
DESCRIPTION
"Selects the WEP encryption method used by this service set.
If set to wep64(1), then 64 bit WEP encryption will be
used with the keys from the wifiGw802dot11BssWep64BitKeyTable.
If set to wep128(2), then 128 bit WEP encryption will be used
with keys from the wifiGw802dot11BssWep128BitKeyTable.
"
::={ wifiGw802dot11BssWepEntry 2 }
wifiGw802dot11BssWepPassPhrase OBJECT-TYPE
SYNTAX DisplayString (SIZE(0..32))
MAX-ACCESS read-write
STATUS current
DESCRIPTION "The passphrase used for WEP security."
::= { wifiGw802dot11BssWepEntry 3 }

wifiGw802dot11BssWepSharedKeyAuthentication OBJECT-TYPE
SYNTAX WepAuthenticationMode
MAX-ACCESS read-write
STATUS current
DESCRIPTION "This object controls the use of Shared Key authentication in WEP protocol.
If this object is set to optional(1), Shared Key authentication is optional.
If set to required(2), Shared Key authentication is required for WEP."
::= { wifiGw802dot11BssWepEntry 4 }

wifiGw802dot11BssWep64BitKeyTable OBJECT-TYPE
SYNTAX SEQUENCE OF WifiGw802dot11BssWep64BitKeyEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "A table of 40 bit key values used when operating in WEP 64 mode."
::= { wifiGw802dot11MgmtBssSecurity 3 }

wifiGw802dot11BssWep64BitKeyEntry OBJECT-TYPE
SYNTAX WifiGw802dot11BssWep64BitKeyEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "A key to be used by the row in the wifiGw802dot11MgmtBssTable identified by the value of ifIndex in the row index."
INDEX { ifIndex, wifiGw802dot11BssWep64BitKeyIndex }
::= { wifiGw802dot11BssWep64BitKeyTable 1 }

WifiGw802dot11BssWep64BitKeyEntry ::= SEQUENCE {
  wifiGw802dot11BssWep64BitKeyIndex Integer32,
  wifiGw802dot11BssWep64BitKeyValue OCTET STRING,
  wifiGw802dot11BssWep64BitKeyStatus RowStatus
}

wifiGw802dot11BssWep64BitKeyIndex OBJECT-TYPE
SYNTAX Integer32 (1..4)
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "Identifies an instance of a key among those used by the service set identified by the value of ifIndex in the row index."
::= { wifiGw0802dot11BssWep64BitKeyEntry 1 }

wifiGw0802dot11BssWep64BitKeyValue OBJECT-TYPE
SYNTAX OCTET STRING (SIZE(5))
MAX-ACCESS read-create
STATUS current
DESCRIPTION "A 40 bit key to be used when the corresponding instance of
wifiGw0802dot11BssSecurityMode is set to wep(1) and the corresponding instance
of wifiGw0802dot11BssWepEncryptionMode is set to wep64(1)."
::= { wifiGw0802dot11BssWep64BitKeyEntry 2 }

wifiGw0802dot11BssWep64BitKeyStatus OBJECT-TYPE
SYNTAX RowStatus
MAX-ACCESS read-create
STATUS current
DESCRIPTION "The status of this conceptual row.

To create a row in this table, a manager must set this object to
either createAndGo(4) or createAndWait(5).

Until instances of all corresponding columns are appropriately
configured, the value of the corresponding instance of the
wifiGw0802dot11BssAccessStatus column is 'notReady'.

In particular, a newly created row cannot be made active until the
corresponding instance of wifiGw0802dot11BssWep64BitKeyValue has been set.

Only rows with a status of active(1) will be applied.

This object may not be set to a value of notInService(2) or destroy(6)
if the corresponding instance of wifiGw0802dot11BssWepDefaultKey identifies
this row as they current key in use. Such an attempt will be rejected
with an inconsistentValue error."
::= { wifiGw0802dot11BssWep64BitKeyEntry 3 }

wifiGw0802dot11BssWep128BitKeyTable OBJECT-TYPE
SYNTAX SEQUENCE OF WifiGw0802dot11BssWep128BitKeyEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "A table of 104 bit key values used when operating in WEP 128 mode."
::= { wifiGw0802dot11MgmtBssSecurity 4 }

wifiGw0802dot11BssWep128BitKeyEntry OBJECT-TYPE
SYNTAX WifiGw0802dot11BssWep128BitKeyEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "A key to be used by the row in the wifiGw0802dot11MgmtBssTable identified by the
value of ifIndex in the row index."
INDEX { ifIndex, wifiGw0802dot11BssWep128BitKeyIndex }
::= { wifiGw802dot11BssWep128BitKeyTable 1 }

WifiGw802dot11BssWep128BitKeyEntry ::=  
SEQUENCE    {
    wifiGw802dot11BssWep128BitKeyIndex           Integer32,
    wifiGw802dot11BssWep128BitKeyValue           OCTET STRING,
    wifiGw802dot11BssWep128BitKeyStatus          RowStatus
    }

wifiGw802dot11BssWep128BitKeyIndex OBJECT-TYPE
SYNTAX    Integer32 (1..4)
MAX-ACCESS not-accessible
STATUS    current
DESCRIPTION  
"Identifies an instance of a key among those used by the service set identified by the value of ifIndex in the row index."
::= { wifiGw802dot11BssWep128BitKeyEntry 1 }

wifiGw802dot11BssWep128BitKeyValue OBJECT-TYPE
SYNTAX    OCTET STRING (SIZE(13))
MAX-ACCESS read-create
STATUS    current
DESCRIPTION  
"A 104 bit key to be used when the corresponding instance of wifiGw802dot11BssSecurityMode is set to wep(1) and the corresponding instance of wifiGw802dot11BssWep1EncryptionMode is set to wep128(128)."
::= { wifiGw802dot11BssWep128BitKeyEntry 2 }

wifiGw802dot11BssWep128BitKeyStatus OBJECT-TYPE
SYNTAX    RowStatus
MAX-ACCESS read-create
STATUS    current
DESCRIPTION  
"The status of this conceptual row. To create a row in this table, a manager must set this object to either createAndGo(4) or createAndWait(5). Until instances of all corresponding columns are appropriately configured, the value of the corresponding instance of the wifiGw802dot11BssAccessStatus column is 'notReady'. In particular, a newly created row cannot be made active until the corresponding instance of wifiGw802dot11BssWep128BitKeyValue has been set. Only rows with a status of active(1) will be applied. This object may not be set to a value of notInService(2) or destroy(6) if the corresponding instance of wifiGw802dot11BssWepDefaultKey identifies this row as the current key in use. Such an attempt will be rejected with an inconsistentValue error."
::= { wifiGw802dot11BssWep128BitKeyEntry 3 }
-- Objects used to control WPA based security modes

wifiGw802dot11BssWpaPskTable OBJECT-TYPE
SYNTAX SEQUENCE OF WifiGw802dot11BssWpaPskEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "A table used to configure settings related to the WPA-PSK security modes."
 ::= { wifiGw802dot11MgmtBssSecurity 5 }

wifiGw802dot11BssWpaPskEntry OBJECT-TYPE
SYNTAX WifiGw802dot11BssWpaPskEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "An entry defining the WPA-PSK characteristics of an individual service set.
A row exists in this table for each row in the wifiGw802dot11MgmtBssTable for which
wifiGw802dot11BssSecurityMode is set to wpaPsK(2), wpa2PsK(3) or
wpaPskwpa2PsK(6)."
INDEX { ifIndex }
 ::= { wifiGw802dot11BssWpaPskTable 1 }

WifiGw802dot11BssWpaPskEntry ::= SEQUENCE {
   wifiGw802dot11BssWpaPskAlgorithm WpaAlgorithm,
   wifiGw802dot11BssWpaPskPreSharedKey OCTET STRING,
   wifiGw802dot11BssWpaPskGroupRekeyInterval Unsigned32
}

wifiGw802dot11BssWpaPskAlgorithm OBJECT-TYPE
SYNTAX WpaAlgorithm
MAX-ACCESS read-write
STATUS current
DESCRIPTION "Controls and reflects the WPA encryption mode used by the service set.

The WPA algorithm for WPA/WPA2-PSK and WPA/WPA2-Enterprise security mode is
either
tkip(1), aes(2) or tkipPlusAes(3)."
 ::= { wifiGw802dot11BssWpaPskEntry 1 }

wifiGw802dot11BssWpaPskPreSharedKey OBJECT-TYPE
SYNTAX OCTET STRING (SIZE (8..64))
MAX-ACCESS read-write
STATUS current
DESCRIPTION "Sets the WPA Pre-Shared Key (PSK) used by this service set. This value
MUST be either a 64 byte hexadecimal number, OR an 8 to 63 character ASCII string.
"
::= { wifiGw802dot11BssWpaPskEntry 2 }

wifiGw802dot11BssWpaPskGroupRekeyInterval OBJECT-TYPE
SYNTAX Unsigned32
UNITS "seconds"
MAX-ACCESS read-write
STATUS current
DESCRIPTION "Sets the WPA Group Rekey Interval for this service set. If set to zero, periodic rekeying is disabled for this service set."
::= { wifiGw802dot11BssWpaPskEntry 3 }

-- Objects used to control RADIUS based security modes

wifiGw802dot11BssWpaEnterpriseTable OBJECT-TYPE
SYNTAX SEQUENCE OF WifiGw802dot11BssWpaEnterpriseEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "A table used to configure settings related to the RADIUS security modes, including WPA-Enterprise, WPA2-Enterprise and RADIUS-WEP."
::= { wifiGw802dot11MgmtBssSecurity 6 }

wifiGw802dot11BssWpaEnterpriseEntry OBJECT-TYPE
SYNTAX WifiGw802dot11BssWpaEnterpriseEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "An entry defining the WPA-Enterprise characteristics of an individual service set. A row exists in this table for each row in the wifiGw802dot11MgmtBssTable for which wifiGw802dot11BssSecurityMode is set to wpaEnterprise(4), wpa2Enterprise(5), or wpaEnterpriseWpa2Enterprise(8)."
INDEX { ifIndex }
::= { wifiGw802dot11BssWpaEnterpriseTable 1 }

WifiGw802dot11BssWpaEnterpriseEntry ::= SEQUENCE {
  wifiGw802dot11BssWpaEnterpriseAlgorithm WpaAlgorithm,
  wifiGw802dot11BssWpaEnterpriseRadiusId Unsigned32,
  wifiGw802dot11BssWpaEnterpriseGroupRekeyInterval Unsigned32
}

wifiGw802dot11BssWpaEnterpriseAlgorithm OBJECT-TYPE
SYNTAX WpaAlgorithm
MAX-ACCESS read-write
STATUS current
DESCRIPTION "Controls and reflects the WPA encryption mode used by the service set."

The WPA algorithm for WPA/WPA2-PSK and WPA/WPA2-Enterprise security mode is either
tkip(1), aes(2) or tkipPlusAes(3).

