

Opsawg Working Group  
Internet-Draft  
Intended status: Experimental  
Expires: August 2, 2018

R. Zhang  
China Telecom  
R. Pazhyannur  
S. Gundavelli  
Cisco  
Z. Cao  
H. Deng  
Z. Du  
Huawei  
January 29, 2018

Alternate Tunnel Encapsulation for Data Frames in CAPWAP  
draft-ietf-opsawg-capwap-alt-tunnel-12

Abstract

Control and Provisioning of Wireless Access Points (CAPWAP) defines a specification to encapsulate a station's data frames between the Wireless Transmission Point (WTP) and Access Controller (AC). Specifically, the station's IEEE 802.11 data frames can be either locally bridged or tunneled to the AC. When tunneled, a CAPWAP data channel is used for tunneling. In many deployments encapsulating data frames to an entity other than the AC (for example to an Access Router (AR)) is desirable. Furthermore, it may also be desirable to use different tunnel encapsulation modes between the WTP and the Access Router. This document defines extension to CAPWAP protocol for supporting this capability and refers to it as alternate tunnel encapsulation. The alternate tunnel encapsulation allows 1) the WTP to tunnel non-management data frames to an endpoint different from the AC and 2) the WTP to tunnel using one of many known encapsulation types such as IP-IP, IP-GRE, CAPWAP. The WTP may advertise support for alternate tunnel encapsulation during the discovery and join process and AC may select one of the supported alternate tunnel encapsulation types while configuring the WTP.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any

time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 2, 2018.

#### Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

#### Table of Contents

1. Introduction . . . . .	3
1.1. Conventions used in this document . . . . .	7
1.2. Terminology . . . . .	7
1.3. History of the document . . . . .	8
2. Alternate Tunnel Encapsulation Overview . . . . .	8
3. CAPWAP Protocol Message Elements Extensions . . . . .	11
3.1. Supported Alternate Tunnel Encapsulations . . . . .	11
3.2. Alternate Tunnel Encapsulations Type . . . . .	11
3.3. IEEE 802.11 WTP Alternate Tunnel Failure Indication . . . . .	12
4. Alternate Tunnel Types . . . . .	13
4.1. CAPWAP based Alternate Tunnel . . . . .	13
4.2. PMIPv6 based Alternate Tunnel . . . . .	14
4.3. GRE based Alternate Tunnel . . . . .	15
5. Alternate Tunnel Information Elements . . . . .	15
5.1. Access Router Information Elements . . . . .	15
5.1.1. AR IPv4 List Element . . . . .	16
5.1.2. AR IPv6 List Element . . . . .	16
5.2. Tunnel DTLS Policy Element . . . . .	17
5.3. IEEE 802.11 Tagging Mode Policy Element . . . . .	19
5.4. CAPWAP Transport Protocol Element . . . . .	20
5.5. GRE Key Element . . . . .	22
5.6. IPv6 MTU Element . . . . .	23
6. IANA Considerations . . . . .	23
7. Security Considerations . . . . .	25
8. Contributors . . . . .	25
9. References . . . . .	25

9.1. Normative References . . . . .	25
9.2. Informative References . . . . .	26
Authors' Addresses . . . . .	27

## 1. Introduction

Service Providers are deploying very large Wi-Fi network containing hundreds of thousands of Access Points (APs), which are referred to as Wireless Transmission Points (WTPs) in Control and Provisioning of Wireless Access Points (CAPWAP) terminology [RFC5415]. These networks are designed to carry traffic generated from mobile users. The volume in mobile user traffic is already very large and expected to continue growing rapidly. As a result, operators are looking for scalable solutions that can meet the increasing demand. The scalability requirement can be met by splitting the control/management plane from the data plane. This enables the data plane to scale independent of the control/management plane. This specification provides a way to enable such separation.

CAPWAP ([RFC5415], [RFC5416]) defines a tunnel mode that describes how the WTP handles the data plane (user traffic). The following types are defined:

- o Local Bridging: All data frames are locally bridged.
- o 802.3 Tunnel: All data frames are tunneled to the Access Controller (AC) in 802.3 format.
- o 802.11 Tunnel: All data frames are tunneled to the AC in 802.11 format.

Figure 1 describes a system with Local Bridging. The AC is in a centralized location. The data plane is locally bridged by the WTPs leading to a system with centralized control plane with distributed data plane. This system has two benefits: 1) reduces the scale requirement on data traffic handling capability of the AC and 2) leads to more efficient/optimal routing of data traffic while maintaining centralized control/management.

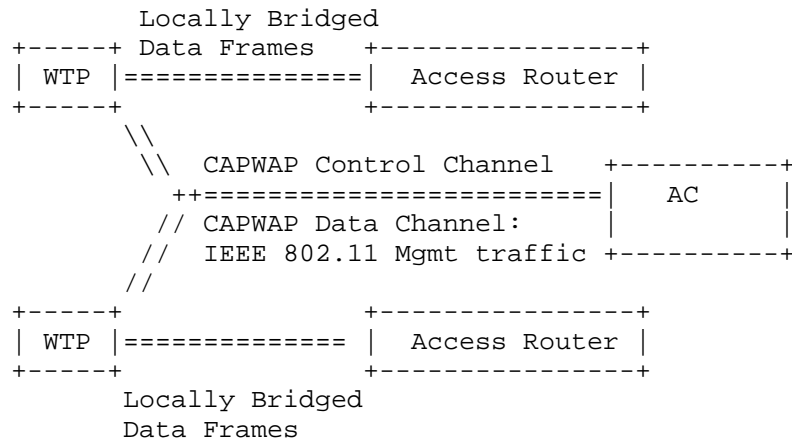


Figure 1: Centralized Control with Distributed Data

The AC handles control of WTPs. In addition, the AC also handles the IEEE 802.11 management traffic to/from the stations. There is CAPWAP Control and Data Channel between the WTP and the AC. Note that even though there is no user traffic transported between the WTP and AC, there is still a CAPWAP Data Channel. The CAPWAP Data Channel carries the IEEE 802.11 management traffic (like IEEE 802.11 Action Frames).

Figure 2 shows a system where the tunnel mode is configured to tunnel data frames between the WTP and the AC either using 802.3 Tunnel or 802.11 Tunnel configurations. Operators deploy this configuration when they need to tunnel the user traffic. The tunneling requirement may be driven by the need to apply policy at the AC. This requirement could be met in the locally bridged system (Figure 1) if the Access Router (AR) implemented the required policy. However, in many deployments the operator managing the WTP is different than the operator managing the Access Router. When the operators are different, the policy has to be enforced in a tunnel termination point in the WTP operator's network.

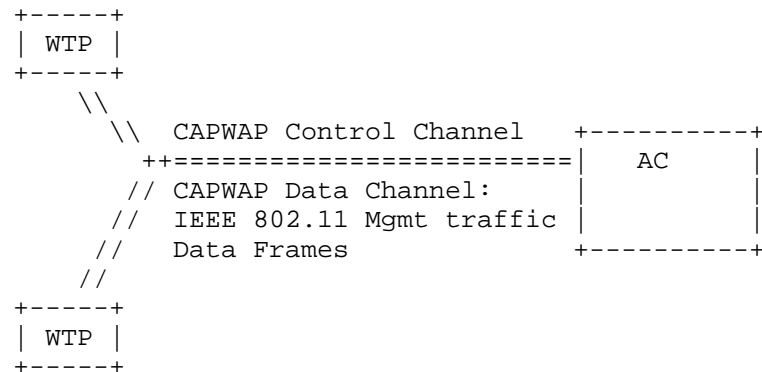


Figure 2: Centralized Control and Centralized Data

The key difference with the locally bridged system is that the data frames are tunneled to the AC instead of being locally bridged. There are two shortcomings with the system in Figure 2. 1) They do not allow the WTP to tunnel data frames to an endpoint different from the AC and 2) They do not allow the WTP to tunnel data frames using any encapsulation other than CAPWAP (as specified in Section 4.4.2 of [RFC5415]).

Figure 3 shows a system where the WTP tunnels data frames to an alternate entity different from the AC. The WTP also uses an alternate tunnel encapsulation such as L2TP, L2TPv3, IP-in-IP, IP/GRE, etc. This enables 1) independent scaling of data plane and 2) leveraging of commonly used tunnel encapsulations such as L2TP, GRE, etc.

Alternate Tunnel to AR (L2TPv3, IP-IP, CAPWAP, etc.)

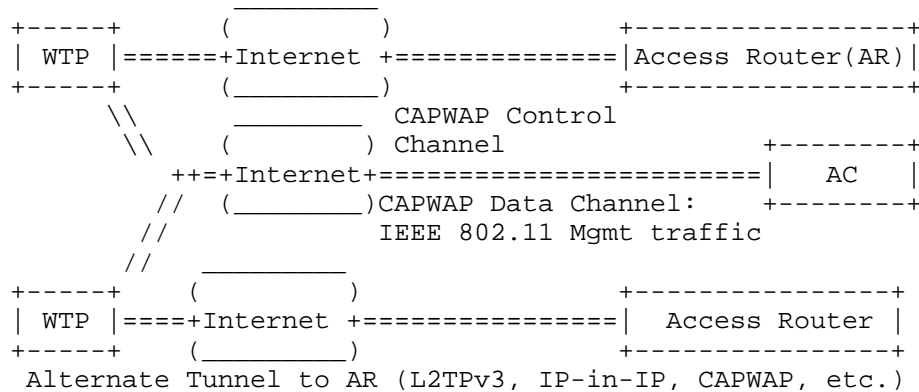


Figure 3: Centralized Control with Alternate Tunnel for Data

The WTP may support widely used encapsulation types such as L2TP, L2TPv3, IP-in-IP, IP/GRE, etc. The WTP advertises the different alternate tunnel encapsulation types it can support. The AC configures one of the advertised types. As shown in the figure there is a CAPWAP control and data channel between the WTP and AC. The CAPWAP data channel carries the stations' management traffic as in the case of the locally bridged system. The main reason to maintain a CAPWAP data channel is to maintain similarity with the locally bridged system. The WTP maintains three tunnels: CAPWAP Control, CAPWAP Data, and another alternate tunnel for the data frames. The data frames are transported by an alternate tunnel between the WTP and a tunnel termination point such as an Access Router. This specification describes how the alternate tunnel can be established. The specification defines message elements for the WTP to advertise support for alternate tunnel encapsulation, for the AC to configure alternate tunnel encapsulation, and for the WTP to report failure of the alternate tunnel.

The alternate tunnel encapsulation also supports the third-party WLAN service provider scenario (i.e. Virtual Network Operator, VNO). Under this scenario, the WLAN provider owns the WTP and AC resources, while the VNOs can rent the WTP resources from the WLAN provider for network access. The AC belonging to the WLAN service provider manages the WTPs in the centralized mode.

As shown in Figure 4, VNO 1&2 don't possess the network access resources, however they provide services by acquiring resources from the WLAN provider. Since a WTP is capable of supporting up to 16 Service Set Identifiers (SSIDs), the WLAN provider may provide network access service for different providers with different SSIDs. For example, SSID1 is advertised by the WTP for VNO1; while SSID2 is advertised by the WTP for VNO2. Therefore the data traffic from the user can be directly steered to the corresponding access router of the VNO who owns that user. As shown in Figure 4, AC can notify multiple AR addresses for load balancing or redundancy.

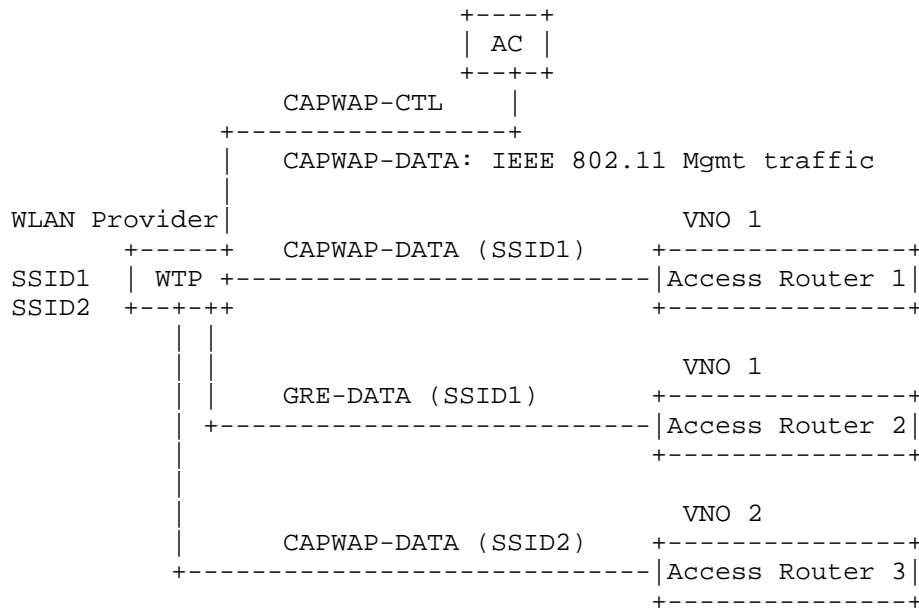


Figure 4: Third-party WLAN Service Provider

### 1.1. Conventions used in this document

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 [RFC2119].

### 1.2. Terminology

**Station (STA):** A device that contains an IEEE 802.11 conformant medium access control (MAC) and physical layer (PHY) interface to the wireless medium (WM).

**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.

**Access Router (AR):** A specialized router usually residing at the edge or boundary of a network. This router ensures the connectivity of its network with external networks, a wide area network or the Internet.

**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.

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. In certain WTP modes, the CAPWAP Data Channel only transports IEEE 802.11 management frames and not the data plane (user traffic).

### 1.3. History of the document

This document was started to accommodate Service Providers' need of a more flexible deployment mode with alternative tunnels [RFC7494]. Experiments and tests have been done for this alt-tunnel network infrastructure. However important, the deployment of relevant technology is yet to complete. This experimental document is intended to serve as an archival record for any future work as to the operational and deployment requirements.

## 2. Alternate Tunnel Encapsulation Overview



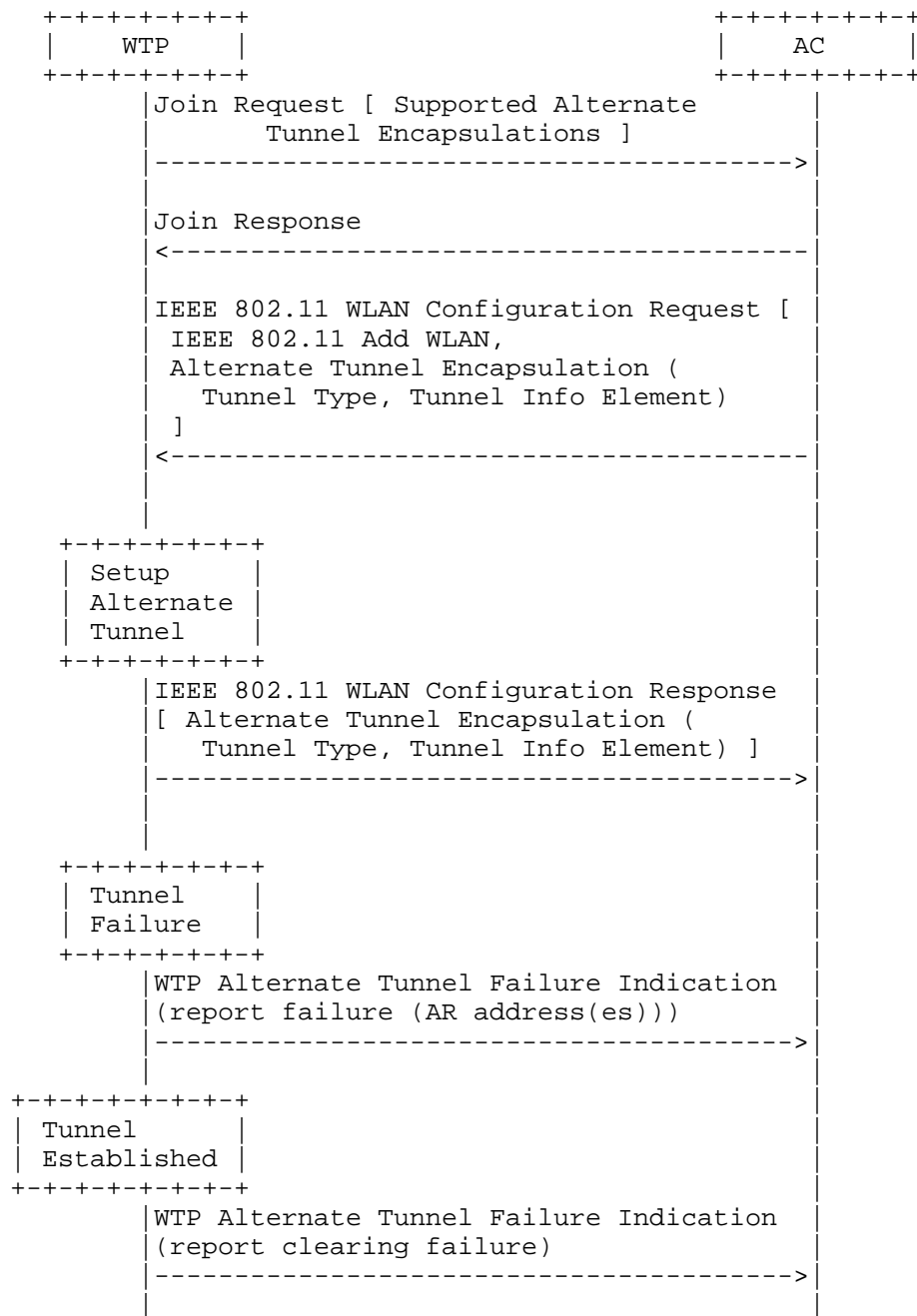


Figure 5: Setup of Alternate Tunnel

The above example describes how the alternate tunnel encapsulation may be established. When the WTP joins the AC, it should indicate its alternate tunnel encapsulation capability. The AC determines whether an alternate tunnel configuration is required. If an appropriate alternate tunnel type is selected, then the AC provides the alternate tunnel encapsulation message element containing the tunnel type and a tunnel-specific information element. The tunnel-specific information element, for example, may contain information like the IP address of the tunnel termination point. The WTP sets up the alternate tunnel using the alternate tunnel encapsulation message element.

Since AC can configure a WTP with more than one AR available for the WTP to establish the data tunnel(s) for user traffic, it may be useful for the WTP to communicate the selected AR. To enable this, the IEEE 802.11 WLAN Configuration Response may carry the alternate tunnel encapsulation message element containing the AR list element corresponding to the selected AR as shown in Figure 5.

On detecting a tunnel failure, WTP SHALL forward data frames to the AC and discard the frames. In addition, WTP may dissociate existing clients and refuse association requests from new clients. Depending on the implementation and deployment scenario, the AC may choose to reconfigure the WLAN (on the WTP) to a local bridging mode or to tunnel frames to the AC. When the WTP detects an alternate tunnel failure, the WTP informs the AC using a message element, WTP Alternate Tunnel Fail Indication (defined in this specification). It MAY be carried in the WTP Event Request message which is defined in [RFC5415].

The WTP also needs to notify the AC of which AR(s) are unavailable. Particularly, in the VNO scenario, the AC of the WLAN service provider needs to maintain the association of the AR addresses of the VNOs and SSIDs, and provide this information to the WTP for the purpose of load balancing or master-slave mode.

The message element has a status field that indicates whether the message denotes reporting a failure or the clearing of the previously reported failure.

For the case where AC is unreachable but the tunnel end point is still reachable, the WTP behavior is up to the implementation. For example, the WTP could either choose to tear down the alternate tunnel or let the existing user's traffic continue to be tunneled.

### 3. CAPWAP Protocol Message Elements Extensions

#### 3.1. Supported Alternate Tunnel Encapsulations

This message element is sent by a WTP to communicate its capability to support alternate tunnel encapsulations. The message element contains the following fields:

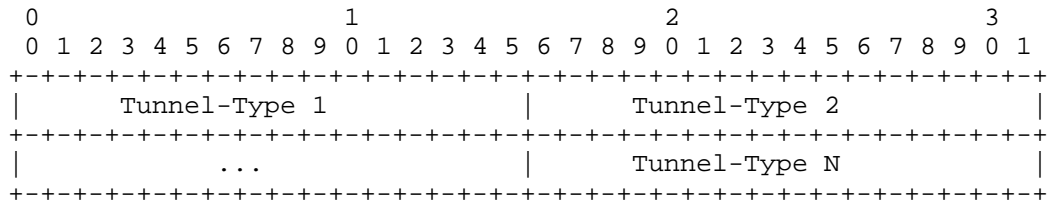


Figure 6: Supported Alternate Tunnel Encapsulations

- o Type: <IANA-1> for Supported Alternate Tunnel Encapsulations
- o Length: The length in bytes, two bytes for each Alternative tunnel type that is included
- o Tunnel-Type: This is identified by value defined in Section 3.2. There may be one or more Tunnel-Types as shows in Figure 6.

#### 3.2. Alternate Tunnel Encapsulations Type

This message element can be sent by the AC. This message element allows the AC to select the alternate tunnel encapsulation. This message element may be provided along with the IEEE 802.11 Add WLAN message element. When the message element is present, the following fields of the IEEE 802.11 Add WLAN element SHALL be set as follows: MAC mode is set to 0 (Local MAC) and Tunnel Mode is set to 0 (Local Bridging). Besides, the message element can also be sent by the WTP to communicate the selected AR(s).

The message element contains the following fields:

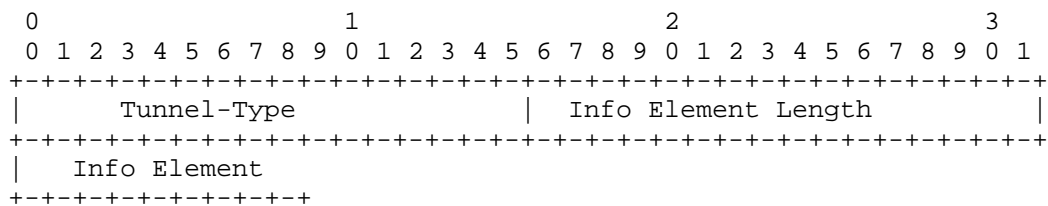


Figure 7: Alternate Tunnel Encapsulations Type

- o Type: <IANA-2> for Alternate Tunnel Encapsulation Type

- o Length: > 4
- o Tunnel-Type: The tunnel type is specified by a 2 byte value. This specification defines the values from zero (0) to six (6) as given below. The remaining values are reserved for future use.
  - \* 0: CAPWAP. This refers to a CAPWAP data channel described in [RFC5415] and [RFC5416].
  - \* 1: L2TP. This refers to tunnel encapsulation described in [RFC2661].
  - \* 2: L2TPv3. This refers to tunnel encapsulation described in [RFC3931].
  - \* 3: IP-in-IP. This refers to tunnel encapsulation described in [RFC2003].
  - \* 4: PMIPv6-UDP. This refers to the UDP encapsulation mode described in [RFC5844]. This encapsulation mode is the basic encapsulation mode and does not include the TLV header specified in section 7.2, of [RFC5845].
  - \* 5: GRE. This refers to GRE tunnel encapsulation as described in [RFC2784].
  - \* 6: GTPv1-U. This refers to GTPv1 user plane mode as described in [TS29281].
- o Info Element: This field contains tunnel specific configuration parameters to enable the WTP to setup the alternate tunnel. This specification provides details for this elements for CAPWAP, PMIPv6, and GRE. This specification reserves the tunnel type values for the key tunnel types and defines the most common message elements. It is anticipated that message elements for the other protocols (like L2TPv3, etc.) will be defined in other specifications in the future.

### 3.3. IEEE 802.11 WTP Alternate Tunnel Failure Indication

The WTP MAY include the Alternate Tunnel Failure Indication message in a WTP Event Request message to inform the AC about the status of the Alternate Tunnel. For the case where WTP establishes data tunnels with multiple ARs (e.g., under VNO scenario), the WTP needs to notify the AC of which AR(s) are unavailable. The message element contains the following fields:

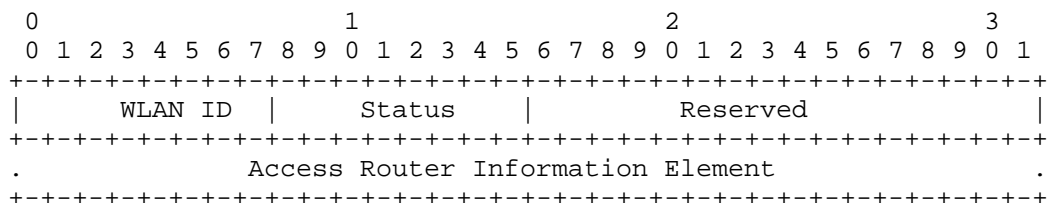


Figure 8: IEEE 802.11 WTP Alternate Tunnel Failure Indication

- o Type: <IANA-3> for IEEE 802.11 WTP Alternate Tunnel Failure Indication
- o Length: > 4
- o WLAN ID: An 8-bit value specifying the WLAN Identifier. The value MUST be between one (1) and 16.
- o Status: An 8-bit boolean indicating whether the radio failure is being reported or cleared. A value of zero is used to clear the event, while a value of one is used to report the event.
- o Reserved: MUST be set to a value of 0 and MUST be ignored by the receiver.
- o Access Router Information Element: IPv4 address or IPv6 address of the Access Router that terminates the alternate tunnel. The Access Router Information Elements allow the WTP to notify the AC of which AR(s) are unavailable.

#### 4. Alternate Tunnel Types

##### 4.1. CAPWAP based Alternate Tunnel

If the CAPWAP encapsulation is selected by the AC and configured by the AC to the WTP, the Info Element field defined in Section 3.2 SHOULD contain the following information:

- o Access Router Information: IPv4 address or IPv6 address of the Access Router for the alternate tunnel.
- o Tunnel DTLS Policy: The CAPWAP protocol allows optional protection of data packets using DTLS. Use of data packet protection on a WTP is not mandatory but determined by the associated AC policy (This is consistent with the WTP behavior described in [RFC5415]).
- o IEEE 802.11 Tagging Mode Policy: It is used to specify how the CAPWAP data channel packet are to be tagged for QoS purposes (see [RFC5416] for more details).
- o CAPWAP Transport Protocol: The CAPWAP protocol supports both UDP and UDP-Lite (see [RFC3828]). When run over IPv4, UDP is used for the CAPWAP data channels. When run over IPv6, the CAPWAP data channel may use either UDP or UDP-lite.

The message element structure for CAPWAP encapsulation is shown in Figure 9:

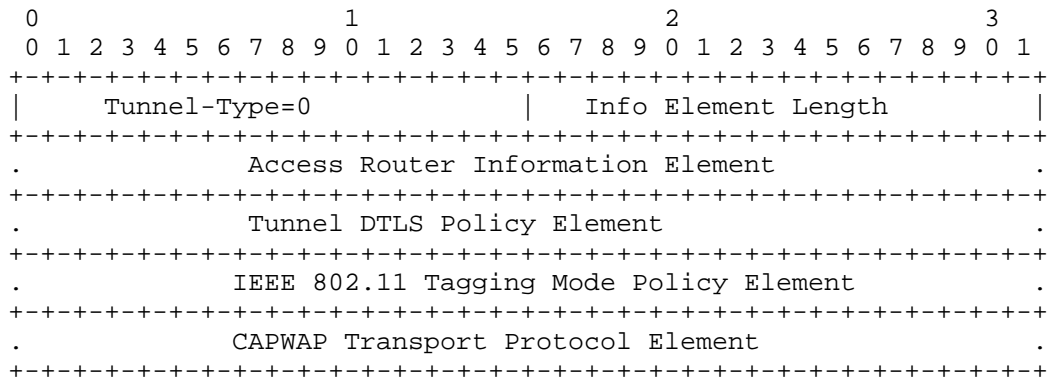


Figure 9: Alternate Tunnel Encapsulation - CAPWAP

#### 4.2. PMIPv6 based Alternate Tunnel

Proxy Mobile IPv6 (PMIPv6) (defined in [RFC5213]) based user plane can also be used as alternate tunnel encapsulation between the WTP and the AR. In this scenario, a WTP acts as the Mobile Access Gateway (MAG) function that manages the mobility-related signaling for a station that is attached to the WTP IEEE 802.11 radio access. The Local Mobility Anchor (LMA) function is at the AR. If PMIPv6 UDP encapsulation is selected by the AC and configured by the AC to a WTP, the Info Element field defined in Section 3.2 SHOULD contain the following information:

- o Access Router (acting as LMA) Information: IPv4 or IPv6 address for the alternate tunnel endpoint.

The message element structure for PMIPv6 encapsulation is shown in Figure 10:

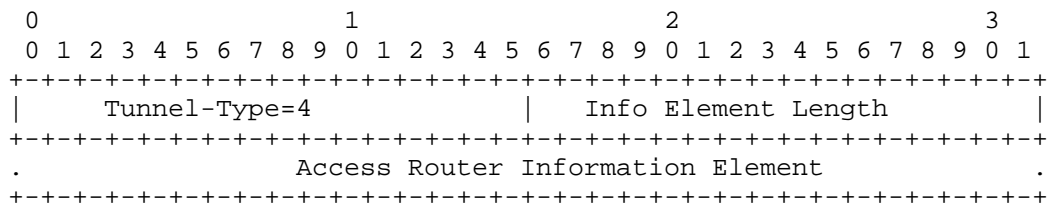


Figure 10: Alternate Tunnel Encapsulation - PMIPv6

### 4.3. GRE based Alternate Tunnel

Generic Routing Encapsulation (defined in [RFC2784]) mode based user plane can also be used as alternate tunnel encapsulation between the WTP and the AR. In this scenario, a WTP and the access router represent the two end points of the GRE tunnel. If GRE encapsulation is selected by the AC and configured by the AC to a WTP, the Info Element field defined in Section 3.2 SHOULD contain the following information:

- o Access Router Information: IPv4 or IPv6 address for the alternate tunnel endpoint.
- o GRE Key Information: The Key field is intended to be used for identifying an individual traffic flow within a tunnel [RFC2890].

The message element structure for GRE encapsulation is shown in Figure 11:

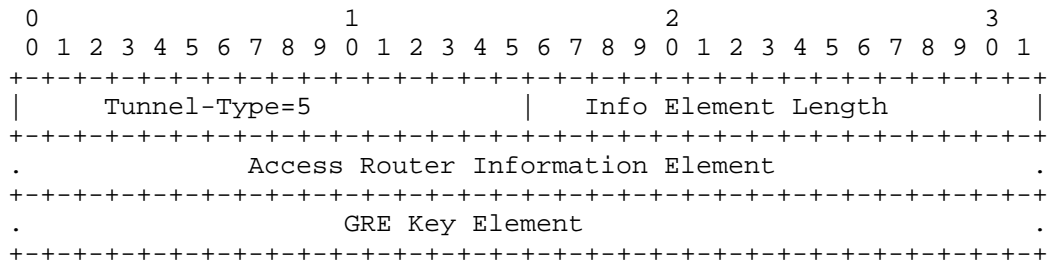


Figure 11: Alternate Tunnel Encapsulation - GRE

## 5. Alternate Tunnel Information Elements

This section defines the various elements described in Section 4.1, Section 4.2, and Section 4.3.

These information elements can only be included in the Alternate Tunnel Encapsulations Type message element, and the IEEE 802.11 WTP Alternate Tunnel Failure Indication message element as their sub-elements.

## 5.1. Access Router Information Elements

The Access Router Information Elements allow the AC to notify a WTP of which AR(s) are available for establishing a data tunnel. The AR information may be IPv4 address, or IPv6 address. This information element SHOULD be contained whatever the tunnel type is.

If the Alternate Tunnel Encapsulations Type message element is sent by the WTP to communicate the selected AR(s), this Access Router Information Element SHOULD be contained.

The following are the Access Router Information Elements defined in this specification. The AC can use one of them to notify the destination information of the data tunnel to the WTP. The Elements containing the AR IPv4 address MUST NOT be used if an IPv6 data channel with IPv6 transport is used.

#### 5.1.1. AR IPv4 List Element

This Element (see Figure 12) is used by the AC to configure a WTP with the AR IPv4 address available for the WTP to establish the data tunnel for user traffic.

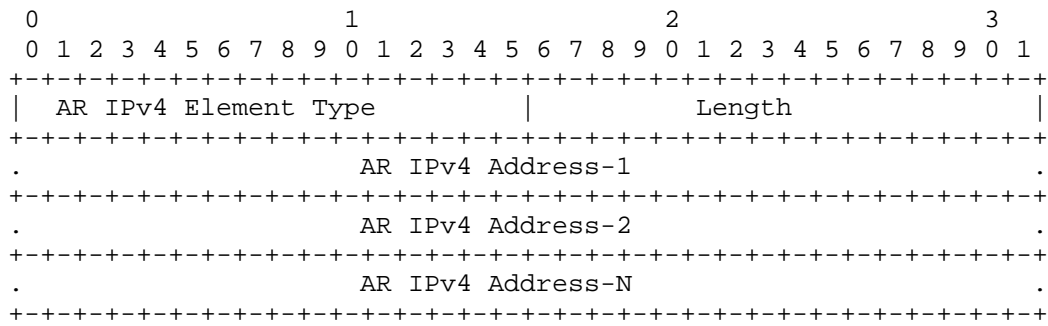


Figure 12: AR IPv4 List Element

Type: 0

Length: This refers to the total length in octets of the element excluding the Type and Length fields.

AR IPv4 Address: The IPv4 address of the AR. At least one IPv4 address SHALL be present. Multiple addresses may be provided for load balancing or redundancy.

#### 5.1.2. AR IPv6 List Element

This Element (see Figure 13) is used by the AC to configure a WTP with the AR IPv6 address available for the WTP to establish the data tunnel for user traffic.



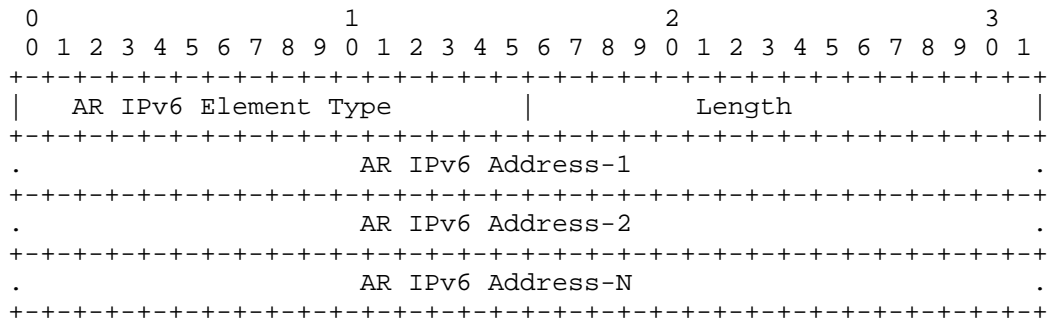


Figure 13: AR IPv6 List Element

Type: 1

Length: This refers to the total length in octets of the element excluding the Type and Length fields.

AR IPv6 Address: The IPv6 address of the AR. At least one IPv6 address SHALL be present. Multiple addresses may be provided for load balancing or redundancy.

## 5.2. Tunnel DTLS Policy Element

The AC distributes its DTLS usage policy for the CAPWAP data tunnel between a WTP and the AR. There are multiple supported options, represented by the bit field below as defined in AC Descriptor message elements. The WTP MUST abide by one of the options for tunneling user traffic with AR. The Tunnel DTLS Policy Element obeys the definition in [RFC5415]. If, for reliability reasons, the AC has provided more than one AR address in the Access Router Information Element, the same Tunnel DTLS Policy (the last one in Figure 14) is generally applied for all tunnels associated with those ARs. Otherwise, Tunnel DTLS Policy MUST be bonded together with each of the Access Router Information Elements, and the WTP will enforce the independent tunnel DTLS policy for each tunnel with a specific AR.

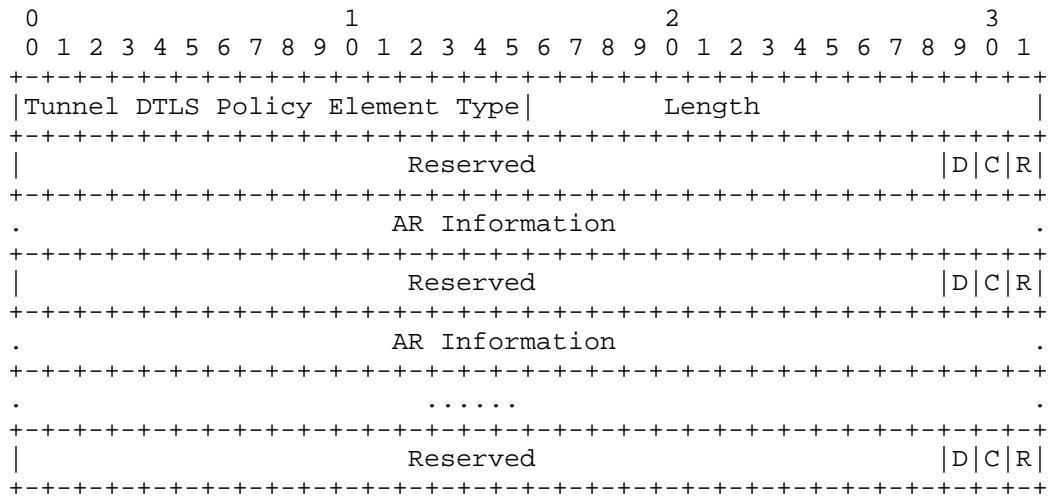


Figure 14: Tunnel DTLS Policy Element

Type: 2

Length: This refers to the total length in octets of the element excluding the Type and Length fields.

Reserved: A set of reserved bits for future use. All implementations complying with this protocol MUST set to zero any bits that are reserved in the version of the protocol supported by that implementation. Receivers MUST ignore all bits not defined for the version of the protocol they support.

D: DTLS-Enabled Data Channel Supported (see [RFC5415]).

C: Clear Text Data Channel Supported (see [RFC5415]).

R: A reserved bit for future use (see [RFC5415]).

AR Information: This means Access Router Information Element. In this context, each address in AR information MUST be one of previously specified AR addresses.

The last element having no AR Information in Figure 14 is the default tunnel DTLS policy, and provides options for any address not previously mentioned. Therefore, the AR information field here is optional. If all ARs share the same tunnel DTLS policy, in this element, there will not be AR information field and its specific tunnel DTLS policy.

### 5.3. IEEE 802.11 Tagging Mode Policy Element

In 802.11 networks, IEEE 802.11 Tagging Mode Policy Element is used to specify how the WTP applies the QoS tagging policy when receiving the packets from stations on a particular radio. When the WTP sends out the packet to data channel to the AR(s), the packets have to be tagged for QoS purposes (see [RFC5416]).

The IEEE 802.11 Tagging Mode Policy abides the IEEE 802.11 WTP Quality of Service defined in Section 6.22 of [RFC5416].

If, for reliability reasons, the AC has provided more than one AR address in the Access Router Information Element, the same IEEE 802.11 Tagging Mode Policy (the last one in Figure 15) is generally applied for all tunnels associated with those ARs. Otherwise, IEEE 802.11 Tagging Mode Policy MUST be bonded together with each of the Access Router Information Elements, and the WTP will enforce the independent IEEE 802.11 Tagging Mode Policy for each tunnel with a specific AR.

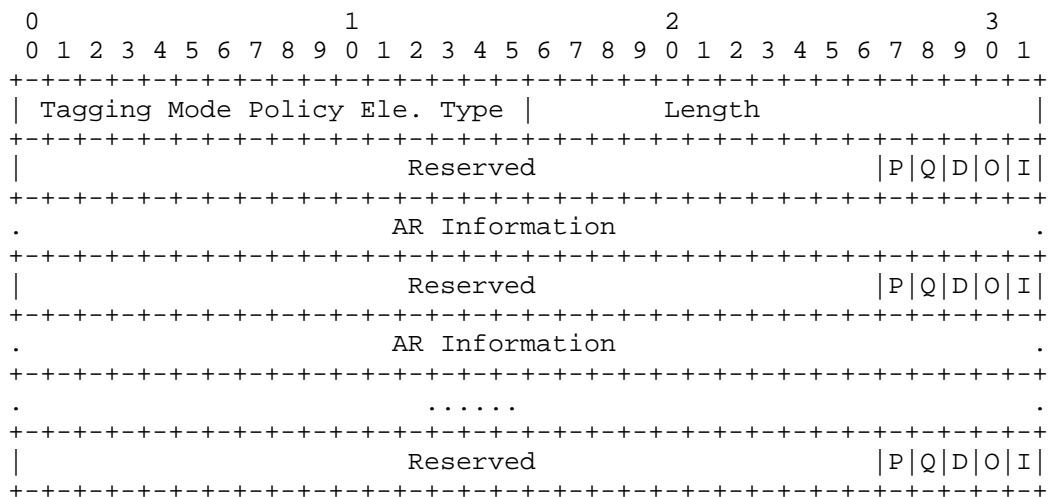


Figure 15: IEEE 802.11 Tagging Mode Policy Element

Type: 3

Length: This refers to the total length in octets of the element excluding the Type and Length fields.

Reserved: A set of reserved bits for future use.

P: When set, the WTP is to employ the 802.1p QoS mechanism (see [RFC5416]).

Q: When the 'P' bit is set, the 'Q' bit is used by the AC to communicate to the WTP how 802.1p QoS is to be enforced (see [RFC5416]).

D: When set, the WTP is to employ the DSCP QoS mechanism (see [RFC5416]).

O: When the 'D' bit is set, the 'O' bit is used by the AC to communicate to the WTP how DSCP QoS is to be enforced on the outer (tunneled) header (see [RFC5416]).

I: When the 'D' bit is set, the 'I' bit is used by the AC to communicate to the WTP how DSCP QoS is to be enforced on the station's packet (inner) header (see [RFC5416]).

AR Information: This means Access Router Information Element. In this context, each address in AR information MUST be one of previously specified AR addresses.

The last element having no AR Information in Figure 15 is the default IEEE 802.11 Tagging Mode Policy, and provides options for any address not previously mentioned. Therefore, the AR information field here is optional. If all ARs share the same IEEE 802.11 Tagging Mode Policy, in this element, there will not be AR information field and its specific IEEE 802.11 Tagging Mode Policy.

#### 5.4. CAPWAP Transport Protocol Element

The CAPWAP data tunnel supports both UDP and UDP-Lite (see [RFC3828]). When run over IPv4, UDP is used for the CAPWAP data channels. When run over IPv6, the CAPWAP data channel may use either UDP or UDP-lite. The AC specifies and configures the WTP for which transport protocol is to be used for the CAPWAP data tunnel.

The CAPWAP Transport Protocol Element abides the definition in Section 4.6.14 of [RFC5415].

If, for reliability reasons, the AC has provided more than one AR address in the Access Router Information Element, the same CAPWAP Transport Protocol (the last one in Figure 16) is generally applied for all tunnels associated with those ARs. Otherwise, CAPWAP Transport Protocol MUST be bonded together with each of the Access Router Information Elements, and the WTP will enforce the independent CAPWAP Transport Protocol for each tunnel with a specific AR.

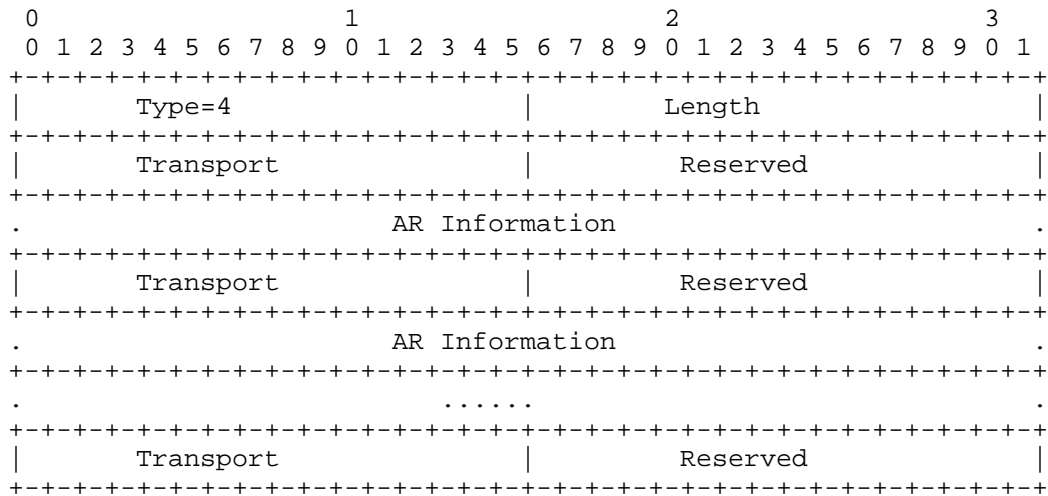


Figure 16: CAPWAP Transport Protocol Element

Type: 4

Length: 1

Transport: The transport to use for the CAPWAP Data channel. The following enumerated values are supported:

1 - UDP-Lite: The UDP-Lite transport protocol is to be used for the CAPWAP Data channel. Note that this option MUST NOT be used if the CAPWAP Control channel is being used over IPv4 and AR address is IPv4 contained in the AR Information Element.

2 - UDP: The UDP transport protocol is to be used for the CAPWAP Data channel.

AR Information: This means Access Router Information Element. In this context, each address in AR information MUST be one of previously specified AR addresses.

The last element having no AR Information in Figure 16 is the default CAPWAP Transport Protocol, and provides options for any address not previously mentioned. Therefore, the AR information field here is optional. If all ARs share the same CAPWAP Transport Protocol, in this element, there will not be AR information field and its specific CAPWAP Transport Protocol.

### 5.5. GRE Key Element

If a WTP receives the GRE Key Element in the Alternate Tunnel Encapsulation message element for GRE selection, the WTP MUST insert the GRE Key to the encapsulation packet (see [RFC2890]). An AR acting as decapsulating tunnel endpoint identifies packets belonging to a traffic flow based on the Key value.

The GRE Key Element field contains a four octet number defined in [RFC2890].

If, for reliability reasons, the AC has provided more than one AR address in the Access Router Information Element, a GRE Key Element MAY be bonded together with each of the Access Router Information Elements, and the WTP will enforce the independent GRE Key for each tunnel with a specific AR.

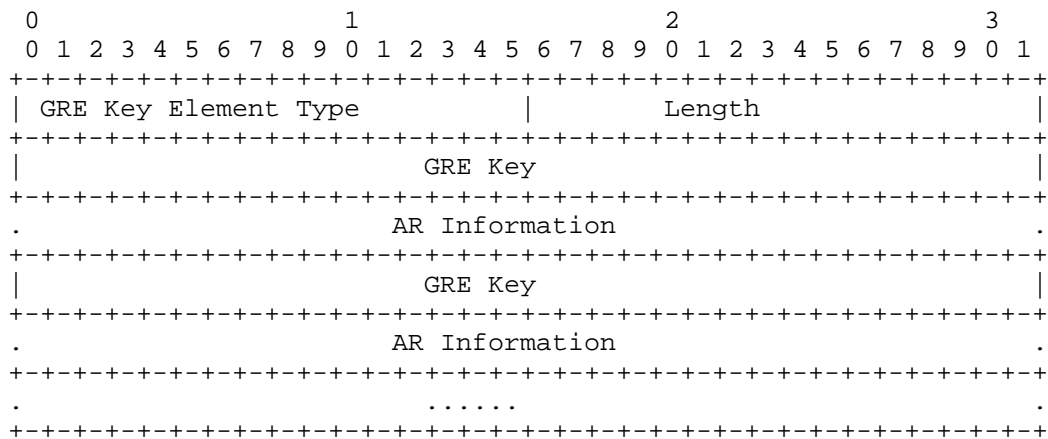


Figure 17: GRE Key Element

Type: 5

Length: This refers to the total length in octets of the element excluding the Type and Length fields.

GRE Key: The Key field contains a four octet number which is inserted by the WTP according to [RFC2890].

AR Information: This means Access Router Information Element. In this context, it SHOULD be restricted to a single address, and MUST be the address of one of previously specified AR addresses.

Any address not explicitly mentioned here does not have a GRE key.

## 5.6. IPv6 MTU Element

If AC has chosen a tunneling mechanism based on IPv6, it SHOULD support the minimum IPv6 MTU requirements [RFC8200]. This issue is described in [I-D.ietf-intarea-tunnels]. AC SHOULD inform the WTP about the IPv6 MTU information in the "Tunnel Info Element" field.

If, for reliability reasons, the AC has provided more than one AR address in the Access Router Information Element, an IPv6 MTU Element MAY be bonded together with each of the Access Router Information Elements, and the WTP will enforce the independent IPv6 MTU for each tunnel with a specific AR.

