Internet Engineering Task Force (IETF)
Request for Comments: 5834
Category: Informational
ISSN: 2070-1721

Y. Shi, Ed.
Hangzhou H3C Tech. Co., Ltd.
D. Perkins, Ed.
C. Elliott, Ed.

Y. Zhang, Ed.
Fortinet, Inc.

May 2010

Control and Provisioning of Wireless Access Points (CAPWAP) Protocol Binding MIB for IEEE 802.11

Abstract

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols. In particular, it describes managed objects for modeling the Control And Provisioning of Wireless Access Points (CAPWAP) protocol for IEEE 802.11 wireless binding. This MIB module is presented as a basis for future work on the management of the CAPWAP protocol using the Simple Network Management Protocol (SNMP).

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc5834.

Table of Contents

1. Introduction

The CAPWAP protocol [RFC5415] defines a standard, interoperable protocol, which enables an Access Controller (AC) to manage a collection of Wireless Termination Points (WTPs). CAPWAP supports the use of various wireless technologies by the WTPs, with one specified in the CAPWAP Protocol Binding for IEEE 802.11 [RFC5416].

This document defines a MIB module that can be used to manage CAPWAP implementations for IEEE 802.11 wireless binding. This MIB module covers both configuration for Wireless Local Area Network (WLAN) and a way to reuse the IEEE 802.11 MIB module [IEEE.802-11.2007]. It is presented as a basis for future work on the SNMP management of the CAPWAP protocol.

2. The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC 3410 [RFC3410].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This memo specifies a MIB module that is compliant to the SMIv2, which is described in STD 58, RFC 2578 [RFC2578], STD 58, RFC 2579 [RFC2579], and STD 58, RFC 2580 [RFC2580].

3. Terminology

This document uses terminology from the CAPWAP protocol specification [RFC5415], the CAPWAP Protocol Binding for IEEE 802.11 [RFC5416], and the CAPWAP Protocol Base MIB [RFC5833].

Access Controller (AC): The network entity that provides WTP access to the network infrastructure in the data plane, control plane, management plane, or a combination therein.

Wireless Termination Point (WTP): The physical or network entity that contains an RF antenna and wireless physical layer (PHY) to transmit and receive station traffic for wireless access networks.

Control And Provisioning of Wireless Access Points (CAPWAP): It is a generic protocol defining AC and WTP control and data plane communication via a CAPWAP protocol transport mechanism. CAPWAP control messages, and optionally CAPWAP data messages, are secured using Datagram Transport Layer Security (DTLS) [RFC4347].

CAPWAP Control Channel: A bi-directional flow defined by the AC IP Address, WTP IP Address, AC control port, WTP control port, and the transport-layer protocol (UDP or UDP-Lite) over which CAPWAP control packets are sent and received.

CAPWAP Data Channel: A bi-directional flow defined by the AC IP Address, WTP IP Address, AC data port, WTP data port, and the transport-layer protocol (UDP or UDP-Lite) over which CAPWAP data packets are sent and received.

Station (STA): A device that contains an interface to a wireless medium (WM).

Split and Local MAC: The CAPWAP protocol supports two modes of operation: Split and Local MAC (medium access control). In Split MAC mode, all Layer 2 wireless data and management frames are encapsulated via the CAPWAP protocol and exchanged between the AC and the WTPs. The Local MAC mode of operation allows the data frames to be either locally bridged or tunneled as 802.3 frames.

Wireless Binding: The CAPWAP protocol is independent of a specific WTP radio technology, as well its associated wireless link layer protocol. Elements of the CAPWAP protocol are designed to accommodate the specific needs of each wireless technology in a standard way. Implementation of the CAPWAP protocol for a particular wireless technology MUST define a binding protocol for it, e.g., the binding for IEEE 802.11, provided in [RFC5416].

Wireless Local Area Network (WLAN): A WLAN refers to a logical component instantiated on a WTP device. A single physical WTP MAY operate a number of WLANs. Each Basic Service Set Identifier (BSSID) and its constituent wireless terminal radios are denoted as a distinct WLAN on a physical WTP. To support a physical WTP with multiple WLANs is an important feature for CAPWAP protocol's 802.11 binding, and it is also for MIB module design.

Wireless Binding MIB Module: Other Standards Development Organizations (SDOs), such as IEEE, already defined MIB modules for specific wireless technologies, e.g., the IEEE 802.11 MIB module [IEEE.802-11.2007]. Such MIB modules are called wireless binding MIB modules.

CAPWAP Protocol Wireless Binding MIB Module: It is a MIB module corresponding to the CAPWAP Protocol Binding for a wireless binding. Sometimes, not all the technology-specific message elements in a CAPWAP binding protocol have MIB objects defined by other SDOs. For example, the protocol of [RFC5416] defines WLAN conception. Also, Local or Split MAC modes could be specified for a WLAN. The MAC mode for a WLAN is not in the scope of IEEE 802.11 [IEEE.802-11.2007]. In such cases, in addition to the existing wireless binding MIB modules defined by other SDOs, a CAPWAP protocol wireless binding MIB module is required to be defined for a wireless binding.

4. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

5. Overview

5.1. WLAN Profile

A WLAN profile stores configuration parameters such as MAC type and tunnel mode for a WLAN. Each WLAN profile is identified by a profile identifier. The operator needs to create WLAN profiles before WTPs connect to the AC. To provide WLAN service, the operator SHOULD bind WLAN profiles to a WTP Virtual Radio Interface that corresponds to a PHY radio. During the binding operation, the AC MUST select an unused WLAN ID between 1 and 16 [RFC5416]. For example, to bind one more WLAN profile to a radio that has been bound with a WLAN profile, the AC SHOULD allocate WLAN ID 2 to the radio. Although the maximum value of a WLAN ID is 16, the operator could configure more than 16 WLAN Profiles on the AC.

5.2. Requirements and Constraints

The IEEE 802.11 MIB module [IEEE.802-11.2007] already defines MIB objects for most IEEE 802.11 Message Elements in the CAPWAP Protocol Binding for IEEE 802.11 [RFC5416]. As a CAPWAP protocol 802.11 binding MIB module, the CAPWAP-DOT11-MIB module MUST be able to reuse such MIB objects in the IEEE 802.11 MIB module and support functions (such as MAC mode for WLAN in the [RFC5416]) that are not in the scope of IEEE 802.11 standard. The CAPWAP-DOT11-MIB module MUST support such functions.

In summary, the CAPWAP-DOT11-MIB module needs to support:

- Reuse of wireless binding MIB modules in the IEEE 802.11 standard;

- Centralized management and configuration of WLAN profiles on the AC;

- Configuration of a MAC type and tunnel mode for a specific WLAN profile.

5.3. Mechanism of Reusing Wireless Binding MIB Module

In the IEEE 802.11 MIB module, the MIB tables such as dot11AuthenticationAlgorithmsTable are able to support WLAN configuration (such as authentication algorithm), and these tables use the ifIndex as the index which works well in the autonomous WLAN architecture.

Reuse of such wireless binding MIB modules is very important to centralized WLAN architectures. The key point is to abstract a WLAN profile as a WLAN Profile Interface on the AC, which could be identified by an ifIndex. The MIB objects in the IEEE 802.11 MIB module which are associated with this interface can be used to configure WLAN parameters for the WLAN, such as authentication algorithm. With the ifIndex of a WLAN Profile Interface, the AC is able to reuse the IEEE 802.11 MIB module.

In the CAPWAP-BASE-MIB module, each PHY radio is identified by a WTP ID and a radio ID, and has a corresponding WTP Virtual Radio Interface on the AC. The IEEE 802.11 MIB module associated with this interface can be used to configure IEEE 802.11 wireless binding parameters for the radio such as RTS Threshold. A WLAN Basic Service Set (BSS) Interface, created by binding a WLAN to a WTP Virtual Radio Interface, is used for data forwarding.

6. Structure of MIB Module

The MIB objects are derived from the CAPWAP protocol binding for IEEE 802.11 document [RFC5416].

capwapDot11WlanTable

The table allows the operator to display and configure WLAN profiles, such as specifying the MAC type and tunnel mode for a WLAN. Also, it helps the AC to configure a WLAN through the IEEE 802.11 MIB module.

capwapDot11WlanBindTable

The table provides a way to bind WLAN profiles to a WTP Virtual Radio Interface, which has a corresponding PHY radio. A binding operation dynamically creates a WLAN BSS Interface, which is used for data forwarding.

7. Relationship to Other MIB Modules

7.1. Relationship to SNMPv2-MIB Module

The CAPWAP-DOT11-MIB module does not duplicate the objects of the 'system' group in the SNMPv2-MIB [RFC3418] that is defined as being mandatory for all systems, and the objects apply to the entity as a whole. The 'system' group provides identification of the management entity and certain other system-wide data.

7.2. Relationship to IF-MIB Module

The Interfaces Group [RFC2863] defines generic managed objects for managing interfaces. This memo contains the media-specific extensions to the Interfaces Group for managing WLAN that are modeled as interfaces.

Each WLAN profile corresponds to a WLAN Profile Interface on the AC. The interface MUST be modeled as an ifEntry, and ifEntry objects such as ifIndex, ifDescr, ifName, and ifAlias are to be used as per [RFC2863]. The WLAN Profile Interface provides a way to configure IEEE 802.11 parameters for a specific WLAN and reuse the IEEE 802.11 MIB module.

To provide data forwarding service, the AC dynamically creates WLAN BSS Interfaces. A WLAN BSS Interface MUST be modeled as an ifEntry, and ifEntry objects such as ifIndex, ifDescr, ifName, and ifAlias are to be used as per [RFC2863]. The interface enables a single physical WTP to support multiple WLANs.