::= { wifiGw802dot11BssWpaEnterpriseEntry 1 }

wifiGw802dot11BssWpaEnterpriseRadiusId OBJECT-TYPE
SYNTAX    Unsigned32 (0..65535)
MAX-ACCESS read-write
STATUS    current
DESCRIPTION
"Defines the reference link to the wifiGw802dot11MgmtRadiusTable, which defines the
parameters to contact a RADIUS server, needed in case the security mode is set to
WPA/WPA2-
Enterprise"
::= { wifiGw802dot11BssWpaEnterpriseEntry 2 }

wifiGw802dot11BssWpaEnterpriseGroupRekeyInterval OBJECT-TYPE
SYNTAX    Unsigned32 (0..86400)
UNITS      "seconds"
MAX-ACCESS read-write
STATUS    current
DESCRIPTION
"Sets the WPA Group Rekey Interval for this service set. If set to zero,
periodic rekeying is disabled for this service set."
::= { wifiGw802dot11BssWpaEnterpriseEntry 3 }

--
-- cmdot11anMbssAccess contains objects used to configure access restrictions
-- of each service set.
--
wifiGw802dot11BssAccessFilterTable  OBJECT-TYPE
SYNTAX    SEQUENCE OF WifiGw802dot11BssAccessFilterEntry
MAX-ACCESS not-accessible
STATUS    current
DESCRIPTION
"A table of MAC addresses which defines the MAC address based access
restrictions for the system."
::= { wifiGw802dot11MgmtBssSecurity 7 }

wifiGw802dot11BssAccessFilterEntry OBJECT-TYPE
SYNTAX    WifiGw802dot11BssAccessFilterEntry
MAX-ACCESS not-accessible
STATUS    current
DESCRIPTION
"A row in the table which specifies a single MAC address for a given
service set."
INDEX    { ifIndex, cmdot11anBssAccessIndex }
::= { wifiGw802dot11BssAccessFilterTable 1 }
WifiGw802dot11BssAccessFilterEntry ::=  
   SEQUENCE  {  
      wifiGw802dot11BssAccessIndex       Integer32,  
      wifiGw802dot11BssAccessStation     MacAddress,  
      wifiGw802dot11BssAccessStatus      RowStatus  
   }

wifiGw802dot11BssAccessIndex OBJECT-TYPE  
   SYNTAX          Integer32 (1..16)  
   MAX-ACCESS      not-accessible  
   STATUS          current  
   DESCRIPTION     "Indicates the instance of this table row."
   ::= { wifiGw802dot11BssAccessFilterEntry 1 }

wifiGw802dot11BssAccessStation OBJECT-TYPE  
   SYNTAX          MacAddress  
   MAX-ACCESS      read-create  
   STATUS          current  
   DESCRIPTION     "A MAC address of a station which will be allowed to connect to the service set if wifiGw802dot11BssAccessMode is set to restrictToList(2), or forbidden if wifiGw802dot11BssAccessMode is set to denyList(3)."
   ::= { wifiGw802dot11BssAccessFilterEntry 2 }

wifiGw802dot11BssAccessStatus OBJECT-TYPE  
   SYNTAX          RowStatus  
   MAX-ACCESS      read-create  
   STATUS          current  
   DESCRIPTION     "The status of this conceptual row. To create a row in this table, a manager must set this object to either createAndGo(4) or createAndWait(5). Until instances of all corresponding columns are appropriately configured, the value of the corresponding instance of the wifiGw802dot11BssAccessStatus column is 'notReady'. In particular, a newly created row cannot be made active until the corresponding instance of wifiGw802dot11BssAccessStation has been set. Only rows with a status of active(1) will be applied."
   ::= { wifiGw802dot11BssAccessFilterEntry 3 }

-- Objects used to control WMM parameters
wifiGw802dot11MgmtBssWmmTable OBJECT-TYPE  
   SYNTAX          SEQUENCE OF WifiGw802dot11MgmtBssWmmEntry  
   MAX-ACCESS      not-accessible  
   STATUS          current
DESCRIPTION

"A table used to configure settings related to WMM on a BSS, which are applicable if QoS is enabled on the BSS (wifiGw802dot11BssWmmEnable)."

::= { wifiGw802dot11MgmtMbss 3 }

wifiGw802dot11MgmtBssWmmEntry OBJECT-TYPE
SYNTAX WifiGw802dot11MgmtBssWmmEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "An entry defining the parameters of WMM on a BSS.

For each BSS (referenced by the ifIndex index), 4 Access Categories can be defined." INDEX { ifIndex, wifiGw802dot11MgmtBssWmmAcId }
::= { wifiGw802dot11MgmtBssWmmTable 1 }

WifiGw802dot11MgmtBssWmmEntry ::= SEQUENCE {
  wifiGw802dot11BssWmmAcId WmmAccessCategory,
  wifiGw802dot11BssWmmAIFSNU Int32,
  wifiGw802dot11BssWmmCWMin Int32,
  wifiGw802dot11BssWmmCWMax Int32,
  wifiGw802dot11BssWmmTXOP Int32
}

wifiGw802dot11BssWmmAcId OBJECT-TYPE
SYNTAX WmmAccessCategory
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "This key represents the identifier for a wifiGw802dot11MgmtBssWmmTable entry.

The enumerated values for the WMM Access Categories are:
- background(1) : AC(background)
- besteffort(2) : AC(best effort)
- video(3) : AC(video)
- voice(4) : AC(voice)"
::= { wifiGw802dot11MgmtBssWmmEntry 1 }

wifiGw802dot11BssWmmDscpClassifier OBJECT-TYPE
SYNTAX OCTET STRING
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "Defines the 6-bit value for the DSCP field that will be used to classify the packet to a specific WMM Access Category."
If a packet can't be classified (because the DSCP value is not defined in this table), the Wi-Fi Gateway must map it to the best effort access category.

The default value for this MIB object depends on the Access Category (wifiGw802dot11WmmAcId):
- '08'H for AC(background)
- '18'H for AC(best effort)
- '28'H for AC(video)
- '38'H for AC(voice)

::= { wifiGw802dot11MgmtBssWmmEntry 2 }

wifiGw802dot11BssWmmAIFSN OBJECT-TYPE
SYNTAX Unsigned32 (2..15)
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "The AIFSN subfield indicates the number of slots after a SIFS duration a STA should defer before either invoking a backoff or starting a transmission. The minimum value for the AIFSN field is 2.

The default value for this MIB object depends on the Access Category (wifiGw802dot11WmmAcId):
- 7 for AC(background)
- 3 for AC(best effort)
- 2 for AC(video)
- 2 for AC(voice)

::= { wifiGw802dot11MgmtBssWmmEntry 3 }

wifiGw802dot11BssWmmCwMin OBJECT-TYPE
SYNTAX Unsigned32 (0..15)
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "This field encodes the value of CWmin (minimum contention window), in an exponent form. The value is defined so that
CWmin = 2^wifiGw802dot11WmmCwMin - 1

Hence the minimum encoded value of CWmin is 0, and the maximum value is 32 767.

The default value for this MIB object depends on the Access Category (wifiGw802dot11WmmAcId):
- 5 for AC(background)
- 5 for AC(best effort)
- 4 for AC(video)
- 3 for AC(voice)

Note: the value of wifiGw802dot11WmmCwMax MUST be larger than the value for wifiGw802dot11WmmCwMin

::= { wifiGw802dot11MgmtBssWmmEntry 4 }
wifiGw802dot11BssWmmCwMax OBJECT-TYPE
SYNTAX Unsigned32 (0..15)
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "This field encodes the value of CWmax (maximum contention window), in an exponent form. The value is defined so that
\[ CWmax = 2^{\text{wifiGw802dot11WmmCwMax}} - 1 \]
Hence the minimum encoded value of CWmax is 0, and the maximum value is 32 767.

The default value for this MIB object depends on the Access Category (wifiGw802dot11WmmAcId):
- 10 for AC(background)
- 10 for AC(best effort)
- 5 for AC(video)
- 4 for AC(voice)

Note: the value of wifiGw802dot11WmmCwMax MUST be larger than the value for wifiGw802dot11WmmCwMin"
::= { wifiGw802dot11MgmtBssWmmEntry 5 }

wifiGw802dot11BssWmmTXOP OBJECT-TYPE
SYNTAX Unsigned32 (0..65535)
UNITS "32 us"
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "The value of the TXOP Limit field is specified as an unsigned integer, in units of 32 us.

A TXOP limit value of 0 indicates that the TXOP holder may transmit or cause to be transmitted (as responses) the following within the current TXOP:
a) A single MSDU, MMPDU, A-MSDU, A-MPDU, or PS-Poll at any rate, subject to the rules in 9.7
b) Any required acknowledgments
c) Any frames required for protection, including one of the following:
   1) An RTS/CTS exchange
   2) CTS to itself
   3) Dual CTS, if supported
d) Any frames required for beamforming, if supported
e) Any frames required for link adaptation, if supported
f) Any number of BlockAckReq frames

The default value for this MIB object depends on the Access Category (wifiGw802dot11WmmAcId):
- 0 for AC(background)
- 0 for AC(best effort)
- 94 for AC(video)
- 47 for AC(voice)"
::= { wifiGw802dot11MgmtBssWmmEntry 6 }

-- Objects used to control WPS parameters
wifiGw802dot11MgmtBssWpsTable OBJECT-TYPE
  SYNTAX  SEQUENCE OF WifiGw802dot11MgmtBssWpsEntry
  MAX-ACCESS  not-accessible
  STATUS    current
  DESCRIPTION
    "A table used to configure settings related to WPS on a BSS,"
  ::= { wifiGw802dot11MgmtMbss 4 }

wifiGw802dot11MgmtBssWpsEntry OBJECT-TYPE
  SYNTAX  WifiGw802dot11MgmtBssWpsEntry
  MAX-ACCESS  not-accessible
  STATUS    current
  DESCRIPTION
    "An entry defining the parameters of WPS on a BSS."
  INDEX  {  ifIndex  }
  ::= { wifiGw802dot11MgmtBssWpsTable 1 }

WifiGw802dot11MgmtBssWpsEntry ::= SEQUENCE  {
  wifiGw802dot11BssWpsEnable TruthValue,
  wifiGw802dot11BssWpsMethod WpsMethod,
  wifiGw802dot11BssWpsClientPin SnmpAdminString,
  wifiGw802dot11BssWpsApPin SnmpAdminString,
  wifiGw802dot11BssWpsAddClient TruthValue
  }

wifiGw802dot11BssWpsEnable OBJECT-TYPE
  SYNTAX  TruthValue
  MAX-ACCESS  read-write
  STATUS    current
  DESCRIPTION
    "Setting this object to true(1) enables the Wireless WPS feature on this BSS.
     Setting this object to false(2) disables the Wireless WPS feature on this BSS."
  ::= { wifiGw802dot11MgmtBssWpsEntry 1 }

wifiGw802dot11BssWpsMethod OBJECT-TYPE
  SYNTAX  WpsMethod
  MAX-ACCESS  read-write
  STATUS    current
  DESCRIPTION
    "WPS method may be choosen from either Push Button Configuration(1) or Personal
     Information Number(2)."
      if set to wpsPBC(1), user has to push a button, either an actual or virtual one,
       on both WPS devices.
      if set to wpsPIN(2), user has to input a wireless client's PIN on the WPS AP."
::= { wifiGw802dot11MgmtBssWpsEntry 2 }

wifiGw802dot11BssWpsClientPin OBJECT-TYPE
SYNTAX       SnmpAdminString (SIZE(8))
MAX-ACCESS   read-write
STATUS       current

DESCRIPTION
"Sets the WPS Client Pin to let it be able to register with the WPS AP. This must be ascii numerical characters"

::= { wifiGw802dot11MgmtBssWpsEntry 3 }

wifiGw802dot11BssWpsApPin OBJECT-TYPE
SYNTAX       SnmpAdminString
MAX-ACCESS   read-only
STATUS       current

DESCRIPTION
"This is wireless AP's PIN number. Users have to input the wireless AP's PIN number into the Registrar. Then the Registrar can configure AP's security successfully."