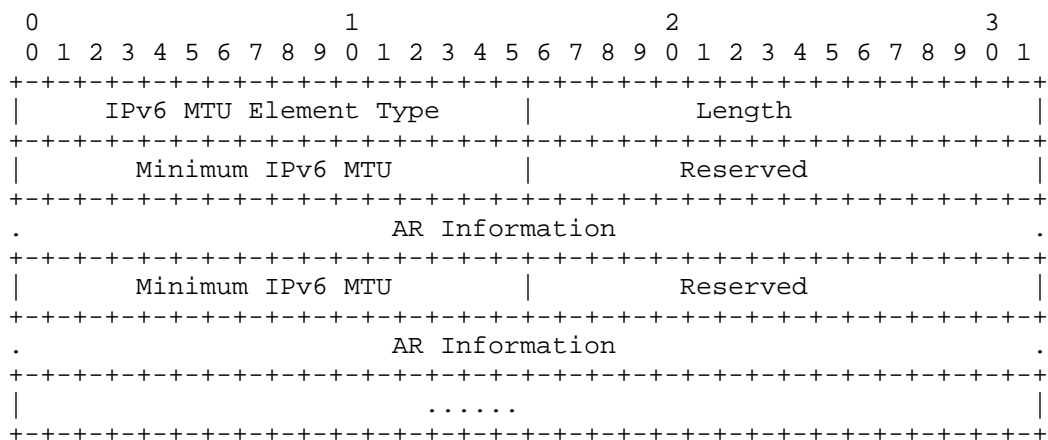


Figure 18: IPv6 MTU Element

Type: 6

Length: This refers to the total length in octets of the element excluding the Type and Length fields.

Minimum IPv6 MTU: The field contains a two octet number indicate the minimum IPv6 MTU in the tunnel.

AR Information: This means Access Router Information Element. In this context, each address in AR information MUST be one of previously specified AR addresses.

## 6. IANA Considerations

This document requires the following IANA considerations.

- o <IANA-1>. This specification defines the Supported Alternate Tunnel Encapsulations Type message element in Section 3.1. This element needs to be registered in the existing CAPWAP Message Element Type registry, defined in [RFC5415]. The Type value for this element needs to be between 1 and 1023 (see Section 15.7 in [RFC5415]).
- o <IANA-2>. This specification defines the Alternate Tunnel Encapsulations Type message element in Section 3.2. This element needs to be registered in the existing CAPWAP Message Element Type registry, defined in [RFC5415]. The Type value for this element needs to be between 1 and 1023.
- o <IANA-3>. This specification defines the IEEE 802.11 WTP Alternate Tunnel Failure Indication message element in Section 3.3. This element needs to be registered in the existing CAPWAP Message Element Type registry, defined in [RFC5415]. The Type value for this element needs to be between 1024 and 2047.
- o Alternate Tunnel-Types Registry: This specification defines the Alternate Tunnel Encapsulations Type message element. This element contains a field Tunnel-Type. The namespace for the field is 16 bits (0-65535). This specification defines values, zero (0) through six (6) and can be found in Section 3.2. Future allocations of values in this name space are to be assigned by IANA using the "Specification Required" policy. IANA needs to create a registry called CAPWAP Alternate Tunnel-Types. The registry format is given below.

Tunnel-Type	Type Value	Reference
CAPWAP	0	[RFC5415],[RFC5416]
L2TP	1	[RFC2661]
L2TPv3	2	[RFC3931]
IP-IP	3	[RFC2003]
PMIPv6-UDP	4	[RFC5844]
GRE	5	[RFC2784]
GTPv1-U	6	[3GPP TS 29.281]

- o Alternate Tunnel Sub-elements Registry: This specification defines the Alternate Tunnel Sub-elements. Currently, these information elements can only be included in the Alternate Tunnel Encapsulations Type message element, and the IEEE 802.11 WTP Alternate Tunnel Failure Indication message element as their sub-elements. These information elements contains a Type field. The namespace for the field is 16 bits (0-65535). This specification defines values, zero (0) through six (6) in Section 5. This namespace is managed by IANA and assignments require an Expert Review.



Type	Type Value
AR IPv4 List	0
AR IPv6 List	1
Tunnel DTLS Policy	2
IEEE 802.11 Tagging Mode Policy	3
CAPWAP Transport Protocol	4
GRE Key	5
IPv6 MTU	6

## 7. Security Considerations

This document introduces three new CAPWAP WTP message elements. These elements are transported within CAPWAP Control messages as the existing message elements. Therefore, this document does not introduce any new security risks to the control plane compared to [RFC5415] and [RFC5416]. In the data plane, if the encapsulation type selected itself is not secured, it is suggested to protect the tunnel by using known secure methods, such as IPsec.

## 8. Contributors

The authors would like to thank Andreas Schultz, Hong Liu, Yifan Chen, Chunju Shao, Li Xue, Jianjie You, Jin Li, Joe Touch, Alexey Melnikov, Kathleen Moriarty, Mirja Kuehlewind, Catherine Meadows, and Paul Kyzivat for their valuable comments.

## 9. References

### 9.1. Normative References

- [RFC2003] Perkins, C., "IP Encapsulation within IP", RFC 2003, DOI 10.17487/RFC2003, October 1996, <<https://www.rfc-editor.org/info/rfc2003>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC2661] Townsley, W., Valencia, A., Rubens, A., Pall, G., Zorn, G., and B. Palter, "Layer Two Tunneling Protocol "L2TP"", RFC 2661, DOI 10.17487/RFC2661, August 1999, <<https://www.rfc-editor.org/info/rfc2661>>.
- [RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, DOI 10.17487/RFC2784, March 2000, <<https://www.rfc-editor.org/info/rfc2784>>.

- [RFC2890] Dommety, G., "Key and Sequence Number Extensions to GRE", RFC 2890, DOI 10.17487/RFC2890, September 2000, <<https://www.rfc-editor.org/info/rfc2890>>.
- [RFC3828] Larzon, L-A., Degermark, M., Pink, S., Jonsson, L-E., Ed., and G. Fairhurst, Ed., "The Lightweight User Datagram Protocol (UDP-Lite)", RFC 3828, DOI 10.17487/RFC3828, July 2004, <<https://www.rfc-editor.org/info/rfc3828>>.
- [RFC3931] Lau, J., Ed., Townsley, M., Ed., and I. Goyret, Ed., "Layer Two Tunneling Protocol - Version 3 (L2TPv3)", RFC 3931, DOI 10.17487/RFC3931, March 2005, <<https://www.rfc-editor.org/info/rfc3931>>.
- [RFC5415] Calhoun, P., Ed., Montemurro, M., Ed., and D. Stanley, Ed., "Control And Provisioning of Wireless Access Points (CAPWAP) Protocol Specification", RFC 5415, DOI 10.17487/RFC5415, March 2009, <<https://www.rfc-editor.org/info/rfc5415>>.
- [RFC5416] Calhoun, P., Ed., Montemurro, M., Ed., and D. Stanley, Ed., "Control and Provisioning of Wireless Access Points (CAPWAP) Protocol Binding for IEEE 802.11", RFC 5416, DOI 10.17487/RFC5416, March 2009, <<https://www.rfc-editor.org/info/rfc5416>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.

## 9.2. Informative References

- [I-D.ietf-intarea-tunnels] Touch, J. and M. Townsley, "IP Tunnels in the Internet Architecture", draft-ietf-intarea-tunnels-08 (work in progress), January 2018.
- [RFC5213] Gundavelli, S., Ed., Leung, K., Devarapalli, V., Chowdhury, K., and B. Patil, "Proxy Mobile IPv6", RFC 5213, DOI 10.17487/RFC5213, August 2008, <<https://www.rfc-editor.org/info/rfc5213>>.
- [RFC5844] Wakikawa, R. and S. Gundavelli, "IPv4 Support for Proxy Mobile IPv6", RFC 5844, DOI 10.17487/RFC5844, May 2010, <<https://www.rfc-editor.org/info/rfc5844>>.

- [RFC5845] Muhanna, A., Khalil, M., Gundavelli, S., and K. Leung, "Generic Routing Encapsulation (GRE) Key Option for Proxy Mobile IPv6", RFC 5845, DOI 10.17487/RFC5845, June 2010, <<https://www.rfc-editor.org/info/rfc5845>>.
- [RFC7494] Shao, C., Deng, H., Pazhyannur, R., Bari, F., Zhang, R., and S. Matsushima, "IEEE 802.11 Medium Access Control (MAC) Profile for Control and Provisioning of Wireless Access Points (CAPWAP)", RFC 7494, DOI 10.17487/RFC7494, April 2015, <<https://www.rfc-editor.org/info/rfc7494>>.
- [TS29281] "3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; General Packet Radio System (GPRS) Tunneling Protocol User Plane (GTPv1-U)", 2016.

## Authors' Addresses

Rong Zhang  
China Telecom  
No.109 Zhongshandadao avenue  
Guangzhou 510630  
China

Email: [zhangr@gsta.com](mailto:zhangr@gsta.com)

Rajesh S. Pazhyannur  
Cisco  
170 West Tasman Drive  
San Jose, CA 95134  
USA

Email: [rpazhyan@cisco.com](mailto:rpazhyan@cisco.com)

Sri Gundavelli  
Cisco  
170 West Tasman Drive  
San Jose, CA 95134  
USA

Email: [sgundave@cisco.com](mailto:sgundave@cisco.com)

Zhen Cao  
Huawei  
Xinxi Rd. 3  
Beijing 100085  
China

Email: [zhencao.ietf@gmail.com](mailto:zhencao.ietf@gmail.com)

Hui Deng  
Huawei  
Xinxi Rd. 3  
Beijing 100085  
China

Email: [denghui02@gmail.com](mailto:denghui02@gmail.com)

Zongpeng Du  
Huawei  
No.156 Beiqing Rd. Z-park, HaiDian District  
Beijing 100095  
China

Email: [duzongpeng@huawei.com](mailto:duzongpeng@huawei.com)

OPSAWG  
Internet-Draft  
Updates: 5416 (if approved)  
Intended status: Standards Track  
Expires: January 7, 2016

Y. Chen  
China Mobile  
D. Liu

H. Deng  
China Mobile  
Lei. Zhu  
Huawei  
July 6, 2015

CAPWAP Extension for 802.11n and Power/channel Autoconfiguration  
draft-ietf-opawg-capwap-extension-06

Abstract

The CAPWAP binding for 802.11 is specified by RFC5416 and it was based on IEEE 802-11.2007 standard. Several new amendments of 802.11 have been published since RFC5416 was published in 2009. 802.11n is one of those amendments and it has been widely used in real deployment. This document extends the CAPWAP binding for 802.11 to support 802.11n and also defines a power and channel auto configuration extension.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 7, 2016.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

(<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Terminology . . . . .	3
3. CAPWAP 802.11n Support . . . . .	3
3.1. CAPWAP Extension for 802.11n Support . . . . .	4
3.1.1. 802.11n Radio Capability Information . . . . .	4
3.1.2. 802.11n Radio Configuration Message Element . . . . .	4
3.1.3. 802.11n Station Information . . . . .	6
4. Power and Channel Autoconfiguration . . . . .	7
4.1. Channel Autoconfiguration When WTP Power On . . . . .	7
4.2. Power Configuration When WTP Power On . . . . .	8
4.3. Channel/Power Auto Adjustment . . . . .	8
4.3.1. IEEE 802.11 Scan Parameters Message Element . . . . .	9
4.3.2. IEEE 802.11 Scan Channel Bind Message Element . . . . .	11
4.3.3. IEEE 802.11 Channel Scan Report . . . . .	12
4.3.4. IEEE 802.11 WTP Neighbor Report . . . . .	14
5. Security Considerations . . . . .	15
6. IANA Considerations . . . . .	15
7. Contributors . . . . .	15
8. Acknowledgements . . . . .	16
9. Normative References . . . . .	16
Authors' Addresses . . . . .	17

## 1. Introduction

IEEE Std 802.11n[TM]-2009 [IEEE 802.11n.2009] was published in 2009 as an amendment to the IEEE 802.11-2007 standard to improve network throughput. The maximum data rate increases to 600Mbps. In the physical layer, 802.11n uses Orthogonal Frequency Division Multiplexing (OFDM) and Multiple Input/Multiple Output (MIMO) to achieve the high throughput. 802.11n uses multiple antennas to form an antenna array which can be dynamically adjusted to improve the signal strength and extend the coverage.

Capabilities of 802.11n such as radio capability, radio configuration and station information need to be supported by CAPWAP control messages. The necessary extensions for this purpose are introduced in Section 3 and specified in Section 4.

For IEEE 802.11 in general, it is desirable to be able to support power and channel auto reconfiguration. Extensions for this purpose are specified in Section 5.

## 2. Terminology

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 [RFC2119].

This document uses the following abbreviations:

- AC Access Controller
- A-MSDU Aggregate MAC Service Data Unit
- A-MPDU Aggregate MAC Protocol Data Unit
- AC Access Controller
- GI Guard Interval
- MCS Maximum Modulation and Coding Scheme
- MIMO Multiple Input/Multiple Output
- MPDU MAC Protocol Data Unit
- MSDU MAC Service Data Unit
- OFDM Orthogonal Frequency Division Multiplexing
- TSF timing synchronization function
- WTP Wireless Termination Point

## 3. CAPWAP 802.11n Support

802.11n supports three modes of channel usage: 20MHz mode, 40MHz mode and mixed mode. 802.11n has a new feature called channel binding. It can bind two adjacent 20MHz channel to one 40MHz channel to improve the throughput. If using 40MHz channel configuration there will be only one non-overlapping channel in the 2.4GHz band. In the large scale deployment scenario, the operator needs to use 20MHz channel configuration in the 2.4GHz band to allow more non-overlapping channels.

In the MAC layer, a new feature of 802.11n is Short Guard Interval (GI). 802.11a/g uses an 800ns guard interval between the adjacent information symbols. In 802.11n, the GI can be configured to 400ns under good wireless conditions.

Another feature in the 802.11 MAC layer is Block ACK. 802.11n can use one ACK frame to acknowledge receipt of several MAC Protocol Data Units (MPDUs).

CAPWAP needs to be extended to support the above new 802.11n features. CAPWAP should allow the access controller to know the supported 802.11n features and the access controller should be able

to configure the different channel binding modes. This document defines extensions of the CAPWAP 802.11 binding to support 802.11n features.

### 3.1. CAPWAP Extension for 802.11n Support

Three 802.11n features need to be supported by CAPWAP 802.11 binding: 802.11n radio capability, 802.11n radio configuration and station information. This section defines the extension of the current CAPWAP 802.11 binding to support the 802.11n features.

#### 3.1.1. 802.11n Radio Capability Information

[RFC5416] defines the IEEE 802.11 binding for the CAPWAP protocol. It defines the IEEE 802.11 Information Element, which is used to communicate any information element (IE) defined in the IEEE 802.11 protocol. This document specifies that the IEEE 802.11 Information Element defined in section 6.6 of [RFC5416] SHALL be used to transport the IEEE 802.11 HT information element defined in section 8.4.2.58 of [IEEE-802.11.2012]. The HT IE MAY in this way be included in CAPWAP Configuration Status Request/Response messages.

#### 3.1.2. 802.11n Radio Configuration Message Element

The 802.11n Radio Configuration message element is used by the AC to provide IEEE 802.11n-specific configuration for a Radio on the WTP, and by the WTP to deliver its radio configuration to the AC. This supplements the IEEE 802.11 WTP WLAN Radio Configuration message element defined in [RFC5416]. The format of the 802.11n Radio Configuration message element is shown in Figure 1. The 802.11n Radio Configuration message element MAY be included in the CAPWAP Configuration Update Request/Response message.

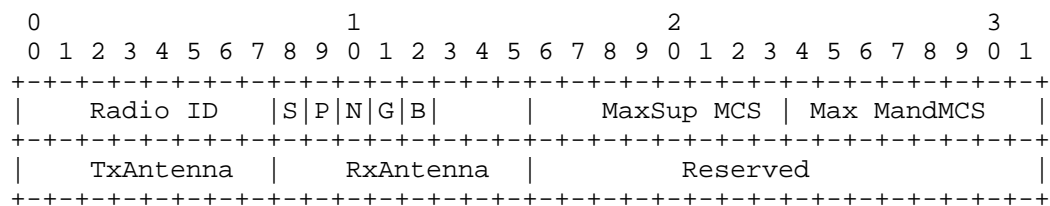


Figure 1: 802.11n Radio Configuration Message Element

Type: TBD1 for 802.11n Radio Configuration Message Element.

Length: 16.



Radio ID: An 8-bit value representing the radio, whose value is between one (1) and 31.

S bit: A-MSDU configuration: Enable/disable Aggregate MAC Service Data Unit (A-MSDU). Set to 0 if disabled. Set to 1 if enabled.

P bit: A-MPDU configuration: Enable/disable Aggregate MAC Protocol Data Unit (A-MPDU). Set to 0 if disabled. Set to 1 if enabled.

N bit: 11n Only configuration: Whether to allow only 11n user access. Set to 0 if non-802.11n user access is allowed. Set to 1 if non-802.11n user access is not allowed.

G bit: Short GI configuration: Set to 0 if Short Guard Interval is disabled. Set to 1 if enabled.

B bit: Bandwidth binding mode configuration: Set to 0 if 40MHz binding mode. Set to 1 if 20MHz binding mode.

Maximum supported MCS: Maximum Modulation and Coding Scheme (MCS) index. It indicates the maximum MCS index that the WTP or the STA can support.

Max Mandatory MCS: Maximum Mandatory Modulation and Coding Scheme (MCS) index. Mandatory rates must be supported by the WTP and the STA that want to associate with the WTP.

TxAntenna: Transmitting antenna configuration. Each TxAntenna bit represents a certain number of antennas. Set to 1 if enabled, set to 0 if disabled.

RxAntenna: Receiving antenna configuration. Each RxAntenna bit represents a certain number of antennas. Set to 1 if enabled, set to 0 if disabled.

The detail definition of TxAntenna/RxAntenna is as follows:

```

      0 1 2 3 4 5 6 7
+---+---+---+---+---+---+
| 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
+---+---+---+---+---+---+

```

Figure 2: Definition of TxAntenna/RxAntenna

Each bit when enabled will represent the number of antennas correspondent to that bit. Only one bit is allowed to be set to 1. For example, when the first bit is enabled, it represents 8 antennas.

### 3.1.3. 802.11n Station Information

The 802.11n Station Information message element is used to deliver IEEE 802.11n station policy from the AC to the WTP. The definition of the 802.11n Station Information message element is in figure 3. The format of 802.11n Station Information MAY be included in the CAPWAP Station Configuration Request message.

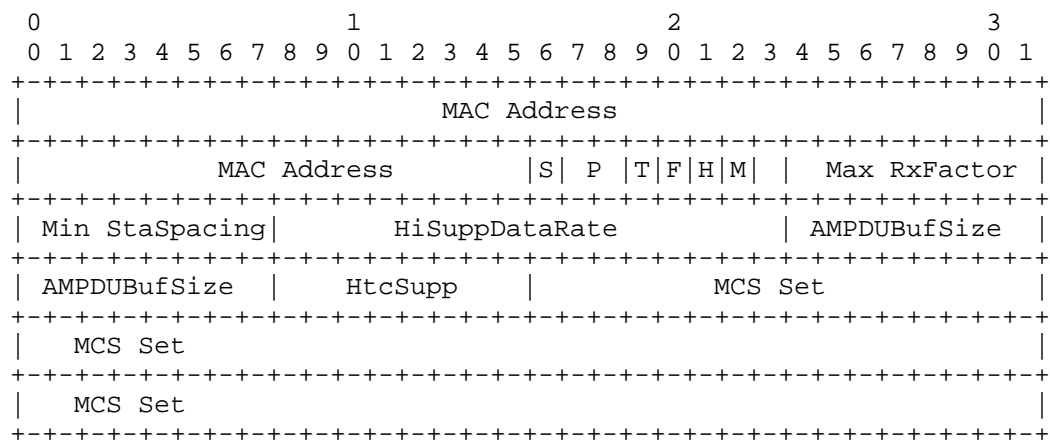


Figure 3: 802.11n Station Information

MAC Address: The station's MAC Address.

Type: TBD2 for 802.11 Station Information.

Length: 24.

S bit: Supporting bandwidth mode. 0x00: 20MHz bandwidth mode. 0x01: 40MHz bandwidth binding mode.

P flag: Power Saving mode: 0x00: Static. 0x01: Dynamic. 0x03: Do not support power saving mode.

T bit: Whether to support short GI in 20MHz bandwidth mode. 0x00: Do not support short GI. 0x01: Support short GI.

F bit: ShortGi40: Whether to support short GI in 40MHz bandwidth mode. 0x00: Do not support short GI. 0x01: Support short GI.

H bit: Whether Block Ack supports delay mode. 0x00: Do not support delay mode. 0x01: Support delay mode.

M bit: The maximal A-MSDU length. 0x00: 3839 bytes. 0x01: 7935 bytes.

Max RxFactor: The maximal receiving A-MPDU factor.

Min StaSpacing: Minimum MPDU Start Spacing.

HiSuppDataRate: Maximal transmission speed (Mbps).

AMPDUBufSize: A-MPDU buffer size (Byte).

HtcSupp: Whether to place HT headers on the packets forwarded from this station.

MCS Set: The MCS bitmap that the station supports.

#### 4. Power and Channel Autoconfiguration

Power and channel autoconfiguration could avoid potential radio interference and improve the WLAN performance. In general, the auto-configuration of radio power and channel could occur at two stages: when the WTP power on or during the WTP running time.

##### 4.1. Channel Autoconfiguration When WTP Power On

Power and channel auto reconfiguration avoids potential radio interference and improves the WLAN performance. In general, the auto-configuration of radio power and channel can occur at two stages: when the WTP powers on or while the WTP is in running state. When the WTP is powered-on, it needs to configure a proper channel. IEEE 802.11 Direct Sequence Control elements or IEEE 802.11 OFDM Control element defined in RFC5416 SHOULD be carried in the Configure Status Response message to offer WTP a channel at this stage. If the channel field of those information element is set to 0, the WTP will need to determine its channel by itself, otherwise the WTP SHOULD be configured according to the provided information element.

When the WTP determines its own channel configuration, it should first scan the channel information, then determine which channel it will work on and form a channel quality scan report. As shown in Figure 3, the AC can control the scanning process by sending the IEEE 802.11 Scan Parameters message element defined in Section 5.1 to the

WTP in a Configure Status Response message or in a WTP Configure Update Request message. The WTP will send the channel quality report to the AC using the WTP Event Request message.

AC will determine whether to change the channel configuration based on the received channel quality report. The AC MAY use a IEEE 802.11 Direct Sequence Control or IEEE 802.11 OFDM Control message element carried by the configure Update Request message to configure a new channel for the WTP.

#### 4.2. Power Configuration When WTP Power On

The IEEE 802.11 Tx Power message element defined in section 6.18 of [RFC5416] is used by the AC to control the transmission power of the WTP. The 802.11 Tx Power information element is carried in the Configure Status Response message or in the Configure Update Request message.

#### 4.3. Channel/Power Auto Adjustment

The Channel Scan Procedure is illustrated by the figure 4.

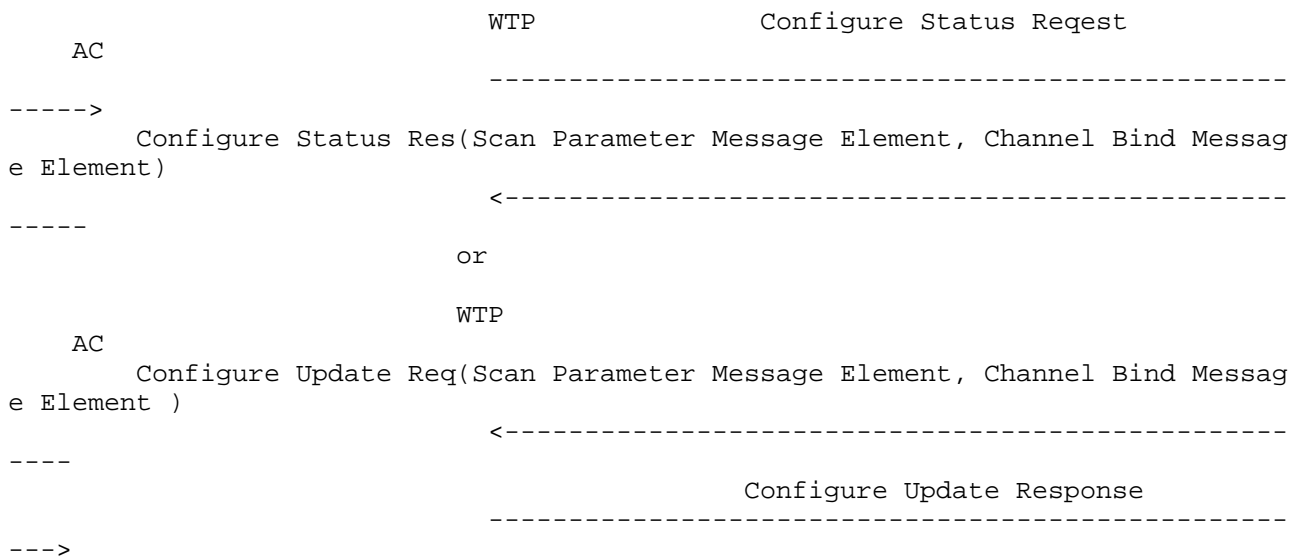


Figure 4: Channel Scan Procedure

The WTP has two work modes: normal mode and scan only mode. In normal mode, the WTP can provide service for station access and scan channels at the same time. Whether the WTP will scan a given set of channels is determined by the Max Cycles field in the IEEE 802.11 Channel Bind message element defined in Section 4.3.2. When this field is set to 0, the WTP will not scan the channel. If this field is set to 255, the WTP will scan the channel continuously. The type of the scan is determined by the Scan Type field. With the passive scan type, the WTP monitors the air interface, using the received

beacon frames to determine the nearby WTPs. With the active scan type, the WTP will send a probe message and receive probe response messages. In this case, the WTP may need to operate in station mode which means it is not a WTP function only device, it also has part of station function.

In normal mode, the WTP behaviour is controlled by three parameters: PrimeChlSrvTime, OnChannelScanTime, and OffChannelScnTime. These are provided by the IEEE 802.11 Scan Parameters message element defined in Section 4.3.1. The WTP will provide access service for stations for the duration given by PrimeChlSrvTime. It then scans the working channel for the duration given by OnChannelScanTime. It returns to servicing station access requests on the working channel for another period of length PrimeChlSrvTime, then moves to a different channel and scans it for duration OffChannelScnTime. It repeats this cycle, scanning a new non-working channel each time, until all the channels have been scanned. This channel scan procedure can be used to determine the interference of both the current working channel and non-working channel to avoid potential interference.

When the WTP works in scan only mode, it does not distinguish between the working channel and scan channel. Every channel's scan duration will be OffChannelScnTime and PrimeChlSrvTime and OnChannelScanTime MUST be set to 0.

As shown in Figure 4, the AC can control the scan behaviour at the WTP by including the IEEE 802.11 Scan Parameters and IEEE 802.11 Channel Bind message elements in a Configure Status Response or WTP Configure Update Request message.

Scan Report. After completing its scan, the WTP MAY send the scan report to the AC using a WTP Event Request message. The scan report information is carried in the IEEE 802.11 Channel Scan Report message element (Section 4.3.3) and an instance of the IEEE 802.11 Information Element message element carrying a copy of the IEEE 802.11 Neighbor WTP Report information element (Section 4.3.4).

#### 4.3.1. IEEE 802.11 Scan Parameters Message Element

The format of the IEEE 802.11 Scan Parameters Message Element is as shown in Figure 5:

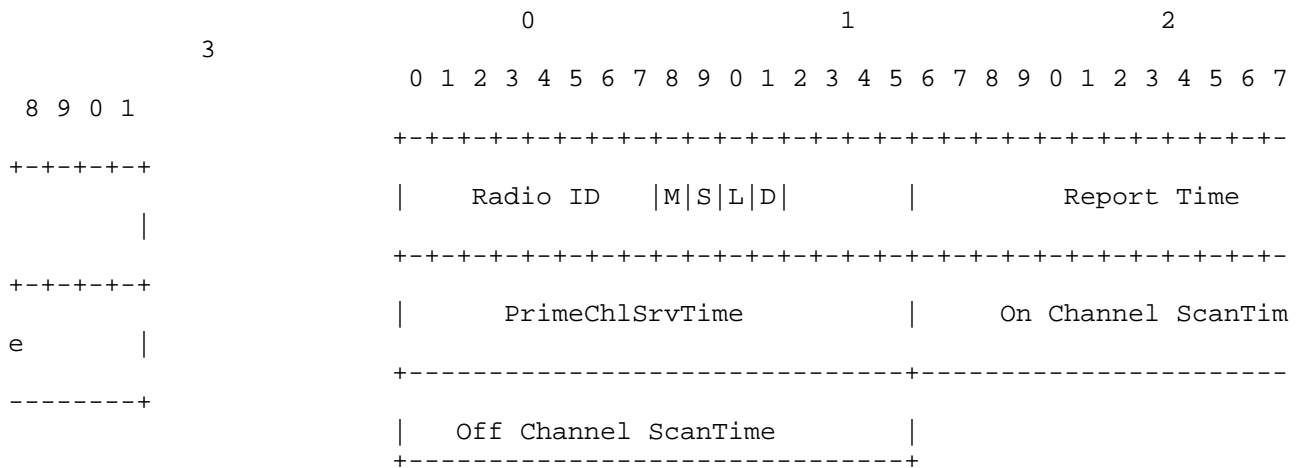


Figure 5: IEEE 802.11 Scan Parameters Message Element

Type: TBD3 for IEEE 802.11 Scan Parameters Message Element.

Length: 10.

Radio ID: An 8-bit value representing the radio, whose value is between one (1) and 31.

M bit: Work mode of the WTP. 0:normal mode. 1: scan only mode, no service is provided in this mode.

S bit: Scan Type: 0: active scan; 1: passive scan.

L bit: L=1: Open Load Balance Scan. L=0: Disable Load Balance Scan.

D bit: D=1: Open Rogue WTP detection scan. D=0: Disable Rouge WTP detection scan.

Report Time: Channel quality report time (unit: second).

PrimeChlSrvTime: Service time (unit: millisecond) on the working scan channel. This segment is invalid(set to 0) when WTP oper mode is set to 1. The maximum value of this segment is 10000, the minimum value of this segment is 5000, the default value is 5000.

On Channel ScanTime: The scan time (unit: millisecond) of the working channel. When the M bit is set to 1 (active scan), this segment is invalid(set to 0). The maximum value of this segment is 120, the minimum value of this segment is 60, the default value is 60.

Off Channel ScanTime: The scan time (unit: millisecond) of the working channel. When the WTP operating mode is set to 2, this segment MUST be set to 0. The maximum value of this segment is 120, the minimum value of this segment is 60, the default value is 60.

#### 4.3.2. IEEE 802.11 Scan Channel Bind Message Element

The format of the IEEE 802.11 Scan Channel Bind Message Element is as follows:

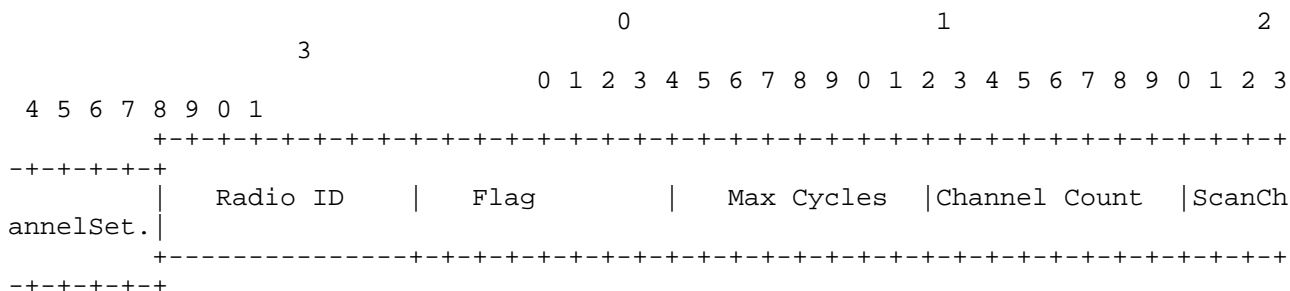


Figure 6: IEEE 802.11 Scan Channel Bind Message Element

Type: TBD4 for IEEE 802.11 Scan Channel Bind Message Element.

Length: variable.

Radio ID: An 8-bit value representing the radio, whose value is between one (1) and 31.

Flag: reserved.

Max Cycles: Number of times the scanning cycle is repeated for the set of channels identified by this message element. 255 means continuous scan.

Channel Count: The number of channels will be scanned.

Scan Channel Set: identifies the members of the set of channels to which this message element instance applies. The format for each channel is as follows:

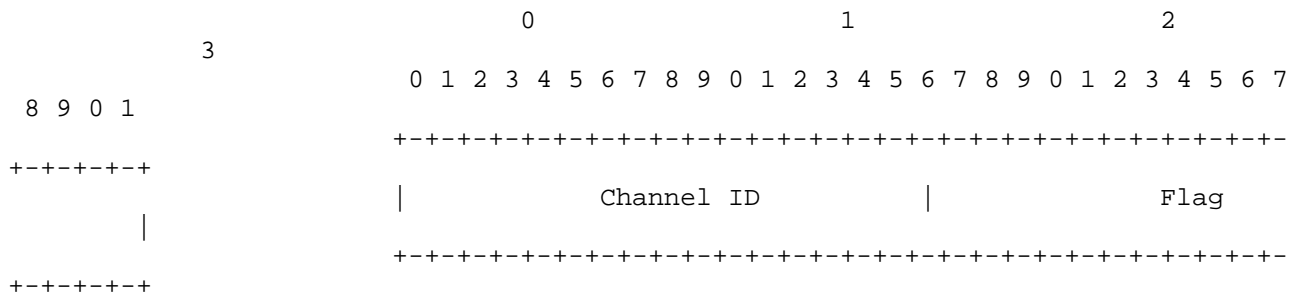


Figure 7: Channel Information Format

Channel ID: the channel ID of the channel which will be scanned.

Flag: Bitmap, reserved for future use.

#### 4.3.3. IEEE 802.11 Channel Scan Report

There are two types of scan report: Channel Scan Report and WTP Neighbor Report. Channel Scan Report is used to channel autoconfiguration while WTP Neighbor Report is used to power autoconfiguration. The WTP send the scan report to the AC through WTP Event Request message. The information element that used to carry the scan report is Channel Scan Report Message Element and WTP Neighbor Report Message Element.

The format of the IEEE 802.11 Channel Scan Report message element is in Figure 8.

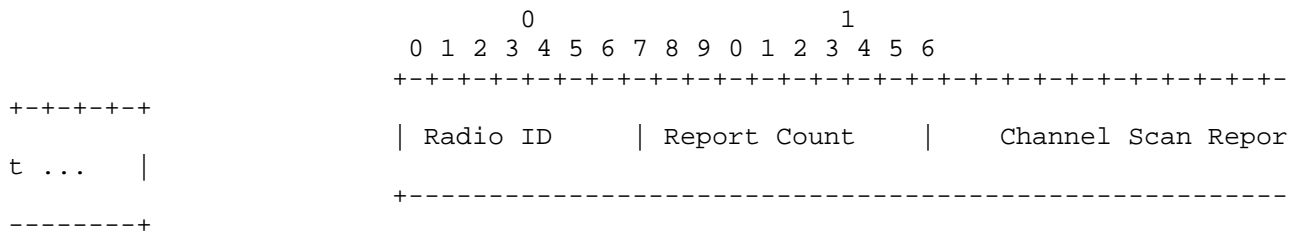


Figure 8: IEEE 802.11 Channel Scan Report Message Element

Type: TBD5 for IEEE 802.11 Channel Scan Report message element.

Length: >=29.

Radio ID: An 8-bit value representing the radio, whose value is between one (1) and 31.

Report Count: The number of channels for which a report is provided.

Channel Scan Report: The format of each Channel Scan Report is shown in Figure 9.



			0										1										2																			
			0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0									
1			+-----+																																							
-+-+			Channel Number										Radar Statistics										Mean																			
			+-----+																																							
-+-+			Time										Mean RSSI										Screen Packet Count																			
			+-----+																																							
---+			NeighborCount										Mean Noise										Interference										WTP Tx Occp									
			+-----+																																							
---+			WTP Rx Occp										Unknown Occp										CRC Err Cnt										Decrypt Err C									
nt			+-----+																																							
---+			Phy Err Cnt										Retrans Cnt																													
			+-----+																																							

Figure 9: Channel Scan Report

Channel Number: The channel number.

Radar Statistics: Whether detect radar signal in this channel. 0x00: detect radar signal. 0x01: no radar signal is detected.

Mean Time: Channel measurement duration (ms).

Mean RSSI: The average signal strength of the scanned channel (dBm(2's complement)).

Screen Packet Count: Received packet number.

Neighbor Count: The neighbor number of this channel.

Mean Noise: the average noise on this channel (dBm(2's complement)).

Interference: The interference of the channel.

WTP Tx Occp: (The WTP transmission time/Monitor time)\*255. The WTP transmission time is the total sending time of the WTP during the period of channel scan.

WTP Rx Occp: (The WTP receiving duration time/Monitor time)\*255. The WTP receiving duration time is the total receiving time of the WTP during the period of channel scan.

Unknown Occp: (All other packet transmission time duration/Monitor time)\*255.

CRC Err Cnt: CRC err packet number.



Decrypt Err Cnt: Decryption err packet number.

Phy Err Cnt: Physical err packet number.

Retrans Cnt: Retransmission packet number.

Note: The values of the above four count fields for a non-operational channel can be ignored

#### 4.3.4. IEEE 802.11 WTP Neighbor Report

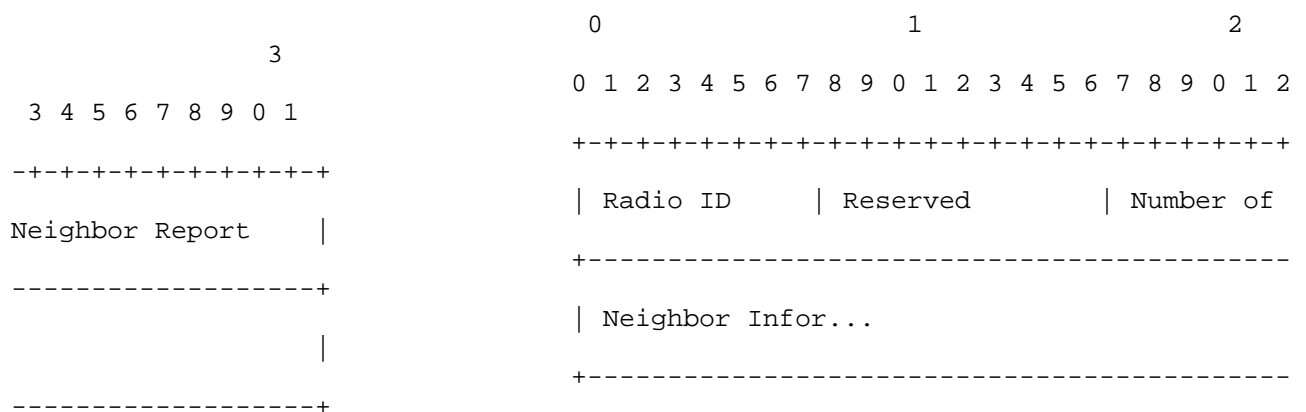


Figure 10: WTP Neighbor Report TLV

The definition of Neighbor info is as follows:

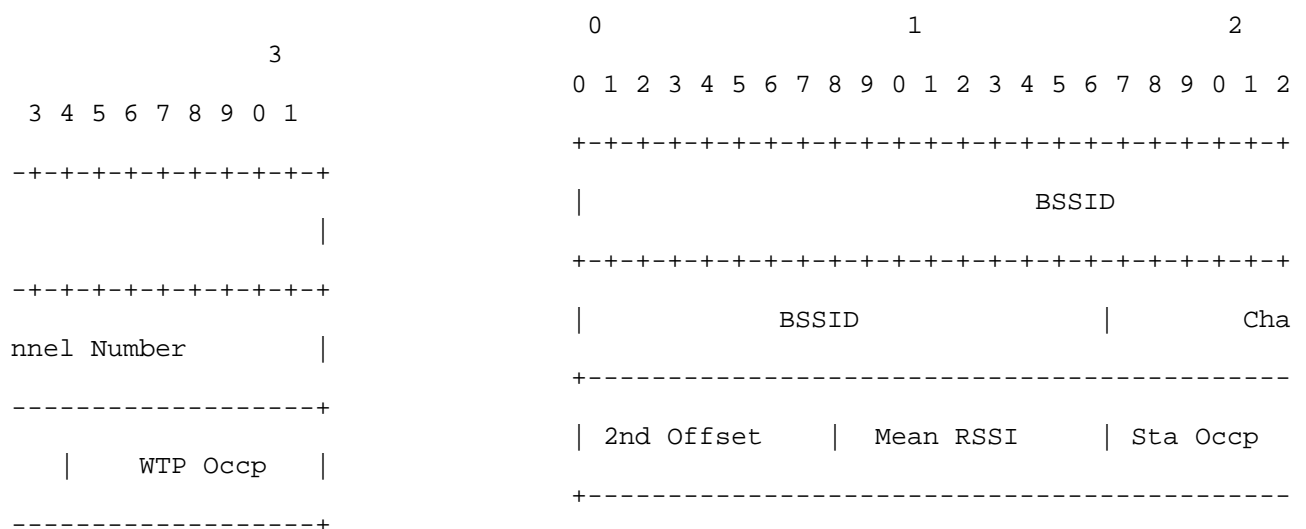


Figure 11: Neighbor info

BSSID: The BSSID of this neighbor WTP.

Channel Number: The channel number of this WTP neighbor.

2nd channel offset: The auxiliary channel offset of this WTP.



Mean RSSI: The average signal strength of this WTP (dbm).

Sta Occp: (The station air interface occupation time/Monitor time)\*255. The station air interface occupation time is the air interface occupation time caused by the stations which are connected to this WTP.

WTP Occp: (The WTP air interface occupation time/Monitor time)\*255. The WTP air interface occupation time is the air interface occupation time caused by the WTP.

## 5. Security Considerations

This document is based on RFC5415/RFC5416 and adds no new security considerations.

## 6. IANA Considerations

The extension defined in this document need to extend CAPWAP IEEE 802.11 binding message element which is defined in section 6 of [RFC5416]. The following IEEE 802.11 specific message element type need to be defined by IANA.

TBD1: 802.11n Radio Configuration Message Element type value described in section 4.1.2.

TBD2: 802.11n Station Message Element type value described in section 4.1.3.

TBD3: 802.11 Scan Parameter Message Element type value described in section 4.3.1.

TBD4: 802.11 Channel Bind Message Element type value described in section 4.3.2.

TBD5: Channel Scan Report Message Element type value described in section 4.3.3.

TBD6 entry for WTP Neighbor Report as described in section 4.3.4 .

## 7. Contributors

This draft is a joint effort from the following contributors:

Gang Chen: China Mobile chengang@chinamobile.com

Naibao Zhou: China Mobile zhounaibao@chinamobile.com

Chunju Shao: China Mobile shaochunju@chinamobile.com

Hao Wang: Huawei3Come hwang@h3c.com

Yakun Liu: AUTELAN liuyk@autelan.com

Xiaobo Zhang: GBCOM

Xiaolong Yu: Ruijie Networks

Song zhao: ZhiDaKang Communications

Yiwen Mo: ZhongTai Networks

Dorothy Stanley: dstanley1389@gmail.com

Tom Taylor: tom.taylor.stds@gmail.com

## 8. Acknowledgements

The authors would like to thanks Ronald Bonica, Romascanu Dan, Benoit Claise, Melinda Shore and Margaret Wasserman for their useful suggestions. The authors also thanks Dorothy Stanley and Tom Taylor for their review and useful comments.

## 9. Normative References

[IEEE-802.11.2009]

"IEEE Standard for 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, Enhancements for Higher Throughput (Amendment 5)", 2009.

[IEEE-802.11.2012]

"IEEE Standard for 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", March 2012.

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

- [RFC4564] Govindan, S., Cheng, H., Yao, ZH., Zhou, WH., and L. Yang, "Objectives for Control and Provisioning of Wireless Access Points (CAPWAP)", RFC 4564, July 2006.
- [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.

## Authors' Addresses

Yifan Chen  
China Mobile  
No.32 Xuanwumen West Street  
Beijing 100053  
China

Email: chen yifan@chinamobile.com

Dapeng Liu  
Beijing  
China

Email: maxpassion@gmail.com

Hui Deng  
China Mobile  
No.32 Xuanwumen West Street  
Beijing 100053  
China

Email: denghui@chinamobile.com

Lei Zhu  
Huawei  
No. 156, Shi-Chuang-Ke-Ji-Shi-Fan-Yuan Beiqing Road, Haidian District  
Beijing 100095  
China

Email: lei.zhu@huawei.com

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: June 21, 2015

C. Shao  
H. Deng  
China Mobile  
R. Pazhyannur  
Cisco Systems  
F. Bari  
AT&T  
R. Zhang  
China Telecom  
S. Matsushima  
SoftBank Telecom  
December 18, 2014

IEEE 802.11 MAC Profile for CAPWAP  
draft-ietf-opsawg-capwap-hybridmac-08

Abstract

The CAPWAP protocol binding for IEEE 802.11 defines two MAC (Medium Access Control) modes for IEEE 802.11 WTP (Wireless Transmission Point): Split and Local MAC. In the Split MAC mode, the partitioning of encryption/decryption functions are not clearly defined. In the Split MAC mode description, IEEE 802.11 encryption is specified as located in either the AC (Access Controller) or the WTP, with no clear way for the AC to inform the WTP of where the encryption functionality should be located. This leads to interoperability issues, especially when the AC and WTP come from different vendors. To prevent interoperability issues, this specification defines an IEEE 802.11 MAC profile message element in which each profile specifies an unambiguous division of encryption functionality between the WTP and AC.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."



This Internet-Draft will expire on June 21, 2015.

#### Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

#### Table of Contents

1. Introduction . . . . .	2
2. IEEE MAC Profile Descriptions . . . . .	4
2.1. Split MAC with WTP encryption . . . . .	4
2.2. Split MAC with AC encryption . . . . .	5
2.3. IEEE 802.11 MAC Profile Frame Exchange . . . . .	6
3. MAC Profile Message Element Definitions . . . . .	7
3.1. IEEE 802.11 Supported MAC Profiles . . . . .	7
3.2. IEEE 802.11 MAC Profile . . . . .	8
4. Security Considerations . . . . .	8
5. IANA Considerations . . . . .	8
6. Contributors . . . . .	9
7. Acknowledgments . . . . .	9
8. Normative References . . . . .	9
Authors' Addresses . . . . .	9

#### 1. Introduction

The CAPWAP protocol supports two MAC modes of operation: Split and Local MAC, as described in [RFC5415], [RFC5416]. However, there are MAC functions that have not been clearly defined. For example IEEE 802.11 encryption is specified as located in either in the AC or the WTP with no clear way to negotiate where it should be located. Because different vendors have different definitions of the MAC mode, many MAC layer functions are mapped differently to either the WTP or the AC by different vendors. Therefore, depending upon the vendor, the operators in their deployments have to perform different configurations based on implementation of the two modes by their vendor. If there is no clear specification, then operators will

experience interoperability issues with WTPs and ACs from different vendors.

Figure 1 from [RFC5416], illustrates how some functions are processed in different places in the Local MAC and Split MAC mode. Specifically, note that in the Split MAC mode the IEEE 802.11 encryption/decryption is specified as WTP/AC implying that it could be at either location. This is not an issue with Local MAC because encryption is always at the WTP.

Functions		Local MAC	Split MAC
Function	Distribution Service	WTP/AC	AC
	Integration Service	WTP	AC
	Beacon Generation	WTP	WTP
	Probe Response Generation	WTP	WTP
	Power Mgmt	WTP	WTP
	/Packet Buffering		
	Fragmentation	WTP	WTP/AC
	/Defragmentation		
	Assoc/Disassoc/Reassoc	WTP/AC	AC
	Classifying	WTP	AC
IEEE 802.11 QoS	Scheduling	WTP	WTP/AC
	Queuing	WTP	WTP
	IEEE 802.1X/EAP	AC	AC
IEEE 802.11 RSN (WPA2)	RSNA Key Management	AC	AC
	IEEE 802.11 Encryption/Decryption	WTP	WTP/AC

Figure 1: Functions in Local MAC and Split MAC

To solve this problem, this specification introduces IEEE 802.11 MAC profile. The MAC profile unambiguously specifies where the various MAC functionality should be located.