Also, the AC MUST have a mechanism that preserves the value of the ifIndexes (of both the WLAN Profile Interfaces and the WLAN BSS Interfaces) in the ifTable at AC reboot.

7.3. Relationship to CAPWAP-BASE-MIB Module

The CAPWAP-BASE-MIB module provides a way to manage and control WTP and radio objects. Especially, it provides the WTP Virtual Radio Interface mechanism to enable the AC to reuse the IEEE 802.11 MIB module. With this mechanism, an operator could configure an IEEE

802.11 radio's parameters and view the radio's traffic statistics on the AC. Based on the CAPWAP-BASE-MIB module, the CAPWAP-DOT11-MIB module provides more WLAN information.

7.4. Relationship to MIB Module in the IEEE 802.11 Standard

With the ifIndex of WLAN Profile Interface and WLAN BSS Interface, the MIB module is able to reuse the IEEE 802.11 MIB module [IEEE.802-11.2007]. The CAPWAP-DOT11-MIB module does not duplicate those objects in the IEEE 802.11 MIB module.

The CAPWAP Protocol Binding for IEEE 802.11 [RFC5416] involves some of the MIB objects defined in the IEEE 802.11 standard. Although CAPWAP-DOT11-MIB module uses it [RFC5416] as a reference, it could reuse all the MIB objects in the IEEE 802.11 standard , and is not limited by the scope of CAPWAP Protocol Binding for IEEE 802.11.

7.5. MIB Modules Required for IMPORTS

The following MIB modules are required for IMPORTS: SNMPv2-SMI [RFC2578], SNMPv2-TC [RFC2579], SNMPv2-CONF [RFC2580], IF-MIB [RFC2863], and CAPWAP-BASE-MIB [RFC5833].

8. Example of CAPWAP-DOT11-MIB Module Usage

1) Create a WTP profile.

Suppose the WTP's base MAC address is '00:01:01:01:01:00'. Creates a WTP profile for it through the capwapBaseWtpProfileTable [RFC5833] as follows:

```
In capwapBaseWtpProfileTable
{
  capwapBaseWtpProfileId                  = 1,
  capwapBaseWtpProfileName                = 'WTP Profile 123456',
  capwapBaseWtpProfileWtpMacAddress       = '00:01:01:01:01:00',
  capwapBaseWtpProfileWTPModelNumber            = 'WTP123',
  capwapBaseWtpProfileWtpName                   = 'WTP 123456',
  capwapBaseWtpProfileWtpLocation               = 'office',
  capwapBaseWtpProfileWtpStaticIpEnable         = true(1),
  capwapBaseWtpProfileWtpStaticIpType           = ipv4(1),
  capwapBaseWtpProfileWtpStaticIpAddress        = '192.0.2.10',
  capwapBaseWtpProfileWtpNetmask                = '255.255.255.0',
  capwapBaseWtpProfileWtpGateway                = '192.0.2.1',
  capwapBaseWtpProfileWtpFallbackEnable         = true(1),
  capwapBaseWtpProfileWtpEchoInterval           = 30,
  capwapBaseWtpProfileWtpIdleTimeout            = 300,
  capwapBaseWtpProfileWtpMaxDiscoveryInterval   = 20,
```

```
   capwapBaseWtpProfileWtpReportInterval          = 120,
   capwapBaseWtpProfileWtpStatisticsTimer         = 120,
   capwapBaseWtpProfileWtpEcnSupport              = limited(0)
 }
```

Suppose the WTP with model number 'WTP123' has one PHY radio and this PHY radio is identified by ID 1. The creation of this WTP profile triggers the AC to automatically create a WTP Virtual Radio Interface and add a new row object to the capwapBaseWirelessBindingTable without manual intervention. Suppose the ifIndex of the WTP Virtual Radio Interface is 10. The following information is stored in the capwapBaseWirelessBindingTable.

```
In capwapBaseWirelessBindingTable
{
  capwapBaseWtpProfileId                          = 1,
  capwapBaseWirelessBindingRadioId                = 1,
  capwapBaseWirelessBindingVirtualRadioIfIndex    = 10,
  capwapBaseWirelessBindingType                   = dot11(2)
}
```

The WTP Virtual Radio Interfaces on the AC correspond to the PHY radios on the WTP. The WTP Virtual Radio Interface is modeled by ifTable [RFC2863].

```
In ifTable
{
  ifIndex              = 10,
  ifDescr              = 'WTP Virtual Radio Interface',
  ifType               = 254,
  ifMtu                = 0,
  ifSpeed              = 0,
  ifPhysAddress        = '00:00:00:00:00:00',
  ifAdminStatus        = true(1),
  ifOperStatus         = false(0),
  ifLastChange         = 0,
  ifInOctets           = 0,
  ifInUcastPkts        = 0,
  ifInDiscards         = 0,
  ifInErrors           = 0,
  ifInUnknownProtos    = 0,
  ifOutOctets          = 0,
  ifOutUcastPkts       = 0,
  ifOutDiscards        = 0,
  ifOutErrors          = 0
 }
```

2) Query the ifIndexes of WTP Virtual Radio Interfaces.