::= { wifiGw802dot11MgmtBssWpsEntry 4 }

wifiGw802dot11BssWpsAddClient OBJECT-TYPE
SYNTAX       TruthValue
MAX-ACCESS   read-write
STATUS       current

DESCRIPTION
"Setting this object to true(1) starts the Wireless WPS procedure on this BSS. Reading this object always returns false(2)."

::= { wifiGw802dot11MgmtBssWpsEntry 5 }

-- Objects used to control RADIUS Accounting parameters
wifiGw802dot11MgmtBssRadiusAccountingTable OBJECT-TYPE
SYNTAX       SEQUENCE OF WifiGw802dot11MgmtBssRadiusAccountingEntry
MAX-ACCESS   not-accessible
STATUS       current

DESCRIPTION
"A table used to configure settings related to RADIUS Accounting on a BSS,"

REFERENCE
"Wi-Fi Requirements for Cable Modem Gateways CEL-WR-SP-WIFI-GW-V0.3, section 6.4."

::= { wifiGw802dot11MgmtMbss 5 }

wifiGw802dot11MgmtBssRadiusAccountingEntry OBJECT-TYPE
SYNTAX       WifiGw802dot11MgmtBssRadiusAccountingEntry
MAX-ACCESS   not-accessible
STATUS       current

DESCRIPTION
"An entry defining the parameters of RADIUS Accounting on a BSS."

INDEX   {   ifIndex }
::= { wifiGw802dot11MgmtBssRadiusAccountingTable 1 }

WifiGw802dot11MgmtBssRadiusAccountingEntry ::=  
SEQUENCE { 
    wifiGw802dot11BssRadiusAccountingEnable TruthValue, 
    wifiGw802dot11BssRadiusAccountingSessionTimeout Unsigned32, 
    wifiGw802dot11BssRadiusAccountingRadiusId Unsigned32 
}

wifiGw802dot11BssRadiusAccountingEnable OBJECT-TYPE  
SYNTAX TruthValue  
MAX-ACCESS read-write  
STATUS current  
DESCRIPTION "Setting this object to true(1) enables RADIUS Accounting on this BSS.  
Setting this object to false(2) disables RADIUS Accounting on this BSS."
::= { wifiGw802dot11MgmtBssRadiusAccountingEntry 1 }

wifiGw802dot11BssRadiusAccountingSessionTimeout OBJECT-TYPE  
SYNTAX Unsigned32 (0..86400)  
UNITS "seconds"  
MAX-ACCESS read-write  
STATUS current  
DESCRIPTION "This MIB object defines the time interval, from the last packet seen from 
a client device, connected on this BSS, after which the Wi-Fi Gateway times  
out the session and concludes that the client device is disconnected and cleans  
up all resources, reserved for this client device."
DEFVAL { 300 }
::= { wifiGw802dot11MgmtBssRadiusAccountingEntry 2 }

wifiGw802dot11BssRadiusAccountingRadiusId OBJECT-TYPE  
SYNTAX Unsigned32  
MAX-ACCESS read-write  
STATUS current  
DESCRIPTION "Defines the reference link to the wifiGw802dot11MgmtRadiusTable, which defines  
the parameters to contact a RADIUS server, needed in case RADIUS Accounting is  
enabled for this BSS"
::= { wifiGw802dot11MgmtBssRadiusAccountingEntry 3 }

-- Objects used to control RADIUS server parameters

wifiGw802dot11MgmtRadiusTable OBJECT-TYPE  
SYNTAX SEQUENCE OF WifiGw802dot11MgmtRadiusEntry  
MAX-ACCESS not-accessible  
STATUS current  
DESCRIPTION "A table used to configure settings related to the RADIUS server,  
which can be used for authentication (BSS with security mode WPA/WPA2-
Enterprise) or accounting purposes.
"
 ::= { wifiGw802dot11MgmtMbss 6 }

wifiGw802dot11MgmtRadiusEntry OBJECT-TYPE
SYNTAX     WifiGw802dot11MgmtRadiusEntry
MAX-ACCESS not-accessible
STATUS     current
DESCRIPTION
"An entry defining the parameters of a RADIUS server."
INDEX     { wifiGw802dot11RadiusId }
 ::= { wifiGw802dot11MgmtRadiusTable 1 }

WifiGw802dot11MgmtRadiusEntry ::= SEQUENCE
{ wifiGw802dot11RadiusId Unsigned32,
  wifiGw802dot11RadiusAddressType InetAddressType,
  wifiGw802dot11RadiusAddress1 InetAddress,
  wifiGw802dot11RadiusAddress2 InetAddress,
  wifiGw802dot11RadiusPort1 InetPortNumber,
  wifiGw802dot11RadiusPort2 InetPortNumber,
  wifiGw802dot11RadiusKey1 DisplayString,
  wifiGw802dot11RadiusKey2 DisplayString,
  wifiGw802dot11RadiusLanRoutingEnabled TruthValue,
  wifiGw802dot11RadiusConnState RadiusConnectivityStatus
}

wifiGw802dot11RadiusId OBJECT-TYPE
SYNTAX     Unsigned32 (0..65535)
MAX-ACCESS not-accessible
STATUS     current
DESCRIPTION
"This key represents the identifier for a wifiGw802dot11MgmtRadiusTable entry."
 ::= { wifiGw802dot11MgmtRadiusEntry 1 }

wifiGw802dot11RadiusAddressType OBJECT-TYPE
SYNTAX     InetAddressType
MAX-ACCESS read-write
STATUS     current
DESCRIPTION
"The type of internet address used for wifiGw802dot11RadiusAddress1 and wifiGw802dot11RadiusAddress2."
 ::= { wifiGw802dot11MgmtRadiusEntry 2 }

wifiGw802dot11RadiusAddress1 OBJECT-TYPE
SYNTAX     InetAddress
MAX-ACCESS read-write
STATUS     current
DESCRIPTION
"The primary internet address of the RADIUS server for this service set.
The failover mechanism, that is used to decide when to switch from wifiGw802dot11RadiusAddress1 to wifiGw802dot11RadiusAddress1 and back, MUST be defined by the vendor.

The type of the address (IPv4|IPv6|FQDN) is defined by the wifiGw802dot11RadiusAddressType

::= { wifiGw802dot11MgmtRadiusEntry 3 }

wifiGw802dot11RadiusAddress2 OBJECT-TYPE
SYNTAX    InetAddress
MAX-ACCESS read-write
STATUS    current
DESCRIPTION
"The secondary internet address of the RADIUS server for this service set."

The failover mechanism, that is used to decide when to switch from wifiGw802dot11RadiusAddress1 to wifiGw802dot11RadiusAddress1 and back, MUST be defined by the vendor.

The type of the address (IPv4|IPv6|FQDN) is defined by the wifiGw802dot11RadiusAddressType

::= { wifiGw802dot11MgmtRadiusEntry 4 }

wifiGw802dot11RadiusPort1 OBJECT-TYPE
SYNTAX    InetPortNumber
MAX-ACCESS read-write
STATUS    current
DESCRIPTION
"The UDP port used to communicate with the RADIUS server, defined by wifiGw802dot11RadiusAddress1, for this service set."

DEFVAL { 1812 }
::= { wifiGw802dot11MgmtRadiusEntry 5 }

wifiGw802dot11RadiusPort2 OBJECT-TYPE
SYNTAX    InetPortNumber
MAX-ACCESS read-write
STATUS    current
DESCRIPTION
"The UDP port used to communicate with the RADIUS server, defined by wifiGw802dot11RadiusAddress2, for this service set."

DEFVAL { 1812 }
::= { wifiGw802dot11MgmtRadiusEntry 6 }

wifiGw802dot11RadiusKey1 OBJECT-TYPE
SYNTAX    DisplayString
MAX-ACCESS read-write
STATUS    current
DESCRIPTION

"The RADIUS key used to communicate with the RADIUS server, defined by wifiGw802dot11RadiusAddress1, for this service set."

::= { wifiGw802dot11MgmtRadiusEntry 7 }

wifiGw802dot11RadiusKey2 OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS read-write
STATUS current
DESCRIPTION

"The RADIUS key used to communicate with the RADIUS server, defined by wifiGw802dot11RadiusAddress2, for this service set."

::= { wifiGw802dot11MgmtRadiusEntry 8 }

wifiGw802dot11RadiusLanRoutingEnabled OBJECT-TYPE
SYNTAX TruthValue
MAX-ACCESS read-write
STATUS current
DESCRIPTION

"Defines whether the RADIUS traffic, generated by the Wi-Fi Gateway for this service set, is switched locally on the Wi-Fi Gateway or always sent in the upstream direction (RF)."

DEFVAL { false }

::= { wifiGw802dot11MgmtRadiusEntry 9 }

wifiGw802dot11RadiusConnState OBJECT-TYPE
SYNTAX RadiusConnectivityStatus
MAX-ACCESS read-write
STATUS current
DESCRIPTION

"Defines the connection status with the configured RADIUS server (through wifiGw802dot11RadiusAddress1 and, optionally, wifiGw802dot11RadiusAddress2). Possible states are:

- 'radiusPrimaryInUse' : primary RADIUS address is in use and has no connectivity issues
- 'radiusSecondaryInUse' : secondary RADIUS address is in use, because there were connectivity issues with the primary RADIUS address
- 'radiusUnreachable' : all configured RADIUS addresses are unreachable"

DEFVAL { false }

::= { wifiGw802dot11MgmtRadiusEntry 10 }

--

-- BSS backwards compatibility Parameters
--
wifiGw802dot11MgmtBssCompatibility \ OBJECT IDENTIFIER ::= 
    { wifiGw802dot11MgmtBss 7 }

-- wifiGw802dot11MgmtBssCompatibilityTable contains objects used to configure backwards compatibility parameters

wifiGw802dot11MgmtBssCompatibilityTable \ OBJECT-TYPE
SYNTAX \ SEQUENCE OF WifiGw802dot11MgmtBssCompatibilityEntry
MAX-ACCESS not-accessible
STATUS \ current
DESCRIPTION
"A table of entries to configure backwards compatibility parameters in a BSS."

REFERENCE
"Wi-Fi Requirements for Cable Modem Gateways CEL-WR-SP-WIFI-GW-V0.3, section 6.1.2."
::= { wifiGw802dot11MgmtBssCompatibility 1 }

WifiGw802dot11MgmtBssCompatibilityEntry \ OBJECT-TYPE
SYNTAX \ WifiGw802dot11MgmtBssCompatibilityEntry
MAX-ACCESS not-accessible
STATUS \ current
DESCRIPTION
"An entry describing the backwards compatibility parameters of an individual BSS. An
entry exists in this table for each entry of type ieee80211 (71) in the ifTable."

INDEX
{ ifIndex,
  wifiGw802dot11BssCompatibilityId }
::= { wifiGw802dot11MgmtBssCompatibilityTable 1 }

WifiGw802dot11MgmtBssCompatibilityEntry ::= \ SEQUENCE
{ wifiGw802dot11BssCompatibilityPointerId \ Unsigned32,
  wifiGw802dot11BssCompatibilityPointer \ RowPointer,
  wifiGw802dot11BssCompatibilityRowStatus \ RowStatus
}

wifiGw802dot11BssCompatibilityPointerId \ OBJECT-TYPE
SYNTAX \ Unsigned32 (0..65535)
MAX-ACCESS not-accessible
STATUS \ current
DESCRIPTION
"This key represents the identifier for a wifiGw802dot11BssCompatibilityPointer entry, which points
to another table (i.e. wifiGw802dot11BssCompNetModeTable or wifiGw802dot11BssCompSnrTable)"
::= { wifiGw802dot11MgmtBssCompatibilityEntry 1 }

wifiGw802dot11BssCompatibilityPointer \ OBJECT-TYPE
SYNTAX \ RowPointer
MAX-ACCESS read-create
STATUS \ current
DESCRIPTION
"A pointer to an entry in a compatibility configuration table. e.g
wifiGw802dot11BssNetModeRowStatus in wifiGw802dot11BssCompNetModeEntry or
wifiGw802dot11BssSnrRowStatus in wifiGw802dot11BssCompSnrEntry.