## 2. IEEE MAC Profile Descriptions

A IEEE MAC Profile refers to a description of how the MAC functionality is split between the WTP and AC shown in Figure 1.

### 2.1. Split MAC with WTP encryption

The functional split for the Split MAC with WTP encryption is provided in Figure 2. This profile is similar to the Split MAC description in [RFC5416], except that IEEE 802.11 encryption/decryption is at the WTP. Note that fragmentation is always done at the same entity as the encryption. Consequently, in this profile fragmentation/defragmentation is also done only at the WTP. Note that scheduling functionality is denoted as WTP/AC. As explained in [RFC5416], this means that the admission control component of IEEE 802.11 resides on the AC, the real-time scheduling and queuing functions are on the WTP.

Functions		Profile
		0
	Distribution Service	AC
	Integration Service	AC
	Beacon Generation	WTP
	Probe Response Generation	WTP
Function	Power Mgmt	WTP
	/Packet Buffering	
	Fragmentation	WTP
	/Defragmentation	
	Assoc/Disassoc/Reassoc	AC
	Classifying	AC
IEEE	Scheduling	WTP/AC
802.11 QoS	Queuing	WTP
	IEEE 802.1X/EAP	AC
IEEE	RSNA Key Management	AC
802.11 RSN	IEEE 802.11	WTP
(WPA2)	Encryption/Decryption	

Figure 2: Functions in Split MAC with WTP Encryption

## 2.2. Split MAC with AC encryption

The functional split for the Split MAC with AC encryption is provided in Figure 3. This profile is similar to the Split MAC in [RFC5416] except that IEEE 802.11 encryption/decryption is at the AC. Since fragmentation is always done at the same entity as the encryption, in this profile, AC does fragmentation/defragmentation.

Functions		Profile
		1
	Distribution Service	AC
	Integration Service	AC
	Beacon Generation	WTP
	Probe Response Generation	WTP
Function	Power Mgmt	WTP
	/Packet Buffering	
	Fragmentation	AC
	/Defragmentation	
	Assoc/Disassoc/Reassoc	AC
	Classifying	AC
IEEE 802.11 QoS	Scheduling	WTP
	Queuing	WTP
	IEEE 802.1X/EAP	AC
IEEE 802.11 RSN (WPA2)	RSNA Key Management	AC
	IEEE 802.11 Encryption/Decryption	AC

Figure 3: Functions in Split MAC with AC encryption

### 2.3. IEEE 802.11 MAC Profile Frame Exchange

An example of message exchange using the IEEE 802.11 MAC Profile message element is shown in Figure 4. The WTP informs the AC of the various MAC profiles it supports. This happens either in a Discovery Request message or the Join Request message. The AC determines the appropriate profile and configures the WTP with the profile while configuring the WLAN.

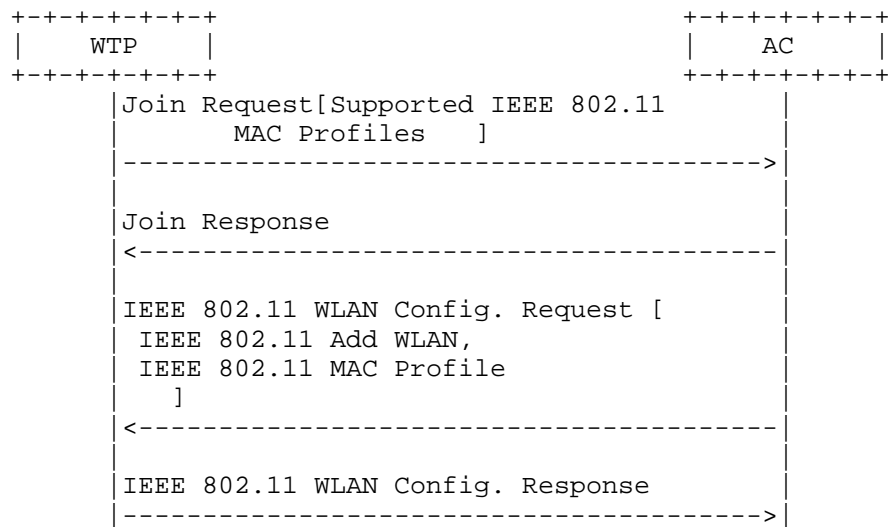


Figure 4: Message Exchange For Negotiating MAC Profile

### 3. MAC Profile Message Element Definitions

#### 3.1. IEEE 802.11 Supported MAC Profiles

The IEEE 802.11 Supported MAC Profile message element allows the WTP to communicate the profiles it supports. The Discovery Request message, Primary Discovery Request message, and Join Request message may include one such message element.

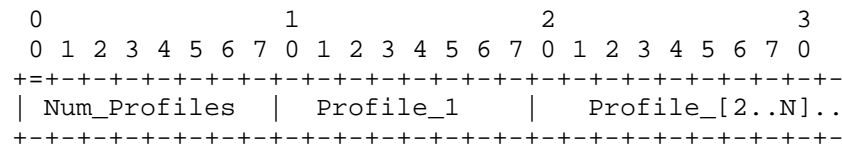


Figure 5: IEEE 802.11 Supported MAC Profiles

- o Type: TBD for IEEE 802.11 Supported MAC Profiles
- o Num\_Profiles >=1: This refers to number of profiles present in this message element. There must be at least one profile.
- o Profile: Each profile is identified by a value specified in Section 3.2.

### 3.2. IEEE 802.11 MAC Profile

The IEEE 802.11 MAC Profile message element allows the AC to select a profile. This message element may be provided along with the IEEE 802.11 ADD WLAN message element while configuring a WLAN on the WTP.

```

    0 1 2 3 4 5 6 7
    +=+--+--+--+--+--+
    |  Profile      |
    +--+--+--+--+--+--+

```

Figure 6: IEEE 802.11 MAC Profile

- o Type: TBD for IEEE 802.11 MAC Profile
- o Profile: The profile is identified by a value as given below
  - \* 0: This refers to the Split MAC Profile with WTP encryption
  - \* 1: This refers to the Split MAC Profile with AC encryption

### 4. Security Considerations

This document does not introduce any new security risks compared to [RFC5416]. The negotiation messages between the WTP and AC have origin authentication and data integrity. As a result an attacker cannot interfere with the messages to force a less secure mode choice. The security considerations described in [RFC5416] apply here as well.

### 5. IANA Considerations

This document requires the following IANA actions:

- o This specification defines two new message elements, IEEE 802.11 Supported MAC Profiles (described in Section 3.1) and IEEE 802.11 MAC Profile (described in Section 3.2). These elements need to be registered in the existing CAPWAP Message Element Type registry, defined in [RFC5415]. The values for these elements need to be between 1024 and 2047 (see Section 15.7 in [RFC5415]).

CAPWAP Protocol Message Element	Type Value
IEEE 802.11 Supported MAC Profiles	TBD1
IEEE 802.11 MAC Profile	TBD2

- o The IEEE 802.11 Supported MAC Profiles message element and IEEE 802.11 MAC Profile message element include a Profile Field (as defined in Section 3.2). The Profile field in the IEEE 802.11 Supported MAC Profiles denotes the MAC profiles supported by the WTP. The profile field in the IEEE MAC profile denotes MAC

profile assigned to the WTP. The namespace for the field is 8 bits (0-255). This specification defines two values, zero (0) and one (1) as described below. The remaining values (2-255) are controlled and maintained by IANA and require an Expert Review. IANA needs to create a new sub-registry called IEEE 802.11 Split MAC Profile and add the new sub-registry to the existing registry "Control And Provisioning of Wireless Access Points (CAPWAP) Parameters". The registry format is given below.

Profile	Type Value	Reference
Split MAC with WTP encryption	0	
Split MAC with AC encryption	1	

## 6. Contributors

Yifan Chen [chenyifan@chinamobile.com](mailto:chenyifan@chinamobile.com)

Naibao Zhou [zhounaibao@chinamobile.com](mailto:zhounaibao@chinamobile.com)

## 7. Acknowledgments

The authors are grateful for extremely valuable suggestions from Dorothy Stanley in developing this specification.

Guidance from management team: Melinda Shore, Scott Bradner, Chris Liljenstolpe, Benoit Claise, Joel Jaeggli, Dan Romascanu are highly appreciated.

## 8. Normative References

- [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.

## Authors' Addresses

Chunju Shao  
China Mobile  
No.32 Xuanwumen West Street  
Beijing 100053  
China

Email: [shaochunju@chinamobile.com](mailto:shaochunju@chinamobile.com)



Hui Deng  
China Mobile  
No.32 Xuanwumen West Street  
Beijing 100053  
China

Email: denghui@chinamobile.com

Rajesh S. Pazhyannur  
Cisco Systems  
170 West Tasman Drive  
San Jose, CA 95134  
USA

Email: rpazhyan@cisco.com

Farooq Bari  
AT&T  
7277 164th Ave NE  
Redmond WA 98052  
USA

Email: farooq.bari@att.com

Rong Zhang  
China Telecom  
No.109 Zhongshandadao avenue  
Guangzhou 510630  
China

Email: zhangr@gsta.com

Satoru Matsushima  
SoftBank Telecom  
1-9-1 Higashi-Shinbashi, Munato-ku  
Tokyo  
Japan

Email: satoru.matsushima@g.softbank.co.jp

Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: September 2, 2015

M. Ersue, Ed.  
Nokia Networks  
D. Romascanu  
Avaya  
J. Schoenwaelder  
Jacobs University Bremen  
U. Herberg  
March 1, 2015

Management of Networks with Constrained Devices: Problem Statement and  
Requirements  
draft-ietf-opsawg-coman-probstate-reqs-05

Abstract

This document provides a problem statement, deployment and management topology options as well as requirements addressing the different use cases of the management of networks where constrained devices are involved.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 2, 2015.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
1.1. Overview . . . . .	3
1.2. Terminology . . . . .	4
1.3. Network Types and Characteristics in Focus . . . . .	5
1.4. Constrained Device Deployment Options . . . . .	9
1.5. Management Topology Options . . . . .	9
1.6. Managing the Constrainedness of a Device or Network . . . .	10
1.7. Configuration and Monitoring Functionality Levels . . . .	13
2. Problem Statement . . . . .	14
3. Requirements on the Management of Networks with Constrained Devices . . . . .	16
3.1. Management Architecture/System . . . . .	17
3.2. Management Protocols and Data Models . . . . .	21
3.3. Configuration Management . . . . .	24
3.4. Monitoring Functionality . . . . .	26
3.5. Self-management . . . . .	31
3.6. Security and Access Control . . . . .	32
3.7. Energy Management . . . . .	34
3.8. Software Distribution . . . . .	36
3.9. Traffic Management . . . . .	36
3.10. Transport Layer . . . . .	37
3.11. Implementation Requirements . . . . .	39
4. IANA Considerations . . . . .	40
5. Security Considerations . . . . .	40
6. Acknowledgments . . . . .	41
7. Informative References . . . . .	41
Appendix A. Change Log . . . . .	42
A.1. draft-ietf-opsawg-coman-probstate-reqs-04 - draft-ietf- opsawg-coman-probstate-reqs-05 . . . . .	42
A.2. draft-ietf-opsawg-coman-probstate-reqs-03 - draft-ietf- opsawg-coman-probstate-reqs-04 . . . . .	42
A.3. draft-ietf-opsawg-coman-probstate-reqs-02 - draft-ietf- opsawg-coman-probstate-reqs-03 . . . . .	42
A.4. draft-ietf-opsawg-coman-probstate-reqs-01 - draft-ietf- opsawg-coman-probstate-reqs-02 . . . . .	43
A.5. draft-ietf-opsawg-coman-probstate-reqs-00 - draft-ietf- opsawg-coman-probstate-reqs-01 . . . . .	43
A.6. draft-ersue-constrained-mgmt-03 - draft-ietf-opsawg- coman-probstate-reqs-00 . . . . .	44
A.7. draft-ersue-constrained-mgmt-02-03 . . . . .	44
A.8. draft-ersue-constrained-mgmt-01-02 . . . . .	45

A.9. draft-ersue-constrained-mgmt-00-01 . . . . .	46
Authors' Addresses . . . . .	46

## 1. Introduction

### 1.1. Overview

Constrained devices, aka. sensor, smart object, or smart device, with limited CPU, memory, and power resources, can constitute a network. Such a network of constrained devices itself may be constrained or challenged, e.g., with unreliable or lossy channels, wireless technologies with limited bandwidth and a dynamic topology, needing the service of a gateway or proxy to connect to the Internet. In other scenarios, the constrained devices can be connected to a non-constrained network using off-the-shelf protocol stacks.

Constrained devices might be in charge of gathering information in diverse settings including natural ecosystems, buildings, and factories, and send the information to one or more server stations. Constrained devices may also work under severe resource constraints such as limited battery and computing power, little memory and insufficient wireless bandwidth, and communication capabilities. A central entity, e.g., a base station or controlling server, might have more computational and communication resources and can act as a gateway between the constrained devices and the application logic in the core network.

Today diverse size of constrained devices with different resources and capabilities are being connected. Mobile personal gadgets, building-automation devices, cellular phones, Machine-to-machine (M2M) devices, etc. benefit from interacting with other "things" in the near or somewhere in the Internet. With this the Internet of Things (IoT) becomes a reality, build up of uniquely identifiable objects (things). And over the next decade, this could grow to trillions of constrained devices and will greatly increase the Internet's size and scope.

Network management is characterized by monitoring network status, detecting faults, and inferring their causes, setting network parameters, and carrying out actions to remove faults, maintain normal operation, and improve network efficiency and application performance. The traditional network monitoring application periodically collects information from a set of elements that are needed to manage, processes the data, and presents them to the network management users. Constrained devices, however, often have limited power, low transmission range, and might be unreliable. They might also need to work in hostile environments with advanced security requirements or need to be used in harsh environments for a

long time without supervision. Due to such constraints, the management of a network with constrained devices faces different type of challenges compared to the management of a traditional IP network.

The IETF has already done substantial standardization work to enable the communication in IP networks and to manage such networks as well as the manifold type of nodes in these networks [RFC6632]. However, the IETF so far has not developed any specific technologies for the management of constrained devices and the networks comprised by constrained devices. IP-based sensors or constrained devices in such an environment, i.e., devices with very limited memory, CPU, and energy resources, use nowadays application-layer protocols in an ad-hoc manner to do simple resource management and monitoring.

This document provides a problem statement and lists requirements for the different use cases of management of a network with constrained devices. Section 1.3 and Section 1.5 describe different topology options for the networking and management of constrained devices. Section 2 provides a problem statement on the issue of the management of networked constrained devices. Section 3 lists requirements on the management of applications and networks with constrained devices. Note that the requirements listed in Section 3 have been separated from the context in which they may appear. Depending on the concrete circumstances, an implementer may decide to address a certain relevant subset of the requirements.

The use cases in the context of networks with constrained devices can be found in the companion document [COM-USE]. This informational document provides a list of objectives for discussions and does not aim to be a strict requirements document for all use cases. In fact, there likely is not a single solution that works equally well for all the use cases.

## 1.2. Terminology

Concerning constrained devices and networks this document generally builds on the terminology defined in [RFC7228], where the terms Constrained Device, Constrained Network, etc. are defined.

The following terms are additionally used throughout this documentation:

AMI: (Advanced Metering Infrastructure) A system including hardware, software, and networking technologies that measures, collects, and analyzes energy usage, and communicates with a hierarchically deployed network of metering devices, either on request or on a schedule.

C0: Class 0 constrained device as defined in Section 3. of [RFC7228].

C1: Class 1 constrained device as defined in Section 3. of [RFC7228].

C2: Class 2 constrained device as defined in Section 3. of [RFC7228].

Network of Constrained Devices: A network to which constrained devices are connected that may or may not be a Constrained Network (see [RFC7228] for the definition of the term Constrained Network).

M2M: (Machine to Machine) stands for the automatic data transfer between devices of different kind. In M2M scenarios a device (such as a sensor or meter) captures an event, which is relayed through a network (wireless, wired or hybrid) to an application.

MANET: Mobile Ad-hoc Networks [RFC2501], a self-configuring and infrastructureless network of mobile devices connected by wireless technologies.

Smart Grid: An electrical grid that uses communication technologies to gather and act on information in an automated fashion to improve the efficiency, reliability and sustainability of the production and distribution of electricity.

Smart Meter: An electrical meter in the context of a Smart Grid.

For a detailed discussion on the constrained networks as well as classes of constrained devices and their capabilities please see [RFC7228].

### 1.3. Network Types and Characteristics in Focus

In this document we differentiate following types of networks concerning their transport and communication technologies:

(Note that a network in general can involve constrained and non-constrained devices.)

1. Wireline non-constrained networks, e.g., an Ethernet-LAN with constrained and non-constrained devices involved.
2. A combination of wireline and wireless networks, possibly with a multi-hop connectivity between constrained devices, utilizing dynamic routing in both the wireless and wireline portions of the

network. Such networks usually support highly distributed applications with many nodes (e.g., environmental monitoring) and tend to deal with large-scale multipoint-to-point systems. Wireless Mesh Networks (WMN), as a specific variant, use off-the-shelf radio technology such as Wi-Fi, WiMax, and cellular 3G/4G. WMNs are reliable based on the redundancy they offer and have often a more planned deployment to provide dynamic and cost effective connectivity over a certain geographic area.

3. A combination of wireline and wireless networks with point-to-point or point-to-multipoint communication generally with single-hop connectivity to constrained devices, utilizing static routing over the wireless network. Such networks support short-range, point-to-point, low-data-rate, source-to-sink type of applications such as RFID systems, light switches, fire and smoke detectors, and home appliances. This type of networks also support confined short-range spaces such as a home, a factory, a building, or the human body. IEEE 802.15.1 (Bluetooth) and IEEE 802.15.4 are well-known examples of applicable standards for such networks. By using 6LowPAN (IPv6 over Low-Power Wireless Personal Area Networks) [RFC4919] and RPL (IPv6 Routing Protocol for Low-Power and Lossy Networks) [RFC6550] on top of IEEE 802.15.4, multi-hop connectivity and dynamic routing can be achieved. With RPL the IETF has specified a proactive route-over architecture where routing and forwarding is implemented at the network layer. The protocol provides a mechanism whereby multipoint-to-point, point-to-multipoint and point-to-point traffic are supported.
4. Self-configuring infrastructureless networks of mobile devices (e.g., Mobile Adhoc networks, MANET) are a particular type of network connected by wireless technologies. Infrastructureless networks are mostly based on point-to-point communications of devices moving independently in any direction and changing the links to other devices frequently. Such devices do act as a router to forward traffic unrelated to their own use.

Wireline non-constrained networks with constrained and non-constrained devices are mainly used for specific applications like Building Automation or Infrastructure Monitoring. Wireline and wireless networks with multi-hop or point-to-multipoint connectivity are used e.g., for environmental monitoring as well as transport and mobile applications.

Furthermore different network characteristics are determined by multiple dimensions: dynamicity of the topology, bandwidth, and loss rate. In the following, each dimension is explained, and networks in scope for this document are outlined:

### Network Topology:

The topology of a network can be represented as a graph, with edges (i.e., links) and vertices (routers and hosts). Examples of different topologies include "star" topologies (with one central node and multiple nodes in one hop distance), tree structures (with each node having exactly one parent), directed acyclic graphs (with each node having one or more parents), clustered topologies (where one or more "cluster heads" are responsible for a certain area of the network), mesh topologies (fully distributed), etc.

Management protocols may take advantage of specific network topologies, for example by distributing large-scale management tasks amongst multiple distributed network management stations (e.g., in case of a mesh topology), or by using a hierarchical management approach (e.g., in case of a tree or clustered topology). These different management topology options are described in Section 1.6.

Note that in certain network deployments, such as community ad hoc networks (see the use case "Community Network Applications" in [COM-USE]), the topology is not pre-planned, and thus may be unknown for management purposes. In other use cases, such as industrial applications (see the use case "Industrial Applications" in [COM-USE]), the topology may be designed in advance and therefore taken advantage of when managing the network.

### Dynamicity of the network topology:

The dynamicity of the network topology determines the rate of change of the graph as a function of time. Such changes can occur due to different factors, such as mobility of nodes (e.g., in MANETs or cellular networks), duty cycles (for low-power devices enabling their network interface only periodically to transmit or receive packets), or unstable links (in particular wireless links with strongly fluctuating link quality).

Examples of different levels of dynamicity of the topology are Ethernets (with typically a very static topology) on the one side, and low-power and lossy networks (LLNs) on the other side. LLNs nodes are often duty-cycled and operate on unreliable wireless links and are potentially mobile (e.g., for sensor networks).

The more dynamic the topology is, the more have routing, transport and application layer protocols to cope with interrupted connectivity and/or longer delays. For example, management protocols (with a given underlying transport protocol) that expect continuous session flows without changes of routes during a communication flow, may fail to operate.



Networks with a very low dynamicity (e.g., Ethernet) with no or infrequent topology changes (e.g., less than once every 30 minutes), are in-scope of this document if they are used with constrained devices (see e.g., the use case "Building Automation" in [COM-USE]).

#### Traffic flows:

The traffic flow in a network determines from which sources data traffic is sent to which destinations in the network. Several different traffic flows are defined in [RFC7102], including "point-to-point" (P2P), "multipoint-to-point" (MP2P), and "point-to-multipoint" (P2MP) flows as:

- o P2P: Point-To-Point. This refers to traffic exchanged between two nodes (regardless of the number of hops between the two nodes).
- o P2MP: Point-to-Multipoint traffic refers to traffic between one node and a set of nodes. This is similar to the P2MP concept in Multicast or MPLS Traffic Engineering.
- o MP2P: Multipoint-to-Point is used to describe a particular traffic pattern (e.g., MP2P flows collecting information from many nodes flowing inwards towards a collecting sink).

If one of these traffic patterns is predominant in a network, protocols (routing, transport, application) may be optimized for the specific traffic flow. For example, in a network with a tree topology and MP2P traffic, collection tree protocols are efficient to send data from the leaves of the tree to the root of the tree, via each node's parent.

#### Bandwidth:

The bandwidth of the network is the amount of data that can be sent per unit of time between two communication end-points. It is usually determined by the link with the minimum bandwidth on the path from the source to the destination of data packets. The bandwidth in networks can range from a few Kilobytes per second (such as on some 802.15.4 link layers) to many Gigabytes per second (e.g., on fiber optics).

For management purposes, the management protocol typically requires to send information between the network management station and the clients, for monitoring or control purposes. If the available bandwidth is insufficient for the management protocol, packets will be buffered and eventually dropped, and thus management is not possible with such a protocol.

Networks without bandwidth limitation (e.g., Ethernet) are in-scope of this document if they are used with constrained devices (see the use case "Building Automation" in [COM-USE]).

Loss rate:

The loss rate (or bit error rate) is the number of bit errors divided by the total number of bits transmitted. For wired networks, loss rates are typically extremely low, e.g., around  $10^{-12}$  or  $10^{-13}$  for the latest 10Gbit Ethernet. For wireless networks, such as 802.15.4, the bit error rate can be as high as  $10^{-1}$  to 1 in case of interferences. Even when using a reliable transport protocol, management operations can fail if the loss rate is too high, unless they are specifically designed to cope with these situations.

#### 1.4. Constrained Device Deployment Options

We differentiate following deployment options for the constrained devices:

- o A network of constrained devices that communicate with each other,
- o Constrained devices, which are connected directly to an IP network,
- o A network of constrained devices which communicate with a gateway or proxy with more communication capabilities acting possibly as a representative of the device to entities in the non-constrained network
- o Constrained devices, which are connected to the Internet or an IP network via a gateway/proxy
- o A hierarchy of constrained devices, e.g., a network of C0 devices connected to one or more C1 devices - connected to one or more C2 devices - connected to one or more gateways - connected to some application servers or NMS system
- o The possibility of device grouping (possibly in a dynamic manner) such as that the grouped devices can act as one logical device at the edge of the network and one device in this group can act as the managing entity

#### 1.5. Management Topology Options

We differentiate following options for the management of networks of constrained devices:

- o A network of constrained devices managed by one central manager. A logically centralized management might be implemented in a hierarchical fashion for scalability and robustness reasons. The manager and the management application logic might have a gateway/proxy in between or might be on different nodes in different networks, e.g., management application running on a cloud server.
- o Distributed management, where a network of constrained devices is managed by more than one manager. Each manager controls a subnetwork and may communicate directly with other manager stations in a cooperative fashion. The distributed management may be weakly distributed, where functions are broken down and assigned to many managers dynamically, or strongly distributed, where almost all managed things have embedded management functionality and explicit management disappears, which usually comes with the price that the strongly distributed management logic now needs to be managed.
- o Hierarchical management, where a hierarchy of networks with constrained devices are managed by the managers at their corresponding hierarchy level. I.e., each manager is responsible for managing the nodes in its sub-network. It passes information from its sub-network to its higher-level manager, and disseminates management functions received from the higher-level manager to its sub-network. Hierarchical management is essentially a scalability mechanism, logically the decision-making may be still centralized.

#### 1.6. Managing the Constrainedness of a Device or Network

The capabilities of a constrained device or network and the constrainedness thereof influence and have an impact on the requirements for the management of such network or devices.

Note that the list below gives examples and does not claim completeness.

A constrained device:

- o might only support an unreliable (e.g. lossy) radio link, i.e., the client and server of a management protocol need to gracefully handle incomplete command exchanges or missing commands.
- o might only be able to go online from time-to-time, where it is reachable, i.e., a command might be necessary to repeat after a longer timeout or the timeout value with which one endpoint waits on a response needs to be sufficiently high.

- o might only be able to support a limited operating time (e.g., based on the available battery), or may behave as 'sleepy endpoints' setting their network links to a disconnected state during long periods of time i.e., the devices need to economize their energy usage with suitable mechanisms and the managing entity needs to monitor and control the energy status of the constrained devices it manages.
- o might only be able to support one simple communication protocol, i.e., the management protocol needs to be possible to downscale from constrained (C2) to very constrained (C0) devices with modular implementation and a very basic version with just a few simple commands.
- o might only be able to support a communication protocol, which is not IP-based.
- o might only be able to support limited or no user and/or transport security, i.e., the management system needs to support a less-costly and simple but sufficiently secure authentication mechanism.
- o might not be able to support compression and decompression of exchanged data based on limited CPU power, i.e., an intermediary entity which is capable of data compression should be able to communicate with both, devices that support data compression (e.g., C2) and devices that do not support data compression (e.g., C1 and C0).
- o might only be able to support a simple encryption, i.e., it would be beneficial if the devices use cryptographic algorithms that are supported in hardware and the encryption used is efficient in terms of memory and CPU usage.
- o might only be able to communicate with one single managing entity and cannot support the parallel access of many managing entities.
- o might depend on a self-configuration feature, i.e., the managing entity might not know all devices in a network and the device needs to be able to initiate connection setup for the device configuration.
- o might depend on self- or neighbor-monitoring feature, i.e., the managing entity might not be able to monitor all devices in a network continuously.
- o might only be able to communicate with its neighbors, i.e., the device should be able to get its configuration from a neighbor.

- o might only be able to support parsing of data models with limited size, i.e., the device data models need to be compact containing the most necessary data and if possible parsable as a stream.
- o might only be able to support a limited or no failure detection, i.e., the managing entity needs to handle the situation, where a failure does not get detected or gets detected late gracefully e.g., with asking repeatedly.
- o might only be able to support the reporting of just one or a limited set failure types.
- o might only be able to support a limited set of notifications, possible only an "I-am-alive" message.
- o might only be able to support a soft-reset from failure recovery.
- o might possibly generate a large amount of redundant reporting data, i.e., the intermediary management entity (see [RFC7252]) should be able to filter and aggregate redundant data.

A network of constrained devices:

- o might only support an unreliable (e.g. lossy) radio link, i.e., the client and server of a management protocol need to repeat commands as necessary or gracefully ignore incomplete commands.
- o might be necessary to manage based on multicast communication, i.e., the managing entity needs to be prepared to configure many devices at once based on the same data model.
- o might have a very large topology supporting 10,000 or more nodes for some applications and as such node naming is a specific issue for constrained networks.
- o needs to support self-organization, i.e., given the large number of nodes and their potential placement in hostile locations and frequently changing topology, manual configuration of nodes is typically not feasible. As such, the network would benefit from the ability to reconfigure itself so that it can continue to operate properly and support reliable connectivity.
- o might need a management solution that is energy-efficient, using as little wireless bandwidth as possible since communication is highly energy demanding.

- o needs to support localization schemes to determine the location of devices since the devices might be moving and location information is important for some applications.
- o needs a management solution that is scalable as the network may consist of thousands of nodes and may need to be extended continuously.
- o needs to provide fault tolerance. Faults in network operation including hardware and software errors or failures detected by the transport protocol should be handled smoothly. In such a case it should be possible to run the protocol possibly at a reduced level but avoiding to fail completely. E.g., self-monitoring mechanisms or graceful degradation of features can be used to provide fault tolerance.
- o might require new management capabilities: for example, network coverage information and a constrained device power-distribution-map.
- o might require a new management function for data management, since the type and amount of data collected in constrained networks is different from those of the traditional networks.
- o might also need energy-efficient key management.

#### 1.7. Configuration and Monitoring Functionality Levels

Devices often differ significantly on the level of configuration management support they provide. This document classifies the configuration management functionality as follows:

CL0: Devices are pre-configured and allow no runtime configuration changes. Configuration parameters are often hard coded and compiled directly into the firmware image.

CL1: Devices have explicit configuration objects. However, changes require a restart of the device to take effect.

CL2: Devices allow management systems to replace the entire configuration (or pre-determined subsets) in bulk. Configuration changes take effect by soft-restarts of the system (or subsystems).

CL3: Devices allow management systems to modify configuration objects without bulk replacements and changes take effect immediately.

CL4: Devices support multiple configuration datastores and they might distinguish between the currently running and the next startup configuration.

CL5: Devices support configuration datastore locking and device-local configuration change transactions, i.e., either all configuration changes are applied or none of them.

CL6: Devices support configuration change transactions across devices.

This document defines a classification of devices with regards to different levels of monitoring support. In general a device may be in several of the levels listed below:

ML0: Devices push pre-defined monitoring data.

ML1: Devices allow management systems to pull pre-defined monitoring data.

ML2: Devices allow management systems to pull user-defined filtered subsets of monitoring data.

ML3: Devices are able to locally process monitoring data in order to detect threshold crossings or to aggregate data.

At the time of this writing, constrained devices often implement a combination of one of CL0-CL2 with one of ML0-ML1.

## 2. Problem Statement

The terminology for the "Internet of Things" is still nascent, and depending on the network type or layer in focus diverse technologies and terms are in use. Common to all these considerations is the "Things" or "Objects" are supposed to have physical or virtual identities using interfaces to communicate. In this context, we need to differentiate between the Constrained and Smart Devices identified by an IP address compared to virtual entities such as Smart Objects, which can be identified as a resource or a virtual object by using a unique identifier. Furthermore, the smart devices usually have a limited memory and CPU power as well as aim to be self-configuring and easy to deploy.

However, the constraints of the network nodes require a rethinking of the protocol characteristics concerning power consumption, performance, bandwidth consumption, memory, and CPU usage. As such, there is a demand for protocol simplification, energy-efficient communication, less CPU usage and smaller memory footprint.

On the application layer the IETF is already developing protocols like the Constrained Application Protocol (CoAP) [RFC7252] enabling the communication of constrained devices and networks e.g., for smart energy applications or home automation environments. The deployment of such an environment involves in fact many, in some scenarios up to million constrained devices (e.g., smart meters), which produce a large amount of data. This data needs to be collected, filtered, and pre-processed for further use in diverse services.

Considering the high number of nodes to deploy, one has to think about the manageability aspects of the smart devices and plan for easy deployment, configuration, and management of the networks of constrained devices as well as the devices themselves. Consequently, seamless monitoring and self-configuration of such network nodes becomes more and more imperative. Self-configuration and self-management is already a reality in the standards of some of the bodies such as 3GPP. To introduce self-configuration of smart devices successfully a device-initiated connection establishment is often required.

A simple and efficient application layer protocol, such as CoAP, is essential to address the issue of efficient object-to-object communication and information exchange. Such an information exchange should be done based on interoperable data models to enable the exchange and interpretation of diverse application and management related data.

In an ideal world, we would have only one network management protocol for monitoring, configuration, and exchanging management data, independently of the type of the network (e.g., Smart Grid, wireless access, or core network). Furthermore, it would be desirable to derive the basic data models for constrained devices from the core models used today to enable reuse of functionality and end-to-end information exchange. However, the current management protocols seem to be too heavyweight compared to the capabilities the constrained devices have and are not applicable directly for the use in a network of constrained devices. Furthermore, the data models addressing the requirements of such smart devices need yet to be designed.

The IETF so far has not developed any specific technologies for the management of constrained devices and the networks comprised by constrained devices. IP-based sensors or constrained devices in such an environment, i.e., devices with very limited memory and CPU resources, use today, e.g., application-layer protocols to do simple resource management and monitoring. This might be sufficient for some basic cases, however, there is a need to reconsider the network management mechanisms based on the new, changed, as well as reduced requirements coming from smart devices and the network of such



constrained devices. Albeit it is questionable whether we can take the same comprehensive approach we use in an IP network also for the management of constrained devices. Hence, the management of a network with constrained devices is necessary to design in a simplified and less complex manner.

As Section 1.6 highlights, there are diverse characteristics of constrained devices or networks, which stem from their constrainedness and therefore have an impact on the requirements for the management of such a network with constrained devices. The use cases discussed in [COM-USE] show that the requirements on constrained networks are manifold and need to be analyzed from different angles, e.g., concerning the design of the management architecture, the selection of the appropriate protocol features as well as the specific issues which are new in the context of constrained devices. Examples of such issues are e.g., the careful management of the scarce energy resources, the necessity for self-organization and self-management of such devices but also the implementation considerations to enable the use of common communication technologies on a constrained hardware in an efficient manner. For an exhaustive list of issues and requirements that need to be addressed for the management of a network with constrained devices please see Section 1.6 and Section 3.

### 3. Requirements on the Management of Networks with Constrained Devices

This section describes the requirements categorized by management areas listed in subsections.

Note that the requirements listed in this section have been separated from the context in which they may appear. This document in general does not recommend the realization of any subset of the described requirements. As such this document avoids selecting any of the requirements as mandatory to implement. A device might be able to provide only a particular selected set of requirements and might not be capable to provide all requirements in this document. On the other hand a device vendor might select a specific relevant subset of the requirements to implement.

The following template is used for the definition of the requirements.

Req-ID: An ID composed by two numbers: section number indicating the topic area and a unique three-digit number per section

Title: The title of the requirement.

Description: The rationale and description of the requirement.

Source: The origin of the requirement and the matching use case or application. For the discussion of referred use cases for constrained management please see [COM-USE].

Requirement Type: Functional Requirement, Non-Functional Requirement. A functional requirement is related to a function or component. As such functional requirements may be technical details, or specific functionality that define what a system is supposed to accomplish. Non-functional requirements (also known as design constraints or quality requirements) impose implementation-related considerations such as performance requirements, security, or reliability.

Device type: The device types by which this requirement can be supported: C0, C1 and/or C2.

Priority: The priority of the requirement showing its importance for a particular type of device: High, Medium, and Low. The priority of a requirement can be High e.g., for a C2 device but Low for a C1 or C0 device as the realization of complex features in a C1 device is in many cases not possible.

### 3.1. Management Architecture/System

Req-ID: 1.001

Title: Support multiple device classes within a single network.

Description: Larger networks usually consist of devices belonging to different device classes (e.g., constrained mesh endpoints and less constrained routers) communicating with each other. Hence, the management architecture must be applicable to networks that have a mix of different device classes. See Section 3. of [RFC7228] for the definition of Constrained Device Classes.

Source: All use cases.

Requirement Type: Non-Functional Requirement

Device type: C1 and/or C2

Priority: High

---

Req-ID: 1.002

Title: Management scalability.

Description: The management architecture must be able to scale with the number of devices involved and operate efficiently in any network size and topology. This implies that e.g., the managing entity is able to handle large amounts of device monitoring data and the management protocol is not sensitive to the decrease of the time between two client requests. To achieve good scalability, caching techniques, in-network data aggregation techniques, hierarchical management models may be used.

Source: General requirement for all use cases to enable large scale networks.

Requirement Type: Non-Functional Requirement

Device type: C0, C1, and C2

Priority: High

---

Req-ID: 1.003

Title: Hierarchical management

Description: Provide a means of hierarchical management, i.e., provide intermediary management entities on different levels, which can take over the responsibility for the management of a sub-hierarchy of the network of constraint devices. The intermediary management entity can e.g., support management data aggregation to handle e.g., high-frequent monitoring data or provide a caching mechanism for the uplink and downlink communication. Hierarchical management contributes to management scalability.

Source: Use cases where a large amount of devices are deployed with a hierarchical topology.

Requirement Type: Non-Functional Requirement

Device type: Managing and intermediary entities.

Priority: Medium

---

Req-ID: 1.004

Title: Minimize state maintained on constrained devices.

Description: The amount of state that needs to be maintained on constrained devices should be minimized. This is important in order to save memory (especially relevant for C0 and C1 devices) and in order to allow devices to restart for example to apply configuration changes or to recover from extended periods of inactivity.

Note: One way to achieve this is to adopt a RESTful architecture that minimizes the amount of state maintained by managed constrained devices and that makes resources of a device addressable via URIs.

Source: Basic requirement which concerns all use cases.

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: High

---

Req-ID: 1.005

Title: Automatic re-synchronization with eventual consistency.

Description: To support large scale networks, where some constrained devices may be offline at any point in time, it is necessary to distribute configuration parameters in a way that allows temporary inconsistencies but eventually converges, after a sufficiently long period of time without further changes, towards global consistency.

Source: Use cases with large scale networks with many devices.

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: High

---

Req-ID: 1.006

Title: Support for lossy links and unreachable devices

Description: Some constrained devices will only be able to support lossy and unreliable links characterized by a limited data rate, a high latency, and a high transmission error rate. Furthermore, constrained devices often duty cycle their radio or the whole device in order to save energy. Some classes of devices labeled as 'sleepy endpoints' set their network links to a disconnected state during long periods of time. In all cases the management system must not assume that constrained devices are always reachable.

Source: Basic requirement for networks of constrained devices with unreliable links and constrained devices that sleep to save energy.

Requirement Type: Non-Functional Requirement

Device type: C0, C1, and C2

Priority: High

---

Req-ID: 1.007

Title: Network-wide configuration

Description: Provide means by which the behavior of the network can be specified at a level of abstraction (network-wide configuration) higher than a set of configuration information specific to individual devices. It is useful to derive the device specific configuration from the network-wide configuration. Such a repository can be used to configure pre-defined device or protocol parameters for the whole network. Furthermore, such a network-wide view can be used to monitor and manage a group of routers or a whole network. E.g., monitoring the performance of a network requires additional information other than what can be acquired from a single router using a management protocol.

Note: The identification of the relevant subset of the policies to be provisioned is according to the capabilities of each device and can be obtained from a pre-configured data-repository.

Source: In general all use cases of network and device configuration based on a network view in a top-down manner.

Requirement Type: Non-Functional Requirement

Device type: C0, C1, and C2

Priority: Medium

---

Req-ID: 1.008

Title: Distributed management

Description: Provide a means of simple distributed management, where a network of constrained devices can be managed or monitored by more than one manager. Since the connectivity to a server cannot be guaranteed at all times, a distributed approach may provide a higher reliability, at the cost of increased complexity. This requirement implies the handling of data consistency in case of concurrent read and write access to the device datastore. It might also happen that no management (configuration) server is accessible and the only reachable node is a peer device. In this case the device should be able to obtain its configuration from peer devices.

Source: Use cases where the count of devices to manage is high.

Requirement Type: Non-Functional Requirement

Device type: C1 and C2

Priority: Medium

### 3.2. Management Protocols and Data Models

Req-ID: 2.001

Title: Modular implementation of management protocols

Description: Management protocols should be specified to allow for modular implementations, i.e., it should be possible to implement only a basic set of protocol primitives on highly constrained devices while devices with additional resources may provide more support for additional protocol primitives. See Section 1.7 for a discussion on the level of configuration management and monitoring support constrained devices may provide.

Source: Basic requirement interesting for all use cases.

Requirement Type: Non-Functional Requirement

Device type: C0, C1, and C2

Priority: High

---

Req-ID: 2.002

Title: Compact encoding of management data

Description: The encoding of management data should be compact and space efficient, enabling small message sizes.

Source: General requirement to save memory for the receiver buffer and on-air bandwidth.

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: High

---

Req-ID: 2.003

Title: Compression of management data or complete messages

Description: Management data exchanges can be further optimized by applying data compression techniques or delta encoding techniques. Compression typically requires additional code size and some additional buffers and/or the maintenance of some additional state information. For C0 devices compression may not be feasible.

Source: Use cases where it is beneficial to reduce transmission time and bandwidth, e.g., mobile applications which require to save on-air bandwidth.

Requirement Type: Functional Requirement

Device type: C1 and C2

Priority: Medium

---

Req-ID: 2.004

Title: Mapping of management protocol interactions

Description: It is desirable to have a lossless automated mapping between the management protocol used to manage constrained devices and the management protocols used to manage regular devices. In the ideal case, the same core management protocol can be used with certain restrictions taking into account the resource limitations of constrained devices. However, for very resource constrained devices, this goal might not be achievable.

Source: Use cases where high-frequent interaction with the management system of a non-constrained network is required.

Requirement Type: Functional Requirement

Device type: C1 and C2

Priority: Medium

---

Req-ID: 2.005

Title: Consistency of data models with the underlying information model

Description: The data models used by the management protocol must be consistent with the information model used to define data models for non-constrained networks. This is essential to facilitate the integration of the management of constrained networks with the management of non-constrained networks. Using an underlying information model for future data model design enables furthermore top-down model design and model reuse as well as data interoperability (i.e., exchange of management information between the constrained and non-constrained networks). This is a strong requirement, even despite the fact that the underlying information models are often not explicitly documented in the IETF.

Source: General requirement to support data interoperability, consistency and model reuse.

Requirement Type: Non-Functional Requirement

Device type: C0, C1, and C2

Priority: High

---

Req-ID: 2.006



Title: Lossless mapping of management data models.

Description: It is desirable to have a lossless automated mapping between the management data models used to manage regular devices and the management data models used for managing constrained devices. In the ideal case, the same core data models can be used with certain restrictions taking into account the resource limitations of constrained devices. However, for very resource constrained devices, this goal might not be achievable.

Source: Use cases where consistent data exchange with the management system of a non-constrained network is required.

Requirement Type: Functional Requirement

Device type: C2

Priority: Medium

---

Req-ID: 2.007

Title: Protocol extensibility

Description: Provide means of extensibility for the management protocol, i.e., by adding new protocol messages or mechanisms that can deal with changing requirements on a supported message and data types effectively, without causing interoperability problems or having to replace/update large amount of deployed devices.

Source: Basic requirement useful for all use cases.

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: High

### 3.3. Configuration Management

Req-ID: 3.001

Title: Self-configuration capability

Description: Automatic configuration and re-configuration of devices without manual intervention. Compared to the traditional management of devices where the management application is the

central entity configuring the devices, in the auto-configuration scenario the device is the active part and initiates the configuration process. Self-configuration can be initiated during the initial configuration or for subsequent configurations, where the configuration data needs to be refreshed. Self-configuration should be also supported during the initialization phase or in the event of failures, where prior knowledge of the network topology is not available or the topology of the network is uncertain.

Source: In general all use cases requiring easy deployment and plug&play behavior as well as easy maintenance of many constrained devices.

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: High for device categories C0 and C1, Medium for C2.

---

Req-ID: 3.002

Title: Capability discovery

Description: Enable the discovery of supported optional management capabilities of a device and their exposure via at least one protocol and/or data model.

Source: Use cases where the device interaction with other devices or applications is a function of the level of support for its capabilities.

Requirement Type: Functional Requirement

Device type: C1 and C2

Priority: Medium

---

Req-ID: 3.003

Title: Asynchronous transaction support

Description: Provide configuration management with asynchronous (event-driven) transaction support. Configuration operations must

support a transactional model, with asynchronous indications that the transaction was completed.

Source: Use cases that require transaction-oriented processing because of reliability or distributed architecture functional requirements.

Requirement Type: Functional Requirement

Device type: C1 and C2

Priority: Medium

---

Req-ID: 3.004

Title: Network reconfiguration

Description: Provide a means of iterative network reconfiguration in order to recover the network from node and communication failures. The network reconfiguration can be failure-driven and self-initiated (automatic reconfiguration). The network reconfiguration can be also performed on the whole hierarchical structure of a network (network topology).

Source: Practically all use cases, as network connectivity is a basic requirement.

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: Medium

### 3.4. Monitoring Functionality

Req-ID: 4.001

Title: Device status monitoring

Description: Provide a monitoring function to collect and expose information about device status and exposing it via at least one management interface. The device monitoring might make use of the hierarchical management through the intermediary entities and the caching mechanism. The device monitoring might also make use of neighbor-monitoring (fault detection in local network) to support fast fault detection and recovery, e.g., in a scenario where a

managing entity is unreachable and a neighbor can take over the monitoring responsibility.

Source: All use cases

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: High, Medium for neighbor-monitoring.

---

Req-ID: 4.002

Title: Energy status monitoring

Description: Provide a monitoring function to collect and expose information about device energy parameters and usage (e.g., battery level and average power consumption).

Source: Use case Energy Management

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: High for energy reporting devices, Low for others.

---

Req-ID: 4.003

Title: Monitoring of current and estimated device availability

Description: Provide a monitoring function to collect and expose information about current device availability (energy, memory, computing power, forwarding plane utilization, queue buffers, etc.) and estimation of remaining available resources.

Source: All use cases. Note that monitoring energy resources (like battery status) may be required on all kinds of devices.

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: Medium

---

Req-ID: 4.004

Title: Network status monitoring

Description: Provide a monitoring function to collect, analyze and expose information related to the status of a network or network segments connected to the interface of the device.

Source: All use cases.

Requirement Type: Functional Requirement

Device type: C1 and C2

Priority: Low, based on the realization complexity.

---

Req-ID: 4.005

Title: Self-monitoring

Description: Provide self-monitoring (local fault detection) feature for fast fault detection and recovery.

Source: Use cases where the devices cannot be monitored centrally in appropriate manner, e.g., self-healing is required.

Requirement Type: Functional Requirement

Device type: C1 and C2

Priority: High for C2, Medium for C1

---

Req-ID: 4.006

Title: Performance monitoring

Description: The device will provide a monitoring function to collect and expose information about the basic performance parameter of the device. The performance management functionality might make use of the hierarchical management through the intermediary devices.

Source: Use cases Building automation, and Transport applications

Requirement Type: Functional Requirement

Device type: C1 and C2

Priority: Low

---

Req-ID: 4.007

Title: Fault detection monitoring

Description: The device will provide fault detection monitoring. The system collects information about network states in order to identify whether faults have occurred. In some cases the detection of the faults might be based on the processing and analysis of the parameters retrieved from the network or other devices. In case of C0 devices the monitoring might be limited to the check whether the device is alive or not.

Source: Use cases Environmental Monitoring, Building Automation, Energy Management, Infrastructure Monitoring

Requirement Type: Functional Requirement

Device type: C0, C1 and C2

Priority: Medium

---

Req-ID: 4.008

Title: Passive and reactive monitoring

Description: The device will provide passive and reactive monitoring capabilities. The system or manager collects information about device components and network states (passive monitoring) and may perform postmortem analysis of collected data. In case events of interest have occurred the system or manager can adaptively react (reactive monitoring), e.g., reconfigure the network. Typically actions (re-actions) will be executed or sent as commands by the management applications.

Source: Diverse use cases relevant for device status and network state monitoring

Requirement Type: Functional Requirement

Device type: C2

Priority: Medium

---

Req-ID: 4.009

Title: Recovery

Description: Provide local, central and hierarchical recovery mechanisms (recovery is in some cases achieved by recovering the whole network of constrained devices).

Source: Use cases Industrial applications, Home and Building Automation, Mobile Applications that involve different forms of clustering or area managers.

Requirement Type: Functional Requirement

Device type: C2

Priority: Medium

---

Req-ID: 4.010

Title: Network topology discovery

Description: Provide a network topology discovery capability (e.g., use of topology extraction algorithms to retrieve the network state) and a monitoring function to collect and expose information about the network topology.

Source: Use cases Community Network Applications and Mobile Applications

Requirement Type: Functional Requirement

Device type: C1 and C2

Priority: Low, based on the realization complexity.

---

Req-ID: 4.011

Title: Notifications

Description: The device will provide the capability of sending notifications on critical events and faults.

Source: All use cases.

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: Medium for C2, Low for C0 and C1

---

Req-ID: 4.012

Title: Logging

Description: The device will provide the capability of building, keeping, and allowing retrieval of logs of events (including but not limited to critical faults and alarms).