Before configuring PHY radios, the operator needs to get the ifIndexes of WTP Virtual Radio Interfaces corresponding to the PHY radios.

As the capwapBaseWirelessBindingTable already stores the mappings between PHY radios (Radio IDs) and the ifIndexes of WTP Virtual Radio Interfaces, the operator can get the ifIndex information by querying this table. Such a query operation SHOULD run from radio ID 1 to radio ID 31 (according to [RFC5415]), and stop when an invalid ifIndex value (0) is returned.

This example uses capwapBaseWtpProfileId = 1 and capwapBaseWirelessBindingRadioId = 1 as inputs to query the capwapBaseWirelessBindingTable, and gets capwapBaseWirelessBindingVirtualRadioIfIndex = 10. Then it uses capwapBaseWtpProfileId = 1 and capwapBaseWirelessBindingRadioId = 2, and gets an invalid ifIndex value (0), so the query operation ends. This method gets not only the ifIndexes of WTP Virtual Radio Interfaces, but also the numbers of PHY radios. Besides checking whether the ifIndex value is valid, the operator SHOULD check whether the capwapBaseWirelessBindingType is the desired binding type.

3) Configure IEEE 802.11 parameters for a WTP Virtual Radio Interface

This configuration is made on the AC through the IEEE 802.11 MIB module.

The following shows an example of configuring parameters for a WTP Virtual Radio Interface with ifIndex 10 through the dot11OperationTable [IEEE.802-11.2007].

```
In dot11OperationTable
{
  ifIndex                                 = 10,
  dot11MACAddress                         = '00:00:00:00:00:00',
  dot11RTSThreshold                       = 2347,
  dot11ShortRetryLimit                    = 7,
  dot11LongRetryLimit                     = 4,
  dot11FragmentationThreshold             = 256,
  dot11MaxTransmitMSDULifetime            = 512,
  dot11MaxReceiveLifetime                 = 512,
  dot11ManufacturerID                     = 'capwap',
  dot11ProductID                          = 'capwap',
  dot11CAPLimit                           = 2,
  dot11HCCWmin                            = 0,
```

```
  dot11HCCWmax                              = 0,
  dot11HCCAIFSN                             = 1,
  dot11ADDBAResponseTimeout                 = 1,
  dot11ADDTSResponseTimeout                 = 1,
  dot11ChannelUtilizationBeaconInterval     = 50,
  dot11ScheduleTimeout                      = 10,
  dot11DLSResponseTimeout                   = 10,
  dot11QAPMissingAckRetryLimit              = 1,
  dot11EDCAAveragingPeriod                  = 5
}
```

4) Configure a WLAN Profile.

WLAN configuration is made on the AC through the CAPWAP-DOT11-MIB module, and IEEE 802.11 MIB module.

The first step is to create a WLAN Profile Interface through the CAPWAP-DOT11-MIB module on the AC.

For example, when you configure a WLAN profile that is identified by capwapDot11WlanProfileId 1, the capwapDot11WlanTable creates the following row object for it.

```
In capwapDot11WlanTable
{
  capwapDot11WlanProfileId          = 1,
  capwapDot11WlanProfileIfIndex     = 20,
  capwapDot11WlanMacType            = splitMAC(2),
  capwapDot11WlanTunnelMode         = dot3Tunnel(2),
  capwapDot11WlanRowStatus          = createAndGo(4)
}
```

The creation of a row object triggers the AC to automatically create a WLAN Profile Interface and it is identified by ifIndex 20 without manual intervention.

A WLAN Profile Interface MUST be modeled as an ifEntry on the AC that provides appropriate interface information. The capwapDot11WlanTable stores the mappings between capwapDot11WlanProfileIds and the ifIndexes of WLAN Profile Interfaces.

```
In ifTable
{
  ifIndex             = 20,
  ifDescr             = 'WLAN Profile Interface',
  ifType              = 252,
  ifMtu               = 0,
```

```
  ifSpeed              = 0,
  ifPhysAddress        = '00:00:00:00:00:00',
  ifAdminStatus        = true(1),
  ifOperStatus         = true(1),
  ifLastChange         = 0,
  ifInOctets           = 0,
  ifInUcastPkts        = 0,
  ifInDiscards         = 0,
  ifInErrors           = 0,
  ifInUnknownProtos    = 0,
  ifOutOctets          = 0,
  ifOutUcastPkts       = 0,
  ifOutDiscards        = 0,
  ifOutErrors          = 0
}
```

The second step is to configure WLAN parameters for the WLAN Profile Interface through the IEEE 802.11 MIB module on the AC.