A value pointing to zeroDotZero, an inactive Row or a
non-existing entry is treated as no compatibility configuration defined for
this
entry."
DEFVAL {zeroDotZero }
::= { wifiGw802dot11MgmtBssCompatibilityEntry 2 }

wifiGw802dot11BssCompatibilityRowStatus OBJECT-TYPE
SYNTAX     RowStatus
MAX-ACCESS read-create
STATUS     current
DESCRIPTION
"This object is to create or delete rows in this table."
::= { wifiGw802dot11MgmtBssCompatibilityEntry 3 }

-- BSS backwards compatibility (netmode) Parameters
wifiGw802dot11MgmtBssCompatibilityNetModeTable   OBJECT-TYPE
SYNTAX      SEQUENCE OF WifiGw802dot11MgmtBssCompatibilityNetModeEntry
MAX-ACCESS not-accessible
STATUS      current
DESCRIPTION
"A table of entries to configure the netmode in a BSS."
""
REFERENCE
"Wi-Fi Requirements for Cable Modem Gateways CEL-SP-WIFI-GW-V0.3,
section 6.1.2."
::= { wifiGw802dot11MgmtBssCompatibility 2 }

WifiGw802dot11MgmtBssCompatibilityNetModeEntry OBJECT-TYPE
SYNTAX      WifiGw802dot11MgmtBssCompatibilityNetModeEntry
MAX-ACCESS not-accessible
STATUS      current
DESCRIPTION
"An entry describing the netmode of a BSS."
INDEX   { wifiGw802dot11BssCompatibilityNetModeId }
::= { wifiGw802dot11MgmtBssCompatibilityNetModeTable 1 }

WifiGw802dot11MgmtBssCompatibilityNetModeEntry ::= SEQUENCE    {
    wifiGw802dot11BssCompatibilityNetModeId      Unsigned32,
    wifiGw802dot11BssCompatibilityNetModeRowStatus RowStatus,
    wifiGw802dot11BssCompatibilityNetMode        CompatibilityNetMode
}

wifiGw802dot11BssCompatibilityNetModeId OBJECT-TYPE
SYNTAX     Unsigned32 (0..65535)
MAX-ACCESS not-accessible
STATUS     current
DESCRIPTION
"This key represents the identifier for a wifiGw802dot11MgmtBssCompatibilityNetModeEntry entry."
::= { wifiGw802dot11MgmtBssCompatibilityNetModeEntry 1 }

wifiGw802dot11BssCompatibilityNetModeRowStatus OBJECT-TYPE
  SYNTAX     RowStatus
  MAX-ACCESS read-create
  STATUS     current
  DESCRIPTION
    "This object is to create or delete rows in this table."
  ::= { wifiGw802dot11MgmtBssCompatibilityNetModeEntry 2 }

wifiGw802dot11BssCompatibilityNetMode OBJECT-TYPE
  SYNTAX     CompatibilityNetMode
  MAX-ACCESS read-create
  STATUS     current
  DESCRIPTION
    "The backwards compatibility netmode that the Wi-Fi Gateway supports
    modeB(1)   : The AP behaves as an 802.11b (2.4GHz only) AP and only supports
                802.11b features.
    modeBG(2)  : The AP behaves as an 802.11g (2.4GHz only) AP and is backwards
                compatible with 802.11b clients.
    modeG(3)   : The AP behaves as an 802.11g (2.4GHz only) AP and does not allow
                legacy 802.11b clients.
    modeGN(4)  : The AP behaves as an 802.11n (2.4GHz only) AP and is backwards
                compatible with 802.11g clients, but does not allow 802.11b clients.
    modeBGN(5): The AP behaves as an 802.11n (2.4GHz only) AP and is backwards
                compatible with 802.11b/g clients
    modeA(6)   : The AP behaves as an 802.11a (5GHz only) AP and only supports
                802.11a features
    modeAN(7)  : The AP behaves as an 802.11n (5GHz only) AP and is backwards
                compatible with 802.11a clients
    modeAG(9)  : The AP behaves as an 802.11a/802.11g (2.4GHz or 5GHz) AP and does
                not allow legacy 802.11b clients
    modeAGN(10): The AP behaves as an 802.11n (2.4GHz or 5GHz) AP and is backwards
                compatible with 802.11a/g clients, but does not allow legacy 802.11b clients
    modeABGN(11): The AP behaves as an 802.11n (2.4GHz or 5GHz) AP and is backwards
                  compatible with all legacy clients
    modeN(12)  : The AP behaves as an 802.11n (2.4GHz or 5GHz) AP and does not
                  allow any legacy client
    "
  DEFVAL { modeABGN }
  ::= { wifiGw802dot11MgmtBssCompatibilityNetModeEntry 3 }

-- BSS backwards compatibility (minimum SNR) Parameters
wifiGw802dot11MgmtBssCompatibilitySnrTable OBJECT-TYPE
  SYNTAX     SEQUENCE OF WifiGw802dot11MgmtBssCompatibilitySnrEntry
  MAX-ACCESS not-accessible
  STATUS     current
  DESCRIPTION
    "A table of entries to configure the minimum SNR in a BSS."
  ::= { wifiGw802dot11MgmtBssCompatibility 3 }
wifiGw802dot11MgmtBssCompatibilitySnrEntry OBJECT-TYPE
SYNTAX     WifiGw802dot11MgmtBssCompatibilitySnrEntry
MAX-ACCESS not-accessible
STATUS     current
DESCRIPTION     "An entry describing the minimum SNR parameters of a BSS."
INDEX   { wifiGw802dot11BssCompatibilitySnrId }
::= { wifiGw802dot11MgmtBssCompatibilitySnrTable 1 }

WifiGw802dot11MgmtBssCompatibilitySnrEntry ::= SEQUENCE    {
  wifiGw802dot11BssCompatibilitySnrId Unsigned32,
  wifiGw802dot11BssCompatibilitySnrRowStatus RowStatus,
  wifiGw802dot11BssCompatibilitySnrMin TenthdB,
  wifiGw802dot11BssCompatibilitySnrAction SnrAction
}

wifiGw802dot11BssCompatibilitySnrId OBJECT-TYPE
SYNTAX     Unsigned32 (0..65535)
MAX-ACCESS not-accessible
STATUS     current
DESCRIPTION     "This key represents the identifier for a
wifiGw802dot11MgmtBssCompatibilitySnrEntry entry."
::= { wifiGw802dot11MgmtBssCompatibilitySnrEntry 1 }

wifiGw802dot11BssCompatibilitySnrRowStatus OBJECT-TYPE
SYNTAX     RowStatus
MAX-ACCESS read-create
STATUS     current
DESCRIPTION     "This object is to create or delete rows in this table."
::= { wifiGw802dot11MgmtBssCompatibilitySnrEntry 2 }

wifiGw802dot11BssCompatibilitySnrMin OBJECT-TYPE
SYNTAX     TenthdB
MAX-ACCESS read-create
STATUS     current
DESCRIPTION     "This object defines the minimum SNR threshold for a client device. When the SNR of a client device drops below this threshold, the Wi-Fi Gateway MUST perform the action, defined by the wifiGw802dot11BssCompatibilitySnrAction object."
::= { wifiGw802dot11MgmtBssCompatibilitySnrEntry 3 }

wifiGw802dot11BssCompatibilitySnrAction OBJECT-TYPE
SYNTAX     SnrAction
MAX-ACCESS read-create
STATUS     current
DESCRIPTION     "This object defines the action that MUST be performed when the SNR of a client device drops below the defined threshold (wifiGw802dot11BssCompatibilitySnrMin)."
Possible actions are:
- noaction(1) : don't take any action (this is the only mandatory supported action)
- denyassociation(2) : don't allow the client device to associate
- disassociate(3) : send a disassociate message to the client device

DEFVAL { noaction }
 ::= { wifiGw802dot11MgmtBssCompatibilitySnrEntry 4 }

--
-- 802.11 Channel survey MIB objects/tables
--

wifiGw802dot11RadioChannelSurveyTable OBJECT-TYPE
  SYNTAX SEQUENCE OF WifiGw802dot11RadioChannelSurveyEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "This table is used to configure channel status surveys."
  ::= { wifiGw802dot11RadioChannelMonitorObjects 1 }

wifiGw802dot11RadioChannelSurveyEntry OBJECT-TYPE
  SYNTAX WifiGw802dot11RadioChannelSurveyEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "Channel survey configuration entry. An entry is available for each 802.11 radio interface
    on the Wi-Fi Gateway."
  INDEX { ifIndex }
  ::= { wifiGw802dot11RadioChannelSurveyTable 1 }

WifiGw802dot11RadioChannelSurveyEntry ::= SEQUENCE
  {
    wifiGw802dot11RadioChannelSurveyMode ChannelSurveyMode,
    wifiGw802dot11RadioChannelSurveyInterval Unsigned32,
    wifiGw802dot11RadioChannelSurveyInitiate TruthValue,
    wifiGw802dot11RadioChannelSurveyLastRunTimestamp DateTime
  }

wifiGw802dot11RadioChannelSurveyMode OBJECT-TYPE
  SYNTAX ChannelSurveyMode
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "This defines the 802.11 channel survey mode for this radio interface
    Possible modes are:
    - manual(1) : the user has to manual start the channel survey by setting
      the correct MIB object (wifiGw802dot11RadioChannelSurveyInitiate)
- periodic(2) : the Wi-Fi Gateway periodically (based on wifiGw802dot11RadioChannelSurveyInterval) schedules a channel survey run.

DEFVAL { manual }
::= { wifiGw802dot11RadioChannelSurveyEntry 1 }

wifiGw802dot11RadioChannelSurveyInterval OBJECT-TYPE
SYNTAX Unsigned32 (0..1440)
UNITS "minutes"
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"This defines the interval between 2 periodically scheduled 802.11 channel survey runs on this radio interface"
DEFVAL { 60 }
::= { wifiGw802dot11RadioChannelSurveyEntry 2 }

wifiGw802dot11RadioChannelSurveyInitiate OBJECT-TYPE
SYNTAX TruthValue
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"Setting this object to true(1) starts a channel survey on this radio interface if wifiGw802dot11RadioChannelSurveyMode is set to 'manual'.