Source: Use cases Industrial Applications, Building Automation, Infrastructure monitoring

Requirement Type: Functional Requirement

Device type: C2

Priority: High for some medical or industrial applications, Medium otherwise

### 3.5. Self-management

Req-ID: 5.001

Title: Self-management - Self-healing

Description: Enable event-driven and/or periodic self-management functionality in a device. The device should be able to react in case of a failure e.g., by initiating a fully or partly reset and initiate a self-configuration or management data update as necessary. A device might be further able to check for failures cyclically or schedule-controlled to trigger self-management as necessary. It is a matter of device design and subject for



discussion how much self-management a C1 device can support. A minimal failure detection and self-management logic is assumed to be generally useful for the self-healing of a device.

Source: The requirement generally relates to all use cases in this document.

Requirement Type: Functional Requirement

Device type: C1 and C2

Priority: High for C2, Medium for C1

### 3.6. Security and Access Control

Req-ID: 6.001

Title: Authentication of management system and devices.

Description: Systems having a management role must be properly authenticated to the device such that the device can exercise proper access control and in particular distinguish rightful management systems from rogue systems. On the other hand managed devices must authenticate themselves to systems having a management role such that management systems can protect themselves from rogue devices. In certain application scenarios, it is possible that a large number of devices need to be (re)started at about the same time. Protocols and authentication systems should be designed such that a large number of devices (re)starting simultaneously does not negatively impact the device authentication process.

Source: Basic security requirement for all use cases.

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: High, Medium for the (re)start of a large number of devices

---

Req-ID: 6.002

Title: Support suitable security bootstrapping mechanisms

Description: Mechanisms should be supported that simplify the bootstrapping of device that is the discovery of newly deployed devices in order to provide them with appropriate access control permissions.

Source: Basic security requirement for all use cases.

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: High

---

Req-ID: 6.003

Title: Access control on management system and devices

Description: Systems acting in a management role must provide an access control mechanism that allows the security administrator to restrict which devices can access the managing system (e.g., using an access control white list of known devices). On the other hand managed constrained devices must provide an access control mechanism that allows the security administrator to restrict how systems in a management role can access the device (e.g., no-access, read-only access, and read-write access).

Source: Basic security requirement for use cases where access control is essential.

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: High

---

Req-ID: 6.004

Title: Select cryptographic algorithms that are efficient in both code space and execution time.

Description: Cryptographic algorithms have a major impact in terms of both code size and overall execution time. It is therefore necessary to select mandatory to implement cryptographic algorithms that are reasonable to implement with the available

code space and that have a small impact at runtime. Furthermore some wireless technologies (e.g., IEEE 802.15.4) require the support of certain cryptographic algorithms. It might be useful to choose algorithms that are likely to be supported in wireless chipsets for certain wireless technologies.

Source: Generic requirement to reduce the footprint and CPU usage of a constrained device.

Requirement Type: Non-Functional Requirement

Device type: C0, C1, and C2

Priority: High, Medium for hardware-supported algorithms.

### 3.7. Energy Management

Req-ID: 7.001

Title: Management of energy resources

Description: Enable managing power resources in the network, e.g., reduce the sampling rate of nodes with critical battery and reduce node transmission power, put nodes to sleep, put single interfaces to sleep, reject a management job based on available energy, criteria e.g., importance levels pre-defined by the management application, etc. (e.g., a task marked as essential can be executed even if the energy level is low). The device may further implement standard data models for energy management and expose it through a management protocol interface, e.g., EMAN MIB modules and extensions (work ongoing). It might be necessary to use a subset of EMAN MIBs for C1 and C2 devices.

Source: Use case Energy Management

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: Medium for the use case Energy Management, Low otherwise.

---

Req-ID: 7.002

Title: Support of energy-optimized communication protocols

Description: Use of an optimized communication protocol to minimize energy usage for the device (radio) receiver/transmitter, on-air bandwidth (protocol efficiency), reduced amount of data communication between nodes (implies data aggregation and filtering but also a compact format for the transferred data).

Source: Use cases Energy Management and Mobile Applications.

Requirement Type: Non-Functional Requirement

Device type: C2

Priority: Medium

---

Req-ID: 7.003

Title: Support for layer 2 energy-aware protocols

Description: The device will support layer 2 energy management protocols (e.g., energy-efficient Ethernet IEEE 802.3az) and be able to report on these.

Source: Use case Energy Management

Requirement Type: Non-Functional Requirement

Device type: C0, C1, and C2

Priority: Medium

---

Req-ID: 7.004

Title: Dying gasp

Description: When energy resources draw below the red line level, the device will send a dying gasp notification and perform if still possible a graceful shutdown including conservation of critical device configuration and status information.

Source: Use case Energy Management

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: Medium

### 3.8. Software Distribution

Req-ID: 8.001

Title: Group-based provisioning

Description: Support group-based provisioning, i.e., firmware update and configuration management, of a large set of constrained devices with eventual consistency and coordinated reload times. The device should accept group-based configuration management based on bulk commands, which aim similar configurations of a large set of constrained devices of the same type in a given group, and which may share a common data model. Activation of configuration may be based on pre-loaded sets of default values.

Source: All use cases

Requirement Type: Non-Functional Requirement

Device type: C0, C1, and C2

Priority: Medium

### 3.9. Traffic Management

Req-ID: 9.001

Title: Congestion avoidance

Description: Support congestion control principles as defined in [RFC2914], e.g., the ability to avoid congestion by modifying the device's reporting rate for periodical data (which is usually redundant) based on the importance and reliability level of the management data. This functionality is usually controlled by the managing entity, where the managing entity marks the data as important or relevant for reliability. However, reducing a device's reporting rate can also be initiated by a device if it is able to detect congestion or has insufficient buffer memory.

Source: Use cases with high reporting rate and traffic e.g., AMI or M2M.

Requirement Type: Non-Functional Requirement

Device type: C1 and C2

Priority: Medium

---

Req-ID: 9.002

Title: Reroute traffic

Description: Provide the ability for network nodes to redirect traffic from overloaded intermediary nodes in a network to another path in order to prevent congestion on a central server and in the primary network.

Source: Use cases with high reporting rate and traffic e.g., AMI or M2M.

Requirement Type: Non-Functional Requirement

Device type: Intermediary entity in the network.

Priority: Medium

---

Req-ID: 9.003

Title: Traffic Shaping.

Description: Provide the ability to apply traffic shaping policies to incoming and outgoing links on an overloaded intermediary node as necessary in order to reduce the amount of traffic in the network.

Source: Use cases with high reporting rate and traffic e.g., AMI or M2M.

Requirement Type: Non-Functional Requirement

Device type: Intermediary entity in the network.

Priority: Medium

### 3.10. Transport Layer

Req-ID: 10.001

Title: Scalable transport layer

Description: Enable the use of a scalable transport layer, i.e., not sensitive to a high rate of incoming client requests, which is useful for applications requiring frequent access to device data.

Source: Applications with high frequent access to the device data.

Requirement Type: Non-Functional Requirement

Device type: C0, C1 and C2

Priority: Medium

---

Req-ID: 10.002

Title: Reliable unicast transport of messages

Description: Diverse applications need a reliable transport of messages. The reliability might be achieved based on a transport protocol such as TCP or can be supported based on message repetition if an acknowledgment is missing.

Source: Generally applications benefit from the reliability of the message transport.

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: High

---

Req-ID: 10.003

Title: Best-effort multicast

Description: Provide best-effort multicast of messages, which is generally useful when devices need to discover a service provided by a server or many devices need to be configured by a managing entity at once based on the same data model.

Source: Use cases where a device needs to discover services as well as use cases with high amount of devices to manage, which are hierarchically deployed, e.g., AMI or M2M.

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: Medium

---

Req-ID: 10.004

Title: Secure message transport

Description: Enable secure message transport providing authentication, data integrity, confidentiality by using existing transport layer technologies with small footprint such as TLS/DTLS.

Source: All use cases.

Requirement Type: Non-Functional Requirements

Device type: C1 and C2

Priority: High

### 3.11. Implementation Requirements

Req-ID: 11.001

Title: Avoid complex application layer transactions requiring large application layer messages.

Description: Complex application layer transactions tend to require large memory buffers that are typically not available on C0 or C1 devices and only by limiting functionality on C2 devices. Furthermore, the failure of a single large transaction requires repeating the whole transaction. On constrained devices, it is often more desirable to split a large transaction into a sequence of smaller transactions that require less resources and allow to make progress using a sequence of smaller steps.

Source: Basic requirement which concerns all use cases with memory constrained devices.

Requirement Type: Non-Functional Requirement

Device type: C0, C1, and C2

Priority: High



---

Req-ID: 11.002

Title: Avoid reassembly of messages at multiple layers in the protocol stack.

Description: Reassembly of messages at multiple layers in the protocol stack requires buffers at multiple layers, which leads to inefficient use of memory resources. This can be avoided by making sure the application layer, the security layer, the transport layer, the IPv6 layer and any adaptation layers are aware of the limitations of each other such that unnecessary fragmentation and reassembly can be avoided. In addition, message size constraints must be announced to protocol peers such that they can adapt and avoid sending messages that can't be processed due to resource constraints on the receiving device.

Source: Basic requirement which concerns all use cases with memory constrained devices.

Requirement Type: Non-Functional Requirement

Device type: C0, C1, and C2

Priority: High

#### 4. IANA Considerations

This document does not introduce any new code-points or namespaces for registration with IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

#### 5. Security Considerations

This document discusses the problem statement and requirements on networks of constrained devices. Section 1.6 mentions a number of limitations that could prevent the implementation of strong cryptographic algorithms. Requirements for security and access control are listed in Section 3.6.

Constrained devices might be deployed often in unsafe environments, where attackers can gain physical access to the devices. As a consequence, it is crucial that devices are robust and tamper resistant, have no backdoors, do not provide services that are not essential for the primary function, and properly protect any security

credentials that may be stored on the device (e.g., by using hardware protection mechanisms). Furthermore, it is important that any credentials leaking from a single device do not simplify the attack on other (similar) devices. In particular, security credentials should never be shared.

Since constrained devices often have limited computational resources, care should be taken in choosing efficient but cryptographically strong cryptographic algorithms. Designers of constrained devices that have a long expected lifetime need to ensure that cryptographic algorithms can be updated once devices have been deployed. The ability to perform secure firmware and software updates is an important management requirement.

Constrained devices might also generate sensitive data or require the processing of sensitive data. It is therefore an important requirement to properly protect access to the data in order to protect the privacy of humans using Internet-enabled devices. For certain types of data, protection during the transmission over the network may not be sufficient and methods should be investigated that provide protection of data while it is cached or stored (e.g., when using a store-and-forward transport mechanism).

## 6. Acknowledgments

Following persons reviewed and provided valuable comments to different versions of this document:

Dominique Barthel, Andy Bierman, Carsten Bormann, Zhen Cao, Benoit Claise, Hui Deng, Bert Greevenbosch, Joel M. Halpern, Ulrich Herberg, James Nguyen, Anuj Sehgal, Zach Shelby, Peter van der Stok, Thomas Watteyne, and Bert Wijnen.

The editors would like to thank the reviewers and the participants on the Coman and OPSAWG mailing lists for their valuable contributions and comments.

## 7. Informative References

- [RFC2914] Floyd, S., "Congestion Control Principles", BCP 41, RFC 2914, September 2000.
- [RFC2501] Corson, M. and J. Macker, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations", RFC 2501, January 1999.
- [RFC6632] Ersue, M. and B. Claise, "An Overview of the IETF Network Management Standards", RFC 6632, June 2012.

- [RFC7102] Vasseur, JP., "Terms Used in Routing for Low-Power and Lossy Networks", RFC 7102, January 2014.
- [RFC7228] Bormann, C., Ersue, M., and A. Keranen, "Terminology for Constrained-Node Networks", RFC 7228, May 2014.
- [RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, June 2014.
- [RFC4919] Kushalnagar, N., Montenegro, G., and C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals", RFC 4919, August 2007.
- [RFC6550] Winter, T., Thubert, P., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, March 2012.
- [COM-USE] Ersue, M., Romascanu, D., and J. Schoenwaelder, "Constrained Management: Use Cases", draft-ietf-opsawg-coman-use-cases (work in progress), July 2014.

#### Appendix A. Change Log

- A.1. draft-ietf-opsawg-coman-probstate-reqs-04 - draft-ietf-opsawg-coman-probstate-reqs-05
  - o Extended Abstract and Overview sections to clarify the type of requirements the draft describes.
  - o Extended security highlighting the devices should make sure credentials are properly protected.
- A.2. draft-ietf-opsawg-coman-probstate-reqs-03 - draft-ietf-opsawg-coman-probstate-reqs-04
  - o Changed in section 1.3 "10<sup>-0</sup>" to "1".
  - o Clarified in section 3 how the Requirements ID is composed.
- A.3. draft-ietf-opsawg-coman-probstate-reqs-02 - draft-ietf-opsawg-coman-probstate-reqs-03
  - o General bug fixing.
  - o Stated in the abstract and introduction section that the requirements listed in the document are potential requirements.

- o Added text in section 1.3 to highlight that with the usage of 6LoWPAN and RPL multi-hop connectivity and dynamic routing can be achieved.
- A.4. draft-ietf-opsawg-coman-probstate-reqs-01 - draft-ietf-opsawg-coman-probstate-reqs-02
- o General bug fixing.
  - o Resolved the use of the term profile of requirements.
  - o Changed requirement title from Redirect traffic to Reroute traffic and the description accordingly.
  - o Changed requirement title from Traffic delay schemes to Traffic Shaping and the description accordingly.
  - o Extended Security Considerations section.
  - o Deleted empty section on Normative References.
- A.5. draft-ietf-opsawg-coman-probstate-reqs-00 - draft-ietf-opsawg-coman-probstate-reqs-01
- o General bug fixing.
  - o Added Section 1.7. on Configuration and Monitoring Functionality Levels.
  - o Changed diverse occurrences of "networks" to "networks with/of constrained devices".
  - o Introduced the term "Self-configuring infrastructureless networks" instead of MANET as it is a superset.
  - o Introduced the term 'sleepy endpoints'.
  - o Changed requirement IDs to be independent of section number.
  - o Introduced notes for parts of the requirements text if it is focusing on implementation or solution.
  - o Extended Security Considerations section.
  - o Deleted Appendix A and B on other SDO's work and related projects as they provided dynamic information and couldn't be kept up-to-date.

A.6. draft-ersue-constrained-mgmt-03 - draft-ietf-opsawg-coman-probstate-reqs-00

- o Reduced the terminology section for terminology addressed in the LWIG terminology draft. Referenced the LWIG terminology draft.
- o Checked and aligned all terminology against the LWIG terminology draft.
- o Moved section 1.4. Constrained Device Deployment Options and section 3. Use Cases to the companion document [COM-USE].
- o Renamed Section 1.3. Class of Networks in Focus to "Network Types in Focus" and removed abbreviations C0, C1 and C2 for network classes as they have not been used.
- o Changed requirement priority classes to be High, Medium and Low.
- o Changed requirement types to be Functional and Non-Functional and added text to explain the requirement types.
- o Reformulation of some text parts for more clarity.

A.7. draft-ersue-constrained-mgmt-02-03

- o Extended the terminology section and removed some of the terminology addressed in the new LWIG terminology draft. Referenced the LWIG terminology draft.
- o Moved Section 1.3. on Constrained Device Classes to the new LWIG terminology draft.
- o Class of networks considering the different type of radio and communication technologies in use and dimensions extended.
- o Extended the Problem Statement in Section 2. following the requirements listed in Section 4.
- o Following requirements, which belong together and can be realized with similar or same kind of solutions, have been merged.
  - \* Distributed Management and Peer Configuration,
  - \* Device status monitoring and Neighbor-monitoring,
  - \* Passive Monitoring and Reactive Monitoring,

- \* Event-driven self-management - Self-healing and Periodic self-management,
  - \* Authentication of management systems and Authentication of managed devices,
  - \* Access control on devices and Access control on management systems,
  - \* Management of Energy Resources and Data models for energy management,
  - \* Software distribution (group-based firmware update) and Group-based provisioning.
- o Deleted the empty section on the gaps in network management standards, as it will be written in a separate draft.
  - o Added links to mentioned external pages.
  - o Added text on OMA M2M Device Classification in appendix.

#### A.8. draft-ersue-constrained-mgmt-01-02

- o Extended the terminology section.
- o Added additional text for the use cases concerning deployment type, network topology in use, network size, network capabilities, radio technology, etc.
- o Added examples for device classes in a use case.
- o Added additional text provided by Cao Zhen (China Mobile) for Mobile Applications and by Peter van der Stok for Building Automation.
- o Added the new use cases 'Advanced Metering Infrastructure' and 'MANET Concept of Operations in Military'.
- o Added the section 'Managing the Constrainedness of a Device or Network' discussing the needs of very constrained devices.
- o Added a note that the requirements in Section 3 need to be seen as standalone requirements and the current document does not recommend any profile of requirements.
- o Added Section 3 on the detailed requirements on constrained management matched to management tasks like fault, monitoring,

configuration management, Security and Access Control, Energy Management, etc.

- o Solved nits and added references.
- o Added Appendix A on the related development in other bodies.
- o Added Appendix B on the work in related research projects.

#### A.9. draft-ersue-constrained-mgmt-00-01

- o Splitted the section on 'Networks of Constrained Devices' into the sections 'Network Topology Options' and 'Management Topology Options'.
- o Added the use case 'Community Network Applications' and 'Mobile Applications'.
- o Provided a Contributors section.
- o Extended the section on 'Medical Applications'.
- o Solved nits and added references.

#### Authors' Addresses

Mehmet Ersue (editor)  
Nokia Networks

Email: mehmet.ersue@nsn.com

Dan Romascanu  
Avaya

Email: dromasca@avaya.com

Juergen Schoenwaelder  
Jacobs University Bremen

Email: j.schoenwaelder@jacobs-university.de

Ulrich Herberg

Email: ulrich@herberg.name

Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: September 2, 2015

M. Ersue, Ed.  
Nokia Networks  
D. Romascanu  
Avaya  
J. Schoenwaelder  
A. Sehgal  
Jacobs University Bremen  
March 1, 2015

Management of Networks with Constrained Devices: Use Cases  
draft-ietf-opsawg-coman-use-cases-05

Abstract

This document discusses use cases concerning the management of networks, where constrained devices are involved. A problem statement, deployment options and the requirements on the networks with constrained devices can be found in the companion document on "Management of Networks with Constrained Devices: Problem Statement and Requirements".

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 2, 2015.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents



carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Access Technologies . . . . .	4
2.1. Constrained Access Technologies . . . . .	4
2.2. Cellular Access Technologies . . . . .	5
3. Device Lifecycle . . . . .	6
3.1. Manufacturing and Initial Testing . . . . .	6
3.2. Installation and Configuration . . . . .	6
3.3. Operation and Maintenance . . . . .	7
3.4. Recommissioning and Decommissioning . . . . .	7
4. Use Cases . . . . .	8
4.1. Environmental Monitoring . . . . .	8
4.2. Infrastructure Monitoring . . . . .	9
4.3. Industrial Applications . . . . .	10
4.4. Energy Management . . . . .	12
4.5. Medical Applications . . . . .	14
4.6. Building Automation . . . . .	15
4.7. Home Automation . . . . .	17
4.8. Transport Applications . . . . .	18
4.9. Community Network Applications . . . . .	20
4.10. Field Operations . . . . .	22
5. IANA Considerations . . . . .	23
6. Security Considerations . . . . .	24
7. Contributors . . . . .	24
8. Acknowledgments . . . . .	24
9. Informative References . . . . .	24
Appendix A. Change Log . . . . .	25
A.1. draft-ietf-opsawg-coman-use-cases-04 - draft-ietf-opsawg-coman-use-cases-05 . . . . .	25
A.2. draft-ietf-opsawg-coman-use-cases-03 - draft-ietf-opsawg-coman-use-cases-04 . . . . .	26
A.3. draft-ietf-opsawg-coman-use-cases-02 - draft-ietf-opsawg-coman-use-cases-03 . . . . .	26
A.4. draft-ietf-opsawg-coman-use-cases-01 - draft-ietf-opsawg-coman-use-cases-02 . . . . .	26
A.5. draft-ietf-opsawg-coman-use-cases-00 - draft-ietf-opsawg-coman-use-cases-01 . . . . .	28
A.6. draft-ersue-constrained-mgmt-03 - draft-ersue-opsawg-coman-use-cases-00 . . . . .	28
A.7. draft-ersue-constrained-mgmt-02-03 . . . . .	28
A.8. draft-ersue-constrained-mgmt-01-02 . . . . .	29

A.9. draft-ersue-constrained-mgmt-00-01 . . . . .	30
Authors' Addresses . . . . .	30

## 1. Introduction

Small devices with limited CPU, memory, and power resources, so called constrained devices (aka. sensor, smart object, or smart device) can be connected to a network. Such a network of constrained devices itself may be constrained or challenged, e.g., with unreliable or lossy channels, wireless technologies with limited bandwidth and a dynamic topology, needing the service of a gateway or proxy to connect to the Internet. In other scenarios, the constrained devices can be connected to a non-constrained network using off-the-shelf protocol stacks. Constrained devices might be in charge of gathering information in diverse settings including natural ecosystems, buildings, and factories and send the information to one or more server stations.

Network management is characterized by monitoring network status, detecting faults, and inferring their causes, setting network parameters, and carrying out actions to remove faults, maintain normal operation, and improve network efficiency and application performance. The traditional network management application periodically collects information from a set of elements that are needed to manage, processes the data, and presents them to the network management users. Constrained devices, however, often have limited power, low transmission range, and might be unreliable. Such unreliability might arise from device itself (e.g., battery exhausted) or from the channel being constrained (i.e., low-capacity and high-latency). They might also need to work in hostile environments with advanced security requirements or need to be used in harsh environments for a long time without supervision. Due to such constraints, the management of a network with constrained devices offers different type of challenges compared to the management of a traditional IP network.

This document aims to understand use cases for the management of a network, where constrained devices are involved. The document lists and discusses diverse use cases for the management from the network as well as from the application point of view. The list of discussed use cases is not an exhaustive one since other scenarios, currently unknown to the authors, are possible. The application scenarios discussed aim to show where networks of constrained devices are expected to be deployed. For each application scenario, we first briefly describe the characteristics followed by a discussion on how network management can be provided, who is likely going to be responsible for it, and on which time-scale management operations are likely to be carried out.

A problem statement, deployment and management topology options as well as the requirements on the networks with constrained devices can be found in the companion document [COM-REQ].

This documents builds on the terminology defined in [RFC7228] and [COM-REQ]. [RFC7228] is a base document for the terminology concerning constrained devices and constrained networks. Some use cases specific to IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs) can be found in [RFC6568].

## 2. Access Technologies

Besides the management requirements imposed by the different use cases, the access technologies used by constrained devices can impose restrictions and requirements upon the Network Management System (NMS) and protocol of choice.

It is possible that some networks of constrained devices might utilize traditional non-constrained access technologies for network access, e.g., local area networks with plenty of capacity. In such scenarios, the constrainedness of the device presents special management restrictions and requirements rather than the access technology utilized.

However, in other situations constrained or cellular access technologies might be used for network access, thereby causing management restrictions and requirements to arise as a result of the underlying access technologies.

A discussion regarding the impact of cellular and constrained access technologies is provided in this section since they impose some special requirements on the management of constrained networks. On the other hand, fixed line networks (e.g., power line communications) are not discussed here since tend to be quite static and do not typically impose any special requirements on the management of the network.

### 2.1. Constrained Access Technologies

Due to resource restrictions, embedded devices deployed as sensors and actuators in the various use cases utilize low-power low data-rate wireless access technologies such as IEEE 802.15.4, DECT ULE or Bluetooth Low-Energy (BT-LE) for network connectivity.

In such scenarios, it is important for the NMS to be aware of the restrictions imposed by these access technologies to efficiently manage these constrained devices. Specifically, such low-power low data-rate access technologies typically have small frame sizes. So

it would be important for the NMS and management protocol of choice to craft packets in a way that avoids fragmentation and reassembly of packets since this can use valuable memory on constrained devices.

Devices using such access technologies might operate via a gateway that translates between these access technologies and more traditional Internet protocols. A hierarchical approach to device management in such a situation might be useful, wherein the gateway device is in-charge of devices connected to it, while the NMS conducts management operations only to the gateway.

## 2.2. Cellular Access Technologies

Machine to machine (M2M) services are increasingly provided by mobile service providers as numerous devices, home appliances, utility meters, cars, video surveillance cameras, and health monitors, are connected with mobile broadband technologies. Different applications, e.g., in a home appliance or in-car network, use Bluetooth, Wi-Fi or ZigBee locally and connect to a cellular module acting as a gateway between the constrained environment and the mobile cellular network.

Such a gateway might provide different options for the connectivity of mobile networks and constrained devices:

- o a smart phone with 3G/4G and WLAN radio might use BT-LE to connect to the devices in a home area network,
- o a femtocell might be combined with home gateway functionality acting as a low-power cellular base station connecting smart devices to the application server of a mobile service provider,
- o an embedded cellular module with LTE radio connecting the devices in the car network with the server running the telematics service,
- o an M2M gateway connected to the mobile operator network supporting diverse IoT connectivity technologies including ZigBee and CoAP over 6LoWPAN over IEEE 802.15.4.

Common to all scenarios above is that they are embedded in a service and connected to a network provided by a mobile service provider. Usually there is a hierarchical deployment and management topology in place where different parts of the network are managed by different management entities and the count of devices to manage is high (e.g. many thousands). In general, the network is comprised by manifold type and size of devices matching to different device classes. As such, the managing entity needs to be prepared to manage devices with diverse capabilities using different communication or management

protocols. In case the devices are directly connected to a gateway they most likely are managed by a management entity integrated with the gateway, which itself is part of the Network Management System (NMS) run by the mobile operator. Smart phones or embedded modules connected to a gateway might be themselves in charge to manage the devices on their level. The initial and subsequent configuration of such a device is mainly based on self-configuration and is triggered by the device itself.

The gateway might be in charge of filtering and aggregating the data received from the device as the information sent by the device might be mostly redundant.

### 3. Device Lifecycle

Since constrained devices deployed in a network might go through multiple phases in their lifetime, it is possible for different managers of networks and/or devices to exist during different parts of the device lifetimes. An in-depth discussion regarding the possible device lifecycles can be found in [IOT-SEC].

#### 3.1. Manufacturing and Initial Testing

Typically, the lifecycle of a device begins at the manufacturing stage. During this phase the manufacturer of the device is responsible for the management and configuration of the devices. It is also possible that a certain use case might utilize multiple types of constrained devices (e.g., temperature sensors, lighting controllers, etc.) and these could be manufactured by different entities. As such, during the manufacturing stage different managers can exist for different devices. Similarly, during the initial testing phase, where device quality assurance tasks might be performed, the manufacturer remains responsible for the management of devices and networks that might comprise them.

#### 3.2. Installation and Configuration

The responsibility of managing the devices must be transferred to the installer during the installation phase. There must exist procedures for transferring management responsibility between the manufacturer and installer. The installer may be the customer or an intermediary contracted to setup the devices and their networks. It is important that the NMS utilized allows devices originating at different vendors to be managed, ensuring interoperability between them and the configuration of trust relationships between them as well.

It is possible that the installation and configuration responsibilities might lie with different entities. For example, the

installer of a device might only be responsible for cabling a network, physically installing the devices and ensuring initial network connectivity between them (e.g., configuring IP addresses). Following such an installation, the customer or a sub-contractor might actually configure the operation of the device. As such, during installation and configuration multiple parties might be responsible for managing a device and appropriate methods must be available to ensure that this management responsibility is transferred suitably.

### 3.3. Operation and Maintenance

At the outset of the operation phase, the operational responsibility of a device and network should be passed on to the customer. It is possible that the customer, however, might contract the maintenance of the devices and network to a sub-contractor. In this case, the NMS and management protocol should allow for configuring different levels of access to the devices. Since different maintenance vendors might be used for devices that perform different functions (e.g., HVAC, lighting, etc.) it should also be possible to restrict management access to devices based on the currently responsible manager.

### 3.4. Recommissioning and Decommissioning

The owner of a device might choose to replace, repurpose or even decommission it. In each of these cases, either the customer or the contracted maintenance agency must ensure that appropriate steps are taken to meet the end goal.

In case the devices needs to be replaced, the manager of the network (customer or contractor responsible) must detach the device from the network, remove all appropriate configuration and discard the device. A new device must then be configured to replace it. The NMS should allow for transferring configuration from and replacing an existing device. The management responsibility of the operation/maintenance manager would end once the device is removed from the network. During the installation of the new replacement device, the same responsibilities would apply as those during the Installation and Configuration phases.

The device being replaced may not have yet reached end-of-life, and as such, instead of being discarded it may be installed in a new location. In this case, the management responsibilities are once again resting in the hands of the entities responsible for the Installation and Configuration phases at the new location.

If a device is repurposed, then it is possible that the management responsibility for this device changes as well. For example, a device might be moved from one building to another. In this case, the managers responsible for devices and networks in each building could be different. As such, the NMS must not only allow for changing configuration but also transferring management responsibilities.

In case a device is decommissioned, the management responsibility typically ends at that point.

#### 4. Use Cases

##### 4.1. Environmental Monitoring

Environmental monitoring applications are characterized by the deployment of a number of sensors to monitor emissions, water quality, or even the movements and habits of wildlife. Other applications in this category include earthquake or tsunami early-warning systems. The sensors often span a large geographic area, they can be mobile, and they are often difficult to replace. Furthermore, the sensors are usually not protected against tampering.

Management of environmental monitoring applications is largely concerned with the monitoring whether the system is still functional and the roll-out of new constrained devices in case the system loses too much of its structure. The constrained devices themselves need to be able to establish connectivity (auto-configuration) and they need to be able to deal with events such as losing neighbors or being moved to other locations.

Management responsibility typically rests with the organization running the environmental monitoring application. Since these monitoring applications must be designed to tolerate a number of failures, the time scale for detecting and recording failures is for some of these applications likely measured in hours and repairs might easily take days. In fact, in some scenarios it might be more cost- and time-effective to not repair such devices at all. However, for certain environmental monitoring applications, much tighter time scales may exist and might be enforced by regulations (e.g., monitoring of nuclear radiation).

Since many applications of environmental monitoring sensors are likely to be in areas that are important to safety (flood monitoring, nuclear radiation monitoring, etc.) it is important for management protocols and network management systems (NMS) to ensure appropriate security protections. These protections include not only access control, integrity and availability of data, but also provide

appropriate mechanisms that can deal with situations that might be categorized as emergencies or when tampering with sensors/data might be detected.

#### 4.2. Infrastructure Monitoring

Infrastructure monitoring is concerned with the monitoring of infrastructures such as bridges, railway tracks, or (offshore) windmills. The primary goal is usually to detect any events or changes of the structural conditions that can impact the risk and safety of the infrastructure being monitored. Another secondary goal is to schedule repair and maintenance activities in a cost effective manner.

The infrastructure to monitor might be in a factory or spread over a wider area but difficult to access. As such, the network in use might be based on a combination of fixed and wireless technologies, which use robust networking equipment and support reliable communication via application layer transactions. It is likely that constrained devices in such a network are mainly C2 devices [RFC7228] and have to be controlled centrally by an application running on a server. In case such a distributed network is widely spread, the wireless devices might use diverse long-distance wireless technologies such as WiMAX, or 3G/LTE. In cases, where an in-building network is involved, the network can be based on Ethernet or wireless technologies suitable for in-building usage.

The management of infrastructure monitoring applications is primarily concerned with the monitoring of the functioning of the system. Infrastructure monitoring devices are typically rolled out and installed by dedicated experts and changes are rare since the infrastructure itself changes rarely. However, monitoring devices are often deployed in unsupervised environments and hence special attention must be given to protecting the devices from being modified.

Management responsibility typically rests with the organization owning the infrastructure or responsible for its operation. The time scale for detecting and recording failures is likely measured in hours and repairs might easily take days. However, certain events (e.g., natural disasters) may require that status information be obtained much more quickly and that replacements of failed sensors can be rolled out quickly (or redundant sensors are activated quickly). In case the devices are difficult to access, a self-healing feature on the device might become necessary. Since infrastructure monitoring is closely related to ensuring safety, management protocols and systems must provide appropriate security



protections to ensure confidentiality, integrity and availability of data.

#### 4.3. Industrial Applications

Industrial Applications and smart manufacturing refer to tasks such as networked control and monitoring of manufacturing equipment, asset and situation management, or manufacturing process control. For the management of a factory it is becoming essential to implement smart capabilities. From an engineering standpoint, industrial applications are intelligent systems enabling rapid manufacturing of new products, dynamic response to product demands, and real-time optimization of manufacturing production and supply chain networks. Potential industrial applications (e.g., for smart factories and smart manufacturing) are:

- o Digital control systems with embedded, automated process controls, operator tools, as well as service information systems optimizing plant operations and safety.
- o Asset management using predictive maintenance tools, statistical evaluation, and measurements maximizing plant reliability.
- o Smart sensors detecting anomalies to avoid abnormal or catastrophic events.
- o Smart systems integrated within the industrial energy management system and externally with the smart grid enabling real-time energy optimization.

Management of Industrial Applications and smart manufacturing may in some situations involve Building Automation tasks such as control of energy, HVAC (heating, ventilation, and air conditioning), lighting, or access control. Interacting with management systems from other application areas might be important in some cases (e.g., environmental monitoring for electric energy production, energy management for dynamically scaling manufacturing, vehicular networks for mobile asset tracking). Management of constrained devices and networks may not only refer to the management of their network connectivity. Since the capabilities of constrained devices are limited, it is quite possible that a management system would even be required to configure, monitor and operate the primary functions that a constrained device is utilized for, besides managing its network connectivity.

Sensor networks are an essential technology used for smart manufacturing. Measurements, automated controls, plant optimization, health and safety management, and other functions are provided by a

large number of networked sectors. Data interoperability and seamless exchange of product, process, and project data are enabled through interoperable data systems used by collaborating divisions or business systems. Intelligent automation and learning systems are vital to smart manufacturing but must be effectively integrated with the decision environment. The NMS utilized must ensure timely delivery of sensor data to the control unit so it may take appropriate decisions. Similarly, relaying of commands must also be monitored and managed to ensure optimal functioning. Wireless sensor networks (WSN) have been developed for machinery Condition-based Maintenance (CBM) as they offer significant cost savings and enable new functionalities. Inaccessible locations, rotating machinery, hazardous areas, and mobile assets can be reached with wireless sensors. WSNs can provide today wireless link reliability, real-time capabilities, and quality-of-service and enable industrial and related wireless sense and control applications.

Management of industrial and factory applications is largely focused on monitoring whether the system is still functional, real-time continuous performance monitoring, and optimization as necessary. The factory network might be part of a campus network or connected to the Internet. The constrained devices in such a network need to be able to establish configuration themselves (auto-configuration) and might need to deal with error conditions as much as possible locally. Access control has to be provided with multi-level administrative access and security. Support and diagnostics can be provided through remote monitoring access centralized outside of the factory.

Factory automation tasks require that continuous monitoring be used to optimize production. Groups of manufacturing and monitoring devices could be defined to establish relationships between them. To ensure timely optimization of processes, commands from the NMS must arrive at all destination within an appropriate duration. This duration could change based on the manufacturing task being performed. Installation and operation of factory networks have different requirements. During the installation phase many networks, usually distributed along different parts of the factory/assembly line, co-exist without a connection to a common backbone. A specialized installation tool is typically used to configure the functions of different types of devices, in different factory location, in a secure manner. At the end of the installation phase, interoperability between these stand-alone networks and devices must be enabled. During the operation phase, these stand-alone networks are connected to a common backbone so that they may retrieve control information from and send commands to appropriate devices.

Management responsibility is typically owned by the organization running the industrial application. Since the monitoring

applications must handle a potentially large number of failures, the time scale for detecting and recording failures is for some of these applications likely measured in minutes. However, for certain industrial applications, much tighter time scales may exist, e.g. in real-time, which might be enforced by the manufacturing process or the use of critical material. Management protocols and NMSs must ensure appropriate access control since different users of industrial control systems will have varying levels of permissions. E.g., while supervisors might be allowed to change production parameters, they should not be allowed to modify the functional configuration of devices like a technician should. It is also important to ensure integrity and availability of data since malfunctions can potentially become safety issues. This also implies that management systems must be able to react to situations that may pose dangers to worker safety.

#### 4.4. Energy Management

The EMAN working group developed an energy management framework [RFC7326] for devices and device components within or connected to communication networks. This document observes that one of the challenges of energy management is that a power distribution network is responsible for the supply of energy to various devices and components, while a separate communication network is typically used to monitor and control the power distribution network. Devices in the context of energy management can be monitored for parameters like power, energy, demand and power quality. If a device contains batteries, they can be also monitored and managed.

Energy devices differ in complexity and may include basic sensors or switches, specialized electrical meters, or power distribution units (PDU), and subsystems inside the network devices (routers, network switches) or home or industrial appliances. The operators of an Energy Management System are either the utility providers or customers that aim to control and reduce the energy consumption and the associated costs. The topology in use differs and the deployment can cover areas from small surfaces (individual homes) to large geographical areas. The EMAN requirements document [RFC6988] discusses the requirements for energy management concerning monitoring and control functions.

It is assumed that energy management will apply to a large range of devices of all classes and networks topologies. Specific resource monitoring like battery utilization and availability may be specific to devices with lower physical resources (device classes C0 or C1 [RFC7228]).

Energy management is especially relevant to the Smart Grid. A Smart Grid is an electrical grid that uses data networks to gather and to act on energy and power-related information in an automated fashion with the goal to improve the efficiency, reliability, economics, and sustainability of the production and distribution of electricity.

Smart Metering is a good example of Smart Grid based energy management applications. Different types of possibly wireless small meters produce all together a large amount of data, which is collected by a central entity and processed by an application server, which may be located within the customer's residence or off-site in a data-center. The communication infrastructure can be provided by a mobile network operator as the meters in urban areas will have most likely a cellular or WiMAX radio. In case the application server is located within the residence, such meters are more likely to use Wi-Fi protocols to interconnect with an existing network.

An Advanced Metering Infrastructure (AMI) network is another example of the Smart Grid that enables an electric utility to retrieve frequent electric usage data from each electric meter installed at a customer's home or business. Unlike Smart Metering, in which case the customer or their agents install appliance level meters, an AMI infrastructure is typically managed by the utility providers and could also include other distribution automation devices like transformers and reclosers. Meters in AMI networks typically contain constrained devices that connect to mesh networks with a low-bandwidth radio. Usage data and outage notifications can be sent by these meters to the utility's headend systems, via aggregation points of higher-end router devices that bridge the constrained network to a less constrained network via cellular, WiMAX, or Ethernet. Unlike meters, these higher-end devices might be installed on utility poles owned and operated by a separate entity.

It thereby becomes important for a management application to not only be able to work with diverse types of devices, but also over multiple links that might be operated and managed by separate entities, each having divergent policies for their own devices and network segments. During management operations, like firmware updates, it is important that the management system performs robustly in order to avoid accidental outages of critical power systems that could be part of AMI networks. In fact, since AMI networks must also report on outages, the management system might have to manage the energy properties of battery operated AMI devices themselves as well.

A management system for home based Smart Metering solutions is likely to have devices laid out in a simple topology. However, AMI networks installations could have thousands of nodes per router, i.e., higher-end device, which organize themselves in an ad-hoc manner. As such,

a management system for AMI networks will need to discover and operate over complex topologies as well. In some situations, it is possible that the management system might also have to setup and manage the topology of nodes, especially critical routers. Encryption key management and sharing in both types of networks is also likely to be important for providing confidentiality for all data traffic. In AMI networks the key may be obtained by a meter only after an end-to-end authentication process based on certificates. Smart Metering solution could adopt a similar approach or the security may be implied due to the encrypted Wi-Fi networks they become part of.

The management of such a network requires end-to-end management of and information exchange through different types of networks. However, as of today there is no integrated energy management approach and no common information model available. Specific energy management applications or network islands use their own management mechanisms.

#### 4.5. Medical Applications

Constrained devices can be seen as an enabling technology for advanced and possibly remote health monitoring and emergency notification systems, ranging from blood pressure and heart rate monitors to advanced devices capable of monitoring implanted technologies, such as pacemakers or advanced hearing aids. Medical sensors may not only be attached to human bodies, they might also exist in the infrastructure used by humans such as bathrooms or kitchens. Medical applications will also be used to ensure treatments are being applied properly and they might guide people losing orientation. Fitness and wellness applications, such as connected scales or wearable heart monitors, encourage consumers to exercise and empower self-monitoring of key fitness indicators. Different applications use Bluetooth, Wi-Fi or ZigBee connections to access the patient's smartphone or home cellular connection to access the Internet.

Constrained devices that are part of medical applications are managed either by the users of those devices or by an organization providing medical (monitoring) services for physicians. In the first case, management must be automatic and/or easy to install and setup by average people. In the second case, it can be expected that devices be controlled by specially trained people. In both cases, however, it is crucial to protect the safety and privacy of the people to which medical devices are attached. Security precautions to protect access (authentication, encryption, integrity protections, etc.) to such devices may be critical to safeguarding the individual. The level of access granted to different users also may need to be

regulated. For example, an authorized surgeon or doctor must be allowed to configure all necessary options on the devices, however, a nurse or technician may only be allowed to retrieve data that can assist in diagnosis. Even though the data collected by a heart beat monitor might be protected, the pure fact that someone carries such a device may need protection. As such, certain medical appliances may not want to participate in discovery and self-configuration protocols in order to remain invisible.

Many medical devices are likely to be used (and relied upon) to provide data to physicians in critical situations since the biggest market is likely elderly and handicapped people. Timely delivery of data can be quite important in certain applications like patient mobility monitoring in old-age homes. Data must reach the physician and/or emergency services within specified limits of time in order to be useful. As such, fault detection of the communication network or the constrained devices becomes a crucial function of the management system that must be carried out with high reliability and, depending on the medical appliance and its application, within seconds.

#### 4.6. Building Automation

Building automation comprises the distributed systems designed and deployed to monitor and control the mechanical, electrical and electronic systems inside buildings with various destinations (e.g., public and private, industrial, institutions, or residential). Advanced Building Automation Systems (BAS) may be deployed concentrating the various functions of safety, environmental control, occupancy, security. More and more the deployment of the various functional systems is connected to the same communication infrastructure (possibly Internet Protocol based), which may involve wired or wireless communications networks inside the building.

Building automation requires the deployment of a large number (10-100.000) of sensors that monitor the status of devices, and parameters inside the building and controllers with different specialized functionality for areas within the building or the totality of the building. Inter-node distances between neighboring nodes vary between 1 to 20 meters. The NMS must, as a result, be able to manage and monitor a large number of devices, which may be organized in multi-hop meshed networks. Distances between the nodes, and the use of constrained protocols, means that networks of nodes might be segmented. The management of such network segments and nodes in these segments should be possible. Contrary to home automation, in building management the devices are expected to be managed assets and known to a set of commissioning tools and a data storage, such that every connected device has a known origin. This requires the management system to be able to discover devices on the

network and ensure that the expected list of devices is currently matched. Management here includes verifying the presence of the expected devices and detecting the presence of unwanted devices.

Examples of functions performed by controllers in building automation are regulating the quality, humidity, and temperature of the air inside the building and lighting. Other systems may report the status of the machinery inside the building like elevators, or inside the rooms like projectors in meeting rooms. Security cameras and sensors may be deployed and operated on separate dedicated infrastructures connected to the common backbone. The deployment area of a BAS is typically inside one building (or part of it) or several buildings geographically grouped in a campus. A building network can be composed of network segments, where a network segment covers a floor, an area on the floor, or a given functionality (e.g., security cameras). It is possible that the management tasks of different types of some devices might be separated from others (e.g., security cameras might operate and be managed via a separate network to the HVAC in a building).

Some of the sensors in Building Automation Systems (for example fire alarms or security systems) register, record and transfer critical alarm information and therefore must be resilient to events like loss of power or security attacks. A management system must be able to deal with unintentional segmentation of networks due to power loss or channel unavailability. It must also be able to detect security events. Due to specific operating conditions required from certain devices, there might be a need to certify components and subsystems operating in such constrained conditions based on specific requirements. Also in some environments, the malfunctioning of a control system (like temperature control) needs to be reported in the shortest possible time. Complex control systems can misbehave, and their critical status reporting and safety algorithms need to be basic and robust and perform even in critical conditions. Providing this monitoring, configuration and notification service is an important task of the management system used in building automation.

Building automation solutions are deployed in some cases in newly designed buildings, in other cases it might be over existing infrastructures. In the first case, there is a broader range of possible solutions, which can be planned for the infrastructure of the building. In the second case the solution needs to be deployed over an existing infrastructure taking into account factors like existing wiring, distance limitations, the propagation of radio signals over walls and floors, thereby making deployment difficult. As a result, some of the existing WLAN solutions (e.g., IEEE 802.11 or IEEE 802.15) may be deployed. In mission-critical or security sensitive environments and in cases where link failures happen often,

topologies that allow for reconfiguration of the network and connection continuity may be required. Some of the sensors deployed in building automation may be very simple constrained devices for which C0 or C1 [RFC7228] may be assumed.

For lighting applications, groups of lights must be defined and managed. Commands to a group of light must arrive within 200 ms at all destinations. The installation and operation of a building network has different requirements. During the installation, many stand-alone networks of a few to 100 nodes co-exist without a connection to the backbone. During this phase, the nodes are identified with a network identifier related to their physical location. Devices are accessed from an installation tool to connect them to the network in a secure fashion. During installation, the setting of parameters of common values to enable interoperability may be required. During operation, the networks are connected to the backbone while maintaining the network identifier to physical location relation. Network parameters like address and name are stored in DNS. The names can assist in determining the physical location of the device.

It is also important for a building automation NMS to take safety and security into account. Ensuring privacy and confidentiality of data, such that unauthorized parties do not get access to it, is likely to be important since users' individual behaviors could be potentially understood via their settings. Appropriate security considerations for authorization and access control to the NMS is also important since different users are likely to have varied levels of operational permissions in the system. E.g., while end users should be able to control lighting systems, HVACs, etc., only qualified technicians should be able to configure parameters that change the fundamental operation of a device. It is also important for devices and the NMS to be able to detect and report any tampering they might detect, since these could lead to potential user safety concerns, e.g., if sensors controlling air quality are tampered with such that the levels of Carbon Monoxide become life threatening. This implies that a NMS should also be able to deal with and appropriately prioritize situations that might potentially lead to safety concerns.

#### 4.7. Home Automation

Home automation includes the control of lighting, heating, ventilation, air conditioning, appliances, entertainment and home security devices to improve convenience, comfort, energy efficiency, and safety. It can be seen as a residential extension of building automation. However, unlike a building automation system, the infrastructure in a home is operated in a considerably more ad-hoc manner. While in some installations it is likely that there is no



centralized management system, akin to a Building Automation System (BAS), available, in other situations outsourced and cloud based systems responsible for managing devices in the home might be used.

Home automation networks need a certain amount of configuration (associating switches or sensors to actuators) that is either provided by electricians deploying home automation solutions, by third party home automation service providers (e.g., small specialized companies or home automation device manufacturers) or by residents by using the application user interface provided by home automation devices to configure (parts of) the home automation solution. Similarly, failures may be reported via suitable interfaces to residents or they might be recorded and made available to services providers in charge of the maintenance of the home automation infrastructure.

The management responsibility lies either with the residents or it may be outsourced to electricians and/or third parties providing management of home automation solutions as a service. A varying combination of electricians, service providers or the residents may be responsible for different aspects of managing the infrastructure. The time scale for failure detection and resolution is in many cases likely counted in hours to days.

#### 4.8. Transport Applications

Transport application is a generic term for the integrated application of communications, control, and information processing in a transportation system. Transport telematics or vehicle telematics are used as a term for the group of technologies that support transportation systems. Transport applications running on such a transportation system cover all modes of the transport and consider all elements of the transportation system, i.e. the vehicle, the infrastructure, and the driver or user, interacting together dynamically. Examples for transport applications are inter and intra vehicular communication, smart traffic control, smart parking, electronic toll collection systems, logistic and fleet management, vehicle control, and safety and road assistance.

As a distributed system, transport applications require an end-to-end management of different types of networks. It is likely that constrained devices in a network (e.g. a moving in-car network) have to be controlled by an application running on an application server in the network of a service provider. Such a highly distributed network including cellular devices on vehicles is assumed to include a wireless access network using diverse long distance wireless technologies such as WiMAX, 3G/LTE or satellite communication, e.g. based on an embedded hardware module. As a result, the management of

constrained devices in the transport system might be necessary to plan top-down and might need to use data models obliged from and defined on the application layer. The assumed device classes in use are mainly C2 [RFC7228] devices. In cases, where an in-vehicle network is involved, C1 devices [RFC7228] with limited capabilities and a short-distance constrained radio network, e.g. IEEE 802.15.4 might be used additionally.