The following example configures an authentication algorithm for a WLAN.

```
In dot11AuthenticationAlgorithmsTable
{
  ifIndex                                = 20,
  dot11AuthenticationAlgorithmsIndex     = 1,
  dot11AuthenticationAlgorithm           = Shared Key(2),
  dot11AuthenticationAlgorithmsEnable    = true(1)
}
```

Here, ifIndex 20 identifies the WLAN Profile Interface, and the index of the configured authentication algorithm is 1.

5) Bind WLAN Profiles to a WTP radio.

On the AC, the capwapDot11WlanBindTable in the CAPWAP-DOT11-MIB stores the bindings between WLAN profiles(identified by capwapDot11WlanProfileId) and WTP Virtual Radio Interfaces (identified by the ifIndex).

For example, after the operator binds a WLAN profile with capwapDot11WlanProfileId 1 to WTP Virtual Radio Interface with ifIndex 10, the capwapDot11WlanBindTable creates the following row object.

```
In capwapDot11WlanBindTable
{
  ifIndex                          = 10,
  capwapDot11WlanProfileId         = 1,
  capwapDot11WlanBindBssIfIndex    = 30,
  capwapDot11WlanBindRowStatus     = createAndGo(4)
}
```

If the capwapDot11WlanMacType of the WLAN is splitMAC(2), the creation of the row object in the capwapDot11WlanBindTable triggers the AC to automatically create a WLAN BSS Interface identified by ifIndex 30 without manual intervention.

The WLAN BSS Interface MUST be modeled as an ifEntry on the AC, which provides appropriate interface information. The capwapDot11WlanBindTable stores the mappings among the ifIndex of a WTP Virtual Radio Interface, WLAN profile ID, WLAN ID, and the ifIndex of a WLAN BSS Interface.

6) Get the current configuration status report from the WTP to the AC.

Before a WTP that has joined the AC gets configuration from the AC, it needs to report its current configuration status by sending a configuration status request message to the AC, which uses the message to update corresponding MIB objects on the AC. For example, for ifIndex 10 (which identifies a WLAN Virtual Radio Interface), its ifOperStatus in the ifTable is updated according to the current radio operational status in the CAPWAP message [RFC5415].

7) Query WTP and radio statistical data.

After WTPs start to run, the operator could query WTP and radio statistics data through the CAPWAP-BASE-MIB and CAPWAP-DOT11-MIB modules. For example, through the dot11CountersTable [IEEE.802-11.2007], the operator could query counter data of a radio that is identified by the ifIndex of the corresponding WLAN Virtual Radio Interface.

8) Query other statistical data.

The operator could query the configuration of a WLAN through the dot11AuthenticationAlgorithmsTable [IEEE.802-11.2007] and the statistical data of a WLAN BSS Interface through the ifTable [RFC2863].