Reading this object always returns false(2).
"
::= { wifiGw802dot11RadioChannelSurveyEntry 3 }

wifiGw802dot11RadioChannelSurveyLastRunTimestamp OBJECT-TYPE
SYNTAX DateAndTime
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The date and time of the last finished channel survey run on this radio interface.
"
::= { wifiGw802dot11RadioChannelSurveyEntry 4 }

-- Channel Survey Results Table
wifiGw802dot11RadioChannelSurveyResultTable OBJECT-TYPE
SYNTAX SEQUENCE OF WifiGw802dot11RadioChannelSurveyResultEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"This table contains the results of the last channel survey per radio interface."
**INDEX**

```text
INDEX { ifIndex,
      wifiGw802dot11RadioChannelSurveyResultId }
::= { wifiGw802dot11RadioChannelSurveyResultTable 1 }
```

**WifiGw802dot11RadioChannelSurveyResultEntry ::=**

```text
SEQUENCE {
  wifiGw802dot11RadioChannelSurveyResultId Unsigned32,
  wifiGw802dot11RadioChannelSurveyResultChannel Unsigned32,
  wifiGw802dot11RadioChannelSurveyResultSsid OCTET STRING,
  wifiGw802dot11RadioChannelSurveyResultBssid MacAddress,
  wifiGw802dot11RadioChannelSurveyResultRssi TenthdB,
  wifiGw802dot11RadioChannelSurveyResultWifiVersion RadioWifiVersion,
  wifiGw802dot11RadioChannelSurveyResultBandwidth RadioBandWidth,
  wifiGw802dot11RadioChannelSurveyResultSidebandPosition RadioSideBand
}
```

**wifiGw802dot11RadioChannelSurveyResultId OBJECT-TYPE**

```text
SYNTAX Unsigned32
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "The index for a unique result in the wifiGw802dot11RadioChannelSurveyResultTable"
::= { wifiGw802dot11RadioChannelSurveyResultEntry 1 }
```

**wifiGw802dot11RadioChannelSurveyResultChannel OBJECT-TYPE**

```text
SYNTAX Unsigned32 (1..165)
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The radio channel on which this result entry is measured"
::= { wifiGw802dot11RadioChannelSurveyResultEntry 2 }
```

**wifiGw802dot11RadioChannelSurveyResultSsid OBJECT-TYPE**

```text
SYNTAX OCTET STRING (SIZE(0..32))
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The service set identifier associated with this service set"
::= { wifiGw802dot11RadioChannelSurveyResultEntry 3 }
```

**wifiGw802dot11RadioChannelSurveyResultBssid OBJECT-TYPE**

```text
SYNTAX MacAddress
MAX-ACCESS read-only
```
The physical address associated with this service set
::= { wifiGw802dot11RadioChannelSurveyResultEntry 4 }

The RSSI of the measured service set
::= { wifiGw802dot11RadioChannelSurveyResultEntry 5 }

The Wi-Fi version of the measured service set
::= { wifiGw802dot11RadioChannelSurveyResultEntry 6 }

The bandwidth of the measured service set
::= { wifiGw802dot11RadioChannelSurveyResultEntry 7 }

The position of the sideband in case the bandwidth of the measured service set is 40 MHz.

Note: In case the bandwidth of the measured service set is 20 MHz, this value is chosen arbitrary.
::= { wifiGw802dot11RadioChannelSurveyResultEntry 8 }

--
-- Wi-Fi Gateway connected clients info

-- wifiGw802dot11BssClientTable OBJECT-TYPE
   SYNTAX SEQUENCE OF WifiGw802dot11BssClientEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "This table contains a list of WiFi clients which associated to the Wi-Fi Gateway, per BSS."

 ::= { wifiGw802dot11BssClientObjects 1 }

wifiGw802dot11BssClientEntry OBJECT-TYPE
   SYNTAX WifiGw802dot11BssClientEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Wi-Fi connected clients entry."
   INDEX { ifIndex, wifiGw802dot11BssClientIndex }
 ::= { wifiGw802dot11BssClientTable 1 }

WifiGw802dot11BssClientEntry ::= SEQUENCE
   {
      wifiGw802dot11BssClientIndex Unsigned32,
      wifiGw802dot11BssClientMacAddress MacAddress,
      wifiGw802dot11BssClientRssi TenthdB,
      wifiGw802dot11BssClientSessionStartTimestamp DateAndTime,
      wifiGw802dot11BssClientSessionTimeout Unsigned32,
      wifiGw802dot11BssClientConnectionState ClientConnectionStatus,
      wifiGw802dot11BssClientWifiVersion RadioWifiVersion,
      wifiGw802dot11BssClientSecurityMode BssSecurityMode,
      wifiGw802dot11BssClientWpaAlgorithm WpaAlgorithm
   }

wifiGw802dot11BssClientIndex OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "The index for the Wi-Fi client Table."
 ::= { wifiGw802dot11BssClientEntry 1 }

wifiGw802dot11BssClientMacAddress OBJECT-TYPE
   SYNTAX MacAddress
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "MAC address of the Wi-Fi client."
 ::= { wifiGw802dot11BssClientEntry 2 }

wifiGw802dot11BssClientRssi OBJECT-TYPE
   SYNTAX TenthdB
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The last measured RSSI of the Wi-Fi client."
"
::= { wifiGw802dot11BssClientEntry 3 }

wifiGw802dot11BssClientSessionStartTimestamp OBJECT-TYPE
SYNTAX DateAndTime
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The date and time of the association of the Wi-Fi client."
"
::= { wifiGw802dot11BssClientEntry 4 }

wifiGw802dot11BssClientSessionTimeout OBJECT-TYPE
SYNTAX Unsigned32 (0..86400)
UNITS "seconds"
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The time interval, from the last packet seen from this Wi-Fi client, after which the Wi-Fi Gateway times out the session and concludes that the client device is disconnected and cleans up all resources, reserved for this client device."
"
::= { wifiGw802dot11BssClientEntry 5 }

wifiGw802dot11BssClientConnectionState OBJECT-TYPE
SYNTAX ClientConnectionStatus
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The current status of the Wi-Fi client
Possible states are:
- connected(1) : client device is connected
- clientDisassociated(2) : client device has sent Disassociate Message
- forcedDisassociatedAuth(3) : client device is disassociated because something went wrong during the security initialization
- forcedDisassociatedTimeout(4) : client device is disassociated because session timed out
- forcedDisassociatedNetMode(5) : client device is disassociated because it was a legacy device which was not supported
- forcedDisassociatedSnr(6) : client device is disassociated because SNR was below threshold
- other(7) : client device is an unspecified other state"
"
::= { wifiGw802dot11BssClientEntry 6 }

wifiGw802dot11BssClientWifiVersion OBJECT-TYPE
SYNTAX RadioWifiVersion
MAX-ACCESS read-only
STATUS current
DESCRIPTION
""
The Wi-Fi version of the Wi-Fi client

::= { wifiGw802dot11BssClientEntry 7 }

wifiGw802dot11BssClientSecurityMode OBJECT-TYPE
SYNTAX BssSecurityMode
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The security mode that is used by the Wi-Fi client.

Possible modes are:
- disabled(0)
  - wep(1)
  - wpaPsk(2)
  - wpa2Psk(3)
  - wpaEnterprise(4)
  - wpa2Enterprise(5)

::= { wifiGw802dot11BssClientEntry 8 }

wifiGw802dot11BssClientWpaAlgorithm OBJECT-TYPE
SYNTAX WpaAlgorithm
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The WPA/WPA2 encryption algorithm that is used by the Wi-Fi client.

Possible algorithms are:
- tkip(0)
  - aes(1)

::= { wifiGw802dot11BssClientEntry 9 }

END

7.3 Forwarding and Classification Objects

WIFI-GW-FWD-MIB DEFINITIONS ::= BEGIN
IMPORTS
 NOTIFICATION-TYPE,
 MODULE-IDENTITY,
 OBJECT-TYPE,
 Integer32,
 Unsigned32,
 Counter32,
 enterprises
 FROM SNMPv2-SMI -- RFC 2578
 TEXTUAL-CONVENTION,
 RowStatus,
 TruthValue,
 MacAddress,
RowStatus,
RowPointer,
DateAndTime,
DisplayString
FROM SNMPv2-TC -- RFC 2579
OBJECT-GROUP,
NOTIFICATION-GROUP,
MODULE-COMPLIENCE
FROM SNMPv2-CONF -- RFC 2580
InetAddressType,
InetAddress,
InetAddressDNS,
InetPortNumber,
InetAddressPrefixLength
FROM INET-ADDRESS-MIB -- RFC 4001
ifIndex
FROM IF-MIB; -- RFC 2863

--
-- Path to root
--

euroCableLabs OBJECT IDENTIFIER ::= { enterprises 24624 }
eclProject OBJECT IDENTIFIER ::= { euroCableLabs 2 }
eclProjWifiGateway OBJECT IDENTIFIER ::= { eclProject 3 }

--
-- Wi-Fi Gateway Forwarding Module
--

wifiGwFwd MODULE-IDENTITY
LAST-UPDATED "201211140000Z" -- November 14, 2012
ORGANIZATION "EuroCableLabs"
CONTACT-INFO
"Editor: Volker Leisse
Postal: EuroCableLabs
Avenue des Arts 36
1040 Brussels
Belgium
Phone: +49 531 391-2478
Fax: +49 531 391-5192
E-mail: mib@eurocablelabs.com"
DESCRIPTION
"This MIB module contains the management objects for the
management of classification and forwarding mechanisms in
the Wi-Fi Gateway devices.

Copyright 2012 EuroCableLabs
All rights reserved."
::= { eclProjWifiGateway 3 }

--
TunnelConnectivityState ::= TEXTUAL-CONVENTION
   STATUS current
   DESCRIPTION
   "This data type defines the connectivity state of a GRE Tunnel Endpoint,
configured
   on the Wi-Fi Gateway.

   The enumerated values associated with the TunnelConnectivityState are:

   'reachable' : indicates that the configured GRE Tunnel Endpoint is in use and
   the Wi-Fi Gateway
does not have any connectivity issues with it.
   'unreachable' : indicates that the configured GRE Tunnel Endpoint was in use
and the Wi-Fi Gateway
   has detected a connectivity issue. If a secondary tunnel endpoint is
defined, the
   Wi-Fi Gateway will try to switch to that tunnel endpoint. If no
   secondary endpoint
   is defined, or the secondary endpoint is unreachable, the Wi-Fi
   Gateway will try to
   re-establish connectivity on this tunnel endpoint.
   'notinuse' : indicates that the GRE Tunnel Endpoint was not configured."
SYNTAX INTEGER { reachable(1),
   unreachable(2),
   notinuse(3) }

--
-- Wi-Fi Gateway MIB Objects
--

wifiGwFwdMibObjects OBJECT IDENTIFIER ::= { wifiGwFwd 1}
wifiGwFwdNotification OBJECT IDENTIFIER ::= { wifiGwFwd 2}
wifiGwFwdConformance OBJECT IDENTIFIER ::= { wifiGwFwd 3}
wifiGwFwdCompliances OBJECT IDENTIFIER ::= { wifiGwFwdConformance 1}
wifiGwFwdGroups OBJECT IDENTIFIER ::= { wifiGwFwdConformance 2}

--
-- Wi-Fi Gateway Classification and Forwarding MIB Objects
--

-- Wi-Fi Gateway Classification Configuration Table

wifiGwClassifierTable OBJECT-TYPE
SYNTAX SEQUENCE OF WifiGwClassifierEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
  "This table defines the (upstream) classifiers that are active on this Wi-Fi
Gateway.
   Each classifier links to a policy where the forwarding parameters are defined for
   the classified (upstream) traffic."
REFERENCE
"Wi-Fi Requirements for Cable Modem Gateways CEL-WR-SP-WIFI-GW-V0.3, section 6.2.1.1."
 ::= { wifiGwPwdMibObjects 1 }

wifiGwClassifierEntry OBJECT-TYPE
 SYNTAX WifiGwClassifierEntry
 MAX-ACCESS not-accessible
 STATUS current
 DESCRIPTION "Objects to define a classifier on the Wi-Fi Gateway"
 INDEX { wifiGwClassifierId }
 ::= { wifiGwClassifierTable 1 }

WifiGwClassifierEntry ::= SEQUENCE {
 wifiGwClassifierId Unsigned32,
 wifiGwClassifierPriority Unsigned32,
 wifiGwClassifierPolicyId Unsigned32,
 wifiGwClassifierRowStatus RowStatus
 }

wifiGwClassifierId OBJECT-TYPE
 SYNTAX Unsigned32 (0..65535)
 MAX-ACCESS not-accessible
 STATUS current
 DESCRIPTION "This key represents the identifier for a wifiGwClassifierTable entry."
 ::= { wifiGwClassifierEntry 1 }

wifiGwClassifierPriority OBJECT-TYPE
 SYNTAX Unsigned32 (0..7)
 MAX-ACCESS read-create
 STATUS current
 DESCRIPTION "This value represents the priority of a classifier. If a packet matches multiple
 classifiers, configured in the wifiGwClassifierTable, the classifier with the highest priority is chosen. If a packet matches multiple
 classifiers and they all share the same priority, the classifier with the lowest wifiGwClassifierId is chosen."
 ::= { wifiGwClassifierEntry 2 }

wifiGwClassifierPolicyId OBJECT-TYPE
 SYNTAX Unsigned32 (0..65535)
 MAX-ACCESS read-create
 STATUS current
 DESCRIPTION "This key represents the link between this classifier and a policy, defined in
 the wifiGwPolicyTable"
 ::= { wifiGwClassifierEntry 3 }

wifiGwClassifierRowStatus OBJECT-TYPE
 SYNTAX RowStatus
 MAX-ACCESS read-create
This object is to create or delete rows in this table.