All Transport Applications will require an IT infrastructure to run on top of, e.g., in public transport scenarios like trains, bus or metro network infrastructure might be provided, maintained and operated by third parties like mobile network or satellite network operators. However, the management responsibility of the transport application typically rests within the organization running the transport application (in the public transport scenario, this would typically be the public transport operator). Different aspects of the infrastructure might also be managed by different entities. For example, the in-car devices are likely to be installed and managed by the manufacturer, while the public works might be responsible for the on-road vehicular communication infrastructure used by these devices. The back-end infrastructure is also likely to be maintained by third party operators. As such, the NMS must be able to deal with different network segments, each being operated and controlled by separate entities, and enable appropriate access control and security as well.

Depending on the type of application domain (vehicular or stationary) and service being provided, it would be important for the NMS to be able to function with different architectures, since different manufacturers might have their own proprietary systems relying on a specific Management Topology Option, as described in [COM-REQ]. Moreover, constituents of the network can be either private, belonging to individuals or private companies, or owned by public institutions leading to different legal and organization requirements. Across the entire infrastructure, a variety of constrained devices are likely to be used, and must be individually managed. The NMS must be able to either work directly with different types of devices, or have the ability to interoperate with multiple different systems.

The challenges in the management of vehicles in a mobile transport application are manifold. The up-to-date position of each node in the network should be reported to the corresponding management entities, since the nodes could be moving within or roaming between different networks. Secondly, a variety of troubleshooting information, including sensitive location information, needs to be reported to the management system in order to provide accurate service to the customer. Management systems dealing with mobile

nodes could possibly exploit specific patterns in the mobility of the nodes. These patterns emerge due to repetitive vehicular usage in scenarios like people commuting to work, logistics supply vehicles transporting shipments between warehouses, etc. The NMS must also be able to handle partitioned networks, which would arise due to the dynamic nature of traffic resulting in large inter-vehicle gaps in sparsely populated scenarios. Since mobile nodes might roam in remote networks, the NMS should be able to provide operating configuration updates regardless of node location.

The constrained devices in a moving transport network might be initially configured in a factory and a reconfiguration might be needed only rarely. New devices might be integrated in an ad-hoc manner based on self-management and -configuration capabilities. Monitoring and data exchange might be necessary to do via a gateway entity connected to the back-end transport infrastructure. The devices and entities in the transport infrastructure need to be monitored more frequently and can be able to communicate with a higher data rate. The connectivity of such entities does not necessarily need to be wireless. The time scale for detecting and recording failures in a moving transport network is likely measured in hours and repairs might easily take days. It is likely that a self-healing feature would be used locally. On the other hand, failures in fixed transport application infrastructure (e.g., traffic-lights, digital signage displays) is likely to be measured in minutes so as to avoid untoward traffic incidents. As such, the NMS must be able to deal with differing timeliness requirements based on the type of devices.

Since transport applications of the constrained devices and networks deal with automotive vehicles, malfunctions and misuse can potentially lead to safety concerns as well. As such, besides access control, privacy of user data and timeliness management systems should also be able to detect situations that are potentially hazardous to safety. Some of these situations could be automatically mitigated, e.g., traffic lights with incorrect timing, but others might require human intervention, e.g., failed traffic lights. The management system should take appropriate actions in these situations. Maintaining data confidentiality and integrity is also an important security aspect of a management system since tampering (or malfunction) can also lead to potentially dangerous situations.

#### 4.9. Community Network Applications

Community networks are comprised of constrained routers in a multi-hop mesh topology, communicating over a lossy, and often wireless channels. While the routers are mostly non-mobile, the topology may be very dynamic because of fluctuations in link quality of the

(wireless) channel caused by, e.g., obstacles, or other nearby radio transmissions. Depending on the routers that are used in the community network, the resources of the routers (memory, CPU) may be more or less constrained - available resources may range from only a few kilobytes of RAM to several megabytes or more, and CPUs may be small and embedded, or more powerful general-purpose processors. Examples of such community networks are the FunkFeuer network (Vienna, Austria), FreiFunk (Berlin, Germany), Seattle Wireless (Seattle, USA), and AWMN (Athens, Greece). These community networks are public and non-regulated, allowing their users to connect to each other and - through an uplink to an ISP - to the Internet. No fee, other than the initial purchase of a wireless router, is charged for these services. Applications of these community networks can be diverse, e.g., location based services, free Internet access, file sharing between users, distributed chat services, social networking, video sharing, etc.

As an example of a community network, the FunkFeuer network comprises several hundred routers, many of which have several radio interfaces (with omnidirectional and some directed antennas). The routers of the network are small-sized wireless routers, such as the Linksys WRT54GL, available in 2011 for less than 50 Euros. These routers, with 16 MB of RAM and 264 MHz of CPU power, are mounted on the rooftops of the users. When new users want to connect to the network, they acquire a wireless router, install the appropriate firmware and routing protocol, and mount the router on the rooftop. IP addresses for the router are assigned manually from a list of addresses (because of the lack of auto-configuration standards for mesh networks in the IETF).

While the routers are non-mobile, fluctuations in link quality require an ad hoc routing protocol that allows for quick convergence to reflect the effective topology of the network (such as NHDP [RFC6130] and OLSRv2 [RFC7181] developed in the MANET WG). Usually, no human interaction is required for these protocols, as all variable parameters required by the routing protocol are either negotiated in the control traffic exchange, or are only of local importance to each router (i.e. do not influence interoperability). However, external management and monitoring of an ad hoc routing protocol may be desirable to optimize parameters of the routing protocol. Such an optimization may lead to a more stable perceived topology and to a lower control traffic overhead, and therefore to a higher delivery success ratio of data packets, a lower end-to-end delay, and less unnecessary bandwidth and energy usage.

Different use cases for the management of community networks are possible:

- o One single Network Management Station, e.g. a border gateway providing connectivity to the Internet, requires managing or monitoring routers in the community network, in order to investigate problems (monitoring) or to improve performance by changing parameters (managing). As the topology of the network is dynamic, constant connectivity of each router towards the management station cannot be guaranteed. Current network management protocols, such as SNMP and NETCONF, may be used (e.g., using interfaces such as the NHDMP-MIB [RFC6779]). However, when routers in the community network are constrained, existing protocols may require too many resources in terms of memory and CPU; and more importantly, the bandwidth requirements may exceed the available channel capacity in wireless mesh networks. Moreover, management and monitoring may be unfeasible if the connection between the network management station and the routers is frequently interrupted.
- o Distributed network monitoring, in which more than one management station monitors or manages other routers. Because connectivity to a server cannot be guaranteed at all times, a distributed approach may provide a higher reliability, at the cost of increased complexity. Currently, no IETF standard exists for distributed monitoring and management.
- o Monitoring and management of a whole network or a group of routers. Monitoring the performance of a community network may require more information than what can be acquired from a single router using a network management protocol. Statistics, such as topology changes over time, data throughput along certain routing paths, congestion etc., are of interest for a group of routers (or the routing domain) as a whole. As of 2014, no IETF standard allows for monitoring or managing whole networks, instead of single routers.

#### 4.10. Field Operations

The challenges of configuration and monitoring of networks operated in the field by rescue and security agencies can be different from the other use cases since the requirements and operating conditions of such networks are quite different.

With technology advancements, field networks operated nowadays are becoming large and can consist of varieties of different types of equipment that run different protocols and tools that obviously increase complexity of these mission-critical networks. In many scenarios, configurations are, most likely, manually performed. Furthermore, some legacy and even modern devices do not even support IP networking. A majority of protocols and tools developed by

vendors that are being used are proprietary, which makes integration more difficult.

The main reason for this disjoint operation scenario is that most equipment is developed with specific task requirements in mind, rather than interoperability of the varied equipment types. For example, the operating conditions experienced by high altitude security equipment is significantly different from that used in desert conditions. Similarly, search and rescue operations equipment used in case of fire rescue has different requirements than flood relief equipment. Furthermore, inter-operation of equipment with telecommunication equipment was not an expected outcome or in some scenarios this may not even be desirable.

Currently, field networks operate with a fixed Network Operations Center (NOC) that physically manages the configuration and evaluation of all field devices. Once configured, the devices might be deployed in fixed or mobile scenarios. Any configuration changes required would need to be appropriately encrypted and authenticated to prevent unauthorized access.

Hierarchical management of devices is a common requirement in such scenarios since local managers or operators may need to respond to changing conditions within their purview. The level of configuration management available at each hierarchy must also be closely governed.

Since many field operation devices are used in hostile environments, a high failure and disconnection rate should be tolerated by the NMS, which must also be able to deal with multiple gateways and disjoint management protocols.

Multi-national field operations involving search, rescue and security are becoming increasingly common, requiring inter-operation of a diverse set of equipment designed with different operating conditions in mind. Furthermore, different intra- and inter-governmental agencies are likely to have a different set of standards, best practices, rules and regulation, and implementation approaches that may contradict or conflict with each other. The NMS should be able to detect these and handle them in an acceptable manner, which may require human intervention.

## 5. IANA Considerations

This document does not introduce any new code-points or namespaces for registration with IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

## 6. Security Considerations

This document discusses use cases for management of networks with constrained devices. The security considerations described throughout the companion document [COM-REQ] apply here as well.

## 7. Contributors

Following persons made significant contributions to and reviewed this document:

- o Ulrich Herberg contributed the Section 4.9 on Community Network Applications.
- o Peter van der Stok contributed to Section 4.6 on Building Automation.
- o Zhen Cao contributed to Section 2.2 Cellular Access Technologies.
- o Gilman Tolle contributed the Section 4.4 on Automated Metering Infrastructure.
- o James Nguyen and Ulrich Herberg contributed to Section 4.10 on Military operations.

## 8. Acknowledgments

Following persons reviewed and provided valuable comments to different versions of this document:

Dominique Barthel, Carsten Bormann, Zhen Cao, Benoit Claise, Bert Greevenbosch, Ulrich Herberg, Ted Lemon, Kathleen Moriarty, James Nguyen, Zach Shelby, Peter van der Stok, and Martin Thomson.

The editors would like to thank the reviewers and the participants on the Coman maillist for their valuable contributions and comments.

## 9. Informative References

- [RFC6130] Clausen, T., Dearlove, C., and J. Dean, "Mobile Ad Hoc Network (MANET) Neighborhood Discovery Protocol (NHDP)", RFC 6130, April 2011.
- [RFC6568] Kim, E., Kaspar, D., and JP. Vasseur, "Design and Application Spaces for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6568, April 2012.

- [RFC6779] Herberg, U., Cole, R., and I. Chakeres, "Definition of Managed Objects for the Neighborhood Discovery Protocol", RFC 6779, October 2012.
- [RFC6988] Quittek, J., Chandramouli, M., Winter, R., Dietz, T., and B. Claise, "Requirements for Energy Management", RFC 6988, September 2013.
- [RFC7181] Clausen, T., Dearlove, C., Jacquet, P., and U. Herberg, "The Optimized Link State Routing Protocol Version 2", RFC 7181, April 2014.
- [RFC7228] Bormann, C., Ersue, M., and A. Keranen, "Terminology for Constrained-Node Networks", RFC 7228, May 2014.
- [RFC7326] Parello, J., Claise, B., Schoening, B., and J. Quittek, "Energy Management Framework", RFC 7326, September 2014.
- [COM-REQ] Ersue, M., Romascanu, D., and J. Schoenwaelder, "Management of Networks with Constrained Devices: Problem Statement and Requirements", draft-ietf-opsawg-coman-probstate-reqs (work in progress), February 2014.
- [IOT-SEC] Garcia-Morchon, O., Kumar, S., Keoh, S., Hummen, R., and R. Struik, "Security Considerations in the IP-based Internet of Things", draft-garcia-core-security-06 (work in progress), September 2013.

## Appendix A. Change Log

- A.1. draft-ietf-opsawg-coman-use-cases-04 - draft-ietf-opsawg-coman-use-cases-05
  - o Added text regarding security and safety considerations to the Environmental Monitoring, Infrastructure Monitoring, Industrial Applications, Medical Applications, Building Automation and Transport Applications section.
  - o Adopted text as per comments received from Kathleen Moriarty during IESG review.
  - o Added security related text to use cases for addressing concerns raised by Ted Lemon during the IESG review.



- A.2. draft-ietf-opsawg-coman-use-cases-03 - draft-ietf-opsawg-coman-use-cases-04
- o Resolved Gen-ART review comments received from Martin Thomson.
  - o Deleted company name for the list of contributors.
  - o Added Martin Thomson to Acknowledgments section.
- A.3. draft-ietf-opsawg-coman-use-cases-02 - draft-ietf-opsawg-coman-use-cases-03
- o Updated references to take into account RFCs that have now been published
  - o Added text to the access technologies section explaining why fixed line technologies (e.g., powerline communications) have not been discussed.
  - o Created a new section, Device Lifecycle, discussing the impact of different device lifecycle stages on the management of constrained networks.
  - o Homogenized usage of device classes to form C0, C1 and C2.
  - o Ensured consistency in usage of Wi-Fi, ZigBee and other terminologies.
  - o Added text clarifying the management aspects of the Building Automation and Industrial Automation use cases.
  - o Clarified the meaning of unreliability in context of constrained devices and networks.
  - o Added information regarding the configuration and operation of factory automation use case, based on the type of information provided in the building automation use case.
  - o Fixed editorial issues discovered by reviewers.
- A.4. draft-ietf-opsawg-coman-use-cases-01 - draft-ietf-opsawg-coman-use-cases-02
- o Renamed Mobile Access Technologies section to Cellular Access Technologies
  - o Changed references to mobile access technologies to now read cellular access technologies.

- o Added text to the introduction to point out that the list of use cases is not exhaustive since others unknown to the authors might exist.
- o Updated references to take into account RFCs that have been now published.
- o Updated Environmental Monitoring section to make it clear that in some scenarios it may not be prudent to repair devices.
- o Added clarification in Infrastructure Monitoring section that reliable communication is achieved via application layer transactions
- o Removed reference to Energy Devices from Energy Management section, instead labeling them as devices within the context of energy management.
- o Reduced descriptive content in Energy Management section.
- o Rewrote text in Energy Management section to highlight management characteristics of Smart Meter and AMI networks.
- o Added text regarding timely delivery of information, and related management system characteristic, to the Medical Applications section
- o Changed subnets to network segment in Building Automation section.
- o Changed structure to infrastructure in Building Automation section, and added text to highlight associated deployment difficulties.
- o Removed Trickle timer as example of common values to be set in Building Automation section.
- o Added text regarding the possible availability of outsourced and cloud based management systems for Home Automation.
- o Added text to Transport Applications section to highlight the requirement of IT infrastructure for such applications to function on top of.
- o Merged the Transport Applications and Vehicular Networks section together. Following changes to the Vehicular Networks section were merged back into Transport Applications

- \* Replaced wireless last hops with wireless access to vehicles in Vehicular Networks.
  - \* Expanded proprietary systems to "systems relying on a specific Management Topology Option, as described in [COM-REQ]." within Vehicular Networks section.
  - \* Added text regarding mobility patterns to Vehicular Networks.
  - o Changed the Military Operations use case to Field Operations and edited the text to be suitable to such scenarios.
- A.5. draft-ietf-opsawg-coman-use-cases-00 - draft-ietf-opsawg-coman-use-cases-01
- o Reordered some use cases to improve the flow.
  - o Added "Vehicular Networks".
  - o Shortened the Military Operations use case.
  - o Started adding substance to the security considerations section.
- A.6. draft-ersue-constrained-mgmt-03 - draft-ersue-opsawg-coman-use-cases-00
- o Reduced the terminology section for terminology addressed in the LWIG and Coman Requirements drafts. Referenced the other drafts.
  - o Checked and aligned all terminology against the LWIG terminology draft.
  - o Spent some effort to resolve the intersection between the Industrial Application, Home Automation and Building Automation use cases.
  - o Moved section section 3. Use Cases from the companion document [COM-REQ] to this draft.
  - o Reformulation of some text parts for more clarity.
- A.7. draft-ersue-constrained-mgmt-02-03
- o Extended the terminology section and removed some of the terminology addressed in the new LWIG terminology draft. Referenced the LWIG terminology draft.

- o Moved Section 1.3. on Constrained Device Classes to the new LWIG terminology draft.
- o Class of networks considering the different type of radio and communication technologies in use and dimensions extended.
- o Extended the Problem Statement in Section 2. following the requirements listed in Section 4.
- o Following requirements, which belong together and can be realized with similar or same kind of solutions, have been merged.
  - \* Distributed Management and Peer Configuration,
  - \* Device status monitoring and Neighbor-monitoring,
  - \* Passive Monitoring and Reactive Monitoring,
  - \* Event-driven self-management - Self-healing and Periodic self-management,
  - \* Authentication of management systems and Authentication of managed devices,
  - \* Access control on devices and Access control on management systems,
  - \* Management of Energy Resources and Data models for energy management,
  - \* Software distribution (group-based firmware update) and Group-based provisioning.
- o Deleted the empty section on the gaps in network management standards, as it will be written in a separate draft.
- o Added links to mentioned external pages.
- o Added text on OMA M2M Device Classification in appendix.

A.8. draft-ersue-constrained-mgmt-01-02

- o Extended the terminology section.
- o Added additional text for the use cases concerning deployment type, network topology in use, network size, network capabilities, radio technology, etc.

- o Added examples for device classes in a use case.
- o Added additional text provided by Cao Zhen (China Mobile) for Mobile Applications and by Peter van der Stok for Building Automation.
- o Added the new use cases 'Advanced Metering Infrastructure' and 'MANET Concept of Operations in Military'.
- o Added the section 'Managing the Constrainedness of a Device or Network' discussing the needs of very constrained devices.
- o Added a note that the requirements in [COM-REQ] need to be seen as standalone requirements and the current document does not recommend any profile of requirements.
- o Added a section in [COM-REQ] for the detailed requirements on constrained management matched to management tasks like fault, monitoring, configuration management, Security and Access Control, Energy Management, etc.
- o Solved nits and added references.
- o Added Appendix A on the related development in other bodies.
- o Added Appendix B on the work in related research projects.

#### A.9. draft-ersue-constrained-mgmt-00-01

- o Split the section on 'Networks of Constrained Devices' into the sections 'Network Topology Options' and 'Management Topology Options'.
- o Added the use case 'Community Network Applications' and 'Mobile Applications'.
- o Provided a Contributors section.
- o Extended the section on 'Medical Applications'.
- o Solved nits and added references.

#### Authors' Addresses

Mehmet Ersue (editor)  
Nokia Networks

Email: mehmet.ersue@nsn.com

Dan Romascanu  
Avaya

Email: dromasca@avaya.com

Juergen Schoenwaelder  
Jacobs University Bremen

Email: j.schoenwaelder@jacobs-university.de

Anuj Sehgal  
Jacobs University Bremen

Email: s.anuj@jacobs-university.de

OPSAWG  
Internet Draft  
Intended status: Informational  
Expires: April 6, 2015

R. Krishnan  
Brocade Communications  
L. Yong  
Huawei USA  
A. Ghanwani  
Dell  
Ning So  
Tata Communications  
B. Khasnabish  
ZTE Corporation  
October 7, 2014

Mechanisms for Optimizing LAG/ECMP Component Link Utilization in  
Networks

draft-ietf-opsawg-large-flow-load-balancing-15.txt

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79. This document may not be modified, and derivative works of it may not be created, except to publish it as an RFC and to translate it into languages other than English.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at  
<http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at  
<http://www.ietf.org/shadow.html>

This Internet-Draft will expire on April 6, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Abstract

Demands on networking infrastructure are growing exponentially due to bandwidth hungry applications such as rich media applications and inter-data center communications. In this context, it is important to optimally use the bandwidth in wired networks that extensively use link aggregation groups and equal cost multi-paths as techniques for bandwidth scaling. This draft explores some of the mechanisms useful for achieving this.

## Table of Contents

1. Introduction.....	3
1.1. Acronyms.....	4
1.2. Terminology.....	4
2. Flow Categorization.....	5
3. Hash-based Load Distribution in LAG/ECMP.....	6
4. Mechanisms for Optimizing LAG/ECMP Component Link Utilization..	7
4.1. Differences in LAG vs ECMP.....	8
4.2. Operational Overview.....	9
4.3. Large Flow Recognition.....	10
4.3.1. Flow Identification.....	10
4.3.2. Criteria and Techniques for Large Flow Recognition..	11
4.3.3. Sampling Techniques.....	11
4.3.4. Inline Data Path Measurement.....	13
4.3.5. Use of Multiple Methods for Large Flow Recognition..	14
4.4. Load Rebalancing Options.....	14
4.4.1. Alternative Placement of Large Flows.....	14
4.4.2. Redistributing Small Flows.....	15
4.4.3. Component Link Protection Considerations.....	15
4.4.4. Load Rebalancing Algorithms.....	15
4.4.5. Load Rebalancing Example.....	16
5. Information Model for Flow Rebalancing.....	17
5.1. Configuration Parameters for Flow Rebalancing.....	17



5.2. System Configuration and Identification Parameters.....	18
5.3. Information for Alternative Placement of Large Flows.....	19
5.4. Information for Redistribution of Small Flows.....	19
5.5. Export of Flow Information.....	20
5.6. Monitoring information.....	20
5.6.1. Interface (link) utilization.....	20
5.6.2. Other monitoring information.....	21
6. Operational Considerations.....	21
6.1. Rebalancing Frequency.....	21
6.2. Handling Route Changes.....	21
6.3. Forwarding Resources.....	22
7. IANA Considerations.....	22
8. Security Considerations.....	22
9. Contributing Authors.....	22
10. Acknowledgements.....	23
11. References.....	23
11.1. Normative References.....	23
11.2. Informative References.....	23

## 1. Introduction

Networks extensively use link aggregation groups (LAG) [802.1AX] and equal cost multi-paths (ECMP) [RFC 2991] as techniques for capacity scaling. For the problems addressed by this document, network traffic can be predominantly categorized into two traffic types: long-lived large flows and other flows. These other flows, which include long-lived small flows, short-lived small flows, and short-lived large flows, are referred to as "small flows" in this document. Long-lived large flows are simply referred to as "large flows."

Stateless hash-based techniques [ITCOM, RFC 2991, RFC 2992, RFC 6790] are often used to distribute both large flows and small flows over the component links in a LAG/ECMP. However the traffic may not be evenly distributed over the component links due to the traffic pattern.

This draft describes mechanisms for optimizing LAG/ECMP component link utilization while using hash-based techniques. The mechanisms comprise the following steps -- recognizing large flows in a router; and assigning the large flows to specific LAG/ECMP component links or redistributing the small flows when a component link on the router is congested.

It is useful to keep in mind that in typical use cases for this mechanism the large flows are those that consume a significant amount of bandwidth on a link, e.g. greater than 5% of link bandwidth. The number of such flows would necessarily be fairly small, e.g. on the

order of 10's or 100's per LAG/ECMP. In other words, the number of large flows is NOT expected to be on the order of millions of flows. Examples of such large flows would be IPsec tunnels in service provider backbone networks or storage backup traffic in data center networks.

### 1.1. Acronyms

DOS: Denial of Service

ECMP: Equal Cost Multi-path

GRE: Generic Routing Encapsulation

LAG: Link Aggregation Group

MPLS: Multiprotocol Label Switching

NVGRE: Network Virtualization using Generic Routing Encapsulation

PBR: Policy Based Routing

QoS: Quality of Service

STT: Stateless Transport Tunneling

TCAM: Ternary Content Addressable Memory

VXLAN: Virtual Extensible LAN

### 1.2. Terminology

Central management entity: Refers to an entity that is capable of monitoring information about link utilization and flows in routers across the network and may be capable of making traffic engineering decisions for placement of large flows. It may include the functions of a collector [RFC 7011].

ECMP component link: An individual nexthop within an ECMP group. An ECMP component link may itself comprise a LAG.

ECMP table: A table that is used as the nexthop of an ECMP route that comprises the set of ECMP component links and the weights associated with each of those ECMP component links. The input for looking up the table is the hash value for the packet, and the weights are used to determine which values of the hash function map to a given ECMP component link.

LAG component link: An individual link within a LAG. A LAG component link is typically a physical link.

LAG table: A table that is used as the output port which is a LAG that comprises the set of LAG component links and the weights associated with each of those component links. The input for looking up the table is the hash value for the packet, and the weights are used to determine which values of the hash function map to a given LAG component link.

Large flow(s): Refers to long-lived large flow(s).

Small flow(s): Refers to any of, or a combination of, long-lived small flow(s), short-lived small flows, and short-lived large flow(s).

## 2. Flow Categorization

In general, based on the size and duration, a flow can be categorized into any one of the following four types, as shown in Figure 1:

- (a) Short-lived Large Flow (SLLF),
- (b) Short-lived Small Flow (SLSF),
- (c) Long-lived Large Flow (LLLF), and
- (d) Long-lived Small Flow (LLSF).

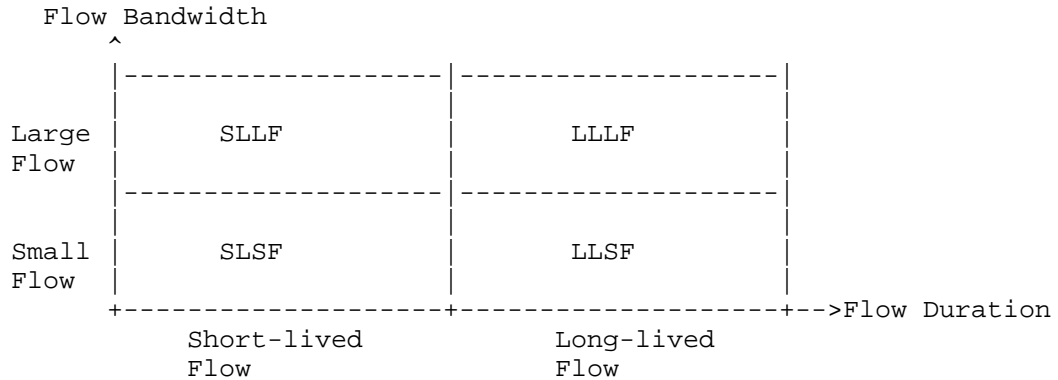


Figure 1: Flow Categorization

In this document, as mentioned earlier, we categorize long-lived large flows as "large flows", and all of the others -- long-lived small flows, short-lived small flows, and short-lived large flows as "small flows".

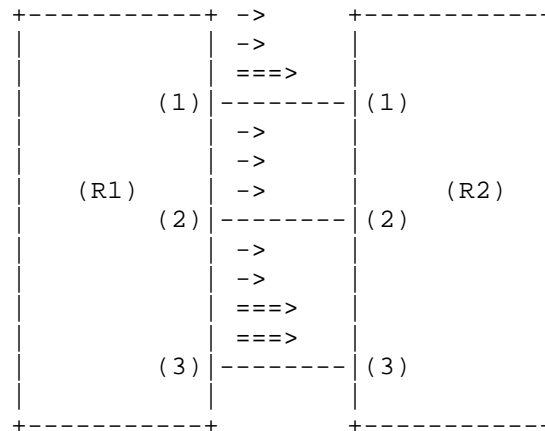
### 3. Hash-based Load Distribution in LAG/ECMP

Hash-based techniques are often used for traffic load balancing to select among multiple available paths within a LAG/ECMP group. The advantages of hash-based techniques for load distribution are the preservation of the packet sequence in a flow and the real-time distribution without maintaining per-flow state in the router. Hash-based techniques use a combination of fields in the packet's headers to identify a flow, and the hash function computed using these fields is used to generate a unique number that identifies a link/path in a LAG/ECMP group. The result of the hashing procedure is a many-to-one mapping of flows to component links.

If the traffic mix constitutes flows such that the result of the hash function across these flows is fairly uniform so that a similar number of flows is mapped to each component link, if the individual flow rates are much smaller as compared to the link capacity, and if the rate differences are not dramatic, hash-based techniques produce good results with respect to utilization of the individual component links. However, if one or more of these conditions are not met, hash-based techniques may result in imbalance in the loads on individual component links.

One example is illustrated in Figure 2. In Figure 2, there are two routers, R1 and R2, and there is a LAG between them which has 3 component links (1), (2), (3). There are a total of 10 flows that need to be distributed across the links in this LAG. The result of applying the hash-based technique is as follows:

- . Component link (1) has 3 flows -- 2 small flows and 1 large flow -- and the link utilization is normal.
- . Component link (2) has 3 flows -- 3 small flows and no large flow -- and the link utilization is light.
  - o The absence of any large flow causes the component link under-utilized.
- . Component link (3) has 4 flows -- 2 small flows and 2 large flows -- and the link capacity is exceeded resulting in congestion.
  - o The presence of 2 large flows causes congestion on this component link.



Where: ->    small flow  
      ==>    large flow

Figure 2: Unevenly Utilized Component Links

This document presents mechanisms for addressing the imbalance in load distribution resulting from commonly used hash-based techniques for LAG/ECMP that were shown in the above example. The mechanisms use large flow awareness to compensate for the imbalance in load distribution.

#### 4. Mechanisms for Optimizing LAG/ECMP Component Link Utilization

The suggested mechanisms in this draft are about a local optimization solution; they are local in the sense that both the identification of large flows and re-balancing of the load can be accomplished completely within individual nodes in the network without the need for interaction with other nodes.

This approach may not yield a global optimization of the placement of large flows across multiple nodes in a network, which may be desirable in some networks. On the other hand, a local approach may be adequate for some environments for the following reasons:

1) Different links within a network experience different levels of utilization and, thus, a "targeted" solution is needed for those hot-spots in the network. An example is the utilization of a LAG between two routers that needs to be optimized.

2) Some networks may lack end-to-end visibility, e.g. when a certain network, under the control of a given operator, is a transit

network for traffic from other networks that are not under the control of the same operator.

#### 4.1. Differences in LAG vs ECMP

While the mechanisms explained herein are applicable to both LAGs and ECMP groups, it is useful to note that there are some key differences between the two that may impact how effective the mechanism is. This relates, in part, to the localized information with which the scheme is intended to operate.

A LAG is usually established across links that are between 2 adjacent routers. As a result, the scope of problem of optimizing the bandwidth utilization on the component links is fairly narrow. It simply involves re-balancing the load across the component links between these two routers, and there is no impact whatsoever to other parts of the network. The scheme works equally well for unicast and multicast flows.

On the other hand, with ECMP, redistributing the load across component links that are part of the ECMP group may impact traffic patterns at all of the nodes that are downstream of the given router between itself and the destination. The local optimization may result in congestion at a downstream node. (In its simplest form, an ECMP group may be used to distribute traffic on component links that are between two adjacent routers, and in that case, the ECMP group is no different than a LAG for the purpose of this discussion. It should be noted that an ECMP component link may itself comprise a LAG, in which case the scheme may be further applied to the component links within the LAG.)

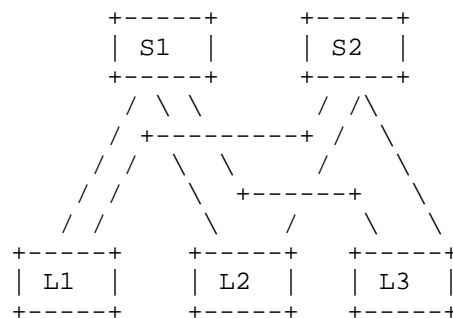


Figure 3: Two-level Clos Network

To demonstrate the limitations of local optimization, consider a two-level Clos network topology as shown in Figure 3 with three leaf nodes (L1, L2, L3) and two spine nodes (S1, S2). Assume all of the links are 10 Gbps.

Let L1 have two flows of 4 Gbps each towards L3, and let L2 have one flow of 7 Gbps also towards L3. If L1 balances the load optimally between S1 and S2, and L2 sends the flow via S1, then the downlink from S1 to L3 would get congested resulting in packet discards. On the other hand, if L1 had sent both its flows towards S1 and L2 had sent its flow towards S2, there would have been no congestion at either S1 or S2.

The other issue with applying this scheme to ECMP groups is that it may not apply equally to unicast and multicast traffic because of the way multicast trees are constructed.

Finally, it is possible for a single physical link to participate as a component link in multiple ECMP groups, whereas with LAGs, a link can participate as a component link of only one LAG.

#### 4.2. Operational Overview

The various steps in optimizing LAG/ECMP component link utilization in networks are detailed below:

Step 1) This involves large flow recognition in routers and maintaining the mapping of the large flow to the component link that it uses. The recognition of large flows is explained in Section 4.3.

Step 2) The egress component links are periodically scanned for link utilization and the imbalance for the LAG/ECMP group is monitored. If the imbalance exceeds a certain imbalance threshold, then rebalancing is triggered. Measurement of the imbalance is discussed further in 5.1. Additional criteria may also be used to determine whether or not to trigger rebalancing, such as the maximum utilization of any of the component links, in addition to the imbalance. The use of sampling techniques for the measurement of egress component link utilization, including the issues of depending on ingress sampling for these measurements, are discussed in Section 4.3.3.

Step 3) As a part of rebalancing, the operator can choose to rebalance the large flows on to lightly loaded component links of the LAG/ECMP group, redistribute the small flows on the congested link to other component links of the group, or a combination of both.

All of the steps identified above can be done locally within the router itself or could involve the use of a central management entity.

Providing large flow information to a central management entity provides the capability to globally optimize flow distribution as described in Section 4.1. Consider the following example. A router may have 3 ECMP nexthops that lead down paths P1, P2, and P3. A couple of hops downstream on path P1 there may be a congested link, while paths P2 and P3 may be under-utilized. This is something that the local router does not have visibility into. With the help of a central management entity, the operator could redistribute some of the flows from P1 to P2 and/or P3 resulting in a more optimized flow of traffic.

The mechanisms described above are especially useful when bundling links of different bandwidths for e.g. 10 Gbps and 100 Gbps as described in [ID.ietf-rtgwg-cl-requirement].

#### 4.3. Large Flow Recognition

##### 4.3.1. Flow Identification

A flow (large flow or small flow) can be defined as a sequence of packets for which ordered delivery should be maintained. Flows are typically identified using one or more fields from the packet header, for example:

- . Layer 2: Source MAC address, destination MAC address, VLAN ID.
- . IP header: IP Protocol, IP source address, IP destination address, flow label (IPv6 only)
- . Transport protocol header: Source port number, destination port number. These apply to protocols such as TCP, UDP, SCTP.
- . MPLS Labels.

For tunneling protocols like Generic Routing Encapsulation (GRE) [RFC 2784], Virtual eXtensible Local Area Network (VXLAN) [RFC 7348], Network Virtualization using Generic Routing Encapsulation (NVGRE) [NVGRE], Stateless Transport Tunneling (STT) [STT], Layer 2 Tunneling Protocol (L2TP) [RFC 3931], etc., flow identification is possible based on inner and/or outer headers as well as fields introduced by the tunnel header, as any or all such fields may be used for load balancing decisions [RFC 5640]. The above list is not exhaustive.



The mechanisms described in this document are agnostic to the fields that are used for flow identification.

This method of flow identification is consistent with that of IPFIX [RFC 7011].

#### 4.3.2. Criteria and Techniques for Large Flow Recognition

From a bandwidth and time duration perspective, in order to recognize large flows we define an observation interval and observe the bandwidth of the flow over that interval. A flow that exceeds a certain minimum bandwidth threshold over that observation interval would be considered a large flow.

The two parameters -- the observation interval, and the minimum bandwidth threshold over that observation interval -- should be programmable to facilitate handling of different use cases and traffic characteristics. For example, a flow which is at or above 10% of link bandwidth for a time period of at least 1 second could be declared a large flow [DevoFlow].

In order to avoid excessive churn in the rebalancing, once a flow has been recognized as a large flow, it should continue to be recognized as a large flow for as long as the traffic received during an observation interval exceeds some fraction of the bandwidth threshold, for example 80% of the bandwidth threshold.

Various techniques to recognize a large flow are described below.

#### 4.3.3. Sampling Techniques

A number of routers support sampling techniques such as sFlow [sFlow-v5, sFlow-LAG], PSAMP [RFC 5475] and NetFlow Sampling [RFC 3954]. For the purpose of large flow recognition, sampling needs to be enabled on all of the egress ports in the router where such measurements are desired.

Using sFlow as an example, processing in a sFlow collector will provide an approximate indication of the large flows mapping to each of the component links in each LAG/ECMP group. It is possible to implement this part of the collector function in the control plane of the router reducing dependence on an external management station, assuming sufficient control plane resources are available.

If egress sampling is not available, ingress sampling can suffice since the central management entity used by the sampling technique typically has multi-node visibility and can use the samples from an

immediately downstream node to make measurements for egress traffic at the local node.

The option of using ingress sampling for this purpose may not be available if the downstream device is under the control of a different operator, or if the downstream device does not support sampling.

Alternatively, since sampling techniques require that the sample be annotated with the packet's egress port information, ingress sampling may suffice. However, this means that sampling would have to be enabled on all ports, rather than only on those ports where such monitoring is desired. There is one situation in which this approach may not work. If there are tunnels that originate from the given router, and if the resulting tunnel comprises the large flow, then this cannot be deduced from ingress sampling at the given router. Instead, if egress sampling is unavailable, then ingress sampling from the downstream router must be used.

To illustrate the use of ingress versus egress sampling, we refer to Figure 2. Since we are looking at rebalancing flows at R1, we would need to enable egress sampling on ports (1), (2), and (3) on R1. If egress sampling is not available, and if R2 is also under the control of the same administrator, enabling ingress sampling on R2's ports (1), (2), and (3) would also work, but it would necessitate the involvement of a central management entity in order for R1 to obtain large flow information for each of its links. Finally, R1 can enable ingress sampling only on all of its ports (not just the ports that are part of the LAG/ECMP group being monitored) and that would suffice if the sampling technique annotates the samples with the egress port information.

The advantages and disadvantages of sampling techniques are as follows.

Advantages:

- . Supported in most existing routers.
- . Requires minimal router resources.

Disadvantages:

- . In order to minimize the error inherent in sampling, there is a minimum delay for the recognition time of large flows, and in the time that it takes to react to this information.

With sampling, the detection of large flows can be done on the order of one second [DevoFlow]. A discussion on determining the appropriate sampling frequency is available in the following reference [SAMP-BASIC].

#### 4.3.4. Inline Data Path Measurement

Implementations may perform recognition of large flows by performing measurements on traffic in the data path of a router. Such an approach would be expected to operate at the interface speed on every interface, accounting for all packets processed by the data path of the router. An example of such an approach is described in IPFIX [RFC 5470].

Using inline data path measurement, a faster and more accurate indication of large flows mapped to each of the component links in a LAG/ECMP group may be possible (as compared to the sampling-based approach).

The advantages and disadvantages of inline data path measurement are:

##### Advantages:

- . As link speeds get higher, sampling rates are typically reduced to keep the number of samples manageable which places a lower bound on the detection time. With inline data path measurement, large flows can be recognized in shorter windows on higher link speeds since every packet is accounted for [NDTM].
- . Eliminates the potential dependence on an external management station for large flow recognition.

##### Disadvantages:

- . It is more resource intensive in terms of the tables sizes required for monitoring all flows in order to perform the measurement.

As mentioned earlier, the observation interval for determining a large flow and the bandwidth threshold for classifying a flow as a large flow should be programmable parameters in a router.

The implementation details of inline data path measurement of large flows is vendor dependent and beyond the scope of this document.

#### 4.3.5. Use of Multiple Methods for Large Flow Recognition

It is possible that a router may have line cards that support a sampling technique while other line cards support inline data path measurement of large flows. As long as there is a way for the router to reliably determine the mapping of large flows to component links of a LAG/ECMP group, it is acceptable for the router to use more than one method for large flow recognition.

If both methods are supported, inline data path measurement may be preferable because of its speed of detection [FLOW-ACC].

#### 4.4. Load Rebalancing Options

Below are suggested techniques for load balancing. Equipment vendors may implement more than one technique, including those not described in this document, and allow the operator to choose between them.

Note that regardless of the method used, perfect rebalancing of large flows may not be possible since flows arrive and depart at different times. Also, any flows that are moved from one component link to another may experience momentary packet reordering.

##### 4.4.1. Alternative Placement of Large Flows

Within a LAG/ECMP group, the member component links with least average port utilization are identified. Some large flow(s) from the heavily loaded component links are then moved to those lightly-loaded member component links using a policy-based routing (PBR) rule in the ingress processing element(s) in the routers.

With this approach, only certain large flows are subjected to momentary flow re-ordering.

When a large flow is moved, this will increase the utilization of the link that it moved to potentially creating imbalance in the utilization once again across the component links. Therefore, when moving large flows, care must be taken to account for the existing load, and what the future load will be after large flow has been moved. Further, the appearance of new large flows may require a rearrangement of the placement of existing flows.

Consider a case where there is a LAG comprising four 10 Gbps component links and there are four large flows, each of 1 Gbps. These flows are each placed on one of the component links. Subsequent, a fifth large flow of 2 Gbps is recognized and to maintain equitable load distribution, it may require placement of one

of the existing 1 Gbps flow to a different component link. And this would still result in some imbalance in the utilization across the component links.

#### 4.4.2. Redistributing Small Flows

Some large flows may consume the entire bandwidth of the component link(s). In this case, it would be desirable for the small flows to not use the congested component link(s). This can be accomplished in one of the following ways.

This method works on some existing router hardware. The idea is to prevent, or reduce the probability, that the small flow hashes into the congested component link(s).

- . The LAG/ECMP table is modified to include only non-congested component link(s). Small flows hash into this table to be mapped to a destination component link. Alternatively, if certain component links are heavily loaded, but not congested, the output of the hash function can be adjusted to account for large flow loading on each of the component links.
- . The PBR rules for large flows (refer to Section 4.4.1) must have strict precedence over the LAG/ECMP table lookup result.

With this approach the small flows that are moved would be subject to reordering.

#### 4.4.3. Component Link Protection Considerations

If desired, certain component links may be reserved for link protection. These reserved component links are not used for any flows in the absence of any failures. In the case when the component link(s) fail, all the flows on the failed component link(s) are moved to the reserved component link(s). The mapping table of large flows to component link simply replaces the failed component link with the reserved link. Likewise, the LAG/ECMP table replaces the failed component link with the reserved link.

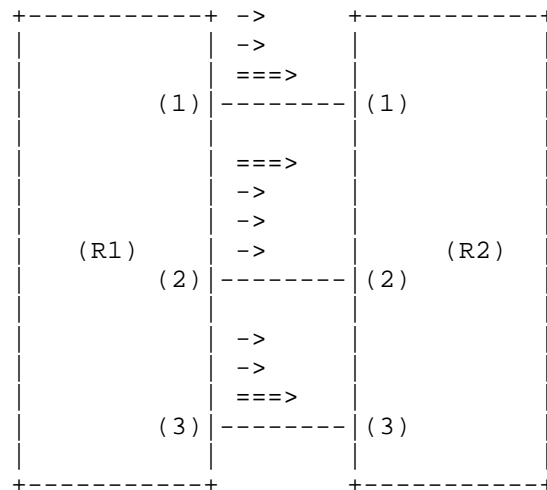
#### 4.4.4. Load Rebalancing Algorithms

Specific algorithms for placement of large flows are out of scope of this document. One possibility is to formulate the problem for large flow placement as the well-known bin-packing problem and make use of the various heuristics that are available for that problem [bin-pack].

## 4.4.5. Load Rebalancing Example

Optimizing LAG/ECMP component utilization for the use case in Figure 2 is depicted below in Figure 4. The large flow rebalancing explained in Section 4.4 is used. The improved link utilization is as follows:

- . Component link (1) has 3 flows -- 2 small flows and 1 large flow -- and the link utilization is normal.
- . Component link (2) has 4 flows -- 3 small flows and 1 large flow -- and the link utilization is normal now.
- . Component link (3) has 3 flows -- 2 small flows and 1 large flow -- and the link utilization is normal now.



Where: ->    small flow  
 ==>    large flow

Figure 4: Evenly Utilized Composite Links

Basically, the use of the mechanisms described in Section 4.4.1 resulted in a rebalancing of flows where one of the large flows on component link (3) which was previously congested was moved to component link (2) which was previously under-utilized.

## 5. Information Model for Flow Rebalancing

In order to support flow rebalancing in a router from an external system, the exchange of some information is necessary between the router and the external system. This section provides an exemplary information model covering the various components needed for the purpose. The model is intended to be informational and may be used as input for development of a data model.

### 5.1. Configuration Parameters for Flow Rebalancing

The following parameters are required the configuration of this feature:

- . Large flow recognition parameters:
  - o Observation interval: The observation interval is the time period in seconds over which the packet arrivals are observed for the purpose of large flow recognition.
  - o Minimum bandwidth threshold: The minimum bandwidth threshold would be configured as a percentage of link speed and translated into a number of bytes over the observation interval. A flow for which the number of bytes received, for a given observation interval, exceeds this number would be recognized as a large flow.
  - o Minimum bandwidth threshold for large flow maintenance: The minimum bandwidth threshold for large flow maintenance is used to provide hysteresis for large flow recognition. Once a flow is recognized as a large flow, it continues to be recognized as a large flow until it falls below this threshold. This is also configured as a percentage of link speed and is typically lower than the minimum bandwidth threshold defined above.
- . Imbalance threshold: A measure of the deviation of the component link utilizations from the utilization of the overall LAG/ECMP group. Since component links can be of a different speed, the imbalance can be computed as follows. Let the utilization of each component link in a LAG/ECMP group with  $n$  links of speed  $b_1, b_2 \dots b_n$ , be  $u_1, u_2 \dots u_n$ . The mean utilization is computed as  $u_{ave} = [ (u_1 \times b_1) + (u_2 \times b_2) + \dots + (u_n \times b_n) ] / [b_1 + b_2 + \dots + b_n]$ . The imbalance is then computed as  $\max_{\{i=1..n\}} | u_i - u_{ave} |$ .

- .    Rebalancing interval: The minimum amount of time between rebalancing events. This parameter ensures that rebalancing is not invoked too frequently as it impacts packet ordering.

These parameters may be configured on a system-wide basis or it may apply to an individual LAG. It may be applied to an ECMP group provided the component links are not shared with any other ECMP group.

## 5.2. System Configuration and Identification Parameters

The following parameters are useful for router configuration and operation when using the mechanisms in this document.

- .    IP address: The IP address of a specific router that the feature is being configured on, or that the large flow placement is being applied to.
- .    LAG ID: Identifies the LAG on a given router. The LAG ID may be required when configuring this feature (to apply a specific set of large flow identification parameters to the LAG) and will be required when specifying flow placement to achieve the desired rebalancing.
- .    Component Link ID: Identifies the component link within a LAG or ECMP group. This is required when specifying flow placement to achieve the desired rebalancing.
- .    Component Link Weight: The relative weight to be applied to traffic for a given component link when using hash-based techniques for load distribution.
- .    ECMP group: Identifies a particular ECMP group. The ECMP group may be required when configuring this feature (to apply a specific set of large flow identification parameters to the ECMP group) and will be required when specifying flow placement to achieve the desired rebalancing. We note that multiple ECMP groups can share an overlapping set (or non-overlapping subset) of component links. This document does not deal with the complexity of addressing such configurations.

The feature may be configured globally for all LAGs and/or for all ECMP groups, or it may be configured specifically for a given LAG or ECMP group.



### 5.3. Information for Alternative Placement of Large Flows

In cases where large flow recognition is handled by an external management station (see Section 4.3.3), an information model for flows is required to allow the import of large flow information to the router.

Typical fields use for identifying large flows were discussed in Section 4.3.1. The IPFIX information model [RFC 7012] can be leveraged for large flow identification.

Large Flow placement is achieved by specifying the relevant flow information along with the following:

- . For LAG: Router's IP address, LAG ID, LAG component link ID.
- . For ECMP: Router's IP address, ECMP group, ECMP component link ID.

In the case where the ECMP component link itself comprises a LAG, we would have to specify the parameters for both the ECMP group as well as the LAG to which the large flow is being directed.

### 5.4. Information for Redistribution of Small Flows

Redistribution of small flows is done using the following:

- . For LAG: The LAG ID and the component link IDs along with the relative weight of traffic to be assigned to each component link ID are required.
- . For ECMP: The ECMP group and the ECMP Nexthop along with the relative weight of traffic to be assigned to each ECMP Nexthop are required.

It is possible to have an ECMP nexthop that itself comprises a LAG. In that case, we would have to specify the new weights for both the ECMP nexthops within the ECMP group as well as the component links within the LAG.

In the case where an ECMP component link itself comprises a LAG, we would have to specify new weights for both the component links within the ECMP group as well as the component links within the LAG.