9. Definitions

```
CAPWAP-DOT11-MIB DEFINITIONS ::= BEGIN

IMPORTS
   RowStatus, TEXTUAL-CONVENTION
       FROM SNMPv2-TC
   OBJECT-GROUP, MODULE-COMPLIANCE
       FROM SNMPv2-CONF
   MODULE-IDENTITY, OBJECT-TYPE, mib-2, Unsigned32
       FROM SNMPv2-SMI
   ifIndex, InterfaceIndex
       FROM IF-MIB
   CapwapBaseMacTypeTC, CapwapBaseTunnelModeTC
       FROM CAPWAP-BASE-MIB;

capwapDot11MIB MODULE-IDENTITY
    LAST-UPDATED "201004300000Z"        -- 30 April 2010
    ORGANIZATION "IETF Control And Provisioning of Wireless Access
                  Points (CAPWAP) Working Group
                  http://www.ietf.org/html.charters/capwap-charter.html"
    CONTACT-INFO
        "General Discussion: capwap@frascone.com
         To Subscribe: http://lists.frascone.com/mailman/listinfo/capwap

         Yang Shi (editor)
         Hangzhou H3C Tech. Co., Ltd.
         Beijing R&D Center of H3C, Digital Technology Plaza
         NO. 9 Shangdi 9th Street, Haidian District
         Beijing  100085
         China
         Phone: +86 010 82775276
         Email: rishyang@gmail.com

         David T. Perkins (editor)
         228 Bayview Dr.
         San Carlos, CA  94070
         USA
         Phone: +1 408 394-8702
         Email:  dperkins@dsperkins.com

         Chris Elliott (editor)
         1516 Kent St.
         Durham, NC  27707
         USA
         Phone: +1 919-308-1216
         Email: chelliot@pobox.com
```

```
          Yong Zhang (editor)
          Fortinet, Inc.
          1090 Kifer Road
          Sunnyvale, CA  94086
          USA
          Email: yzhang@fortinet.com"

   DESCRIPTION
       "Copyright (c) 2010 IETF Trust and the persons identified as
        authors of the code.  All rights reserved.

        Redistribution and use in source and binary forms, with or
        without modification, is permitted pursuant to, and subject
        to the license terms contained in, the Simplified BSD License
        set forth in Section 4.c of the IETF Trust's Legal Provisions
        Relating to IETF Documents
        (http://trustee.ietf.org/license-info).

        This version of this MIB module is part of RFC 5834;
        see the RFC itself for full legal notices.

        This MIB module contains managed object definitions for
        CAPWAP Protocol binding for IEEE 802.11."
   REVISION    "201004300000Z"
   DESCRIPTION
       "Initial version, published as RFC 5834"
        ::= { mib-2 195 }

-- Textual conventions

CapwapDot11WlanIdTC ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS      current
    DESCRIPTION
        "Represents the unique identifier of a Wireless Local Area
         Network (WLAN)."
    SYNTAX      Unsigned32 (1..16)

CapwapDot11WlanIdProfileTC ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS      current
    DESCRIPTION
        "Represents the unique identifier of a WLAN profile."
    SYNTAX      Unsigned32 (1..512)

-- Top level components of this MIB module

-- Tables, Scalars
```

```
capwapDot11Objects OBJECT IDENTIFIER
    ::= { capwapDot11MIB 1 }
-- Conformance
capwapDot11Conformance OBJECT IDENTIFIER
    ::= { capwapDot11MIB 2 }

-- capwapDot11WlanTable Table

capwapDot11WlanTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF CapwapDot11WlanEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table that allows the operator to display and configure
         WLAN profiles, such as specifying the MAC type and tunnel mode
         for a WLAN.  Also, it helps the AC to configure a WLAN through
         the IEEE 802.11 MIB module.
         Values of all objects in this table are persistent at
         restart/reboot."
    ::= { capwapDot11Objects 1 }

capwapDot11WlanEntry  OBJECT-TYPE
    SYNTAX      CapwapDot11WlanEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A set of objects that stores the settings of a WLAN profile."
    INDEX { capwapDot11WlanProfileId }
    ::= { capwapDot11WlanTable 1 }

CapwapDot11WlanEntry ::=
    SEQUENCE {
      capwapDot11WlanProfileId          CapwapDot11WlanIdProfileTC,
      capwapDot11WlanProfileIfIndex     InterfaceIndex,
      capwapDot11WlanMacType            CapwapBaseMacTypeTC,
      capwapDot11WlanTunnelMode         CapwapBaseTunnelModeTC,
      capwapDot11WlanRowStatus          RowStatus
    }

capwapDot11WlanProfileId OBJECT-TYPE
    SYNTAX      CapwapDot11WlanIdProfileTC
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Represents the identifier of a WLAN profile that has a
         corresponding capwapDot11WlanProfileIfIndex."
    ::= { capwapDot11WlanEntry 1 }
```

```
capwapDot11WlanProfileIfIndex OBJECT-TYPE
    SYNTAX      InterfaceIndex
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the index value that uniquely identifies a
         WLAN Profile Interface.  The interface identified by a
         particular value of this index is the same interface as
         identified by the same value of the ifIndex.
         The creation of a row object in the capwapDot11WlanTable
         triggers the AC to automatically create an WLAN Profile
         Interface identified by an ifIndex without manual
         intervention.
         Most MIB tables in the IEEE 802.11 MIB module
         [IEEE.802-11.2007] use an ifIndex to identify an interface
         to facilitate the configuration and maintenance, for example,
         dot11AuthenticationAlgorithmsTable.
         Using the ifIndex of a WLAN Profile Interface, the Operator
         could configure a WLAN through the IEEE 802.11 MIB module."
    ::= { capwapDot11WlanEntry 2 }

capwapDot11WlanMacType OBJECT-TYPE
    SYNTAX      CapwapBaseMacTypeTC
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Represents whether the WTP SHOULD support the WLAN in
         Local or Split MAC modes."
    REFERENCE
        "Section 6.1 of CAPWAP Protocol Binding for IEEE 802.11,
         RFC 5416."
    ::= { capwapDot11WlanEntry 3 }

capwapDot11WlanTunnelMode OBJECT-TYPE
    SYNTAX      CapwapBaseTunnelModeTC
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Represents the frame tunneling mode to be used for IEEE 802.11
         data frames from all stations associated with the WLAN.
         Bits are exclusive with each other for a specific WLAN profile,
         and only one tunnel mode could be configured.
         If the operator set more than one bit, the value of the
         Response-PDU's error-status field is set to 'wrongValue',
         and the value of its error-index field is set to the index of
         the failed variable binding."
    REFERENCE
        "Section 6.1 of CAPWAP Protocol Binding for IEEE 802.11,