 ::= { wifiGwClassifierEntry 4 }

-- Wi-Fi Gateway Classifier Attributes Link Table

wifiGwClassifierAttributesLinkTable OBJECT-TYPE
SYNTAX SEQUENCE OF WifiGwClassifierAttributesLinkEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"This table links to specific classifier attributes tables and
is referenced by the wifiGwClassifierTable to link to these classifier
attributes."

 ::= { wifiGwFdMibObjects 2 }

wifiGwClassifierAttributesLinkEntry OBJECT-TYPE
SYNTAX WifiGwClassifierAttributesLinkEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"Objects to configure links to classifier attributes"
INDEX { wifiGwClassifierId, wifiGwClassifierAttributesPointerId }
 ::= { wifiGwClassifierAttributesLinkTable 1 }

WifiGwClassifierAttributesLinkEntry ::= SEQUENCE {
    wifiGwClassifierAttributesPointerId Unsigned32,
    wifiGwClassifierAttributesPointer RowPointer,
    wifiGwClassifierAttributesLinkRowStatus RowStatus
}

wifiGwClassifierAttributesPointerId OBJECT-TYPE
SYNTAX Unsigned32 (0..65535)
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"This key represents the identifier for a wifiGwClassifierAttributesPointer
entry, which points
to another table (i.e. wifiGwClassifierAttributesTable)"
 ::= { wifiGwClassifierAttributesLinkEntry 1 }

wifiGwClassifierAttributesPointer OBJECT-TYPE
SYNTAX RowPointer
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"A pointer to an entry in a classifier attribute configuration table. e.g
 wifiGwClassifierAttributesRowStatus in wifiGwClassifierAttributesEntry.
A value pointing to zeroDotZero, an inactive Row or a
non-existing entry is treated as no attributes defined for this

© Cable Europe Labs - CONFIDENTIAL
entry."
DEFVAL {zeroDotZero }
::= { wifiGwClassifierAttributesLinkEntry 2 }

wifiGwClassifierAttributesLinkRowStatus OBJECT-TYPE
SYNTAX RowStatus
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"This object is to create or delete rows in this table."
::= { wifiGwClassifierAttributesLinkEntry 3 }

-- Wi-Fi Gateway Classifier Attributes Table
wifiGwClassifierAttributesTable OBJECT-TYPE
SYNTAX SEQUENCE OF WifiGwClassifierAttributesEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"This table configures the classifier attributes."
::= { wifiGwFwdMibObjects 3 }

wifiGwClassifierAttributesEntry OBJECT-TYPE
SYNTAX WifiGwClassifierAttributesEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"Objects to configure the classifier attributes."
INDEX { wifiGwClassifierAttributesId }
::= { wifiGwClassifierAttributesTable 1 }

WifiGwClassifierAttributesEntry ::= SEQUENCE {
  wifiGwClassifierAttributesId Unsigned32,
  wifiGwClassifierAttributesRowStatus RowStatus,
  wifiGwClassifierIfIndex Unsigned32,
  wifiGwClassifierSourceMac MacAddress,
  wifiGwClassifierSourceMacMask MacAddress,
  wifiGwClassifierDestinationMac MacAddress,
  wifiGwClassifierDestinationMacMask MacAddress,
  wifiGwClassifierIpAddressType InetAddressType,
  wifiGwClassifierSourceIpAddress InetAddress,
  wifiGwClassifierSourceIpPrefixLength InetAddressPrefixLength,
  wifiGwClassifierDestinationIpAddress InetAddress,
  wifiGwClassifierDestinationIpPrefixLength InetAddressPrefixLength,
  wifiGwClassifierIpProtocol Unsigned32,
  wifiGwClassifierSourcePortNumberStart InetPortNumber,
  wifiGwClassifierSourcePortNumberEnd InetPortNumber,
  wifiGwClassifierDestinationPortNumberStart InetPortNumber,
  wifiGwClassifierDestinationPortNumberEnd InetPortNumber
}

wifiGwClassifierAttributesId OBJECT-TYPE
SYNTAX Unsigned32 (0..65535)
MAX-ACCESS not-accessible
STATUS     current
DESCRIPTION
   "This key represents the identifier for this entry in the classifier attributes
table."
::= { wifiGwClassifierAttributesEntry 1 }

wifiGwClassifierAttributesRowStatus OBJECT-TYPE
SYNTAX     RowStatus
MAX-ACCESS read-create
STATUS     current
DESCRIPTION
   "This object is to create or delete rows in this table."
::= { wifiGwClassifierAttributesEntry 2 }

wifiGwClassifierIfIndex OBJECT-TYPE
SYNTAX     Unsigned32 (0..65535)
MAX-ACCESS read-create
STATUS     current
DESCRIPTION
   "A packet matches an entry when it is received on the interface where the ifIndex
equals wifiGwClassifierIfIndex.

   Possible interfaces include all Ethernet interfaces, all wireless radio
interfaces,
   all SSIDs and the eRouter interface"
::= { wifiGwClassifierAttributesEntry 3 }

wifiGwClassifierSourceMac OBJECT-TYPE
SYNTAX     MacAddress
MAX-ACCESS read-create
STATUS     current
DESCRIPTION
   "An Ethernet packet matches an entry when its
source MAC address bitwise ANDed with
wifiGwClassifierSourceMacMask equals the value of
wifiGwClassifierSourceMac.

   If the referenced parameter is not present
in a classifier, this object reports the value of
'000000000000'H."
::= { wifiGwClassifierAttributesEntry 4 }

wifiGwClassifierSourceMacMask OBJECT-TYPE
SYNTAX     MacAddress
MAX-ACCESS read-create
STATUS     current
DESCRIPTION
   "An Ethernet packet matches an entry when its
source MAC address bitwise ANDed with
wifiGwClassifierSourceMacMask equals the value of
wifiGwClassifierSourceMac.

   If the referenced parameter is not present
in a classifier, this object reports the value of
'000000000000'H.

 ::= { wifiGwClassifierAttributesEntry 5 }

wifiGwClassifierDestinationMac OBJECT-TYPE
SYNTAX   MacAddress
MAX-ACCESS read-create
STATUS   current
DESCRIPTION "An Ethernet packet matches an entry when its
destination MAC address bitwise ANDed with
wifiGwClassifierDestinationMacMask equals the value of
wifiGwClassifierDestinationMac.

If the referenced parameter is not present
in a classifier, this object reports the value of
'000000000000'H.
 ::= { wifiGwClassifierAttributesEntry 6 }

wifiGwClassifierDestinationMacMask OBJECT-TYPE
SYNTAX   MacAddress
MAX-ACCESS read-create
STATUS   current
DESCRIPTION "An Ethernet packet matches an entry when its
destination MAC address bitwise ANDed with
wifiGwClassifierDestinationMacMask equals the value of
wifiGwClassifierDestinationMac.

If the referenced parameter is not present
in a classifier, this object reports the value of
'000000000000'H.
 ::= { wifiGwClassifierAttributesEntry 7 }

wifiGwClassifierIpAddressType OBJECT-TYPE
SYNTAX   InetAddressType
MAX-ACCESS read-create
STATUS   current
DESCRIPTION "The address type for the source/destination IP address.

The allowed enumerated values associated with the InetAddressType are:

'ipv4' (1)
indicates that the wifiGwClassifierSourceIpAddress and
wifiGwClassifierDestinationIpAddress entries will be of type InetAddressIPv4.
'ipv6' (2)
indicates that the wifiGwClassifierSourceIpAddress and
wifiGwClassifierDestinationIpAddress entries will be of type InetAddressIPv6.

 ::= { wifiGwClassifierAttributesEntry 8 }

wifiGwClassifierSourceIpAddress OBJECT-TYPE
SYNTAX   InetAddress
MAX-ACCESS read-create
STATUS    current
DESCRIPTION
 "This object specifies the value of the IP
 Source Address required for packets to
 match this rule. An IP packet matches the rule
 when the IP version equals the wifiGwClassifierIpAddressType
 and when the wifiGwClassifierSourceIpPrefixLength most significant
 bits of the packet IP source address equals the
 wifiGwClassifierSourceIpPrefixLength value.

If the referenced parameter is not present
in a classifier, this object reports the value of
0.0.0.0 (in case wifiGwClassifierIpAddressType equals ipv4(1))
or 0000:0000:0000:0000:0000:0000 (in case wifiGwClassifierIpAddressType
equals ipv6(2))
::= { wifiGwClassifierAttributesEntry 9 }

wifiGwClassifierSourceIpPrefixLength OBJECT-TYPE
SYNTAX     InetAddressPrefixLength
MAX-ACCESS read-create
STATUS     current
DESCRIPTION
 "An IP packet matches the rule when the IP version
 equals the wifiGwClassifierIpAddressType and when
 the wifiGwClassifierSourceIpPrefixLength most significant
 bits of the packet IP source address equals the
 wifiGwClassifierSourceIpPrefixLength value.

If the referenced parameter is not present
in a classifier, this object reports the value of
0"
::= { wifiGwClassifierAttributesEntry 10 }

wifiGwClassifierDestinationIpAddress OBJECT-TYPE
SYNTAX     InetAddress
MAX-ACCESS read-create
STATUS     current
DESCRIPTION
 "This object specifies the value of the IP
 Destination Address required for packets to
 match this rule. An IP packet matches the rule
 when the IP version equals the wifiGwClassifierIpAddressType
 and when the wifiGwClassifierDestinationIpPrefixLength most significant
 bits of the packet IP destination address equals the
 wifiGwClassifierDestinationIpPrefixLength value.

If the referenced parameter is not present
in a classifier, this object reports the value of
0.0.0.0.0 (in case wifiGwClassifierIpAddressType equals ipv4(1))
or 0000:0000:0000:0000:0000:0000 (in case wifiGwClassifierIpAddressType
equals ipv6(2))
::= { wifiGwClassifierAttributesEntry 11 }

wifiGwClassifierDestinationIpPrefixLength OBJECT-TYPE
SYNTAX InetAddressPrefixLength
MAX-ACCESS read-create
STATUS current

DESCRIPTION
"An IP packet matches the rule when the IP version equals the wifiGwClassifierIpAddressType and when the wifiGwClassifierDestinationIpPrefixLength most significant bits of the packet IP destination address equals the wifiGwClassifierDestinationIpAddress value.