## 5.5. Export of Flow Information

Exporting large flow information is required when large flow recognition is being done on a router, but the decision to rebalance is being made in an external management station. Large flow information includes flow identification and the component link ID that the flow currently is assigned to. Other information such as flow QoS and bandwidth may be exported too.

The IPFIX information model [RFC 7012] can be leveraged for large flow identification.

## 5.6. Monitoring information

### 5.6.1. Interface (link) utilization

The incoming bytes (ifInOctets), outgoing bytes (ifOutOctets) and interface speed (ifSpeed) can be obtained, for example, from the Interface table (iftable) MIB [RFC 1213].

The link utilization can then be computed as follows:

Incoming link utilization =  $(\text{delta\_ifInOctets} * 8) / (\text{ifSpeed} * T)$

Outgoing link utilization =  $(\text{delta\_ifOutOctets} * 8) / (\text{ifSpeed} * T)$

Where T is the interval over which the utilization is being measured, delta\_ifInOctets is the change in ifInOctets over that interval, and delta\_ifOutOctets is the change in ifOutOctets over that interval.

For high speed Ethernet links, the etherStatsHighCapacityTable MIB [RFC 3273] can be used.

Similar results may be achieved using the corresponding objects of other interface management data models such as YANG [RFC 7223] if those are used instead of MIBs.

For scalability, it is recommended to use the counter push mechanism in [sflow-v5] for the interface counters. Doing so would help avoid counter polling through the MIB interface.

The outgoing link utilization of the component links within a LAG/ECMP group can be used to compute the imbalance (See Section 5.1) for the LAG/ECMP group.

#### 5.6.2. Other monitoring information

Additional monitoring information that is useful includes:

- .    Number of times rebalancing was done.
- .    Time since the last rebalancing event.
- .    The number of large flows currently rebalanced by the scheme.
- .    A list of the large flows that have been rebalanced including
  - o the rate of each large flow at the time of the last rebalancing for that flow,
  - o the time that rebalancing was last performed for the given large flow, and
  - o the interfaces that the large flows was (re)directed to.
- .    The settings for the weights of the interfaces within a LAG/ECMP used by the small flows which depend on hashing.

### 6. Operational Considerations

#### 6.1. Rebalancing Frequency

Flows should be rebalanced only when the imbalance in the utilization across component links exceeds a certain threshold. Frequent rebalancing to achieve precise equitable utilization across component links could be counter-productive as it may result in moving flows back and forth between the component links impacting packet ordering and system stability. This applies regardless of whether large flows or small flows are redistributed. It should be noted that reordering is a concern for TCP flows with even a few packets because three out-of-order packets would trigger sufficient duplicate ACKs to the sender resulting in a retransmission [RFC 5681].

The operator would have to experiment with various values of the large flow recognition parameters (minimum bandwidth threshold, observation interval) and the imbalance threshold across component links to tune the solution for their environment.

#### 6.2. Handling Route Changes

Large flow rebalancing must be aware of any changes to the FIB. In cases where the nexthop of a route no longer points to the LAG, or

to an ECMP group, any PBR entries added as described in Section 4.4.1 and 4.4.2 must be withdrawn in order to avoid the creation of forwarding loops.

### 6.3. Forwarding Resources

Hash-based techniques used for load balancing with LAG/ECMP are usually stateless. The mechanisms described in this document require additional resources in the forwarding plane of routers for creating PBR rules that are capable of overriding the forwarding decision from the hash-based approach. These resources may limit the number of flows that can be rebalanced and may also impact the latency experienced by packets due to the additional lookups that are required.

### 7. IANA Considerations

This memo includes no request to IANA.

### 8. Security Considerations

This document does not directly impact the security of the Internet infrastructure or its applications. In fact, it could help if there is a DOS attack pattern which causes a hash imbalance resulting in heavy overloading of large flows to certain LAG/ECMP component links.

An attacker with knowledge of the large flow recognition algorithm and any stateless distribution method can generate flows that are distributed in a way that overloads a specific path. This could be used to cause the creation of PBR rules that exhaust the available rule capacity on nodes. If PBR rules are consequently discarded, this could result in congestion on the attacker-selected path. Alternatively, tracking large numbers of PBR rules could result in performance degradation.

### 9. Contributing Authors

Sanjay Khanna  
Cisco Systems  
Email: sanjakha@gmail.com

## 10. Acknowledgements

The authors would like to thank the following individuals for their review and valuable feedback on earlier versions of this document: Shane Amante, Fred Baker, Michael Bugenhagen, Zhen Cao, Brian Carpenter, Benoit Claise, Michael Fargano, Wes George, Sriganesh Kini, Roman Krzanowski, Andrew Malis, Dave McDysan, Pete Moyer, Peter Phaal, Dan Romascanu, Curtis Villamizar, Jianrong Wong, George Yum, and Weifeng Zhang. As a part of the IETF Last Call process, valuable comments were received from Martin Thomson and Carlos Pignatiro.

## 11. References

### 11.1. Normative References

[802.1AX] IEEE Standards Association, "IEEE Std 802.1AX-2008 IEEE Standard for Local and Metropolitan Area Networks - Link Aggregation", 2008.

[RFC 2991] Thaler, D. and C. Hopps, "Multipath Issues in Unicast and Multicast," November 2000.

[RFC 7011] Claise, B. et al., "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information," September 2013.

[RFC 7012] Claise, B. and B. Trammell, "Information Model for IP Flow Information Export (IPFIX)," September 2013.

### 11.2. Informative References

[bin-pack] Coffman, Jr., E., M. Garey, and D. Johnson. Approximation Algorithms for Bin-Packing -- An Updated Survey. In Algorithm Design for Computer System Design, ed. by Ausiello, Lucertini, and Serafini. Springer-Verlag, 1984.

[CAIDA] "Caida Internet Traffic Analysis," <http://www.caida.org/home>.

[DevoFlow] Mogul, J., et al., "DevoFlow: Cost-Effective Flow Management for High Performance Enterprise Networks," Proceedings of the ACM SIGCOMM, August 2011.

[FLOW-ACC] Zseby, T., et al., "Packet sampling for flow accounting: challenges and limitations," Proceedings of the 9th international conference on Passive and active network measurement, 2008.

[ID.ietf-rtgwg-cl-requirement] Villamizar, C. et al., "Requirements for MPLS over a Composite Link," September 2013.

[ITCOM] Jo, J., et al., "Internet traffic load balancing using dynamic hashing with flow volume," SPIE ITCOM, 2002.

[NDTM] Estan, C. and G. Varghese, "New directions in traffic measurement and accounting," Proceedings of ACM SIGCOMM, August 2002.

[NVGRE] Sridharan, M. et al., "NVGRE: Network Virtualization using Generic Routing Encapsulation," draft-sridharan-virtualization-nvgre-06, January 2015.

[RFC 2784] Farinacci, D. et al., "Generic Routing Encapsulation (GRE)," March 2000.

[RFC 6790] Kompella, K. et al., "The Use of Entropy Labels in MPLS Forwarding," November 2012.

[RFC 1213] McCloghrie, K., "Management Information Base for Network Management of TCP/IP-based internets: MIB-II," March 1991.

[RFC 2992] Hopps, C., "Analysis of an Equal-Cost Multi-Path Algorithm," November 2000.

[RFC 3273] Waldbusser, S., "Remote Network Monitoring Management Information Base for High Capacity Networks," July 2002.

[RFC 3931] Lau, J. (Ed.), M. Townsley (Ed.), and I. Goyret (Ed.), "Layer 2 Tunneling Protocol - Version 3," March 2005.

[RFC 3954] Claise, B., "Cisco Systems NetFlow Services Export Version 9," October 2004.

[RFC 5470] G. Sadasivan et al., "Architecture for IP Flow Information Export," March 2009.

[RFC 5475] Zseby, T. et al., "Sampling and Filtering Techniques for IP Packet Selection," March 2009.

[RFC 5640] Filsfils, C., P. Mohapatra, and C. Pignataro, "Load Balancing for Mesh Softwires," August 2009.

[RFC 5681] Allman, M. et al., "TCP Congestion Control," September 2009.

[RFC 7223] Bjorklund, M., "A YANG Data Model for Interface Management," May 2014.

[SAMP-BASIC] Phaal, P. and S. Panchen, "Packet Sampling Basics," <http://www.sflow.org/packetSamplingBasics/>.

[sFlow-v5] Phaal, P. and M. Lavine, "sFlow version 5," [http://www.sflow.org/sflow\\_version\\_5.txt](http://www.sflow.org/sflow_version_5.txt), July 2004.

[sFlow-LAG] Phaal, P. and A. Ghanwani, "sFlow LAG counters structure," [http://www.sflow.org/sflow\\_lag.txt](http://www.sflow.org/sflow_lag.txt), September 2012.

[STT] Davie, B. (Ed.) and J. Gross, "A Stateless Transport Tunneling Protocol for Network Virtualization (STT)," draft-davie-stt-06, March 2014.

[RFC 7348] Mahalingam, M. et al., "VXLAN: A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks," August 2014.

[YONG] Yong, L., "Enhanced ECMP and Large Flow Aware Transport," draft-yong-pwe3-enhance-ecmp-lfat-01, September 2010.

#### Appendix A. Internet Traffic Analysis and Load Balancing Simulation

Internet traffic [CAIDA] has been analyzed to obtain flow statistics such as the number of packets in a flow and the flow duration. The five tuples in the packet header (IP addresses, TCP/UDP Ports, and IP protocol) are used for flow identification. The analysis indicates that < ~2% of the flows take ~30% of total traffic volume while the rest of the flows (> ~98%) contributes ~70% [YONG].

The simulation has shown that given Internet traffic pattern, the hash-based technique does not evenly distribute the flows over ECMP paths. Some paths may be > 90% loaded while others are < 40% loaded. The more ECMP paths exist, the more severe the misbalancing. This implies that hash-based distribution can cause some paths to become congested while other paths are underutilized [YONG].

The simulation also shows substantial improvement by using the large flow-aware hash-based distribution technique described in this document. In using the same simulated traffic, the improved rebalancing can achieve < 10% load differences among the paths. It proves how large flow-aware hash-based distribution can effectively compensate the uneven load balancing caused by hashing and the traffic characteristics [YONG].

#### Authors' Addresses

Ram Krishnan  
Brocade Communications  
San Jose, 95134, USA  
Phone: +1-408-406-7890  
Email: ramkri123@gmail.com

Lucy Yong  
Huawei USA  
5340 Legacy Drive  
Plano, TX 75025, USA  
Phone: +1-469-277-5837  
Email: lucy.yong@huawei.com

Anoop Ghanwani  
Dell  
San Jose, CA 95134  
Phone: +1-408-571-3228  
Email: anoop@alumni.duke.edu

Ning So  
Tata Communications  
Plano, TX 75082, USA  
Phone: +1-972-955-0914  
Email: ning.so@tatacommunications.com

Bhumip Khasnabish  
ZTE Corporation  
New Jersey, 07960, USA  
Phone: +1-781-752-8003



Internet-Draft    Optimizing Load Distribution over LAG/ECMP    October 2014

Email: [vumip1@gmail.com](mailto:vumip1@gmail.com)



OPSAWG  
Internet-Draft  
Intended status: Standards Track  
Expires: February 6, 2016

H. Asai  
Univ. of Tokyo  
M. MacFaden  
VMware Inc.  
J. Schoenwaelder  
Jacobs University  
K. Shima  
IIJ Innovation Institute Inc.  
T. Tsou  
Huawei Technologies (USA)  
August 5, 2015

Management Information Base for Virtual Machines Controlled by a  
Hypervisor  
draft-ietf-opsawg-vmm-mib-04

Abstract

This document defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, this specifies objects for managing virtual machines controlled by a hypervisor (a.k.a. virtual machine monitor).

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on February 6, 2016.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. The Internet-Standard Management Framework . . . . .	3
3. Overview and Objectives . . . . .	3
4. Structure of the VM-MIB Module . . . . .	5
5. Relationship to Other MIB Modules . . . . .	7
6. Definitions . . . . .	8
6.1. VM-MIB . . . . .	8
6.2. IANA-STORAGE-MEDIA-TYPE-MIB . . . . .	43
7. IANA Considerations . . . . .	44
8. Security Considerations . . . . .	44
9. Contributors . . . . .	46
10. Acknowledgements . . . . .	46
11. References . . . . .	46
11.1. Normative References . . . . .	46
11.2. Informative References . . . . .	48
Appendix A. State Transition Table . . . . .	48
Authors' Addresses . . . . .	50

## 1. Introduction

This document defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, this specifies objects for managing virtual machines controlled by a hypervisor (a.k.a. virtual machine monitor). A hypervisor controls multiple virtual machines on a single physical machine by allocating resources to each virtual machine using virtualization technologies. Therefore, this MIB module contains information on virtual machines and their resources controlled by a hypervisor as well as hypervisor's hardware and software information.

The design of this MIB module has been derived from product-specific MIB modules, namely a MIB module for managing guests of the Xen hypervisor, a MIB module for managing virtual machines controlled by the VMware hypervisor, and a MIB module using the libvirt programming interface to access different hypervisors. However, this MIB module

attempts to generalize the managed objects to support other implementations of hypervisors.

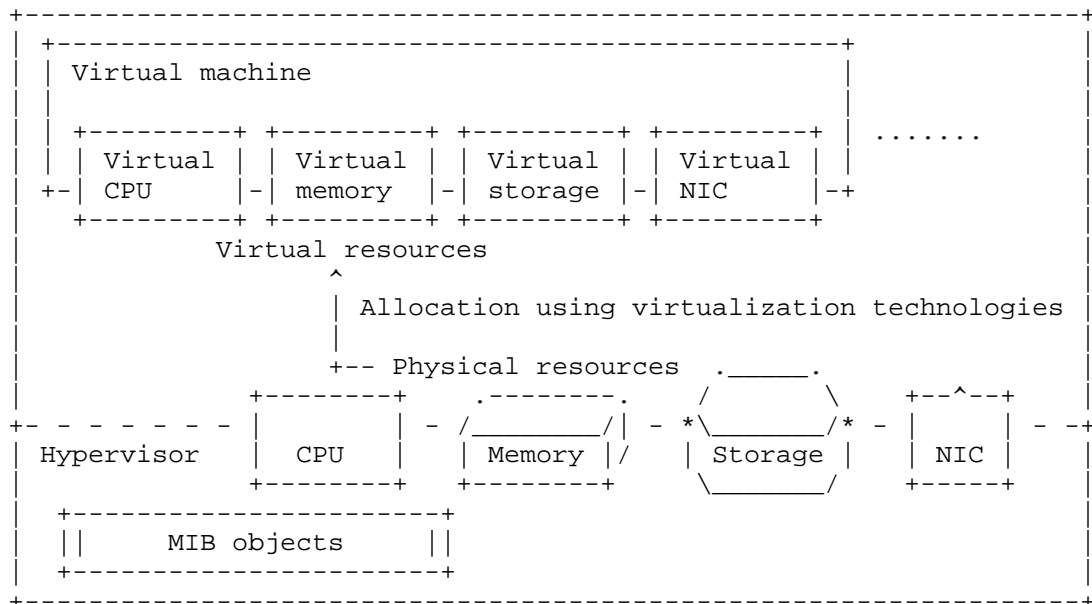
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

## 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. Overview and Objectives

This document defines a portion of MIB for the management of virtual machines controlled by a hypervisor. This MIB module consists of the managed objects related to system and software information of a hypervisor, the list of virtual machines controlled by the hypervisor, and information of virtual resources allocated to virtual machines by the hypervisor. This document specifies four specific types of virtual resources that are common to many hypervisor implementations; processors (CPUs), memory, network interfaces (NICs), and storage devices. These managed objects are independent of the families of hypervisors or operating systems running on virtual machines.



A hypervisor allocates virtual resources such as virtual CPUs, virtual memory, virtual storage devices, and virtual network interfaces to virtual machines from physical resources.

Figure 1: An example of a virtualization environment

On the common implementations of hypervisors, a hypervisor allocates virtual resources from physical resources; virtual CPUs, virtual memory, virtual storage devices, and virtual network interfaces to virtual machines as shown in Figure 1. Since the virtual resources allocated to virtual machines are managed by the hypervisor, the MIB objects are managed at the hypervisor. In case that the objects are accessed through the SNMP, an SNMP agent is launched at the hypervisor to provide access to the objects.

The objects are managed from the viewpoint of the operators of hypervisors, but not the operators of virtual machines; i.e., the objects do not take into account the actual resource utilization on each virtual machine but the resource allocation from the physical resources. For example, `vmNetworkIfIndex` indicates the virtual interface associated with an interface of a virtual machine at the hypervisor, and consequently, the 'in' and 'out' directions denote 'from a virtual machine to the hypervisor' and 'from the hypervisor to a virtual machine', respectively. Moreover, `vmStorageAllocatedSize` denotes the size allocated by the hypervisor, but not the size actually used by the operating system on the virtual

machine. This means that `vmStorageDefinedSize` and `vmStorageAllocatedSize` do not take different values when the `vmStorageSourceType` is 'block' or 'raw'.

The objectives of this document are the followings: 1) This document defines the MIB objects common to many hypervisors for the management of virtual machines controlled by a hypervisor. 2) This document clarifies the relationship with other MIB modules for managing host computers and network devices.

#### 4. Structure of the VM-MIB Module

The MIB module is organized into a group of scalars and tables. The scalars below 'vmHypervisor' provide basic information about the hypervisor. The 'vmTable' lists the virtual machines (guests) that are known to the hypervisor. The 'vmCpuTable' provides the mapping table of virtual CPUs to virtual machines, including CPU time used by each virtual CPU. The 'vmCpuAffinityTable' provides the affinity of each virtual CPU to a physical CPU. The 'vmStorageTable' provides the list of virtual storage devices and their mapping to virtual machines. In case that an entry in the 'vmStorageTable' has a corresponding parent physical storage device managed in 'vmStorageTable' of HOST-RESOURCES-MIB [RFC2790], the entry contains a pointer 'vmStorageParent' to the physical storage device. The 'vmNetworkTable' provides the list of virtual network interfaces and their mapping to virtual machines. Each entry in the 'vmNetworkTable' also provides a pointer 'vmNetworkIfIndex' to the corresponding entry in the 'ifTable' of IF-MIB [RFC2863]. In case that an entry in the 'vmNetworkTable' has a corresponding parent physical network interface managed in the 'ifTable' of IF-MIB, the entry contains a pointer 'vmNetworkParent' to the physical network interface.

Notation:

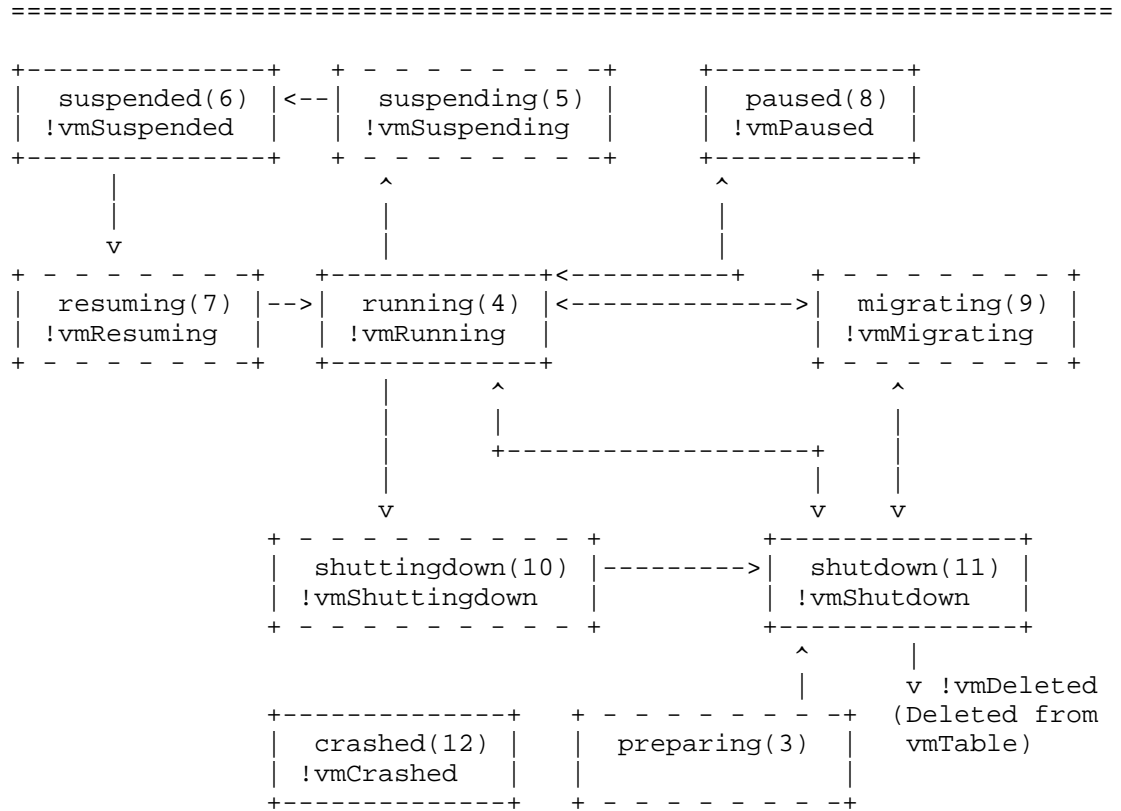
```

+-----+
| vmOperState | : Finite state; the first line presents the
+-----+      : 'vmOperState', and the second line presents a
                  : notification generated if applicable.

+ - - - - - +
| vmOperState | : Transient state; first line presents the
+ - - - - - +   : 'vmOperState', and the second line presents a
                  : notification generated if applicable.

!               : Notification; a text followed by the symbol "!"
                  : denotes a notification generated.

```



The overview of the state transition of a virtual machine

Figure 2: State transition of a virtual machine



The 'vmAdminState' and 'vmOperState' textual conventions define an administrative state and an operational state model for virtual machines. Events causing transitions between major operational states will cause the generation of notifications. Per virtual machine (per-VM) notifications (vmRunning, vmShutdown, vmPaused, vmSuspended, vmCrashed, vmDeleted) are generated if vmPerVMNotificationsEnabled is true(1). Bulk notifications (vmBulkRunning, vmBulkShutdown, vmBulkPaused, vmBulkSuspended, vmBulkCrashed, vmBulkDeleted) are generated if vmBulkNotificationsEnabled is true(1). The overview of the transition of 'vmOperState' by the write access to 'vmAdminState' and the notifications generated by the operational state changes are illustrated in Figure 2. The detailed state transition is summarized in Appendix A. Note that the notifications shown in this figure are per-VM notifications. In the case of Bulk notifications, the prefix 'vm' is replaced with 'vmBulk'.

The bulk notification mechanism is designed to reduce the number of notifications that are trapped by an SNMP manager. This is because the number of virtual machines managed by a bunch of hypervisors in a datacenter possibly becomes several thousands or more, and consequently, many notifications could be trapped if these virtual machines frequently change their administrative state. The per-VM notifications carry more detailed information, but the scalability is a problem. The notification filtering mechanism described in section 6 of RFC 3413 [RFC3413] is used by the management applications to control the notifications.

## 5. Relationship to Other MIB Modules

The HOST-RESOURCES-MIB [RFC2790] defines the MIB objects for managing host systems. On systems implementing the HOST-RESOURCES-MIB, the objects of HOST-RESOURCES-MIB indicate resources of a hypervisor. Some objects of HOST-RESOURCES-MIB are used to indicate physical resources through indexes. On systems implementing HOST-RESOURCES-MIB, the 'vmCpuPhysIndex' points to the processor's 'hrDeviceIndex' in the 'hrProcessorTable'. The 'vmStorageParent' also points to the storage device's 'hrStorageIndex' in the 'hrStorageTable'.

The IF-MIB [RFC2863] defines the MIB objects for managing network interfaces. Both physical and virtual network interfaces are required to be contained in the 'ifTable' of IF-MIB. The virtual network interfaces in the 'ifTable' of IF-MIB are pointed from the 'vmNetworkTable' defined in this document through a pointer 'vmNetworkIfIndex'. In case that an entry in the 'vmNetworkTable' has a corresponding parent physical network interface managed in the 'ifTable' of IF-MIB, the entry contains a pointer 'vmNetworkParent' to the physical network interface.

The objects related to virtual switches are not included in the MIB module defined in this document though virtual switches MAY be placed on a hypervisor. This is because the virtual network interfaces are the lowest abstraction of network resources allocated to a virtual machine. Instead of including the objects related to virtual switches, for example, IEEE8021-BRIDGE-MIB [IEEE8021-BRIDGE-MIB] and IEEE8021-Q-BRIDGE-MIB [IEEE8021-Q-BRIDGE-MIB] could be used.

The other objects related to virtual machines such as management IP addresses of a virtual machine are not included in this MIB module because this MIB module defines the objects common to general hypervisors but they are specific to some hypervisors. They may be included in the entLogicalTable of ENTITY-MIB [RFC6933].

## 6. Definitions

### 6.1. VM-MIB

```
VM-MIB DEFINITIONS ::= BEGIN
```

```
IMPORTS
```

```
    MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE, TimeTicks,  
    Counter64, Integer32, mib-2
```

```
    FROM SNMPv2-SMI
```

```
    OBJECT-GROUP, MODULE-COMPLIANCE, NOTIFICATION-GROUP
```

```
    FROM SNMPv2-CONF
```

```
    TEXTUAL-CONVENTION, PhysAddress, TruthValue
```

```
    FROM SNMPv2-TC
```

```
    SnmpAdminString
```

```
    FROM SNMP-FRAMEWORK-MIB
```

```
    UUIDorZero
```

```
    FROM UUID-TC-MIB
```

```
    InterfaceIndexOrZero
```

```
    FROM IF-MIB
```

```
    IANAStorageMediaType
```

```
    FROM IANA-STORAGE-MEDIA-TYPE-MIB;
```

```
vmMIB MODULE-IDENTITY
```

```
    LAST-UPDATED "201508050000Z"          -- 5 August 2015
```

```
    ORGANIZATION "IETF Operations and Management Area Working Group"
```

```
    CONTACT-INFO
```

```
    "
```

```
    WG E-mail: opsawg@ietf.org
```

```
    Mailing list subscription info:
```

```
    https://www.ietf.org/mailman/listinfo/opsawg
```

```
    Hirochika Asai
```

```
    The University of Tokyo
```

7-3-1 Hongo  
Bunkyo-ku, Tokyo 113-8656  
JP  
Phone: +81 3 5841 6748  
Email: panda@hongo.wide.ad.jp

Michael MacFaden  
VMware Inc.  
Email: mrm@vmware.com

Juergen Schoenwaelder  
Jacobs University  
Campus Ring 1  
Bremen 28759  
Germany  
Email: j.schoenwaelder@jacobs-university.de

Keiichi Shima  
IIJ Innovation Institute Inc.  
3-13 Kanda-Nishikicho  
Chiyoda-ku, Tokyo 101-0054  
JP  
Email: keiichi@iijlab.net

Tina Tsou  
Huawei Technologies (USA)  
2330 Central Expressway  
Santa Clara CA 95050  
USA  
Email: tina.tsou.zouting@huawei.com  
"

#### DESCRIPTION

"This MIB module is for use in managing a hypervisor and virtual machines controlled by the hypervisor.

Copyright (c) 2015 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>)."

REVISION "201508050000Z"  
DESCRIPTION

-- 5 August 2015

```

        "The initial version of this MIB, published as
        RFCXXXX."
 ::= { mib-2 yyy }

-- RFC Ed.: replace XXXX with RFC number and remove this note
-- RFC Ed.: replace yyy with actual number and remove this note

vmNotifications OBJECT IDENTIFIER ::= { vmMIB 0 }
vmObjects        OBJECT IDENTIFIER ::= { vmMIB 1 }
vmConformance    OBJECT IDENTIFIER ::= { vmMIB 2 }

-- Textual conversion definitions
--
VirtualMachineIndex ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS        current
    DESCRIPTION
        "A unique value, greater than zero, identifying a
        virtual machine. The value for each virtual machine
        MUST remain constant at least from one re-initialization
        of the hypervisor to the next re-initialization."
    SYNTAX        Integer32 (1..2147483647)

VirtualMachineIndexOrZero ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS        current
    DESCRIPTION
        "This textual convention is an extension of the
        VirtualMachineIndex convention. This extension permits
        the additional value of zero. The meaning of the value
        zero is object-specific and MUST therefore be defined as
        part of the description of any object which uses this
        syntax. Examples of the usage of zero might include
        situations where a virtual machine is unknown, or when
        none or all virtual machines need to be referenced."
    SYNTAX        Integer32 (0..2147483647)

VirtualMachineAdminState ::= TEXTUAL-CONVENTION
    STATUS        current
    DESCRIPTION
        "The administrative state of a virtual machine:

        running(1)    The administrative state of the virtual
                        machine indicating the virtual machine
                        is currently online or should be brought
                        online.
```

suspended(2) The administrative state of the virtual machine where its memory and CPU execution state has been saved to persistent store and will be restored at next running(1).

paused(3) The administrative state indicating the virtual machine is resident in memory but is no longer scheduled to execute by the hypervisor.

shutdown(4) The administrative state of the virtual machine indicating the virtual machine is currently offline or should be taken shutting down."

```
SYNTAX      INTEGER {  
              running(1),  
              suspended(2),  
              paused(3),  
              shutdown(4)  
            }
```

VirtualMachineOperState ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"The operational state of a virtual machine:

unknown(1) The operational state of the virtual machine is unknown, e.g., because the implementation failed to obtain the state from the hypervisor.

other(2) The operational state of the virtual machine indicating that an operational state is obtained from the hypervisor but it is not a state defined in this MIB module.

preparing(3) The operational state of the virtual machine indicating the virtual machine is currently in the process of preparation, e.g., allocating and initializing virtual storage after creating (defining) virtual machine.

running(4) The operational state of the virtual machine indicating the virtual machine is currently executed but it is not in the process of preparing(3), suspending(5),

resuming(7), migrating(9), and  
shuttingdown(10).

suspending(5) The operational state of the virtual machine indicating the virtual machine is currently in the process of suspending to save its memory and CPU execution state to persistent store. This is a transient state from running(4) to suspended(6).

suspended(6) The operational state of the virtual machine indicating the virtual machine is currently suspended, which means the memory and CPU execution state of the virtual machine are saved to persistent store. During this state, the virtual machine is not scheduled to execute by the hypervisor.

resuming(7) The operational state of the virtual machine indicating the virtual machine is currently in the process of resuming to restore its memory and CPU execution state from persistent store. This is a transient state from suspended(6) to running(4).

paused(8) The operational state of the virtual machine indicating the virtual machine is resident in memory but no longer scheduled to execute by the hypervisor.

migrating(9) The operational state of the virtual machine indicating the virtual machine is currently in the process of migration from/to another hypervisor.

shuttingdown(10) The operational state of the virtual machine indicating the virtual machine is currently in the process of shutting down. This is a transient state from running(4) to shutdown(11).

shutdown(11) The operational state of the virtual machine indicating the virtual machine is down, and CPU execution is no longer

scheduled by the hypervisor and its memory is not resident in the hypervisor.

crashed(12) The operational state of the virtual machine indicating the virtual machine has crashed."

```
SYNTAX      INTEGER {
                unknown(1),
                other(2),
                preparing(3),
                running(4),
                suspending(5),
                suspended(6),
                resuming(7),
                paused(8),
                migrating(9),
                shuttingdown(10),
                shutdown(11),
                crashed(12)
            }
```

VirtualMachineAutoStart ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"The autostart configuration of a virtual machine:

unknown(1) The autostart configuration is unknown, e.g., because the implementation failed to obtain the autostart configuration from the hypervisor.

enabled(2) The autostart configuration of the virtual machine is enabled. The virtual machine should be automatically brought online at the next re-initialization of the hypervisor.

disabled(3) The autostart configuration of the virtual machine is disabled. The virtual machine should not be automatically brought online at the next re-initialization of the hypervisor."

```
SYNTAX      INTEGER {
                unknown(1),
                enabled(2),
                disabled(3)
            }
```

```
VirtualMachinePersistent ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "This value indicates whether a virtual machine has a
        persistent configuration which means the virtual machine
        will still exist after shutting down:

        unknown(1)    The persistent configuration is unknown,
                       e.g., because the implementation failed
                       to obtain the persistent configuration
                       from the hypervisor. (read-only)

        persistent(2) The virtual machine is persistent, i.e.,
                       the virtual machine will exist after its
                       shutting down.

        transient(3)  The virtual machine is transient, i.e.,
                       the virtual machine will not exist after
                       its shutting down."
    SYNTAX      INTEGER {
                    unknown(1),
                    persistent(2),
                    transient(3)
                }
```

```
VirtualMachineCpuIndex ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS      current
    DESCRIPTION
        "A unique value for each virtual machine, greater than
        zero, identifying a virtual CPU assigned to a virtual
        machine. The value for each virtual CPU MUST remain
        constant at least from one re-initialization of the
        hypervisor to the next re-initialization."
    SYNTAX      Integer32 (1..2147483647)
```

```
VirtualMachineStorageIndex ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS      current
    DESCRIPTION
        "A unique value for each virtual machine, greater than
        zero, identifying a virtual storage device allocated to
        a virtual machine. The value for each virtual storage
        device MUST remain constant at least from one
        re-initialization of the hypervisor to the next
        re-initialization."
    SYNTAX      Integer32 (1..2147483647)
```



```
VirtualMachineStorageSourceType ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "The source type of a virtual storage device:

        unknown(1)    The source type is unknown, e.g., because
                       the implementation failed to obtain the
                       media type from the hypervisor.

        other(2)      The source type is other than those
                       defined in this conversion.

        block(3)      The source type is a block device.

        raw(4)        The source type is a raw-formatted file.

        sparse(5)     The source type is a sparse file.

        network(6)    The source type is a network device."
    SYNTAX      INTEGER {
        unknown(1),
        other(2),
        block(3),
        raw(4),
        sparse(5),
        network(6)
    }

VirtualMachineStorageAccess ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "The access permission of a virtual storage:

        unknown(1)    The access permission of the virtual
                       storage is unknown.

        readwrite(2)   The virtual storage is a read-write
                       device.

        readonly(3)    The virtual storage is a read-only
                       device."
    SYNTAX      INTEGER {
        unknown(1),
        readwrite(2),
        readonly(3)
    }

VirtualMachineNetworkIndex ::= TEXTUAL-CONVENTION
```

```

DISPLAY-HINT "d"
STATUS      current
DESCRIPTION
    "A unique value for each virtual machine, greater than
    zero, identifying a virtual network interface allocated
    to the virtual machine.  The value for each virtual
    network interface MUST remain constant at least from one
    re-initialization of the hypervisor to the next
    re-initialization."
SYNTAX      Integer32 (1..2147483647)

VirtualMachineList ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "1x"
    STATUS      current
    DESCRIPTION
        "Each octet within this value specifies a set of eight
        virtual machine vmIndex values, with the first octet
        specifying virtual machine 1 through 8, the second octet
        specifying virtual machine 9 through 16, etc.  Within
        each octet, the most significant bit represents the
        lowest numbered vmIndex, and the least significant bit
        represents the highest numbered vmIndex.  Thus, each
        virtual machine of the host is represented by a single
        bit within the value of this object.  If that bit has
        a value of '1', then that virtual machine is included
        in the set of virtual machines; the virtual machine is
        not included if its bit has a value of '0'."
    SYNTAX      OCTET STRING

-- The hypervisor group
--
-- A collection of objects common to all hypervisors.
--
vmHypervisor      OBJECT IDENTIFIER ::= { vmObjects 1 }

vmHvSoftware OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE (0..255))
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "A textual description of the hypervisor software.  This
        value SHOULD NOT include its version as it SHOULD be
        included in 'vmHvVersion'."
    ::= { vmHypervisor 1 }

vmHvVersion OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE (0..255))
    MAX-ACCESS   read-only

```

```
STATUS          current
DESCRIPTION
    "A textual description of the version of the hypervisor
    software."
::= { vmHypervisor 2 }

vmHvObjectID OBJECT-TYPE
SYNTAX          OBJECT IDENTIFIER
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "The vendor's authoritative identification of the
    hypervisor software contained in the entity. This value
    is allocated within the SMI enterprises
    subtree (1.3.6.1.4.1). Note that this is different from
    sysObjectID in the SNMPv2-MIB [RFC3418] because
    sysObjectID is not the identification of the hypervisor
    software but the device, firmware, or management
    operating system."
::= { vmHypervisor 3 }

vmHvUpTime OBJECT-TYPE
SYNTAX          TimeTicks
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "The time (in centi-seconds) since the hypervisor was
    last re-initialized. Note that this is different from
    sysUpTime in the SNMPv2-MIB [RFC3418] and hrSystemUptime
    in the HOST-RESOURCES-MIB [RFC2790] because sysUpTime is
    the uptime of the network management portion of the
    system, and hrSystemUptime is the uptime of the
    management operating system but not the hypervisor
    software."
::= { vmHypervisor 4 }

-- The virtual machine information
--
-- A collection of objects common to all virtual machines.
--
vmNumber OBJECT-TYPE
SYNTAX          Integer32 (0..2147483647)
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "The number of virtual machines (regardless of their
    current state) present on this hypervisor."
```

```

 ::= { vmObjects 2 }

vmTableLastChange OBJECT-TYPE
    SYNTAX      TimeTicks
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The value of vmHvUpTime at the time of the last creation
        or deletion of an entry in the vmTable."
    ::= { vmObjects 3 }

vmTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF VmEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "A list of virtual machine entries. The number of
        entries is given by the value of vmNumber."
    ::= { vmObjects 4 }

vmEntry OBJECT-TYPE
    SYNTAX      VmEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "An entry containing management information applicable
        to a particular virtual machine."
    INDEX      { vmIndex }
    ::= { vmTable 1 }

VmEntry ::=
    SEQUENCE {
        vmIndex          VirtualMachineIndex,
        vmName            SnmpAdminString,
        vmUUID            UUIDorZero,
        vmOSType          SnmpAdminString,
        vmAdminState      VirtualMachineAdminState,
        vmOperState       VirtualMachineOperState,
        vmAutoStart       VirtualMachineAutoStart,
        vmPersistent       VirtualMachinePersistent,
        vmCurCpuNumber    Integer32,
        vmMinCpuNumber     Integer32,
        vmMaxCpuNumber     Integer32,
        vmMemUnit          Integer32,
        vmCurMem          Integer32,
        vmMinMem           Integer32,
        vmMaxMem           Integer32,
        vmUpTime           TimeTicks,

```

```
        vmCpuTime          Counter64
    }

vmIndex OBJECT-TYPE
    SYNTAX      VirtualMachineIndex
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A unique value, greater than zero, identifying the
        virtual machine. The value assigned to a given virtual
        machine may not persist across re-initialization of the
        hypervisor. A command generator MUST use the vmUUID to
        identify a given virtual machine of interest."
    ::= { vmEntry 1 }

vmName OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE (0..255))
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "A textual name of the virtual machine."
    ::= { vmEntry 2 }

vmUUID OBJECT-TYPE
    SYNTAX      UUIDorZero
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The virtual machine's 128-bit UUID or the zero-length
        string when a UUID is not available. The UUID if set
        MUST uniquely identify a virtual machine from all other
        virtual machines in an administrative domain. A
        zero-length octet string is returned if no UUID
        information is known."
    ::= { vmEntry 3 }

vmOSType OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE (0..255))
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "A textual description containing operating system
        information installed on the virtual machine. This
        value corresponds to the operating system the hypervisor
        assumes to be running when the virtual machine is
        started. This may differ from the actual operating
        system in case the virtual machine boots into a
        different operating system."
```

```
 ::= { vmEntry 4 }

vmAdminState OBJECT-TYPE
    SYNTAX      VirtualMachineAdminState
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The administrative state of the virtual machine."
    ::= { vmEntry 5 }

vmOperState OBJECT-TYPE
    SYNTAX      VirtualMachineOperState
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The operational state of the virtual machine."
    ::= { vmEntry 6 }

vmAutoStart OBJECT-TYPE
    SYNTAX      VirtualMachineAutoStart
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The autostart configuration of the virtual machine.  If
        this value is enable(2), the virtual machine
        automatically starts at the next initialization of the
        hypervisor."
    ::= { vmEntry 7 }

vmPersistent OBJECT-TYPE
    SYNTAX      VirtualMachinePersistent
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "This value indicates whether the virtual machine has a
        persistent configuration which means the virtual machine
        will still exist after its shutdown."
    ::= { vmEntry 8 }

vmCurCpuNumber OBJECT-TYPE
    SYNTAX      Integer32 (0..2147483647)
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of virtual CPUs currently assigned to the
        virtual machine."
    ::= { vmEntry 9 }
```

vmMinCpuNumber OBJECT-TYPE  
SYNTAX Integer32 (-1|0..2147483647)  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
"The minimum number of virtual CPUs that are assigned to the virtual machine when it is in a power-on state. The value -1 indicates that there is no hard boundary for the minimum number of virtual CPUs."  
 ::= { vmEntry 10 }

vmMaxCpuNumber OBJECT-TYPE  
SYNTAX Integer32 (-1|0..2147483647)  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
"The maximum number of virtual CPUs that are assigned to the virtual machine when it is in a power-on state. The value -1 indicates that there is no limit."  
 ::= { vmEntry 11 }

vmMemUnit OBJECT-TYPE  
SYNTAX Integer32 (1..2147483647)  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
"The multiplication unit in byte for vmCurMem, vmMinMem, and vmMaxMem. For example, when this value is 1024, the memory size unit for vmCurMem, vmMinMem, and vmMaxMem is KiB."  
 ::= { vmEntry 12 }

vmCurMem OBJECT-TYPE  
SYNTAX Integer32 (0..2147483647)  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
"The current memory size currently allocated to the virtual memory module in the unit designated by vmMemUnit."  
 ::= { vmEntry 13 }

vmMinMem OBJECT-TYPE  
SYNTAX Integer32 (-1|0..2147483647)  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
"The minimum memory size defined to the virtual machine

in the unit designated by vmMemUnit. The value -1 indicates that there is no hard boundary for the minimum memory size."  
 ::= { vmEntry 14 }

## vmMaxMem OBJECT-TYPE

SYNTAX Integer32 (-1|0..2147483647)  
 MAX-ACCESS read-only  
 STATUS current  
 DESCRIPTION  
 "The maximum memory size defined to the virtual machine in the unit designated by vmMemUnit. The value -1 indicates that there is no limit."  
 ::= { vmEntry 15 }

## vmUpTime OBJECT-TYPE

SYNTAX TimeTicks  
 MAX-ACCESS read-only  
 STATUS current  
 DESCRIPTION  
 "The time (in centi-seconds) since the administrative state of the virtual machine was last changed from shutdown(4) to running(1)."  
 ::= { vmEntry 16 }

## vmCpuTime OBJECT-TYPE

SYNTAX Counter64  
 UNITS "microsecond"  
 MAX-ACCESS read-only  
 STATUS current  
 DESCRIPTION  
 "The total CPU time used in microsecond. If the number of virtual CPUs is larger than 1, vmCpuTime may exceed real time.  
  
 Discontinuities in the value of this counter can occur at re-initialization of the hypervisor, and administrative state (vmAdminState) changes of the virtual machine."  
 ::= { vmEntry 17 }

-- The virtual CPU on each virtual machines

## vmCpuTable OBJECT-TYPE

SYNTAX SEQUENCE OF VmCpuEntry  
 MAX-ACCESS not-accessible  
 STATUS current  
 DESCRIPTION



```

        "The table of virtual CPUs provided by the hypervisor."
 ::= { vmObjects 5 }

vmCpuEntry OBJECT-TYPE
    SYNTAX      VmCpuEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "An entry for one virtual processor assigned to a
        virtual machine."
    INDEX { vmIndex, vmCpuIndex }
    ::= { vmCpuTable 1 }

VmCpuEntry ::=
    SEQUENCE {
        vmCpuIndex          VirtualMachineCpuIndex,
        vmCpuCoreTime       Counter64
    }

vmCpuIndex OBJECT-TYPE
    SYNTAX      VirtualMachineCpuIndex
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "A unique value identifying a virtual CPU assigned to
        the virtual machine."
    ::= { vmCpuEntry 1 }

vmCpuCoreTime OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "microsecond"
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "The total CPU time used by this virtual CPU in
        microsecond.

        Discontinuities in the value of this counter can occur
        at re-initialization of the hypervisor, and
        administrative state (vmAdminState) changes of the
        virtual machine."
    ::= { vmCpuEntry 2 }

-- The virtual CPU affinity on each virtual machines
vmCpuAffinityTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF VmCpuAffinityEntry
    MAX-ACCESS   not-accessible
    STATUS      current

```

```

DESCRIPTION
    "A list of CPU affinity entries of a virtual CPU."
 ::= { vmObjects 6 }

vmCpuAffinityEntry OBJECT-TYPE
    SYNTAX      VmCpuAffinityEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "An entry containing CPU affinity associated with a
         particular virtual machine."
    INDEX       { vmIndex, vmCpuIndex, vmCpuPhysIndex }
    ::= { vmCpuAffinityTable 1 }

VmCpuAffinityEntry ::=
    SEQUENCE {
        vmCpuPhysIndex      Integer32,
        vmCpuAffinity        Integer32
    }

vmCpuPhysIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..2147483647)
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "A value identifying a physical CPU on the hypervisor.
         On systems implementing the HOST-RESOURCES-MIB, the
         value MUST be the same value that is used as the index
         in the hrProcessorTable (hrDeviceIndex)."
    ::= { vmCpuAffinityEntry 2 }

vmCpuAffinity OBJECT-TYPE
    SYNTAX      INTEGER {
                    unknown(0),  -- unknown
                    enable(1),  -- enabled
                    disable(2)  -- disabled
                }
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The CPU affinity of this virtual CPU to the physical
         CPU represented by 'vmCpuPhysIndex'."
    ::= { vmCpuAffinityEntry 3 }

-- The virtual storage devices on each virtual machine.  This
-- document defines some overlapped objects with hrStorage in
-- HOST-RESOURCES-MIB [RFC2790], because virtual resources are

```

```

-- allocated from the hypervisor's resources, which is the 'host
-- resources'
vmStorageTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF VmStorageEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "The conceptual table of virtual storage devices
        attached to the virtual machine."
    ::= { vmObjects 7 }

vmStorageEntry OBJECT-TYPE
    SYNTAX      VmStorageEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "An entry for one virtual storage device attached to the
        virtual machine."
    INDEX { vmStorageVmIndex, vmStorageIndex }
    ::= { vmStorageTable 1 }

VmStorageEntry ::=
    SEQUENCE {
        vmStorageVmIndex      VirtualMachineIndexOrZero,
        vmStorageIndex        VirtualMachineStorageIndex,
        vmStorageParent        Integer32,
        vmStorageSourceType    VirtualMachineStorageSourceType,
        vmStorageSourceTypeString
                               SnmpAdminString,
        vmStorageResourceID    SnmpAdminString,
        vmStorageAccess        VirtualMachineStorageAccess,
        vmStorageMediaType     IANAStorageMediaType,
        vmStorageMediaTypeString
                               SnmpAdminString,
        vmStorageSizeUnit      Integer32,
        vmStorageDefinedSize    Integer32,
        vmStorageAllocatedSize  Integer32,
        vmStorageReadIOs        Counter64,
        vmStorageWriteIOs       Counter64,
        vmStorageReadOctets     Counter64,
        vmStorageWriteOctets    Counter64,
        vmStorageReadLatency    Counter64,
        vmStorageWriteLatency   Counter64
    }

vmStorageVmIndex OBJECT-TYPE
    SYNTAX      VirtualMachineIndexOrZero
    MAX-ACCESS   not-accessible

```

```
STATUS          current
DESCRIPTION
    "This value identifies the virtual machine (guest) this
    storage device has been allocated to.  The value zero
    indicates that the storage device is currently not
    allocated to any virtual machines."
 ::= { vmStorageEntry 1 }

vmStorageIndex OBJECT-TYPE
    SYNTAX      VirtualMachineStorageIndex
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "A unique value identifying a virtual storage device
        allocated to the virtual machine."
    ::= { vmStorageEntry 2 }

vmStorageParent OBJECT-TYPE
    SYNTAX      Integer32 (0..2147483647)
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "The value of hrStorageIndex which is the parent (i.e.,
        physical) device of this virtual device on systems
        implementing the HOST-RESOURCES-MIB.  The value zero
        denotes this virtual device is not any child represented
        in the hrStorageTable."
    ::= { vmStorageEntry 3 }

vmStorageSourceType OBJECT-TYPE
    SYNTAX      VirtualMachineStorageSourceType
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "The source type of the virtual storage device."
    ::= { vmStorageEntry 4 }

vmStorageSourceTypeString OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE (0..255))
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "A (detailed) textual string of the source type of the
        virtual storage device.  For example, this represents
        the specific format name of the sparse file."
    ::= { vmStorageEntry 5 }

vmStorageResourceID OBJECT-TYPE
```

```
SYNTAX      SnmpAdminString (SIZE (0..255))
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "A textual string that represents the resource
    identifier of the virtual storage.  For example, this
    contains the path to the disk image file that
    corresponds to the virtual storage."
::= { vmStorageEntry 6 }

vmStorageAccess OBJECT-TYPE
    SYNTAX      VirtualMachineStorageAccess
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The access permission of the virtual storage device."
    ::= { vmStorageEntry 7 }

vmStorageMediaType OBJECT-TYPE
    SYNTAX      IANAStorageMediaType
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The media type of the virtual storage device."
    ::= { vmStorageEntry 8 }

vmStorageMediaTypeString OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE (0..255))
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "A (detailed) textual string of the virtual storage
        media.  For example, this represents the specific driver
        name of the emulated media such as 'IDE' and 'SCSI'."
    ::= { vmStorageEntry 9 }

vmStorageSizeUnit OBJECT-TYPE
    SYNTAX      Integer32 (1..2147483647)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The multiplication unit in byte for
        vmStorageDefinedSize and vmStorageAllocatedSize.  For
        example, when this value is 1048576, the storage size
        unit for vmStorageDefinedSize and vmStorageAllocatedSize
        is MiB."
    ::= { vmStorageEntry 10 }
```

## vmStorageDefinedSize OBJECT-TYPE

SYNTAX Integer32 (-1|0..2147483647)

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The defined virtual storage size defined in the unit designated by vmStorageSizeUnit. If this information is not available, this value MUST be -1."