```

```
          RFC 5416."
    ::= { capwapDot11WlanEntry 4 }

capwapDot11WlanRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This variable is used to create, modify, and/or delete a row
         in this table.
         All the objects in a row can be modified only when the value
         of this object in the corresponding conceptual row is not
         'active'.  Thus, to modify one or more of the objects in
         this conceptual row:
              a. change the row status to 'notInService',
              b. change the values of the row
              c. change the row status to 'active'
         The capwapDot11WlanRowStatus may be changed to 'active'
         if all the managed objects in the conceptual row with
         MAX-ACCESS read-create have been assigned valid values.

         When the operator deletes a WLAN profile, the AC SHOULD
         check whether the WLAN profile is bound with a radio.
         If yes, the value of the Response-PDU's error-status field
         is set to 'inconsistentValue', and the value of its
         error-index field is set to the index of the failed variable
         binding.  If not, the row object could be deleted."
    ::= { capwapDot11WlanEntry 5 }

-- End of capwapDot11WlanTable Table


-- capwapDot11WlanBindTable Table

capwapDot11WlanBindTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF CapwapDot11WlanBindEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table that stores bindings between WLAN profiles
         (identified by capwapDot11WlanProfileId) and WTP Virtual Radio
         Interfaces.  The WTP Virtual Radio Interfaces on the AC
         correspond to physical layer (PHY) radios on the WTPs.
         It also stores the mappings between WLAN IDs and WLAN
         Basic Service Set (BSS) Interfaces.
         Values of all objects in this table are persistent at
         restart/reboot."
    REFERENCE
```

```
        "Section 6.1 of CAPWAP Protocol Binding for IEEE 802.11,
         RFC 5416."
    ::= { capwapDot11Objects 2 }

capwapDot11WlanBindEntry OBJECT-TYPE
    SYNTAX      CapwapDot11WlanBindEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A set of objects that stores the binding of a WLAN profile
         to a WTP Virtual Radio Interface.  It also stores the mapping
         between WLAN ID and WLAN BSS Interface.
         The INDEX object ifIndex is the ifIndex of a WTP Virtual
         Radio Interface."
    INDEX { ifIndex, capwapDot11WlanProfileId }
    ::= { capwapDot11WlanBindTable 1 }

CapwapDot11WlanBindEntry ::=
    SEQUENCE {
      capwapDot11WlanBindWlanId        CapwapDot11WlanIdTC,
      capwapDot11WlanBindBssIfIndex    InterfaceIndex,
      capwapDot11WlanBindRowStatus     RowStatus
    }

capwapDot11WlanBindWlanId OBJECT-TYPE
    SYNTAX      CapwapDot11WlanIdTC
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the WLAN ID of a WLAN.
         During a binding operation, the AC MUST select an unused
         WLAN ID from between 1 and 16 [RFC5416].  For example, to bind
         another WLAN profile to a radio that has been bound with
         a WLAN profile, WLAN ID 2 should be assigned."
    REFERENCE
        "Section 6.1 of CAPWAP Protocol Binding for IEEE 802.11,
         RFC 5416."
    ::= { capwapDot11WlanBindEntry 1 }

capwapDot11WlanBindBssIfIndex OBJECT-TYPE
    SYNTAX      InterfaceIndex
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the index value that uniquely identifies a
         WLAN BSS Interface.  The interface identified by a
         particular value of this index is the same interface as
         identified by the same value of the ifIndex.
```

```
          The ifIndex here is for a WLAN BSS Interface.
          The creation of a row object in the capwapDot11WlanBindTable
          triggers the AC to automatically create a WLAN BSS Interface
          identified by an ifIndex without manual intervention.
          The PHY address of the capwapDot11WlanBindBssIfIndex is the
          BSSID.  While manufacturers are free to assign BSSIDs by using
          any arbitrary mechanism, it is advised that where possible the
          BSSIDs are assigned as a contiguous block.
          When assigned as a block, implementations can still assign
          any of the available BSSIDs to any WLAN.  One possible method
          is for the WTP to assign the address using the following
          algorithm: base BSSID address + WLAN ID."
      REFERENCE
          "Section 2.4 of CAPWAP Protocol Binding for IEEE 802.11,
          RFC 5416."
      ::= { capwapDot11WlanBindEntry 2 }

capwapDot11WlanBindRowStatus OBJECT-TYPE
      SYNTAX      RowStatus
      MAX-ACCESS  read-create
      STATUS      current
      DESCRIPTION
          "This variable is used to create, modify, and/or delete a row
          in this table.
          All the objects in a row can be modified only when the value
          of this object in the corresponding conceptual row is not
          'active'.  Thus, to modify one or more of the objects in
          this conceptual row:
               a. change the row status to 'notInService',
               b. change the values of the row
               c. change the row status to 'active'"
      ::= { capwapDot11WlanBindEntry 3 }

-- End of capwapDot11WlanBindTable Table


-- Module compliance

capwapDot11Groups OBJECT IDENTIFIER
      ::= { capwapDot11Conformance 1 }

capwapDot11Compliances OBJECT IDENTIFIER
      ::= { capwapDot11Conformance 2 }

capwapDot11Compliance MODULE-COMPLIANCE
      STATUS current
      DESCRIPTION
          "Describes the requirements for conformance to the
```

```
          CAPWAP-DOT11-MIB module."

    MODULE -- this module
      MANDATORY-GROUPS {
        capwapDot11WlanGroup,
        capwapDot11WlanBindGroup
      }
    ::= { capwapDot11Compliances 1 }

capwapDot11WlanGroup    OBJECT-GROUP
    OBJECTS {
      capwapDot11WlanProfileIfIndex,
      capwapDot11WlanMacType,
      capwapDot11WlanTunnelMode,
      capwapDot11WlanRowStatus
    }
    STATUS  current
    DESCRIPTION
        "A collection of objects that is used to configure
         the properties of a WLAN profile."
    ::= { capwapDot11Groups 1 }

capwapDot11WlanBindGroup    OBJECT-GROUP
    OBJECTS {
      capwapDot11WlanBindWlanId,
      capwapDot11WlanBindBssIfIndex,
      capwapDot11WlanBindRowStatus
    }
    STATUS  current
    DESCRIPTION
        "A collection of objects that is used to bind the
         WLAN profiles with a radio."
    ::= { capwapDot11Groups 2 }

END
```