If the referenced parameter is not present in a classifier, this object reports the value of 0"

::= { wifiGwClassifierAttributesEntry 12 }

wifiGwClassifierIpProtocol OBJECT-TYPE
SYNTAX Unsigned32 (0..258)
MAX-ACCESS read-create
STATUS current

DESCRIPTION
"This object indicates the value of the IP Protocol field required for IP packets to match this rule.

The value 256 matches traffic with any IP Protocol value. The value 257 by convention matches both TCP and UDP.

If the referenced parameter is not present in a classifier, this object reports the value of 258"

::= { wifiGwClassifierAttributesEntry 13 }

wifiGwClassifierSourcePortNumberStart OBJECT-TYPE
SYNTAX InetPortNumber
MAX-ACCESS read-create
STATUS current

DESCRIPTION
"This object specifies the low end inclusive range of TCP/UDP source port numbers to which a packet is compared. This object is irrelevant for non-TCP/UDP IP packets.

If the referenced parameter is not present in a classifier, this object reports the value of 0."

::= { wifiGwClassifierAttributesEntry 14 }

wifiGwClassifierSourcePortNumberEnd OBJECT-TYPE
SYNTAX InetPortNumber
MAX-ACCESS read-create
STATUS current

DESCRIPTION
"This object specifies the high end inclusive range of TCP/UDP source port numbers to which a packet is compared. This object is irrelevant
for non-TCP/UDP IP packets.

If the referenced parameter is not present
in a classifier, this object reports the value of 65535."
::= { wifiGwClassifierAttributesEntry 15 }

wifiGwClassifierDestinationPortNumberStart OBJECT-TYPE
SYNTAX     InetPortNumber
MAX-ACCESS read-create
STATUS     current
DESCRIPTION
"This object specifies the low end inclusive
range of TCP/UDP destination port numbers to which
a packet is compared. This object is irrelevant
for non-TCP/UDP IP packets.

If the referenced parameter is not present
in a classifier, this object reports the value of 0."  
::= { wifiGwClassifierAttributesEntry 16 }

wifiGwClassifierDestinationPortNumberEnd OBJECT-TYPE
SYNTAX     InetPortNumber
MAX-ACCESS read-create
STATUS     current
DESCRIPTION
"This object specifies the high end inclusive
range of TCP/UDP destination port numbers to which
a packet is compared. This object is irrelevant
for non-TCP/UDP IP packets.

If the referenced parameter is not present
in a classifier, this object reports the value of 65535."  
::= { wifiGwClassifierAttributesEntry 17 }

-- Wi-Fi Gateway Policy Configuration Table
wifiGwPolicyTable OBJECT-TYPE
SYNTAX     SEQUENCE OF WifiGwPolicyEntry
MAX-ACCESS not-accessible
STATUS     current
DESCRIPTION
"This table defines the (upstream) forwarding policies that are active on this Wi-Fi Gateway.

Each policy defines the forwarding parameters which are defined for (upstream) traffic that is
classified, according to the classifiers that link to this table.

By default, a single row in this table must be present, which has the
wifiGwPolicyIsDefault value equal
true (1). The attributes of this default policy can be changed, but trying to
destroy this row must return an error.

All other rows, created by the user, must have the wifiGwPolicyIsDefault value
equal false (2).
REFERENCE
"Wi-Fi Requirements for Cable Modem Gateways CEL-WR-SP-WIFI-GW-V0.3,
section 6.2.1.2."
 ::= { wifiGwFwdMibObjects 4 }

wifiGwPolicyEntry OBJECT-TYPE
SYNTAX WifiGwPolicyEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "Objects to define a forwarding policy on the Wi-Fi Gateway"
INDEX { wifiGwPolicyId }
 ::= { wifiGwPolicyTable 1 }

WifiGwPolicyEntry ::= SEQUENCE {
 wifiGwPolicyId Unsigned32,
 wifiGwPolicyIsDefault TruthValue,
 wifiGwPolicyRowStatus RowStatus
}

wifiGwPolicyId OBJECT-TYPE
SYNTAX Unsigned32 (0..65535)
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "This key represents the identifier for a wifiGwPolicyTable entry."
 ::= { wifiGwPolicyEntry 1 }

wifiGwPolicyIsDefault OBJECT-TYPE
SYNTAX TruthValue
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This value indicates if this policy is the default one.
The default policy is the policy that is chosen if the packet
does not match any classifier.

By default, a single row in this table must be present, which has the
wifiGwPolicyIsDefault value equal
to (1). The attributes of this default policy can be changed, but trying to
destroy this row must return an error.

All rows, created by the user, must have a wifiGwPolicyIsDefault value which
equals false (2) and this attribute
must be read-only.
"
DEFVAL { false }
 ::= { wifiGwPolicyEntry 2 }

wifiGwPolicyRowStatus OBJECT-TYPE
SYNTAX RowStatus
MAX-ACCESS read-create
STATUS current
DESCRIPTION

"This object is to create or delete rows in this table.

By default, a single row in this table must be present, which has the
wifiGwPolicyIsDefault value equal
to true (1). Trying to destroy this row must return an error.

Other policy rows can be created/destroyed by the user.
"
 ::= { wifiGwPolicyEntry 3 }

-- Wi-Fi Gateway Policy Attributes Link Table
wifiGwPolicyAttributesLinkTable OBJECT-TYPE
SYNTAX       SEQUENCE OF WifiGwPolicyAttributesLinkEntry
MAX-ACCESS   not-accessible
STATUS       current
DESCRIPTION
 "This table links to specific forwarding policy attributes tables and
 is referenced by the wifiGwPolicyTable to link to these policy attributes.
"
 ::= { wifiGwFwdMibObjects 5 }

wifiGwPolicyAttributesLinkEntry OBJECT-TYPE
SYNTAX       WifiGwPolicyAttributesLinkEntry
MAX-ACCESS   not-accessible
STATUS       current
DESCRIPTION
 "Objects to configure links to policy attributes"
INDEX { wifiGwPolicyId,
         wifiGwPolicyAttributesPointerId }
 ::= { wifiGwPolicyAttributesLinkTable 1 }

WifiGwPolicyAttributesLinkEntry ::= SEQUENCE {
  wifiGwPolicyAttributesPointerId Unsigned32,
  wifiGwPolicyAttributesPointer RowPointer,
  wifiGwPolicyAttributesLinkRowStatus RowStatus
}

wifiGwPolicyAttributesPointerId OBJECT-TYPE
SYNTAX       Unsigned32 (0..65535)
MAX-ACCESS   not-accessible
STATUS       current
DESCRIPTION
 "This key represents the identifier for a wifiGwPolicyAttributesPointer entry,
 which points
to another table (i.e. wifiGwPolicyAttributesTable)"
 ::= { wifiGwPolicyAttributesLinkEntry 1 }

wifiGwPolicyAttributesPointer OBJECT-TYPE
SYNTAX       RowPointer
MAX-ACCESS   read-create
STATUS       current
DESCRIPTION
 "A pointer to an entry in a policy attribute configuration table. e.g
wifiGwPolicyAttributesRowStatus in wifiGwPolicyAttributesEntry.

A value pointing to zeroDotZero, an inactive Row or a non-existing entry is treated as no attributes defined for this entry."
DEFVAL {zeroDotZero }
 ::= { wifiGwPolicyAttributesLinkEntry 2 }

wifiGwPolicyAttributesLinkRowStatus OBJECT-TYPE
SYNTAX    RowStatus
MAX-ACCESS read-create
STATUS     current
DESCRIPTION "This object is to create or delete rows in this table."
 ::= { wifiGwPolicyAttributesLinkEntry 3 }

-- Wi-Fi Gateway Policy Attributes Table
wifiGwPolicyAttributesTable OBJECT-TYPE
SYNTAX     SEQUENCE OF WifiGwPolicyAttributesEntry
MAX-ACCESS not-accessible
STATUS     current
DESCRIPTION "This table configures the policy attributes."
 ::= { wifiGwFwdMibObjects 6 }

wifiGwPolicyAttributesEntry OBJECT-TYPE
SYNTAX     WifiGwPolicyAttributesEntry
MAX-ACCESS not-accessible
STATUS     current
DESCRIPTION "Objects to configure the forwarding policy attributes."
INDEX { wifiGwPolicyAttributesId }
 ::= { wifiGwPolicyAttributesTable 1 }

WifiGwPolicyAttributesEntry ::= SEQUENCE {
    wifiGwPolicyAttributesId Unsigned32,
    wifiGwPolicyAttributesRowStatus RowStatus,
    wifiGwPolicyGreTunnelEndpointId Unsigned32,
    wifiGwPolicyVlanId Unsigned32,
    wifiGwPolicyVlanPriority Unsigned32,
    wifiGwPolicyDscpSet OCTET STRING,
    wifiGwPolicyIpv6FlowLabelSet Unsigned32,
    wifiGwPolicyDrop TruthValue
}

wifiGwPolicyAttributesId OBJECT-TYPE
SYNTAX    Unsigned32 (0..65535)
MAX-ACCESS not-accessible
STATUS     current
DESCRIPTION "This key represents the identifier for this entry in the policy attributes table."
 ::= { wifiGwPolicyAttributesEntry 1 }
wifiGwPolicyAttributesRowStatus OBJECT-TYPE
SYNTAX           RowStatus
MAX-ACCESS       read-create
STATUS           current
DESCRIPTION      "This object is to create or delete rows in this table."
 ::= { wifiGwPolicyAttributesEntry 2 }

wifiGwPolicyGreTunnelEndpointId OBJECT-TYPE
SYNTAX           Unsigned32 (0..65535)
MAX-ACCESS       read-create
STATUS           current
DESCRIPTION      "This key represents the link to a GRE encapsulation configuration
table, defined in the wifiGwGreConfigurationTable
If this object is set to the value of 0, there is no
matching GRE configuration, and wifiGwGreConfigurationTable
MUST NOT be consulted."
DEFVAL { 0 }
 ::= { wifiGwPolicyAttributesEntry 3 }

wifiGwPolicyVlanId OBJECT-TYPE
SYNTAX           Unsigned32 (0..4095)
MAX-ACCESS       read-create
STATUS           current
DESCRIPTION      "This value defines the 12-bit value for the VLAN identifier,
in case 802.1Q tagging is performed on packets, assigned to
this policy.
If this object is set to the value of 0, no 802.1Q tagging
is performed on packets, assigned to this policy."
DEFVAL { 0 }
 ::= { wifiGwPolicyAttributesEntry 4 }

wifiGwPolicyVlanPriority OBJECT-TYPE
SYNTAX           Unsigned32 (0..7)
MAX-ACCESS       read-create
STATUS           current
DESCRIPTION      "This value defines 3-bit value for the Priority Code Point,
in case 802.1Q tagging is performed on packets, assigned to
this policy."
DEFVAL { 0 }
 ::= { wifiGwPolicyAttributesEntry 5 }

wifiGwPolicyDscpSet OBJECT-TYPE
SYNTAX           OCTET STRING
MAX-ACCESS       read-create
STATUS           current
DESCRIPTION      "Defines the 6-bit value for the DSCP field that will be over-
written for each IPv4/IPv6 packet, assigned to this policy.

In case encapsulation techniques (e.g. GRE) are defined for this policy, the Traffic Class field will be set in the outer IP header.

This attribute can not be disabled, so the (outer) IP header is always overwritten for packets, assigned to this policy."