::= { vmStorageEntry 11 }

## vmStorageAllocatedSize OBJECT-TYPE

SYNTAX Integer32 (-1|0..2147483647)

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The storage size allocated to the virtual storage from a physical storage in the unit designated by vmStorageSizeUnit. When the virtual storage is block device or raw file, this value and vmStorageDefinedSize are supposed to equal. This value MUST NOT be different from vmStorageDefinedSize when vmStorageSourceType is 'block' or 'raw'. If this information is not available, this value MUST be -1."

::= { vmStorageEntry 12 }

## vmStorageReadIOs OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The number of read I/O requests.

Discontinuities in the value of this counter can occur at re-initialization of the hypervisor, and administrative state (vmAdminState) changes of the virtual machine."

::= { vmStorageEntry 13 }

## vmStorageWriteIOs OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The number of write I/O requests.

Discontinuities in the value of this counter can occur at re-initialization of the hypervisor, and administrative state (vmAdminState) changes of the

```
        virtual machine."
 ::= { vmStorageEntry 14 }

vmStorageReadOctets OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The total number of bytes read from this device.

        Discontinuities in the value of this counter can occur
        at re-initialization of the hypervisor, and
        administrative state (vmAdminState) changes of the
        virtual machine."
 ::= { vmStorageEntry 15 }

vmStorageWriteOctets OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The total number of bytes written to this device.

        Discontinuities in the value of this counter can occur
        at re-initialization of the hypervisor, and
        administrative state (vmAdminState) changes of the
        virtual machine."
 ::= { vmStorageEntry 16 }

vmStorageReadLatency OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The total number of microseconds read requests have
        been queued for this device.
        This would typically be implemented by storing the high
        precision system time stamp of when the request is
        received from the virtual machine with the request, the
        difference between this initial timestamp and the time
        at which the requested operation has completed SHOULD be
        converted to microseconds and accumulated.
        Discontinuities in the value of this counter can occur at
        re-initialization of the hypervisor, and administrative
        state (vmAdminState) changes of the virtual machine."
 ::= { vmStorageEntry 17 }

vmStorageWriteLatency OBJECT-TYPE
```

```

SYNTAX          Counter64
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "The total number of microseconds write requests have
    been queued for this device.
    This would typically be implemented by storing the high
    precision system time stamp of when the request is
    received from the virtual machine with the request, the
    difference between this initial timestamp and the time
    at which the requested operation has completed SHOULD be
    converted to microseconds and accumulated.
    Discontinuities in the value of this counter can occur
    at re-initialization of the hypervisor, and
    administrative state (vmAdminState) changes of the
    virtual machine."
 ::= { vmStorageEntry 18 }

```

```
-- The virtual network interfaces on each virtual machine.
```

```

vmNetworkTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF VmNetworkEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "The conceptual table of virtual network interfaces
        attached to the virtual machine."
    ::= { vmObjects 8 }

```

```

vmNetworkEntry OBJECT-TYPE
    SYNTAX          VmNetworkEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "An entry for one virtual network interfaces attached to
        the virtual machine."
    INDEX { vmIndex, vmNetworkIndex }
    ::= { vmNetworkTable 1 }

```

```

VmNetworkEntry ::=
    SEQUENCE {
        vmNetworkIndex          VirtualMachineNetworkIndex,
        vmNetworkIfIndex        InterfaceIndexOrZero,
        vmNetworkParent          InterfaceIndexOrZero,
        vmNetworkModel           SnmpAdminString,
        vmNetworkPhysAddress     PhysAddress
    }

```



```
vmNetworkIndex OBJECT-TYPE
    SYNTAX      VirtualMachineNetworkIndex
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "A unique value identifying a virtual network interface
        allocated to the virtual machine."
    ::= { vmNetworkEntry 1 }

vmNetworkIfIndex OBJECT-TYPE
    SYNTAX      InterfaceIndexOrZero
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "The value of ifIndex which corresponds to this virtual
        network interface.  If this device is not represented in
        the ifTable, then this value MUST be zero."
    ::= { vmNetworkEntry 2 }

vmNetworkParent OBJECT-TYPE
    SYNTAX      InterfaceIndexOrZero
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "The value of ifIndex which corresponds to the parent
        (i.e., physical) device of this virtual device on.  The
        value zero denotes this virtual device is not any child
        represented in the ifTable."
    ::= { vmNetworkEntry 3 }

vmNetworkModel OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE (0..255))
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "A textual string containing the (emulated) model of
        virtual network interface.  For example, this value is
        'virtio' when the emulation driver model is virtio."
    ::= { vmNetworkEntry 4 }

vmNetworkPhysAddress OBJECT-TYPE
    SYNTAX      PhysAddress
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "The MAC address of the virtual network interface."
    ::= { vmNetworkEntry 5 }
```

-- Notification definitions:

vmPerVMNotificationsEnabled OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Indicates if notification generator will send notifications per virtual machine. Changes to this object MUST NOT persist across re-initialization of the management system, e.g., SNMP agent."

::= { vmObjects 9 }

vmBulkNotificationsEnabled OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Indicates if notification generator will send notifications per set of virtual machines. Changes to this object MUST NOT persist across re-initialization of the management system, e.g., SNMP agent."

::= { vmObjects 10 }

vmAffectedVMs OBJECT-TYPE

SYNTAX VirtualMachineList

MAX-ACCESS accessible-for-notify

STATUS current

DESCRIPTION

"A complete list of virtual machines whose state has changed. This object is the only object sent with bulk notifications."

::= { vmObjects 11 }

vmRunning NOTIFICATION-TYPE

OBJECTS {  
    vmName,  
    vmUUID,  
    vmOperState  
}

STATUS current

DESCRIPTION

"This notification is generated when the operational state of a virtual machine has been changed to running(4) from some other state. The other state is indicated by the included value of vmOperState."

::= { vmNotifications 1 }

```
vmShuttingdown NOTIFICATION-TYPE
  OBJECTS      {
                vmName,
                vmUUID,
                vmOperState
              }
  STATUS       current
  DESCRIPTION   "This notification is generated when the operational
                state of a virtual machine has been changed to
                shuttingdown(10) from some other state. The other state
                is indicated by the included value of vmOperState."
  ::= { vmNotifications 2 }

vmShutdown NOTIFICATION-TYPE
  OBJECTS      {
                vmName,
                vmUUID,
                vmOperState
              }
  STATUS       current
  DESCRIPTION   "This notification is generated when the operational
                state of a virtual machine has been changed to
                shutdown(11) from some other state. The other state is
                indicated by the included value of vmOperState."
  ::= { vmNotifications 3 }

vmPaused NOTIFICATION-TYPE
  OBJECTS      {
                vmName,
                vmUUID,
                vmOperState
              }
  STATUS       current
  DESCRIPTION   "This notification is generated when the operational
                state of a virtual machine has been changed to
                paused(8) from some other state. The other state is
                indicated by the included value of vmOperState."
  ::= { vmNotifications 4 }

vmSuspending NOTIFICATION-TYPE
  OBJECTS      {
                vmName,
                vmUUID,
                vmOperState
              }
```

```
STATUS          current
DESCRIPTION
    "This notification is generated when the operational
    state of a virtual machine has been changed to
    suspending(5) from some other state.  The other state is
    indicated by the included value of vmOperState."
 ::= { vmNotifications 5 }

vmSuspended NOTIFICATION-TYPE
OBJECTS      {
    vmName,
    vmUUID,
    vmOperState
}
STATUS      current
DESCRIPTION
    "This notification is generated when the operational
    state of a virtual machine has been changed to
    suspended(6) from some other state.  The other state is
    indicated by the included value of vmOperState."
 ::= { vmNotifications 6 }

vmResuming NOTIFICATION-TYPE
OBJECTS      {
    vmName,
    vmUUID,
    vmOperState
}
STATUS      current
DESCRIPTION
    "This notification is generated when the operational
    state of a virtual machine has been changed to
    resuming(7) from some other state.  The other state is
    indicated by the included value of vmOperState."
 ::= { vmNotifications 7 }

vmMigrating NOTIFICATION-TYPE
OBJECTS      {
    vmName,
    vmUUID,
    vmOperState
}
STATUS      current
DESCRIPTION
    "This notification is generated when the operational
    state of a virtual machine has been changed to
    migrating(9) from some other state.  The other state is
    indicated by the included value of vmOperState."
```

```
 ::= { vmNotifications 8 }

vmCrashed NOTIFICATION-TYPE
  OBJECTS      {
    vmName,
    vmUUID,
    vmOperState
  }
  STATUS      current
  DESCRIPTION
    "This notification is generated when a virtual machine
    has been crashed. The previous state of the virtual
    machine is indicated by the included value of
    vmOperState."
  ::= { vmNotifications 9 }

vmDeleted NOTIFICATION-TYPE
  OBJECTS      {
    vmName,
    vmUUID,
    vmOperState,
    vmPersistent
  }
  STATUS      current
  DESCRIPTION
    "This notification is generated when a virtual machine
    has been deleted. The prior state of the virtual
    machine is indicated by the included value of
    vmOperState."
  ::= { vmNotifications 10 }

vmBulkRunning NOTIFICATION-TYPE
  OBJECTS      {
    vmAffectedVMs
  }
  STATUS      current
  DESCRIPTION
    "This notification is generated when the operational
    state of one or more virtual machine has been changed to
    running(4) from all prior states except for
    running(4). Management stations are encouraged to
    subsequently poll the subset of virtual machines of
    interest for vmOperState."
  ::= { vmNotifications 11 }

vmBulkShuttingdown NOTIFICATION-TYPE
  OBJECTS      {
    vmAffectedVMs
```

```

    }
    STATUS      current
    DESCRIPTION
        "This notification is generated when the operational
        state of one or more virtual machine has been changed to
        shuttingdown(10) from a state other than
        shuttingdown(10).  Management stations are encouraged to
        subsequently poll the subset of virtual machines of
        interest for vmOperState."
    ::= { vmNotifications 12 }

vmBulkShutdown NOTIFICATION-TYPE
    OBJECTS      {
        vmAffectedVMs
    }
    STATUS      current
    DESCRIPTION
        "This notification is generated when the operational
        state of one or more virtual machine has been changed to
        shutdown(11) from a state other than shutdown(11).
        Management stations are encouraged to subsequently poll
        the subset of virtual machines of interest for
        vmOperState."
    ::= { vmNotifications 13 }

vmBulkPaused NOTIFICATION-TYPE
    OBJECTS      {
        vmAffectedVMs
    }
    STATUS      current
    DESCRIPTION
        "This notification is generated when the operational
        state of one or more virtual machines have been changed
        to paused(8) from a state other than paused(8).
        Management stations are encouraged to subsequently poll
        the subset of virtual machines of interest for
        vmOperState."
    ::= { vmNotifications 14 }

vmBulkSuspending NOTIFICATION-TYPE
    OBJECTS      {
        vmAffectedVMs
    }
    STATUS      current
    DESCRIPTION
        "This notification is generated when the operational
        state of one or more virtual machines have been changed

```

```

        to suspending(5) from a state other than suspending(5).
        Management stations are encouraged to subsequently poll
        the subset of virtual machines of interest for
        vmOperState."
 ::= { vmNotifications 15 }

vmBulkSuspended NOTIFICATION-TYPE
OBJECTS      {
                vmAffectedVMs
            }
STATUS      current
DESCRIPTION
    "This notification is generated when the operational
    state of one or more virtual machines have been changed
    to suspended(6) from a state other than suspended(6).
    Management stations are encouraged to subsequently poll
    the subset of virtual machines of interest for
    vmOperState."
 ::= { vmNotifications 16 }

vmBulkResuming NOTIFICATION-TYPE
OBJECTS      {
                vmAffectedVMs
            }
STATUS      current
DESCRIPTION
    "This notification is generated when the operational
    state of one or more virtual machines have been changed
    to resuming(7) from a state other than resuming(7).
    Management stations are encouraged to subsequently poll
    the subset of virtual machines of interest for
    vmOperState."
 ::= { vmNotifications 17 }

vmBulkMigrating NOTIFICATION-TYPE
OBJECTS      {
                vmAffectedVMs
            }
STATUS      current
DESCRIPTION
    "This notification is generated when the operational
    state of one or more virtual machines have been changed
    to migrating(9) from a state other than migrating(9).
    Management stations are encouraged to subsequently poll
    the subset of virtual machines of interest for
    vmOperState."
 ::= { vmNotifications 18 }
```

```

vmBulkCrashed NOTIFICATION-TYPE
    OBJECTS      {
                    vmAffectedVMs
                }
    STATUS        current
    DESCRIPTION   "This notification is generated when one or more virtual
                    machines have been crashed.  Management stations are
                    encouraged to subsequently poll the subset of virtual
                    machines of interest for vmOperState."
    ::= { vmNotifications 19 }

vmBulkDeleted NOTIFICATION-TYPE
    OBJECTS      {
                    vmAffectedVMs
                }
    STATUS        current
    DESCRIPTION   "This notification is generated when one or more virtual
                    machines have been deleted.  Management stations are
                    encouraged to subsequently poll the subset of virtual
                    machines of interest for vmOperState."
    ::= { vmNotifications 20 }

-- Compliance definitions:
vmCompliances OBJECT IDENTIFIER ::= { vmConformance 1 }
vmGroups      OBJECT IDENTIFIER ::= { vmConformance 2 }

vmFullCompliances MODULE-COMPLIANCE
    STATUS        current
    DESCRIPTION   "Compliance statement for implementations supporting
                    read/write access, according to the object definitions."
    MODULE        -- this module
    MANDATORY-GROUPS {
        vmHypervisorGroup,
        vmVirtualMachineGroup,
        vmCpuGroup,
        vmCpuAffinityGroup,
        vmStorageGroup,
        vmNetworkGroup
    }
    GROUP vmPerVMNotificationOptionalGroup
    DESCRIPTION   "Support for per-VM notifications is optional.  If not
                    implemented then vmPerVMNotificationsEnabled MUST report
                    false(2)."
```

```

    GROUP vmBulkNotificationsVariablesGroup
```



```
DESCRIPTION
    "Necessary only if vmPerVMNotificationOptionalGroup is
    implemented."
GROUP vmBulkNotificationOptionalGroup
DESCRIPTION
    "Support for bulk notifications is optional.  If not
    implemented then vmBulkNotificationsEnabled MUST report
    false(2)."
```

::= { vmCompliances 1 }

```
vmReadOnlyCompliances MODULE-COMPLIANCE
STATUS      current
DESCRIPTION
    "Compliance statement for implementations supporting
    only readonly access."
MODULE      -- this module
MANDATORY-GROUPS {
    vmHypervisorGroup,
    vmVirtualMachineGroup,
    vmCpuGroup,
    vmCpuAffinityGroup,
    vmStorageGroup,
    vmNetworkGroup
}

OBJECT vmPerVMNotificationsEnabled
MIN-ACCESS read-only
DESCRIPTION
    "Write access is not required."

OBJECT vmBulkNotificationsEnabled
MIN-ACCESS read-only
DESCRIPTION
    "Write access is not required."
 ::= { vmCompliances 2 }
```

```
vmHypervisorGroup OBJECT-GROUP
OBJECTS {
    vmHvSoftware,
    vmHvVersion,
    vmHvObjectID,
    vmHvUpTime,
    vmNumber,
    vmTableLastChange,
    vmPerVMNotificationsEnabled,
    vmBulkNotificationsEnabled
}
```

```
STATUS          current
DESCRIPTION
    "A collection of objects providing insight into the
    hypervisor itself."
 ::= { vmGroups 1 }

vmVirtualMachineGroup OBJECT-GROUP
OBJECTS {
    -- vmIndex
    vmName,
    vmUUID,
    vmOSType,
    vmAdminState,
    vmOperState,
    vmAutoStart,
    vmPersistent,
    vmCurCpuNumber,
    vmMinCpuNumber,
    vmMaxCpuNumber,
    vmMemUnit,
    vmCurMem,
    vmMinMem,
    vmMaxMem,
    vmUpTime,
    vmCpuTime
}
STATUS          current
DESCRIPTION
    "A collection of objects providing insight into the
    virtual machines) controlled by a hypervisor."
 ::= { vmGroups 2 }

vmCpuGroup OBJECT-GROUP
OBJECTS {
    -- vmCpuIndex,
    vmCpuCoreTime
}
STATUS          current
DESCRIPTION
    "A collection of objects providing insight into the
    virtual machines) controlled by a hypervisor."
 ::= { vmGroups 3 }

vmCpuAffinityGroup OBJECT-GROUP
OBJECTS {
    -- vmCpuPhysIndex,
    vmCpuAffinity
}
}
```

```
STATUS          current
DESCRIPTION
    "A collection of objects providing insight into the
    virtual machines) controlled by a hypervisor."
 ::= { vmGroups 4 }

vmStorageGroup OBJECT-GROUP
OBJECTS {
    -- vmStorageVmIndex,
    -- vmStorageIndex,
    vmStorageParent,
    vmStorageSourceType,
    vmStorageSourceTypeString,
    vmStorageResourceID,
    vmStorageAccess,
    vmStorageMediaType,
    vmStorageMediaTypeString,
    vmStorageSizeUnit,
    vmStorageDefinedSize,
    vmStorageAllocatedSize,
    vmStorageReadIOs,
    vmStorageWriteIOs,
    vmStorageReadOctets,
    vmStorageWriteOctets,
    vmStorageReadLatency,
    vmStorageWriteLatency
}
STATUS          current
DESCRIPTION
    "A collection of objects providing insight into the
    virtual storage devices controlled by a hypervisor."
 ::= { vmGroups 5 }

vmNetworkGroup OBJECT-GROUP
OBJECTS {
    -- vmNetworkIndex,
    vmNetworkIfIndex,
    vmNetworkParent,
    vmNetworkModel,
    vmNetworkPhysAddress
}
STATUS          current
DESCRIPTION
    "A collection of objects providing insight into the
    virtual network interfaces controlled by a hypervisor."
 ::= { vmGroups 6 }

vmPerVMNotificationOptionalGroup NOTIFICATION-GROUP
```

```
NOTIFICATIONS {
    vmRunning,
    vmShuttingdown,
    vmShutdown,
    vmPaused,
    vmSuspending,
    vmSuspended,
    vmResuming,
    vmMigrating,
    vmCrashed,
    vmDeleted
}
STATUS          current
DESCRIPTION
    "A collection of notifications for per-VM notification
    of changes to virtual machine state (vmOperState) as
    reported by a hypervisor."
 ::= { vmGroups 7 }

vmBulkNotificationsVariablesGroup OBJECT-GROUP
OBJECTS {
    vmAffectedVMs
}
STATUS          current
DESCRIPTION
    "The variables used in vmBulkNotificationOptionalGroup
    virtual network interfaces controlled by a hypervisor."
 ::= { vmGroups 8 }

vmBulkNotificationOptionalGroup NOTIFICATION-GROUP
NOTIFICATIONS {
    vmBulkRunning,
    vmBulkShuttingdown,
    vmBulkShutdown,
    vmBulkPaused,
    vmBulkSuspending,
    vmBulkSuspended,
    vmBulkResuming,
    vmBulkMigrating,
    vmBulkCrashed,
    vmBulkDeleted
}
STATUS          current
DESCRIPTION
    "A collection of notifications for bulk notification of
    changes to virtual machine state (vmOperState) as
    reported by a given hypervisor."
 ::= { vmGroups 9 }
```

END

## 6.2. IANA-STORAGE-MEDIA-TYPE-MIB

IANA-STORAGE-MEDIA-TYPE-MIB DEFINITIONS ::= BEGIN

### IMPORTS

MODULE-IDENTITY, mib-2  
FROM SNMPv2-SMI  
TEXTUAL-CONVENTION  
FROM SNMPv2-TC;

ianaStorageMediaTypeMIB MODULE-IDENTITY

LAST-UPDATED "201508050000Z" -- 5 August 2015  
ORGANIZATION "IANA"  
CONTACT-INFO

"Internet Assigned Numbers Authority  
Postal: ICANN  
12025 Waterfront Drive, Suite 300  
Los Angeles, CA 90094-2536  
Tel: +1 310-301-5800  
E-Mail: iana@iana.org"

### DESCRIPTION

"This MIB module defines Textual Conventions  
representing the media type of a storage device.

Copyright (c) 2015 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>)."

REVISION "201508050000Z" -- 5 August 2015

### DESCRIPTION

"The initial version of this MIB, published as  
RFCXXXX."  
::= { mib-2 zzz }

-- RFC Ed.: replace XXXX with RFC number and remove this note  
-- RFC Ed.: replace zzz with actual number and remove this note

IANAStorageMediaType ::= TEXTUAL-CONVENTION  
STATUS current

## DESCRIPTION

"The media type of a storage device:

```

unknown(1)      The media type is unknown, e.g., because
                  the implementation failed to obtain the
                  media type from the hypervisor.

other(2)         The media type is other than those
                  defined in this conversion.

hardDisk(3)      The media type is hard disk.

opticalDisk(4)   The media type is optical disk.

floppyDisk(5)    The media type is floppy disk."
```

## SYNTAX

```

INTEGER {
    other(1),
    unknown(2),
    hardDisk(3),
    opticalDisk(4),
    floppyDisk(5)
}
```

END

## 7. IANA Considerations

This document defines the first version of the IANA-maintained IANA-STORAGE-MEDIA-TYPE-MIB module, which allows new storage media types to be added to the enumeration in IANASStorageMediaType. An Expert Review, as defined in RFC 5226 [RFC5226], is REQUIRED for each modification.

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

Descriptor -----	OBJECT IDENTIFIER value -----
vmMIB	{ mib-2 yyy }
IANASStorageMediaTypeMIB	{ mib-2 zzz }

## 8. Security Considerations

This MIB module is typically implemented on the hypervisor not inside a virtual machine. Virtual machines, possibly under other

administrative domains, would not have access to this MIB as the SNMP service would typically operate in a separate management network.

There are two objects defined in this MIB module, `vmPerVMNotificationsEnabled` and `vmBulkNotificationsEnabled`, that have a MAX-ACCESS clause of read-write. Enabling notifications can lead to a substantial number of notifications if many virtual machines change their state concurrently. Hence, 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 the management system. It is RECOMMENDED that these objects have access of read-only instead of read-write on deployments where SNMPv3 strong security (i.e., authentication and encryption) is not used.

There are a number of managed objects in this MIB that may contain sensitive information. The objects in the `vmHvSoftware` and `vmHvVersion` list information about the hypervisor's software and version. Some may wish not to disclose to others which software they are running. Further, an inventory of the running software and versions may be helpful to an attacker who hopes to exploit software bugs in certain applications. Moreover, the objects in the `vmTable`, `vmCpuTable`, `vmCpuAffinityTable`, `vmStorageTable` and `vmNetworkTable` list information about the virtual machines and their virtual resource allocation. Some may wish not to disclose to others how many and what virtual machines they are operating.

It is thus important to control even GET access to these objects and possibly to even encrypt the values of these object when sending them over the network via SNMP. Not all versions of SNMP provide features for such a secure environment.

SNMPv1 by itself is not a secure environment. 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.

It is recommended that the implementers consider using the security features as provided by the SNMPv3 framework. Specifically, the use of the User-based Security Model [RFC3414] and the View-based Access Control Model [RFC3415] is recommended.

It is then a customer/user responsibility to ensure that the SNMP entity giving access to an instance of this MIB, 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.

## 9. Contributors

Yuji Sekiya  
The University of Tokyo  
2-11-16 Yayoi  
Bunkyo-ku, Tokyo 113-8658  
Japan

Email: [sekiya@wide.ad.jp](mailto:sekiya@wide.ad.jp)

Cathy Zhou  
Huawei Technologies  
Bantian, Longgang District  
Shenzhen 518129  
P.R. China

Email: [cathyzhou@huawei.com](mailto:cathyzhou@huawei.com)

Hiroshi Esaki  
The University of Tokyo  
7-3-1 Hongo  
Bunkyo-ku, Tokyo 113-8656  
Japan

Email: [hiroshi@wide.ad.jp](mailto:hiroshi@wide.ad.jp)

## 10. Acknowledgements

The authors like to thank Andy Bierman, David Black, Joe Marcus Clarke, C.M. Heard, Joel Jaeggli, Tom Petch, Randy Presuhn, and Ian West for providing helpful comments during the development of this specification.

Juergen Schoenwaelder was partly funded by Flamingo, a Network of Excellence project (ICT-318488) supported by the European Commission under its Seventh Framework Programme.

## 11. References

### 11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.



- [RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIV2)", STD 58, RFC 2578, DOI 10.17487/RFC2578, April 1999, <<http://www.rfc-editor.org/info/rfc2578>>.
- [RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Textual Conventions for SMIV2", STD 58, RFC 2579, DOI 10.17487/RFC2579, April 1999, <<http://www.rfc-editor.org/info/rfc2579>>.
- [RFC2580] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Conformance Statements for SMIV2", STD 58, RFC 2580, DOI 10.17487/RFC2580, April 1999, <<http://www.rfc-editor.org/info/rfc2580>>.
- [RFC2790] Waldbusser, S. and P. Grillo, "Host Resources MIB", RFC 2790, DOI 10.17487/RFC2790, March 2000, <<http://www.rfc-editor.org/info/rfc2790>>.
- [RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, DOI 10.17487/RFC2863, June 2000, <<http://www.rfc-editor.org/info/rfc2863>>.
- [RFC3413] Levi, D., Meyer, P., and B. Stewart, "Simple Network Management Protocol (SNMP) Applications", STD 62, RFC 3413, DOI 10.17487/RFC3413, December 2002, <<http://www.rfc-editor.org/info/rfc3413>>.
- [RFC3414] Blumenthal, U. and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", STD 62, RFC 3414, DOI 10.17487/RFC3414, December 2002, <<http://www.rfc-editor.org/info/rfc3414>>.
- [RFC3415] Wijnen, B., Presuhn, R., and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3415, DOI 10.17487/RFC3415, December 2002, <<http://www.rfc-editor.org/info/rfc3415>>.
- [RFC3418] Presuhn, R., Ed., "Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3418, DOI 10.17487/RFC3418, December 2002, <<http://www.rfc-editor.org/info/rfc3418>>.

- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <<http://www.rfc-editor.org/info/rfc5226>>.
- [RFC6933] Bierman, A., Romascanu, D., Quittek, J., and M. Chandramouli, "Entity MIB (Version 4)", RFC 6933, DOI 10.17487/RFC6933, May 2013, <<http://www.rfc-editor.org/info/rfc6933>>.

## 11.2. Informative References

- [RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, DOI 10.17487/RFC3410, December 2002, <<http://www.rfc-editor.org/info/rfc3410>>.
- [IEEE8021-BRIDGE-MIB]  
IEEE, "IEEE8021-BRIDGE-MIB", October 2008, <<http://www.ieee802.org/1/files/public/MIBs/IEEE8021-BRIDGE-MIB-200810150000Z.txt>>.
- [IEEE8021-Q-BRIDGE-MIB]  
IEEE, "IEEE8021-BRIDGE-MIB", October 2008, <<http://www.ieee802.org/1/files/public/MIBs/IEEE8021-Q-BRIDGE-MIB-200810150000Z.txt>>.

## Appendix A. State Transition Table

State	Change to vmAdminState at the hypervisor or (Event)	Next state	Notification
suspended	running	resuming	vmResuming   vmBulkResuming
suspending	(suspend operation completed)	suspended	vmSuspended   vmBulkSuspended
running	suspended	suspending	vmSuspending   vmBulkSuspending
	shutdown	shuttingdown	vmShuttingdown

				vmBulkShuttingdown
		(migration to other hypervisor initiated)	migrating	vmMigrating   vmBulkMigrating
	resuming	(resume operation completed)	running	vmRunning   vmBulkRunning
	paused	running	running	vmRunning   vmBulkRunning
	shuttingdown	(shutdown operation completed)	shutdown	vmShutdown   vmBulkShutdown
	shutdown	running	running	vmRunning   vmBulkRunning
		(if this state entry is created by a migration operation (*))	migrating	vmMigrating   vmBulkMigrating
		(deletion operation completed)	(no state)	vmDeleted   vmBulkDeleted
	migrating	(migration from other hypervisor completed)	running	vmRunning   vmBulkRunning
		(migration to other hypervisor completed)	shutdown	vmShutdown   vmBulkShutdown
	preparing	(preparation completed)	shutdown	vmShutdown   vmBulkShutdown
	crashed	-	-	-
		(crashed)	crashed	vmCrashed   vmBulkCrashed

(no state)	(preparation initiated)	preparing	-
	(migrate from other hypervisor initiated)	shutdown (*)	vmShutdown   vmBulkShutdown

State transition table for vmOperState

## Authors' Addresses

Hirochika Asai  
The University of Tokyo  
7-3-1 Hongo  
Bunkyo-ku, Tokyo 113-8656  
JP

Phone: +81 3 5841 6748  
Email: panda@hongo.wide.ad.jp

Michael MacFaden  
VMware Inc.  
  
Email: mrm@vmware.com

Juergen Schoenwaelder  
Jacobs University  
Campus Ring 1  
Bremen 28759  
Germany

Email: j.schoenwaelder@jacobs-university.de

Keiichi Shima  
IIJ Innovation Institute Inc.  
2-10-2 Fujimi  
Chiyoda-ku, Tokyo 102-0071  
JP

Email: keiichi@iijlab.net

Tina Tsou  
Huawei Technologies (USA)  
2330 Central Expressway  
Santa Clara CA 95050  
USA

Email: [tina.tsou.zouting@huawei.com](mailto:tina.tsou.zouting@huawei.com)

OPSAWG  
Internet-Draft  
Intended status: Informational  
Expires: January 22, 2015

T. Taylor, Ed.  
PT Taylor Consulting  
D. Romascanu  
Avaya  
July 21, 2014

Transferring MIB Work from IETF Ethernet Interfaces and Hub MIB WG to  
IEEE 802.3 WG  
draft-taylor-opsawg-mibs-to-ieee80231-01

## Abstract

This document records the transfer of ownership of the Ethernet-related MIB modules DOT3-OAM-MIB, SNMP-REPEATER-MIB, POWER-ETHERNET-MIB, DOT3-EPON-MIB, EtherLike-MIB, EFM-CU-MIB, ETHER-WIS and MAU-MIB from the IETF to the IEEE 802.3 Working Group. This document also describes the procedures associated with the transfer, relying heavily on RFC 4663 (which records an earlier transfer to the IEEE 802.1 Working Group) as the primary source.

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 22, 2015.

## Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. IETF and Corresponding IEEE 802.3 MIB modules . . . . .	2
3. Procedural Aspects Of the Transfer . . . . .	4
3.1. IEEE MIB Modules in ASCII Format . . . . .	4
3.2. OID Registration for New MIB Modules . . . . .	4
3.3. Mailing List Discussions . . . . .	4
3.4. IETF MIB Doctor Reviews . . . . .	5
4. Security Considerations . . . . .	5
5. IANA Considerations . . . . .	5
6. IPR Considerations . . . . .	5
7. Acknowledgements . . . . .	5
8. Informative References . . . . .	5
Authors' Addresses . . . . .	6

## 1. Introduction

[RFC4663], published in September, 2006, described a plan for transferring four MIB modules related to bridges from IETF to IEEE 802.1 ownership. Some years later, ownership of eight more MIB modules was transferred from the IETF Ethernet Interfaces and Hub MIB (hubmib) Working Group to the IEEE 802.3 Working Group. The MIB modules concerned are tabulated below (Section 2). [RFC4663] clearly enunciates the motivation for both transfers and also provides an introduction to IEEE standardization procedures. The discussions of those topics will not be repeated here.

The IEEE version of this second lot of transferred MIB modules was published as 802.3.1-2011 in February, 2011. The IEEE 802.3.1 specification was subsequently updated. The latest version, IEEE 802.3.1-2013 [IEEE802.3.1-2013], is the basis for this document.

## 2. IETF and Corresponding IEEE 802.3 MIB modules

This section tabulates the MIB modules that were transferred to IEEE 802.3, identifying the IETF source document, the corresponding clause of [IEEE802.3.1-2013], and the location of the MIB itself in ASCII format.

IETF MIB Name: DOT3-OAM-MIB

IETF Reference: Definitions and Managed Objects for Operations, Administration, and Maintenance (OAM) Functions on Ethernet-Like Interfaces [RFC4878]

IEEE 802.3 MIB Name: IEEE8023-DOT3-OAM-MIB

IEEE 802.3.1-2013 description: Clause 6, Ethernet operations, administration, and maintenance (OAM) MIB module

MIB Location: [http://www.ieee802.org/3/1/public/mib\\_modules/20130411/802dot3dot1C6mib.txt](http://www.ieee802.org/3/1/public/mib_modules/20130411/802dot3dot1C6mib.txt)

IETF MIB Name: SNMP-REPEATER-MIB

IETF Reference: Definitions of Managed Objects for IEEE 802.3

Repeater Devices using SMiv2 [RFC2108]

IEEE 802.3 MIB Name: IEEE8023-SNMP-REPEATER-MIB

IEEE 802.3.1-2013 description: Clause 7, Ethernet repeater device MIB module

MIB Location: [http://www.ieee802.org/3/1/public/mib\\_modules/20130411/802dot3dot1C7mib.txt](http://www.ieee802.org/3/1/public/mib_modules/20130411/802dot3dot1C7mib.txt)

IETF MIB Name: POWER-ETHERNET-MIB

IETF Reference: Power Ethernet MIB [RFC3621]

IEEE 802.3 MIB Name: IEEE8023-POWER-ETHERNET-MIB

IEEE 802.3.1-2013 description: Clause 8, Ethernet data terminal equipment (DTE) power via medium dependent interface (MDI) MIB module

MIB Location: [http://www.ieee802.org/3/1/public/mib\\_modules/20130411/802dot3dot1C8mib.txt](http://www.ieee802.org/3/1/public/mib_modules/20130411/802dot3dot1C8mib.txt)

IETF MIB Name: DOT3-EPON-MIB

IETF Reference: Managed Objects of Ethernet Passive Optical Networks (EPON) [RFC4837]

IEEE 802.3 MIB Name: IEEE8023-DOT3-EPON-MIB

IEEE 802.3.1-2013 description: Clause 9, Ethernet passive optical networks (EPON) MIB module

MIB Location: [http://www.ieee802.org/3/1/public/mib\\_modules/20130411/802dot3dot1C9mib.txt](http://www.ieee802.org/3/1/public/mib_modules/20130411/802dot3dot1C9mib.txt)

IETF MIB Name: EtherLike-MIB

IETF Reference: Definitions of Managed Objects for the Ethernet-like Interface Types [RFC3635]

IEEE 802.3 MIB Name: ieee8023etherMIB

IEEE 802.3.1-2013 description: Clause 10, Ethernet-like interface MIB module

MIB Location: [http://www.ieee802.org/3/1/public/mib\\_modules/20130411/802dot3dot1C10mib.txt](http://www.ieee802.org/3/1/public/mib_modules/20130411/802dot3dot1C10mib.txt)

IETF MIB Name: EFM-CU-MIB

IETF Reference: Ethernet in the First Mile Copper (EFMCu) Interfaces MIB [RFC5066]

IEEE 802.3 MIB Name: IEEE8023-EFM-CU-MIB



IEEE 802.3.1-2013 description: Clause 11, Ethernet in the First Mile copper (EFMCu) interfaces MIB module

MIB Location: [http://www.ieee802.org/3/1/public/mib\\_modules/20130411/802dot3dot1C11mib.tx](http://www.ieee802.org/3/1/public/mib_modules/20130411/802dot3dot1C11mib.tx)

IETF MIB Name: ETHER-WIS

IETF Reference: Definitions of Managed Objects for the Ethernet WAN Interface Sublayer [RFC3637]

IEEE 802.3 MIB Name: IEEE8023-ETHER-WIS-MIB

IEEE 802.3.1-2013 description: Clause 12, Ethernet wide area network (WAN) interface sublayer (WIS) MIB module

MIB Location: [http://www.ieee802.org/3/1/public/mib\\_modules/20130411/802dot3dot1C12mib.txt](http://www.ieee802.org/3/1/public/mib_modules/20130411/802dot3dot1C12mib.txt)

IETF MIB Name: MAU-MIB

IETF Reference: Definitions of Managed Objects for IEEE 802.3 Medium Attachment Units (MAUs) [RFC4836]

IEEE 802.3 MIB Name: IEEE8023-MAU-MIB

IEEE 802.3.1-2013 description: Clause 13, Ethernet medium attachment units (MAUs) MIB module

MIB Location: [http://www.ieee802.org/3/1/public/mib\\_modules/20130411/802dot3dot1C13mib.txt](http://www.ieee802.org/3/1/public/mib_modules/20130411/802dot3dot1C13mib.txt)

### 3. Procedural Aspects Of the Transfer

#### 3.1. IEEE MIB Modules in ASCII Format

The content of Section 2.2 of [RFC4663] is accurate also for this document.

#### 3.2. OID Registration for New MIB Modules

The IEEE 802.3 WG adopted the approach recommended in [RFC4663], Section 2.3 of developing an IEEE MIB module and defining new compliance clauses under the IEEE OID branch. Information about the IEEE 802.3 Management Registration Arcs can be found at <http://www.ieee802.org/3/arcs/index.html>.

#### 3.3. Mailing List Discussions

The Ethernet Interfaces and Hub MIB WG has completed its documents, and the WG was closed in September 2007. The mailing list stayed open for a while, and was closed a few years later. The appropriate mailing list for IEEE 802.3 MIB modules discussion is STDS-802-3-MIB@LISTSERV.IEEE.ORG.

To see general information about 802.3, including how they work and how to participate, go to <http://www.ieee802.org/3/>.

### 3.4. IETF MIB Doctor Reviews

The content of Section 5 of [RFC4663] is accurate also for this document, noting that from the point of view of the present document, 802.3 should replace 802.1 wherever it occurs in the text.

### 4. Security Considerations

This document records the transfer of ownership of Ethernet-related MIB modules to IEEE 802.3.1 several years ago. The transfer has no security implications.

### 5. IANA Considerations

This document requires no actions by IANA.

### 6. IPR Considerations

See Section 9 of [RFC4663].

### 7. Acknowledgements

Thanks to Juergen Schoenwaelder and Howard Frazier for their reviews and comments on both the initial and the present versions of this document.

### 8. Informative References

- [IEEE802.3.1-2013]  
IEEE Computer Society, "IEEE Standard for Management Information Base (MIB) Definitions for Ethernet", June 2013.
- [RFC2108] de Graaf, K., Romascanu, D., McMaster, D., and K. McCloghrie, "Definitions of Managed Objects for IEEE 802.3 Repeater Devices using SMIV2", RFC 2108, February 1997.
- [RFC3621] Berger, A. and D. Romascanu, "Power Ethernet MIB", RFC 3621, December 2003.
- [RFC3635] Flick, J., "Definitions of Managed Objects for the Ethernet-like Interface Types", RFC 3635, September 2003.
- [RFC3637] Heard, C., "Definitions of Managed Objects for the Ethernet WAN Interface Sublayer", RFC 3637, September 2003.

- [RFC4663] Harrington, D., "Transferring MIB Work from IETF Bridge MIB WG to IEEE 802.1 WG", RFC 4663, September 2006.
- [RFC4836] Beili, E., "Definitions of Managed Objects for IEEE 802.3 Medium Attachment Units (MAUs)", RFC 4836, April 2007.
- [RFC4837] Khernmash, L., "Managed Objects of Ethernet Passive Optical Networks (EPON)", RFC 4837, July 2007.
- [RFC4878] Squire, M., "Definitions and Managed Objects for Operations, Administration, and Maintenance (OAM) Functions on Ethernet-Like Interfaces", RFC 4878, June 2007.
- [RFC5066] Beili, E., "Ethernet in the First Mile Copper (EFMCu) Interfaces MIB", RFC 5066, November 2007.

Authors' Addresses

Tom Taylor (editor)  
PT Taylor Consulting  
Ottawa  
Canada

Email: tom.taylor.stds@gmail.com

Dan Romascanu  
Avaya  
Park Atidim, Bldg. #3  
Tel Aviv 61581  
Israel

Phone: +972-3-6458414  
Email: dromasca@avaya.com

Network Working Group  
Internet-Draft  
Intended status: Experimental  
Expires: August 17, 2014

A. Capello  
M. Cociglio  
L. Castaldelli  
Telecom Italia  
A. Tempia Bonda

February 13, 2014

A packet based method for passive performance monitoring  
draft-tempia-opsawg-p3m-04.txt

## Abstract

This document describes a passive method to perform packet loss, delay and jitter measurements on live traffic. Implementation and deployment details are also explained in order to clarify how the tools and features currently available on existing routing platforms can be used to implement the method. This method has been invented and engineered in Telecom Italia and it's currently being used in Telecom Italia's network.

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 17, 2014.

## Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document.

## Table of Contents

1. Introduction . . . . .	2
2. Overview of the method . . . . .	4
3. Detailed description of the method . . . . .	5
3.1. Packet loss measurement . . . . .	5
3.2. One-way delay measurement . . . . .	9
3.2.1. Average delay . . . . .	10
3.3. Delay variation measurement . . . . .	11
4. Implementation and deployment . . . . .	11
4.1. Coloring the packets . . . . .	13
4.2. Counting the packets . . . . .	14
4.3. Collecting data and calculating packet loss . . . . .	15
5. Compliance with RFC6390 guidelines . . . . .	15
6. Security Considerations . . . . .	17
7. Conclusions . . . . .	17
8. IANA Considerations . . . . .	18
9. Acknowledgements . . . . .	18
10. References . . . . .	18
10.1. Normative References . . . . .	18
10.2. Informative References . . . . .	19
Authors' Addresses . . . . .	19

## 1. Introduction

Nowadays, most of the traffic in Service Providers' networks carries multimedia content. Video contents are highly sensitive to packet loss [RFC2680], while interactive contents are sensitive to delay [RFC2679], and jitter [RFC3393].

In front of this scenario, Service Providers need methodologies and tools to monitor and measure network performances with an adequate accuracy, in order to constantly control the quality of experience perceived by their customers. On the other hand, performance monitoring provides useful information for improving network management (e.g. isolation of network problems, troubleshooting, etc.).

A lot of work related to OAM, that includes also performance monitoring techniques, has been done by Standards Developing Organizations: [I-D.ietf-opsawg-oam-overview] provides a good overview of existing OAM mechanisms defined in IETF, ITU-T and IEEE. Considering IETF, a lot of work has been done on fault detection and connectivity verification, while a minor effort has been dedicated so far to performance monitoring. The IPPM WG has defined standard

metrics to measure network performance; however, the methods developed in the WG mainly refer to active measurement techniques. More recently, the MPLS WG has defined mechanisms for measuring packet loss, one-way and two-way delay, and delay variation in MPLS networks[RFC6374], but their applicability to passive measurements has some limitations, especially for pure connection-less networks.

The lack of adequate tools to measure packet loss with the desired accuracy drove an effort in Telecom Italia to design a new method for the performance monitoring of live traffic, possibly easy to implement and deploy. The effort led to the method described in this document: basically, it is a passive performance monitoring technique, potentially applicable to any kind of packet based traffic, including Ethernet, IP, and MPLS, both unicast and multicast. The method addresses primarily packet loss measurement, but it can be easily extended to one-way delay and delay variation measurements as well. It doesn't require any protocol extension or interaction with existing protocols, thus avoiding any interoperability issue. Even if the method doesn't raise any specific need for standardization, it could be further improved by means of some extension to existing protocols, but this aspect is left for further study and it is out of the scope of this document.

The method has been explicitly designed for passive measurements but it can also be used with active probes. Passive measurements are usually more easily understood by customers and provide a much better accuracy, especially for packet loss measurements.

The method described in this document has been invented and engineered in Telecom Italia and it's currently being used in Telecom Italia's network.

This document is organized as follows:

- o Section 2 gives an overview of the method, including a comparison with alternate measurement strategies;
- o Section 3 describes the method in detail
- o Section 4 discusses implementation and deployment considerations, with special regard to the choices adopted in Telecom Italia's own implementation;
- o Section 5 includes some considerations about security aspects;
- o Section 6 finally summarizes some concluding remarks.

## 2. Overview of the method

In order to perform packet loss measurements on a live traffic flow, different approaches exist. The most intuitive one consists in numbering the packets, so that each router that receives the flow can immediately detect a packet missing. This approach, though very simple in theory, is not simple to achieve: it requires the insertion of a sequence number into each packet and the devices must be able to extract the number and check it in real time. Such a task can be difficult to implement on live traffic: if UDP is used as the transport protocol, the sequence number is not available; on the other hand, if a higher layer sequence number (e.g. in the RTP header) is used, extracting that information from each packet and process it in real time could overload the device.

An alternate approach is to count the number of packets sent on one end, the number of packets received on the other end, and to compare the two values. This operation is much simpler to implement, but requires that the devices performing the measurement are in sync: in order to compare two counters it is required that they refer exactly to the same set of packets. Since a flow is continuous and cannot be stopped when a counter has to be read, it could be difficult to determine exactly when to read the counter. A possible solution to overcome this problem is to virtually split the flow in consecutive blocks by inserting periodically a delimiter so that each counter refers exactly to the same block of packets. The delimiter could be for example a special packet inserted artificially into the flow. However, delimiting the flow using specific packets has some limitations. First, it requires generating additional packets within the flow and requires the equipment to be able to process those packets. In addition, the method is vulnerable to out of order reception of delimiting packets and, to a lesser extent, to their loss.

The method proposed in this document follows the second approach, but it doesn't use additional packets to virtually split the flow in blocks. Instead, it "colors" the packets so that the packets belonging to the same block will have the same color, whilst consecutive blocks will have different colors. Each change of color represents a sort of auto-synchronization signal that guarantees the consistency of measurements taken by different devices along the path.

Figure 1 represents a very simple network and shows how the method can be used to measure packet loss on different network segments: by enabling the measurement on several interfaces along the path, it is possible to perform link monitoring, node monitoring or end-to-end monitoring. The method is flexible enough to measure packet loss on

any segment of the network and can be used to isolate the faulty element.

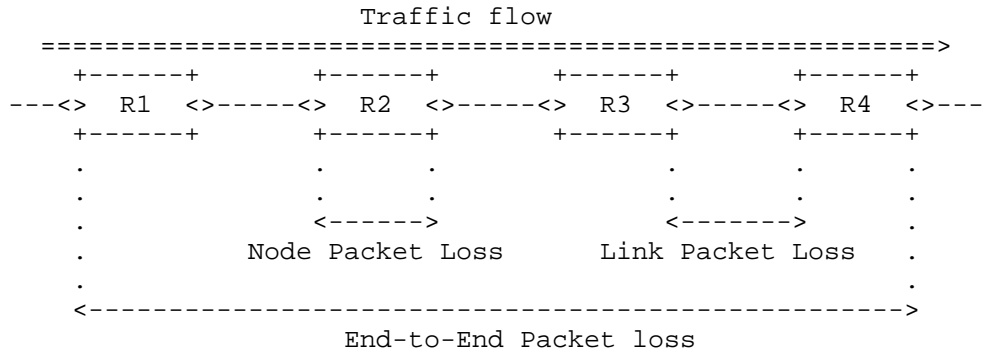


Figure 1: Available measurements

### 3. Detailed description of the method

This section describes in detail how the method. A special emphasis is given to the measurement of packet loss, that represents the core application of the method, but applicability to delay and jitter measurements is also considered.