10. Security Considerations

There are a number of management objects defined in this MIB module with a MAX-ACCESS clause of read-write and/or read-create. Such objects MAY be considered sensitive or vulnerable in some network environments. The support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations. The following are the tables and objects and their sensitivity/vulnerability:

- Unauthorized changes to the capwapDot11WlanTable and capwapDot11WlanBindTable MAY disrupt allocation of resources in the network, and also change the behavior of the WLAN system such as MAC type.

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPSec), even then, there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB module.

It is RECOMMENDED that implementers consider the security features as provided by the SNMPv3 framework (see [RFC3410], section 8), including full support for the SNMPv3 cryptographic mechanisms (for authentication and privacy).

Further, deployment of SNMP versions prior to SNMPv3 is NOT RECOMMENDED. Instead, it is RECOMMENDED to deploy SNMPv3 and to enable cryptographic security. It is then a customer/operator responsibility to ensure that the SNMP entity giving access to an instance of this MIB module is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

11. IANA Considerations

11.1. IANA Considerations for CAPWAP-DOT11-MIB Module

The MIB module in this document uses the following IANA-assigned OBJECT IDENTIFIER value recorded in the SMI Numbers registry:

Descriptor	OBJECT IDENTIFIER value
capwapDot11MIB	{ mib-2 195 }

11.2. IANA Considerations for ifType

IANA has assigned the following ifTypes:

Decimal	Name	Description
252	capwapDot11Profile	WLAN Profile Interface
253	capwapDot11Bss	WLAN BSS Interface

12. Contributors

This MIB module is based on contributions from Long Gao.

13. Acknowledgements

Thanks to David Harrington, Dan Romascanu, Abhijit Choudhury, and Elwyn Davies for helpful comments on this document and guiding some technical solutions.

The authors also thank their friends and coworkers Fei Fang, Xuebin Zhu, Hao Song, Yu Liu, Sachin Dutta, Ju Wang, Yujin Zhao, Haitao Zhang, Xiansen Cai, and Xiaolan Wan.

14. References

14.1. Normative References

[IEEE.802-11.2007] "Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications", IEEE Standard 802.11, 2007, <http://standards.ieee.org/getieee802/download/802.11-2007.pdf>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

[RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.

[RFC2580] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.

[RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, June 2000.

[RFC3418] Presuhn, R., "Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3418, December 2002.

[RFC5415] Calhoun, P., Montemurro, M., and D. Stanley, "Control And Provisioning of Wireless Access Points (CAPWAP) Protocol Specification", RFC 5415, March 2009.

[RFC5416] Calhoun, P., Montemurro, M., and D. Stanley, "Control and Provisioning of Wireless Access Points (CAPWAP) Protocol Binding for IEEE 802.11", RFC 5416, March 2009.

[RFC5833] Shi, Y., Ed., Perkins, D., Ed., Elliott, C., Ed., and Y. Zhang, Ed., "Control and Provisioning of Wireless Access Points (CAPWAP) Protocol Base MIB", RFC 5833, May 2010.

14.2. Informative References

[RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.

[RFC4347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security", RFC 4347, April 2006.

Authors' Addresses

Yang Shi (editor)
Hangzhou H3C Tech. Co., Ltd.
Beijing R&D Center of H3C, Digital Technology Plaza
NO. 9 Shangdi 9th Street, Haidian District
Beijing 100085
China

Phone: +86 010 82775276
EMail: rishyang@gmail.com

David T. Perkins (editor)
228 Bayview Dr.
San Carlos, CA 94070
USA

Phone: +1 408 394-8702
EMail: dperkins@dsperkins.com

Chris Elliott (editor)
1516 Kent St.
Durham, NC 27707
USA

Phone: +1 919-308-1216
EMail: chelliot@pobox.com

Yong Zhang (editor)
Fortinet, Inc.
1090 Kifer Road
Sunnyvale, CA 94086
USA

EMail: yzhang@fortinet.com