DEFVAL { '00'H }
 ::= { wifiGwPolicyAttributesEntry 6 }

wifiGwPolicyIpv6FlowLabelSet OBJECT-TYPE
 SYNTAX     Unsigned32 (0..1048575)
 MAX-ACCESS read-create
 STATUS     current
 DESCRIPTION
 "Defines the 20-bit value for the Flow Label field that will be overwritten for each IPv6 packet, assigned to this policy.

In case encapsulation techniques (e.g. GRE) are defined for this policy, the Flow Label will be set in the outer IPv6 header.

This attribute can not be disabled, so the (outer) IPv6 header is always overwritten for packets, assigned to this policy."

DEFVAL { 0 }
 ::= { wifiGwPolicyAttributesEntry 7 }

wifiGwPolicyDrop OBJECT-TYPE
 SYNTAX     TruthValue
 MAX-ACCESS read-create
 STATUS     current
 DESCRIPTION
 "Defines wether packets, classified for this policy are dropped by the Wi-Fi Gateway.

A value of true indicates that all packets, assigned to this policy, are dropped. In this case, all other parameters of this policy are ignored."

DEFVAL { false }
 ::= { wifiGwPolicyAttributesEntry 8 }

-- Wi-Fi Gateway GRE Configuration Table
wifiGwGreConfigurationTable OBJECT-TYPE
 SYNTAX      SEQUENCE OF WifiGwGreConfigurationEntry
 MAX-ACCESS  not-accessible
 STATUS      current
 DESCRIPTION
 "This table configures GRE Tunnel Endpoints."

REFERENCE
 "Wi-Fi Requirements for Cable Modem Gateways CEL-WR-SP-WIFI-GW-V0.3, section 6.2.1.2.1."
 ::= { wifiGwFwdMibObjects 7 }
wifiGwGreConfigurationEntry OBJECT-TYPE
  SYNTAX      WifiGwGreConfigurationEntry
  MAX-ACCESS  not-accessible
  STATUS      current
  DESCRIPTION
    "Objects to configure the GRE Tunnel Endpoint attributes."
  INDEX { wifiGwGreTunnelEndpointId }
  ::= { wifiGwGreConfigurationTable 1 }

WifiGwGreConfigurationEntry ::= SEQUENCE {
  wifiGwGreTunnelEndpointId          Unsigned32,
  wifiGwGreTunnelEndpointRowStatus   RowStatus,
  wifiGwGreTunnelEndpointAddressType InetAddressType,
  wifiGwGreTunnelEndpointAddress1    InetAddress,
  wifiGwGreTunnelEndpointAddress2    InetAddress,
  wifiGwGreFragmentationEnabled      TruthValue,
  wifiGwGreFailoverEnabled           TruthValue,
  wifiGwGreKeepaliveEnabled          TruthValue,
  wifiGwGreKeepaliveInterval         Unsigned32,
  wifiGwGreReviveInterval            Unsigned32,
  wifiGwGreReviveRetries             Unsigned32,
  wifiGwGreSecretKey                 DisplayString,
  wifiGwGreTunnelEndpointConnectivityState1 TunnelConnectivityState,
  wifiGwGreTunnelEndpointConnectivityState2 TunnelConnectivityState
}

wifiGwGreTunnelEndpointId OBJECT-TYPE
  SYNTAX     Unsigned32 (1..65535)
  MAX-ACCESS not-accessible
  STATUS     current
  DESCRIPTION
    "This key represents the identifier for this entry in the GRE configuration table.

    The value 0 is not allowed as index for this table."
  ::= { wifiGwGreConfigurationEntry 1 }

wifiGwGreTunnelEndpointRowStatus OBJECT-TYPE
  SYNTAX     RowStatus
  MAX-ACCESS read-create
  STATUS     current
  DESCRIPTION
    "This object is to create or delete rows in this table."
  ::= { wifiGwGreConfigurationEntry 2 }

wifiGwGreTunnelEndpointAddressType OBJECT-TYPE
  SYNTAX     InetAddressType
  MAX-ACCESS read-create
  STATUS     current
  DESCRIPTION
    "The address type for the GRE Tunnel Endpoint, configured through this entry.

    The allowed enumerated values associated with the InetAddressType are:
'ipv4' (1) : indicates that the wifiGwGreTunnelEndpointAddress1 and wifiGwGreTunnelEndpointAddress2 entries will be of type InetAddressIPv4.

'ipv6' (2) : indicates that the wifiGwGreTunnelEndpointAddress1 and wifiGwGreTunnelEndpointAddress2 entries will be of type InetAddressIPv6.

dns' (16) : indicates that the wifiGwGreTunnelEndpointAddress1 and wifiGwGreTunnelEndpointAddress2 entries will be of type InetAddressDNS.

::= { wifiGwGreConfigurationEntry 3 }

wifiGwGreTunnelEndpointAddress1 OBJECT-TYPE
SYNTAX InetAddress
MAX-ACCESS read-create
STATUS current
DESCRIPTION "This object defines the primary address for the GRE Tunnel Endpoint, configured through this entry.

The type of the address (IPv4|IPv6|FQDN) is defined by the wifiGwGreTunnelEndpointAddressType.

::= { wifiGwGreConfigurationEntry 4 }

wifiGwGreTunnelEndpointAddress2 OBJECT-TYPE
SYNTAX InetAddress
MAX-ACCESS read-create
STATUS current
DESCRIPTION "This object defines an (optional) secondary address for the GRE Tunnel Endpoint, configured through this entry. This address will only be used if it is defined, wifiGwGreFailoverEnabled is set to 'true' and the Wi-Fi Gateway detects that the address, defined in wifiGwGreTunnelEndpointAddress1 is unreachable.

The type of the address (IPv4|IPv6|FQDN) is defined by the wifiGwGreTunnelEndpointAddressType.

::= { wifiGwGreConfigurationEntry 5 }

wifiGwGreFragmentationEnabled OBJECT-TYPE
SYNTAX TruthValue
MAX-ACCESS read-create
STATUS current
DESCRIPTION "This object defines whether the Wi-Fi Gateway must support fragmentation for packets that are sent to the GRE Tunnel Endpoint, configured through this entry.

If this object is set to 'true' and the GRE-encapsulated packet exceeds the upstream MTU,
the Wi-Fi Gateway MUST fragment the GRE-encapsulated packet and support reassembly of received downstream fragmented packets from the GRE Tunnel Endpoint, configured through this entry.

If this object is set to 'false' and the GRE-encapsulated packet exceeds the upstream MTU, the Wi-Fi Gateway MUST drop the packet.

DEFVAL { true }
 ::= { wifiGwGreConfigurationEntry 6 }

wifiGwGreFailoverEnabled OBJECT-TYPE
SYNTAX     TruthValue
MAX-ACCESS read-create
STATUS     current
DESCRIPTION
 "This object defines whether the Wi-Fi Gateway must support switching to the other GRE Tunnel Endpoint address if the GRE Tunnel Endpoint address, currently in use, becomes unreachable (e.g. to wifiGwGreTunnelEndpointAddress2 if wifiGwGreTunnelEndpointAddress1 becomes unreachable and to wifiGwGreTunnelEndpointAddress1 if wifiGwGreTunnelEndpointAddress2 becomes unreachable).

The failover detection mechanism is defined by the vendor and can be based on the keepalive mechanism, defined in this table or based on ICMP-Unreachable messages for the configured GRE Tunnel Endpoint.

This failover mechanism is only enabled if this object is set to 'true' and both wifiGwGreTunnelEndpointAddress1 and wifiGwGreTunnelEndpointAddress2 are configured."
DEFVAL { false }
 ::= { wifiGwGreConfigurationEntry 7 }

wifiGwGreKeepaliveEnabled OBJECT-TYPE
SYNTAX     TruthValue
MAX-ACCESS read-create
STATUS     current
DESCRIPTION
 "This object defines whether the Wi-Fi Gateway must support a keepalive mechanism for the configured GRE Tunnel Endpoint, in case no downstream traffic is received for an extensible amount of time (equal to the wifiGwGreKeepaliveInterval value).

The keepalive mechanism is only enabled if this object is set to 'true' and the wifiGwGreKeepaliveInterval does not equal '0'."
DEFVAL { false }
 ::= { wifiGwGreConfigurationEntry 8 }

wifiGwGreKeepaliveInterval OBJECT-TYPE
SYNTAX     Unsigned32 (0..86400)
This object defines the interval between keepalive messages, generated from the Wi-Fi Gateway and destined for the configured GRE Tunnel Endpoint. This object also defines the interval, after which the first keepalive message is sent. This interval is measured from the arrival of the last downstream GRE-encapsulated message, received from the configured GRE Tunnel Endpoint.

A value '0' disables the keepalive mechanism.

```
DEFVAL { 300 }
 ::= { wifiGwGreConfigurationEntry 9 }
```

This object defines how many times the Wi-Fi Gateway must try to revive the connectivity with the configured GRE Tunnel Endpoint after the Wi-Fi Gateway detects that the configured GRE Tunnel Endpoint is unreachable. The unreachable detection mechanism is defined by the vendor and can be based on the keepalive mechanism, defined in this table or based on ICMP-Unreachable messages for the configured GRE Tunnel Endpoint.

```
DEFVAL { 30 }
 ::= { wifiGwGreConfigurationEntry 10 }
```

This object defines the secret key that the Wi-Fi Gateway must include in
intercepted DHCP messages from connected client devices, before forwarding them through the configured GRE Tunnel.

The secret key must be inserted in the DHCP messages according to the corresponding entry in the wifiGwDhcpAttributesLinkTable.

If the DHCP message is coming from an interface that has no DHCP intercept parameters defined or DHCP intercept is disabled (wifiGwDhcpInterceptConfigurationTable), the Wi-Fi Gateway must not insert this secret key.

```
::= { wifiGwGreConfigurationEntry 12 }

wifiGwGreTunnelEndpointConnectivityState1 OBJECT-TYPE
SYNTAX TunnelConnectivityState
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"This object displays the current state of the configured GRE Tunnel Endpoint (wifiGwGreTunnelEndpointAddress1).

The enumerated values associated with the TunnelConnectivityState are:

'reachable' : indicates that the configured GRE Tunnel Endpoint is in use and the Wi-Fi Gateway does not have any connectivity issues with it.
'unreachable' : indicates that the configured GRE Tunnel Endpoint was in use and the Wi-Fi Gateway has detected a connectivity issue. If a secondary tunnel endpoint is defined, the Wi-Fi Gateway will try to switch to that tunnel endpoint. If no secondary endpoint is defined, or the secondary endpoint is unreachable, the Wi-Fi Gateway will try to re-establish connectivity on this tunnel endpoint.

'notinuse' : indicates that the GRE Tunnel Endpoint was not configured."
::= { wifiGwGreConfigurationEntry 13 }

wifiGwGreTunnelEndpointConnectivityState2 OBJECT-TYPE
SYNTAX TunnelConnectivityState
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"This object displays the current state of the configured GRE Tunnel Endpoint (wifiGwGreTunnelEndpointAddress2).

The enumerated values associated with the TunnelConnectivityState are:

'reachable' : indicates that the configured GRE Tunnel Endpoint is in use and the Wi-Fi Gateway does not have any connectivity issues with it.
'unreachable' : indicates that the configured GRE Tunnel Endpoint was in use and the Wi-Fi Gateway has detected a connectivity issue. If a secondary tunnel endpoint is defined, the
Wi-Fi Gateway will try to switch to that tunnel endpoint. If no secondary endpoint is defined, or the secondary endpoint is unreachable, the Wi-Fi Gateway will try to re-establish connectivity on this tunnel endpoint.

'notinuse' : indicates that the GRE Tunnel Endpoint was not configured.

::= { wifiGwGreConfigurationEntry 14 }