#### 3.1. Packet loss measurement

The basic idea is to virtually split traffic flows into consecutive blocks: each block represents a measurable entity unambiguously recognizable by all network devices along the path. By counting the number of packets in each block and comparing the values measured by different network devices along the path, it is possible to measure packet loss occurred in any single block between any two points.

As discussed in the previous section, a simple way to create the blocks is to "color" the traffic (two colors are sufficient) so that packets belonging to different consecutive blocks will have different colors. Whenever the color changes, the previous block terminates and the new one begins. Hence, all the packets belonging to the same block will have the same color and packets of different consecutive blocks will have different colors. The number of packets in each block depends on the criterion used to create the blocks: if the color is switched after a fixed number of packets, then each block will contain the same number of packets (except for any losses); but if the color is switched according to a fixed timer, then the number of packets may be different in each block depending on the packet rate.



The following figure shows how a flow looks like when it is split in traffic blocks with colored packets.

A: packet with A coloring  
B: packet with B coloring

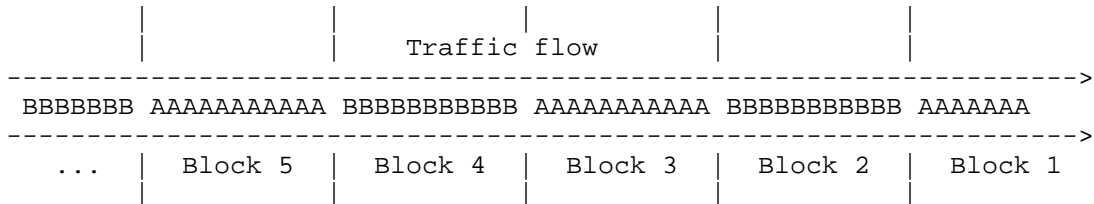


Figure 2: Traffic coloring

Figure 3 shows how the method can be used to measure link packet loss between two adjacent nodes.

Referring to the figure, let's assume we want to monitor the packet loss on the link between two routers: router R1 and router R2. According to the method, the traffic is colored alternatively with two different colors, A and B. Whenever the color changes, the transition generates a sort of square-wave signal, as depicted in the following figure.

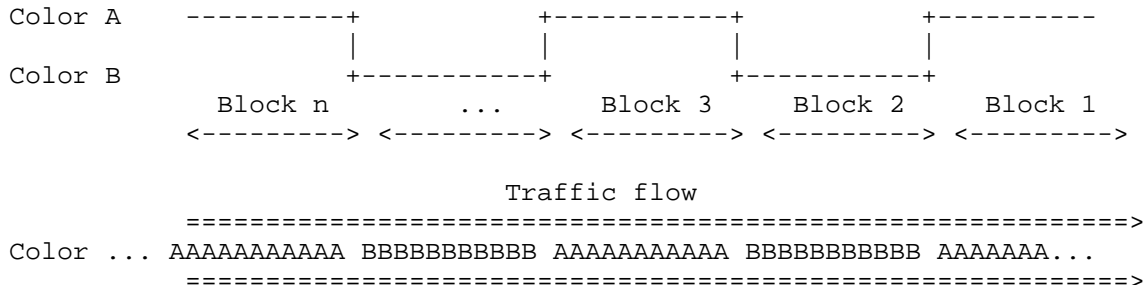


Figure 3: Application of the method to compute link packet loss

Traffic coloring could be done by R1 itself or by an upward router. R1 needs two counters, C(A)R1 and C(B)R1, on its egress interface: C(A)R1 counts the packets with color A and C(B)R1 counts those with color B. As long as traffic is colored A, only counter C(A)R1 will be incremented, while C(B)R1 is not incremented; vice versa, when the traffic is colored as B, only C(B)R1 is incremented. C(A)R1 and C(B)R1 can be used as reference values to determine the packet loss

from R1 to any other measurement point down the path. Router R2, similarly, will need two counters on its ingress interface, C(A)R2 and C(B)R2, to count the packets received on that interface and colored with color A and B respectively. When an A block ends, it is possible to compare C(A)R1 and C(A)R2 and calculate the packet loss within the block; similarly, when the successive B block terminates, it is possible to compare C(B)R1 with C(B)R2, and so on for every successive block.

Likewise, by using two counters on R2 egress interface it is possible to count the packets sent out of R2 interface and use them as reference values to calculate the packet loss from R2 to any measurement point down R2.

Using a fixed timer for color switching offers a better control over the method: the (time) length of the blocks can be chosen large enough to simplify the collection and the comparison of measures taken by different network devices. It's preferable to read the value of the counters not immediately after the color switch: some packets could arrive out of order and increment the counter associated to the previous block (color), so it is worth waiting for some seconds. The drawback is that the longer the duration of the block, the less frequent the measurement can be taken.

The following table shows how the counters can be used to calculate the packet loss between R1 and R2. The first column lists the sequence of traffic blocks while the other columns contain the counters of A-colored packets and B-colored packets for R1 and R2. In this example, we assume that the values of the counters are reset to zero whenever a block ends and its associated counter has been read: with this assumption, the table shows only relative values, that is the exact number of packets of each color within each block. If the values of the counters were not reset, the table would contain cumulative values, but the relative values could be determined simply by difference from the value of the previous block of the same color.

The color is switched on the basis of a fixed timer (not shown in the table), so the number of packets in each block is different.

Block	C(A)R1	C(B)R1	C(A)R2	C(B)R2	Loss
1	375	0	375	0	0
2	0	388	0	388	0
3	382	0	381	0	1
4	0	377	0	374	3
...	...	...	...	...	...
n	0	387	0	387	0
n+1	379	0	377	0	2

Table 1: Evaluation of counters for packet loss measurements

During an A block (blocks 1, 3 and n+1), all the packets are A-colored, therefore the C(A) counters are incremented to the number seen on the interface, while C(B) counters are zero. Vice versa, during a B block (blocks 2, 4 and n), all the packets are B-colored: C(A) counters are zero, while C(B) counters are incremented.

When a block ends (because of color switching) the relative counters stop incrementing and it is possible to read them, compare the values measured on router R1 and R2 and calculate the packet loss within that block.

For example, looking at the table above, during the first block (A-colored), C(A)R1 and C(A)R2 have the same value (375), which corresponds to the exact number of packets of the first block (no loss). Also during the second block (B-colored) R1 and R2 counters have the same value (388), which corresponds to the number of packets of the second block (no loss). During blocks three and four, R1 and R2 counters are different, meaning that some packets have been lost: in the example, one single packet (382-381) was lost during block three and three packets (377-374) were lost during block four.

The method applied to R1 and R2 can be extended to any other router and applied to more complex networks, as far as the measurement is enabled on the path followed by the traffic flow(s) being observed.

### 3.2. One-way delay measurement

The same principle used to measure packet loss can be applied also to one-way delay measurement: the alternation of colors can be used as a time reference to calculate the delay. Whenever the color changes (that means that a new block has started) a network device can store the timestamp of the first packet of the new block; that timestamp can be compared with the timestamp of the same packet on a second router to compute packet delay. Considering Figure 4, R1 stores a timestamp TS(A1)R1 when it sends the first packet of block 1 (A-colored), a timestamp TS(B2)R1 when it sends the first packet of block 2 (B-colored) and so on for every other block. R2 performs the same operation on the receiving side, recording TS(A1)R2, TS(B2)R2 and so on. Since the timestamps refer to specific packets (the first packet of each block) we are sure that timestamps compared to compute delay refer to the same packets. By comparing TS(A1)R1 with TS(A1)R2 (and similarly TS(B2)R1 with TS(B2)R2 and so on) it is possible to measure the delay between R1 and R2. In order to have more measurements, it is possible to take and store more timestamps, referring to other packets within each block.

In order to coherently compare timestamps collected on different routers, the network nodes must be in sync. Furthermore, a measurement is valid only if no packet loss occurs and if packet misordering can be avoided, otherwise the first packet of a block on R1 could be different from the first packet of the same block on R2 (f.i. if that packet is lost between R1 and R2 or it arrives after the next one).

The following table shows how timestamps can be used to calculate the delay between R1 and R2. The first column lists the sequence of blocks while other columns contain the timestamp referring to the first packet of each block on R1 and R2. The delay is computed as a difference between timestamps. For the sake of simplicity, all the values are expressed in milliseconds.

Block	TS(A)R1	TS(B)R1	TS(A)R2	TS(B)R2	Delay R1-R2
1	12.483	-	15.591	-	3.108
2	-	6.263	-	9.288	3.025
3	27.556	-	30.512	-	2.956
	-	18.113	-	21.269	3.156
...	...	...	...	...	...
n	77.463	-	80.501	-	3.038
n+1	-	24.333	-	27.433	3.100

Table 2: Evaluation of timestamps for delay measurements

The first row shows timestamps taken on R1 and R2 respectively and referring to the first packet of block 1 (which is A-colored). Delay can be computed as a difference between the timestamp on R2 and the timestamp on R1. Similarly, the second row shows timestamps (in milliseconds) taken on R1 and R2 and referring to the first packet of block 2 (which is B-colored). Comparing timestamps taken on different nodes in the network and referring to the same packets (identified using the alternation of colors) it is possible to measure delay on different network segments.

For the sake of simplicity, in the above example a single measurement is provided within a block, taking into account only the first packet of each block. The number of measurements can be easily increased by considering multiple packets in the block: for instance, a timestamp could be taken every N packets, thus generating multiple delay measurements. Taking this to the limit, in principle the delay could be measured for each packet, by taking and comparing the corresponding timestamps (possible but impractical from an implementation point of view).

### 3.2.1. Average delay

As mentioned before, the method previously exposed for measuring the delay is sensitive to out of order reception of packets. In order to overcome this problem, a different approach has been considered: it is based on the concept of average delay. The average delay is calculated by considering the average arrival time of the packets within a single block. The network device locally stores a timestamp

for each packet received within a single block: summing all the timestamps and dividing by the total number of packets received, the average arrival time for that block of packets can be calculated. By subtracting the average arrival times of two adjacent devices it is possible to calculate the average delay between those nodes. This method is robust to out of order packets and also to packet loss (only a small error is introduced). Moreover, it greatly reduces the number of timestamps (only one per block for each network device) that have to be collected by the management system. On the other hand, it only gives one measure for the duration of the block (f.i. 5 minutes), and it doesn't give the minimum and maximum delay values. This limitation could be overcome by reducing the duration of the block (f.i. from 5 minutes to a few seconds) by means of an highly optimized implementation of the method.

By summing the average delays of the two directions of a path, it is also possible to measure the two-way delay (round-trip delay).

### 3.3. Delay variation measurement

Similarly to one-way delay measurement, the method can also be used to measure the inter-arrival jitter. The alternation of colors can be used as a time reference to measure delay variations. Considering the example depicted in Figure 4, R1 stores a timestamp TS(A)R1 whenever it sends the first packet of a block and R2 stores a timestamp TS(B)R2 whenever it receives the first packet of a block. The inter-arrival jitter can be easily derived from one-way delay measurement, by evaluating the delay variation of consecutive samples.

The concept of average delay can also be applied to delay variation, by evaluating the variation of consecutive measures of the average delay.

## 4. Implementation and deployment

The methodology described in the previous sections has been implemented in Telecom Italia by leveraging functions and tools available on IP routers and it's currently being used to monitor packet loss in some portions of Telecom Italia's network. The application of the method to delay measurement is currently being evaluated in Telecom Italia's labs.

The fundamental steps for the implementation of the method can be summarized in the following items:

- o coloring the packets;

- o counting the packets;
- o collecting data and calculating the packet loss.

Before going deeper into the implementation details, it's worth mentioning two different strategies that can be used when implementing the method:

- o flow-based: the flow-based strategy is used when only a limited number of traffic flows need to be monitored. This could be the case, for example, of IPTV channels or other specific applications traffic with high QoS requirements. According to this strategy, only a subset of the flows is colored. Counters for packet loss measurements can be instantiated for each single flow, or for the set as a whole, depending on the desired granularity. A relevant problem with this approach is the necessity to know in advance the path followed by flows that are subject to measurement. Path rerouting and traffic load-balancing increase the issue complexity, especially for unicast traffic. The problem is easier to solve for multicast traffic where load balancing is seldom used, especially for IPTV traffic where static joins are frequently used to force traffic forwarding and replication.
- o link-based: measurements are performed on all the traffic on a link by link basis. The link could be a physical link or a logical link (for instance an Ethernet VLAN or a MPLS PW). Counters could be instantiated for the traffic as a whole or for each traffic class (in case it is desired to monitor each class separately), but in the second case a couple of counters is needed for each class.

The current implementation in Telecom Italia uses the first strategy. As mentioned, the flow-based measurement requires the identification of the flow to be monitored and the discovery of the path followed by the selected flow. It is possible to monitor a single flow or multiple flows grouped together, but in this case measurement is consistent only if all the flows in the group follow the same path. Moreover, a Service Provider should be aware that, if a measurement is performed by grouping many flows, it is not possible to determine exactly which flow was affected by packets loss. In order to have measures per single flow it is necessary to configure counters for each specific flow. Once the flow(s) to be monitored have been identified, it is necessary to configure the monitoring on the proper nodes. Configuring the monitoring means configuring the policy to intercept the traffic and configuring the counters to count the packets. To have just an end-to-end monitoring, it is sufficient to enable the monitoring on the first and the last hop routers of the path: the mechanism is completely transparent to intermediate nodes

and independent from the path followed by traffic flows. On the contrary, to monitor the flow on a hop-by-hop basis along its whole path it is necessary to enable the monitoring on every node from the source to the destination. In case the exact path followed by the flow is not known a priori (i.e. the flow has multiple paths to reach the destination) it is necessary to enable the monitoring system on every path: counters on interfaces traversed by the flow will report packet count, counters on other interfaces will be null.

#### 4.1. Coloring the packets

The coloring operation is fundamental in order to create packet blocks. This implies choosing where to activate the coloring and how to color the packets.

In case of flow-based measurements, it is desirable, in general, to have a single coloring node because it is easier to manage and doesn't rise any risk of conflict (consider the case where two nodes color the same flow). Thus it is necessary to color the flow as close as possible to the source. In addition, coloring a flow close to the source allows an end-to-end measure if a measurement point is enabled on the last-hop router as well. The only requirement is that the coloring must change periodically and every node along the path must be able to identify unambiguously the colored packets. For link-based measurements, all traffic needs to be colored when transmitted on the link. If the traffic had already been colored, then it has to be re-colored because the color must be consistent on the link. This means that each hop along the path must (re-)color the traffic; the color is not required to be consistent along different links.

Traffic coloring can be implemented by setting a specific bit in the packet header and changing the value of that bit periodically. With current router implementations, only QoS-related fields and features offer the required flexibility to explicitly set the value of some bits in the packet header from the Command Line Interface (CLI). In case a Service Provider only uses the three most significant bits of the DSCP field (corresponding to IP Precedence) for QoS classification and queuing, it is possible to use the two less significant bits of the DSCP field (bit 0 and bit 1) to implement the method without affecting QoS policies. One of the two bits (bit 0) could be used to identify flows subject to traffic monitoring (set to 1 if the flow is under monitoring, otherwise it is set to 0), while the second (bit 1) can be used for coloring the traffic (switching between values 0 and 1, corresponding to color A and B) and creating the blocks.



In practice, coloring the traffic using the DSCP field can be implemented by configuring on the router output interface an access list that intercepts the flow(s) to be monitored and applies to them a policy that sets the DSCP field accordingly. Since traffic coloring has to be switched between the two values over time, the policy needs to be modified periodically: an automatic script can be used to perform this task on the basis of a fixed timer. In Telecom Italia's implementation this timer is set to 5 minutes: this value showed to be a good compromise between measurement frequency and stability of the measurement (i.e. possibility to collect all the measures referring to the same block).

#### 4.2. Counting the packets

Assuming that the coloring of the packets is performed only by the source node, the nodes between source and destination (included) have to count the colored packets that they receive and forward: this operation can be enabled on every router along the path or only on a subset, depending on which network segment is being monitored (a single link, a particular metro area, the backbone, the whole path).

Since the color switches periodically between two values, two counters (one for each value) are needed: one counter for packets with color A and one counter for packets with color B. For each flow (or group of flows) being monitored and for every interface where the monitoring is active, a couple of counters is needed. For example, in order to monitor separately 3 flows on a router with 4 interfaces involved, 24 counters are needed (2 counters for each of the 3 flows on each of the 4 interfaces). If traffic is colored using the DSCP field, as in Telecom Italia's implementation, an access-list that matches specific DSCP values can be used to count the packets of the flow(s) being monitored.

In case of link-based measurements the behavior is similar except that coloring and counting operations are performed on a link by link basis at each endpoint of the link.

Another important aspect to take into consideration is when to read the counters: in order to count the exact number of packets of a block the routers must perform this operation when that block has ended: in other words, the counter for color A must be read when the current block has color B, in order to be sure that the value of the counter is stable. This task can be accomplished in two ways. The general approach suggests to read the counters periodically, many times during a block duration, and to compare these successive readings: when the counter stops incrementing means that the current block has ended and its value can be elaborated safely. Alternatively, if the coloring operation is performed on the basis of

a fixed timer, it is possible to configure the reading of the counters according to that timer: for example, if each block is 5 minutes long, reading the counter for color A every 5 minute in the middle of the subsequent block (with color B) is a safe choice. A sufficient margin should be considered between the end of a block and the reading of the counter, in order to take into account any out-of-order packets. The choice of a 5 minutes timer for color switching was also suggested by these considerations

#### 4.3. Collecting data and calculating packet loss

The nodes enabled to perform performance monitoring collect the value of the counters, but they are not able to directly use this information to measure packet loss, because they only have their own samples. For this reason, an external Network Management System (NMS) is required to collect and elaborate data and to perform packet loss calculation. The NMS compares the values of counters from different nodes and can calculate if some packets were lost (even a single packet) and also where packets were lost.

The value of the counters needs to be transmitted to the NMS as soon as it has been read. This can be accomplished by using SNMP or FTP and can be done in Push Mode or Polling Mode. In the first case, each router periodically sends the information to the NMS, in the latter case it is the NMS that periodically polls routers to collect information. In any case, the NMS has to collect all the relevant values from all the routers within one cycle of the timer (5 minutes).

#### 5. Compliance with RFC6390 guidelines

RFC6390 [RFC6390] defines a framework and a process for developing Performance Metrics for protocols above and below the IP layer (such as IP-based applications that operate over reliable or datagram transport protocols).

This document doesn't aim to propose a new Performance Metric but a new method of measurement for a few Performance Metrics that have already been standardized. Nevertheless, it's worth applying RFC6390 guidelines to the present document, in order to provide a more complete and coherent description of the proposed method. We used a subset of the Performance Metric Definition template defined by RFC6390.

- o Metric name and description: as already stated, this document doesn't propose any new Performance Metric. On the contrary, it describes a novel method for measuring packet loss[RFC2680]. The same concept, with small differences, can also be used to measure

delay[RFC2679], and jitter[RFC3393]. The document mainly describes the applicability to packet loss measurement.

- o Method of Measurement or Calculation: according to the method described in the previous sections, the number of packets lost is calculated by subtracting the value of the counter on the source node from the value of the counter on the destination node. Both counters must refer to the same color. The calculation is performed when the value of the counters is in a steady state.
- o Units of Measurement: the method calculates and reports the exact number of packets sent by the source node and not received by the destination node.
- o Measurement Points: the measurement can be performed between adjacent nodes, on a per-link basis, or along a multi-hop path, provided that the traffic under measurement follows that path. In case of a multi-hop path, the measurements can be performed both end-to-end and hop-by-hop.
- o Measurement Timing: the method have a constraint on the frequency of measurements. In order to perform a measure, the counter must be in a steady state: this happens when the traffic is being colored with the alternate color; in the current implementation the time interval is set to 5 minutes.
- o Implementation: the current implementation of the method uses two encodings of the DSCP field to color the packets; this enables the use of policy configurations on the router to color the packets and accordingly configure the counter for each color. The path followed by traffic being measured should be known in advance in order to configure the counters along the path and be able to compare the correct values.
- o Use and Applications: the method can be used to measure packet loss with high precision (i.e.  $10\exp(-7)$ ) on live traffic; moreover, by combining end-to-end and per-link measurements, the method is useful to pinpoint the single link that is experiencing loss events.
- o Reporting Model: the value of the counters has to be sent to a centralized management system that perform the calculations; such samples must contain a reference to the time interval they refer to, so that the management system can perform the correct correlation; the samples have to be sent while the corresponding counter is in a steady state (within a time interval), otherwise the value of the sample should be stored locally.

- o Dependencies: the values of the counters have to be correlated to the time interval they refer to; moreover, as far the current implementation is based on DSCP values, there are significant dependencies on the usage of the DSCP field: it must be possible to rely on unused DSCP values without affecting QoS-related configuration and behavior; moreover, the intermediate nodes must not change the value of the DSCP field not to alter the measurement.
- o Organization of Results: the method of measurement produces singletons
- o Parameters: currently, the main parameter of the method is the time interval used to alternate the colors and read the counters.

## 6. Security Considerations

This document specifies a method to perform measurements in the context of a Service Provider's network and has not been developed to conduct Internet measurements, so it does not directly affect Internet security nor applications which run on the Internet. However, implementation of this method must be mindful of security and privacy concerns.

There are two types of security concerns: potential harm caused by the measurements and potential harm to the measurements. For what concerns the first point, the measurements described in this document are passive, so there are no packets injected into the network causing potential harm to the network itself and to data traffic. Nevertheless, the method implies modifications on the fly to the IP header of data packets: this must be performed in a way that doesn't alter the quality of service experienced by packets subject to measurements and that preserve stability and performance of routers doing the measurements. The measurements themselves could be harmed by routers altering the coloring of the packets, or by an attacker injecting artificial traffic. Authentication techniques, such as digital signatures, may be used where appropriate to guard against injected traffic attacks.

The privacy concerns of network measurement are limited because the method only relies on information contained in the IP header without any release of user data.

## 7. Conclusions

The advantages of the method described in this document are:

- o easy implementation: it can be implemented using features already available on major routing platforms;
- o low computational effort: the additional load on processing is negligible;
- o accurate packet loss measurement: single packet loss granularity is achieved with a passive measurement;
- o potential applicability to any kind of packet/frame -based traffic: Ethernet, IP, MPLS, etc., both unicast and multicast;
- o robustness: the method can tolerate out of order packets and it's not based on "special" packets whose loss could have a negative impact;
- o no interoperability issues: the features required to implement the method are available on all current routing platforms.

The method doesn't raise any specific need for standardization, but it could be further improved by means of some extension to existing protocols. Specifically, the use of DiffServ bits for coloring the packets could not be a viable solution in some cases: a standard method to color the packets for this specific application could be beneficial.

## 8. IANA Considerations

There are no IANA actions required.

## 9. Acknowledgements

The authors would like to thank Domenico Laforgia, Daniele Accetta and Mario Bianchetti for their contribution to the definition and the implementation of the method.

## 10. References

### 10.1. Normative References

- [RFC2679] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way Delay Metric for IPPM", RFC 2679, September 1999.
- [RFC2680] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way Packet Loss Metric for IPPM", RFC 2680, September 1999.

- [RFC3393] Demichelis, C. and P. Chimento, "IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)", RFC 3393, November 2002.

## 10.2. Informative References

- [I-D.ietf-opsawg-oam-overview]  
Mizrahi, T., Sprecher, N., Bellagamba, E., and Y. Weingarten, "An Overview of Operations, Administration, and Maintenance (OAM) Tools", draft-ietf-opsawg-oam-overview-13 (work in progress), January 2014.
- [RFC6374] Frost, D. and S. Bryant, "Packet Loss and Delay Measurement for MPLS Networks", RFC 6374, September 2011.
- [RFC6390] Clark, A. and B. Claise, "Guidelines for Considering New Performance Metric Development", BCP 170, RFC 6390, October 2011.

## Authors' Addresses

Alessandro Capello  
Telecom Italia  
Via Reiss Romoli, 274  
Torino 10148  
Italy

Email: [alessandro.capello@telecomitalia.it](mailto:alessandro.capello@telecomitalia.it)

Mauro Cociglio  
Telecom Italia  
Via Reiss Romoli, 274  
Torino 10148  
Italy

Email: [mauro.cociglio@telecomitalia.it](mailto:mauro.cociglio@telecomitalia.it)

Luca Castaldelli  
Telecom Italia  
Via Reiss Romoli, 274  
Torino 10148  
Italy

Email: [luca.castaldelli@telecomitalia.it](mailto:luca.castaldelli@telecomitalia.it)

Alberto Tempia Bonda

Email: [alberto.tempia@gmail.com](mailto:alberto.tempia@gmail.com)

Operations Area Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: January 7, 2016

S. Winter  
RESTENA  
July 06, 2015

A Configuration File Format for Extensible Authentication Protocol (EAP)  
Deployments  
draft-winter-opsawg-eap-metadata-02

Abstract

This document specifies a YANG module and derived XML and JSON file formats for transferring configuration information of deployments of the Extensible Authentication Protocol (EAP). Such configuration files are meant to be discovered, consumed and used by EAP supplicant software to achieve secure and automatic EAP configuration on the consuming device.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 7, 2016.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.



## Table of Contents

1. Introduction . . . . .	2
1.1. Problem Statement . . . . .	2
1.2. Other Approaches . . . . .	4
1.3. Requirements Language . . . . .	4
1.4. Terminology . . . . .	4
2. YANG module for EAP Metadata . . . . .	4
2.1. Location of the YANG module and derived XML Schema . . .	4
2.2. Description of YANG Module Elements . . . . .	4
2.2.1. Overall structure . . . . .	4
2.2.2. The 'AuthenticationMethods' container . . . . .	5
2.2.3. The 'ProviderInfo' container . . . . .	9
2.3. Internationalisation / Multi-language support . . . . .	10
3. Derivation of formats from YANG source . . . . .	11
4. Issuer Authentication, Integrity Protection and Encryption of EAP Metadata configuration files . . . . .	11
5. XML Farget Format: File Discovery . . . . .	11
5.1. By MIME-Type: application/eap-config-xml . . . . .	11
5.2. By filename extension: .eap-config-xml . . . . .	11
5.3. By network location: SCAD . . . . .	12
6. Design Decisions . . . . .	12
6.1. Why YANG and not directly XML, JSON or \$FOO? . . . . .	12
6.2. Shallow vs. Deep definition of EAP method properties . .	12
6.3. EAP tunneling inside EAP tunnels . . . . .	12
6.4. Placement of 'OuterIdentity' inside 'AuthenticationMethod' . . . . .	12
7. Implementation Status . . . . .	12
8. Security Considerations . . . . .	15
9. IANA Considerations . . . . .	15
10. Contributors . . . . .	16
11. References . . . . .	16
11.1. Normative References . . . . .	16
11.2. Informative References . . . . .	16
Appendix A. Appendix A: MIME Type Registration Template . . . .	18

## 1. Introduction

## 1.1. Problem Statement

The IETF has produced the Extensible Authentication Protocol (EAP, [RFC3748] and numerous EAP methods (for example EAP-TTLS [RFC5281], EAP-TLS [RFC5216] and EAP-pwd [RFC5931]); the methods have many properties which need to be setup on the EAP server and matched as configuration items on the EAP peer for a secure EAP deployment.

Setting up these configuration items is comparatively easy if the end-user devices which implement the EAP peer functionality are under

central administrative control, e.g. in closed enterprise environments. Group policies or device provisioning by the IT department can push the settings to user devices.

In other environments, for example "BYOD" scenarios where users bring their own devices which are not under enterprise control, or in EAP-based WISP environments (see e.g. [HS20] and [I-D.wierenga-ietf-eduroam]) where it is not desired neither for the ISP nor for his user that the device control is in the ISPs hands, configuration of EAP is significantly harder as it has to be done by potentially very non-technical end users.

Correct configuration of all EAP deployment parameters is required to make the resulting authentications

- o functional (i.e. the end user can authenticate to an EAP server at all)
- o secure (i.e. the end user device can unambiguously authenticate the EAP server prior to releasing any sensitive client-side credentials)
- o privacy-preserving (i.e. the end user is able to conceal his username from the EAP authenticator)

It would be desirable to be able to convey the EAP configuration information of a deployment in a machine parseable way to the end-user device, so that all the gory details need not be known/understood by the user. Instead, the EAP peer software on the device could consume the configuration information and set up all EAP authentication details automatically.

However, there is currently no standard way of communicating configuration parameters about an EAP setup to the EAP peer.

This specification defines such file formats for EAP configuration metadata. The source definition is a YANG module which allows for automatic derivation of XML and JSON formats.

The specification allows for unique identification of an EAP identity provider by scoping it into a namespace and giving it a unique name inside that namespace. Using this unique identification, other configuration files (which e.g. detail the wireless media properties of an Enterprise Wi-Fi setup) can then refer to this particular instance of EAP identity information as authentication source. The contents of the EAP configuration file may also be an embedded part of those other configuration files.

## 1.2. Other Approaches

Device manufacturers sometimes have developed their own proprietary configuration formats, examples include Apple's "mobileconfig" (MIME type application/x-apple-aspen-config), Microsoft's XML schemata for EAP methods for use with the command-line "netsh" tool, or Intel's "PRO/Set Wireless" binary configuration files. The multitude of proprietary file formats and their different levels of richness in expression of EAP details create a very heterogenous and non-interoperable landscape.

New devices which would like to benefit from machine-parseable EAP configuration currently either have to choose to follow a competitor's approach and use that competitor's file format or have to develop their own. This situation is very unsatisfactory.

## 1.3. Requirements Language

In this document, several words are used to signify the requirements of the specification. 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]

## 1.4. Terminology

## 2. YANG module for EAP Metadata

### 2.1. Location of the YANG module and derived XML Schema

The schema files are currently hosted on this location:

- o YANG module: <https://www.suplicants.net/site/standardisation/eap-metadata-02.yang>
- o XML Schema: <https://www.suplicants.net/site/standardisation/eap-metadata-02.xml>

### 2.2. Description of YANG Module Elements

#### 2.2.1. Overall structure

The root element is the container 'EAPIdentityProviderList', which contains a list of 'EAPIdentityProvider' elements; these carry the actual EAP configuration information for this identity provider. In most practical applications, the 'EAPIdentityProviderList' will contain only a single element; a longer list can be used for metadata

transfers between systems or to allow users to select from a set of providers in one file.

The global uniqueness of each 'EAPIdentityProvider' is ensured by the combination of the two leafs 'NameIDFormat' which provides a namespace identifier, and 'ID' which specifies the unique name inside the namespace. The other leafs and containers in the 'EAPIdentityProvider' list are:

- o zero or one 'ValidUntil' date-and-time timestamp with an indication of possible expiry of the information in the configuration file. EAP peers importing the configuration file can use this information for example to re-assess whether the account is still valid (e.g. if the ValidUntil timestamp has passed, and authentication attempts consistently fail, the supplicant should consider the information stale and ask the user to verify his access authorisation with the EAP identity provider)
- o exactly one 'AuthenticationMethods' container with a list of EAP methods which the EAPIdentityProvider supports. This container is described in more detail in section Section 2.2.2
- o zero or one 'ProviderInfo' container can provide additional information about the EAPIdentityProvider, e.g. a logo to allow visual identification of the provider to the user in a user interface, or Acceptable Use Policies pertaining to the use of this EAP identity. This element is described in more detail in section Section 2.2.3

#### 2.2.2. The 'AuthenticationMethods' container

'AuthenticationMethods' contains a sequence of 'AuthenticationMethod' groupings. Each such grouping specifies the properties of one supported authentication method of an EAPIdentityProvider. The content of this grouping is enumerated in section Section 2.2.2.1 The set of configuration parameters specified in the grouping depends on the particular EAP method to be configured.

For instance, EAP-PWD [RFC5931] does not require any server certificate parameters; EAP-FAST and TEAP are the only ones making use of Protected Access Credential (PAC) provisioning. On the other hand, properties such as outer ("anonymous") identity or the need for a trusted root Certification Authority are common to several EAP methods. The server- and client-side credential types of EAP methods are defined as a flat list of elements to choose from (see 'ServerSideCredential' and 'ClientSideCredential' below); see section Section 6.2 for a rationale.

Where the sequence of 'AuthenticationMethod' groupings contains more than one element, the order of appearance in the file indicates the server operator's preference for the supported EAP types; occurrences earlier in the file indicate a more preferred authentication method.

When a consuming device receives multiple 'AuthenticationMethod' groupings inside 'AuthenticationMethods', it should attempt to install more preferred methods first. During interactive provisioning of EAP properties, if the configuration information for a preferred method is insufficient (e.g. the 'AuthenticationMethod' is EAP-TLS, but the configuration file does not contain the client certificate/private key and the device's credential store is not pre-loaded with the client's certificate), the device should query whether this more preferred method should be used (requiring the user to supplement the missing data) or whether a less-preferred method should be configured instead. In non-interactive provisioning scenarios, all methods should be tried non-interactively in order until one method can be installed; if no method can be installed in a fully automated way, provisioning is aborted.

#### 2.2.2.1. Authentication Method Properties

The 'AuthenticationMethod' grouping contains

- o exactly one 'EAPMethod' leaf, which is an enumerated integer of the EAP method identifier as assigned by IANA (typedef eap-method)
- o zero or one container 'ServerSideCredential' which defines means to authenticate the EAP server to the EAP peer (for a list of the elements comprising this container, see section Section 2.2.2.2)
- o zero or one container 'ClientSideCredential' which defines means to authenticate the EAP peer to the EAP server (for a list of the elements comprising this container, see section Section 2.2.2.3)
- o zero or more 'InnerAuthenticationMethod' lists. Occurrence of this list indicates that a tunneled EAP method is in use, and that further server-side and/or client-side credentials are defined inside the tunnel. The presence of more than one 'InnerAuthenticationMethod' indicates that EAP Method Chaining is in use, i.e. that several inner EAP methods are to be executed in sequence inside the tunnel. The order of occurrence of the inner EAP methods defines the chaining order of the methods.

The 'InnerAuthenticationMethod' list itself contains the same 'EAPMethod', 'ServerSideCredentials' and 'ClientSideCredentials' elements as described in the preceding list, but differs in two points:

- o It can optionally contain the leaf 'NonEAPAuthMethod' (an enumerated integer of authentication methods not based on EAP) instead of 'EAPMethod' because some tunneled EAP types do not necessarily contain EAP inside the tunnel (e.g. TTLS-PAP, TEAP). The YANG definition ensures that EAPMethod and NonEAPAuthMethod are mutually exclusive in instantiations of the YANG module.
- o It can NOT contain a further 'InnerAuthenticationMethod' because establishing a secure tunnel inside an already established secure tunnel is considered a pathological case which needs not be considered. See section Section 6.3 for a rationale.

#### 2.2.2.2. The 'ServerSideCredential' container

The server-side authentication of a mutually authenticating EAP method is typically based on X.509 certificates, which requires the EAP peer to be pre-provisioned with one or more trusted root Certification Authority (CA) prior to authenticating. A server is uniquely identified by presenting a certificate which is signed by these trusted CAs, and by the EAP peer verifying that the name of the server matches the expected one. Consequently, a (set of) CAs and a (set of) server names make up the ServerSideCredentials block.

Note that different EAP methods use different terminology when referring to trusted CA roots, server certificates, and server name identification. They also differ or have inherent ambiguity in their interpretation on where to extract the server name from (e.g. is the server name the CN part of the DistinguishedName, or is the server name one of the subjectAltName:DNS entries; what to do if there is a mismatch?). This specification introduces one single element for CA trust roots and naming; these notions map into the naming of the particular EAP methods very naturally. This specification can not remove the CN vs. sAN:DNS ambiguity in many EAP methods.

- o zero or more 'CA' lists: a Certification Authority which is trusted to sign the expected server certificate. The set of 'CA' occurrences SHOULD contain self-signed root certificates to establish trust, and MAY contain additional intermediate CA certificates which ultimately root in these self-signed root CAs. A configuration file can, but SHOULD NOT include only an intermediate CA certificate (i.e. without also including the corresponding self-signed root) because trusting only an intermediate CA without being able to verify to a self-signed root is an unsupported notion in many EAP peers.
- o zero or more 'ServerID' leafs: these leafs contain the expected server names in incoming X.509 EAP server certificates. For EAP methods not using X.509 certificates for their mutual

authentication, these elements contain other string-based handles which identify the server (Example: EAP-pwd).

#### 2.2.2.3. The 'ClientSideCredential' container

There is a variety of means to identify the EAP peer to the EAP server. EAP methods use a subset of these criteria. As with server-side credentials, the terminology for the credential type may differ slightly between EAP types. The naming convention in this specification maps nicely into the method-specific terminology. Not all the criteria make sense in all contexts; for EAP methods which do not support a criterion, configuration files SHOULD NOT contain the corresponding elements, and consumers of the file MUST ignore these elements.

Specifying any one of these elements is optional and they can occur at most once. Consumers of configuration files MUST be able to fall back to user-interactive configuration for these parts if they are not specified (e.g. ask for the username and password for an EAP method during import of the EAP configuration data). Configuration files which contain sensitive elements such as 'Password' MUST be handled with due care after the import on the device (e.g. ensure minimal file permissions, or delete the source file after installing). See also the leaf 'allow-save' below.

The leaf 'allow-save' specifies whether consumers should allow the user to save the credential persistently; if it is set to false, sensitive parts of the client-side credentials MUST NOT be persistently saved on the device. See also section Section 4 for transport security considerations.

Leaf 'AnonymousIdentity' is typically used on the outside of a tunneled EAP method and allows to specify which user identity should be used outside the tunnel. This string is not used for actual user authentication, but may contain routing hints to send the request to the right EAP server.

'UserName' contains the actual username to be used for user authentication. For tunneled EAP methods, this element SHOULD only occur in the 'InnerAuthenticationMethod's 'ClientSideCredentials' - if differing outer identities are not desired in the deployment, the 'OuterIdentity' element should be populated for the 'AuthenticationMethod' element but be populated with the actual username then.

The 'ClientCertificate' container holds a X.509 certificate and private key; if the key is protected, the 'Passphrase' leaf MAY be used to indicate the passphrase, see below

'Passphrase' contains the passphrase needed to unlock a cryptographic credential internally on the device (i.e. it is not used itself for the actual authentication during the EAP conversation)

'Password' contains the user's password, or an otherwise secret string which the user needs to authenticate to the EAP server

'PAC' contains the Protected Access Credential, typically used in EAP-FAST and TEAP.

'ProvisionPAC' is a boolean which indicates whether a PAC should be provisioned on the first connection. Note that this specification allows to use 'ProvisionPAC' without a CA nor ServerID in 'ServerSideCredential'. While this allows the operation mode of "Anonymous PAC Provisioning" as used in many field deployments of EAP-FAST (and is thus supported here), due to the known security vulnerabilities of anonymous PAC provisioning, this combination SHOULD NOT be used.

### 2.2.3. The 'ProviderInfo' container

This specification needs to consider that user interaction during the installation time may be required; the user at the very least must be empowered to decide whether the configuration file was issued by a provider he has an account with; the provider may have hints for the user (e.g. which password to use for the login), or may want to display links to helpdesk pages in case the user has problems with the setup or use of his identity.

The 'ProviderInfo' container allows to specify a range of potentially useful information for display to the user (some of which is relevant only during installation time, other pieces of information could be retained by the EAP peer implementation and displayed e.g. in case of failed authentication):

- o 'DisplayName' specifies a user-friendly name for the EAP Identity Provider. Consumers of this specification should be aware that this is simple text, and self-asserted by the producer of the configuration file. If more authoritative information about the issuer is available (e.g. if the file is signed with S/MIME and carries an Organisation name (O attribute) in the signing certificate) then the more authoritative information should be displayed with more prominence than the self-asserted one.
- o 'Description' specifies a generic descriptive text which should be displayed to the user prior to the installation of the configuration data.



- o 'ProviderLocation' specifies the approximate geographic location(s) of the EAP Identity Provider and/or his Points of Presence. This can be useful if the configuration file contains multiple 'EAPIdentityProvider' elements; the user device can then make an informed guess which of the Identity Providers could be a good match to suggest to the user.
- o 'ProviderLogo' specifies the logo of the EAP Identity Provider. The same self-assertion considerations as for 'DisplayName' above apply.
- o 'TermsOfUse' contains terms of use to be displayed to and acknowledged by the user prior to the installation of the configuration on the user's system
- o 'Helpdesk' is a container with three possible sub-elements: 'EmailAddress', 'WebAddress' and 'Phone', all of which can be displayed to the user and possibly retained for future debugging hints.

### 2.3. Internationalisation / Multi-language support

Some elements in this specification contain text to be displayed in User Interfaces; depending on the user's language preferences, it would be desirable to present the information in a local language. Other elements contain contact information, and those contact points may only be able to handle requests in a number of languages; it may be desirable to present only contact points to the user which are compatible with his language capabilities.

All elements which either contain localisable text, or which point to external resources in localised languages, use the grouping 'localized-non-interactive' or 'localized-interactive'. These groupings can occur more than once in the specification, which enables an iteration of all applicable languages. If the grouping is omitted or its 'lang' leaf is set to "C", the instance of the element is considered a default choice which is to be displayed if no other language is a better match.

If the entire file content consistently uses only one language set, e.g. all the elements are to be treated as "default" choices, the language can also be set for the entire 'EAPIdentityProvider' element in its own 'lang-tag' leaf.

### 3. Derivation of formats from YANG source

The utility 'pyang' is used to derive XML Schema (XSD) from the YANG source. The Schema for this Internet-Draft was generated with pyang 1.4.1.

### 4. Issuer Authentication, Integrity Protection and Encryption of EAP Metadata configuration files

S/MIME or underlying transport security. Nuff said :-)

### 5. XML Farget Format: File Discovery

#### 5.1. By MIME-Type: application/eap-config-xml

For transports where the categorisation of file types via MIME types is possible (e.g. HTTP, E-Mail), this document assigns the MIME type

application/eap-config-xml

Edge devices can associate this MIME type to incoming files on such transports, and register the application which can consume the EAP Metadata in XML format as the default handler for this file type. By doing so, for example a single click or tap on a link to the file in the device's browser will invoke the configuration process.

This method of discovery is analogous to the Apple "mobileconfig" discovery on recent versions of Mac OS and iOS.

#### 5.2. By filename extension: .eap-config-xml

In situations where file types can not be determined by MIME type meta-information (e.g. when the file gets stored on a local filesystem), this document RECOMMENDs that EAP Metadata in XML format files be stored with the extension

.eap-config-xml

to identify the file as containing EAP Metadata configuration information in XML format. Edge devices can register the application which can consume the EAP Metadata with this file extension. By doing so, for example a single click or tap on the filename in the device's User Interface will invoke the configuration process.

### 5.3. By network location: SCAD

## 6. Design Decisions

### 6.1. Why YANG and not directly XML, JSON or \$FOO?

XML is a popular choice for EAP configurations: Microsoft's "netsh" files, Apple's "mobileconfig" files, the Wi-Fi Alliance's "PerProviderSubscription Managed Object", and other vendor/SDO definitions are all using XML.

JSON file formats for EAP configuration exist as well; most notable are Google's most recent efforts for their Chromebook Operating system.

YANG has a very rich feature set, and can codify restrictions on which element is allowed when in a much more fine-grained way than XML Schema could. Since YANG modules can be converted to XML Schema and be instantiated as XML or JSON, they can serve as an abstract notion of EAP configuration which can be deployed on consumer devices in either of those two more popular formats as needed by the device in question.

### 6.2. Shallow vs. Deep definition of EAP method properties

### 6.3. EAP tunneling inside EAP tunnels

### 6.4. Placement of 'OuterIdentity' inside 'AuthenticationMethod'

## 7. Implementation Status

RFC Editor Note: Please remove this section and the reference to [RFC6982] prior to publication.

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in [RFC6982]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to [RFC6982], "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

All of the implementations listed below interoperate from producer- to consumer-side of the EAP metadata specification.

#### Producers of the configuration files

- o eduroam Configuration Assistant Tool

Organisation: Nicolaus Copernicus University, Torun, Poland

Implementation Name: eduroam Configuration Assistant Tool

This existing tool already produces EAP configuration files in various proprietary formats for hundreds of EAP Identity Providers. A module which produces configuration files in the XML variant as specified in an earlier revision of this draft (-00) is in production deployment.

Link to production version: <https://cat.eduroam.org>

Maturity: production

Coverage: entire specification; XML structure aligns with version -00 of this draft

Licensing: freely distributable with acknowledgement (BSD style)

Implementation experience: given that the specification is XML, it is easy to produce a configuration file with common XML libraries. The CAT Framework is written in PHP, which provides ample procedures to produce well-formed XML.

Contact Information: Tomasz Wolniewicz (see Section 10); the CAT software homepage at <http://forge.geant.net/CAT/>

#### Consumers of the configuration files

- o Android

Organisation: Swansea University, Swansea, Wales, U.K.

Implementation Name: eduroam CAT app

An Android app, compatible with API level 18 of Android (i.e. version 4.3 and above); the app consumes the -00 revision of this specification. The information in the config files is used to push settings to the SSID 'eduroam' (hard-coded) via the WifiEnterpriseConfig API. The app is in production deployment, with a 4-four digit amount of downloads one month after launch.

Link to production version: <https://play.google.com/store/apps/details?id=uk.ac.swansea.eduroamcat>

Maturity: production

Coverage: entire specification; XML structure aligns with version -00 of this draft

Licensing: Apache 2.0

Implementation experience: parsing XML is rather straightforward. The ability to verify signatures on XML files (S/MIME vs. XMLDSIG as discussed in Section 4) remains unclear at this point.

Contact Information: eduroam CAT Play Store app contact address ( [playstore@eduroam.org](mailto:playstore@eduroam.org) )

o Windows

Organisation: Amebis, d.o.o.i, Kamnik, Slovenia

Implementation Name: ArnesLink

A Windows supplicant/Enterprise WiFi installer/debugging assistant. The application consumes the -02 revision of this specification. The information from the XML variant of this specification is embedded in a larger XML file. The additional parts of the overall configuration file include information regarding the SSID to configure and other useful, but not EAP-specific information. The complete set of information is used to push settings into the Windows Wi-Fi configuration via the 'netsh' tool. The app is in production deployment.

Link to production version: <http://ftp.arnes.si/software/eduroam/ArnesLink/>

Maturity: production

Coverage: entire specification; XML structure aligns with version -02 of this draft

Licensing: GPL

Implementation experience: parsing XML is rather straightforward. For Wi-Fi configuration use, the lack of 802.11 specific details in the config file is an issue.

Contact Information: info@amebis.si

- o Linux: the authors of this specification are currently developing an application for UNIX-like operating systems which configure enterprise networks via the NetworkManager daemon; the application can consume the file format as defined in this draft specification (XML format) and configure the settings via Networkmanager's D-BUS interface.

## 8. Security Considerations

## 9. IANA Considerations

IANA is requested to allocate the MIME type "application/eap-config-xml" in the MIME Media Types / application registry (see section Section 5.1). The allocation should contain the following values:

- o Name: eap-config-xml
- o Template: see Appendix A (RFC editor note: remove this appendix prior to publication; replace this line with the URL to the application as posted online)
- o Reference: RFCabcd (RFC editor note: replace with the RFC number of this document)

IANA is requested to allocate the location "TBD" in the "well-known URIs" registry. The allocation should contain the following values:

- o URI Suffix: TBD
- o Change Controller: IETF
- o Reference: RFCabcd (RFC editor note: replace with the RFC number of this document)
- o Related Information: none

IANA is requested to register the XML namespace "urn:ietf:params:xml:ns:eap-config-xml" in the "IETF XML Registry / ns". The allocation should contain the following values:

- o ID: eap-config-xml
- o URI: urn:ietf:params:xml:ns:eap-config-xml
- o Filename: <https://www.iana.org/assignments/xml-registry/ns/eap-config-xml.txt> (to be created by IANA)
- o Reference: RFCabcd (RFC editor note: replace with the RFC number of this document)

IANA is requested to register the XML schema "urn:ietf:params:xml:schema:eap-config-xml" in the "IETF XML Registry / schema". The allocation should contain the following values:

- o ID: eap-config-xml
- o URI: urn:ietf:params:xml:schema:eap-config-xml
- o Filename: <https://www.iana.org/assignments/xml-registry/schema/eap-config-xml.xsd> (to be created by IANA; current XSD file is linked to in section Section 2.1)
- o Reference: RFCabcd (RFC editor note: replace with the RFC number of this document)

## 10. Contributors

Tomasz Wolniewicz of Nicolaus Copernicus University in Torun, Poland, provided significant input into this specification.

## 11. References

### 11.1. Normative References

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

### 11.2. Informative References

- [RFC3748] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowitz, "Extensible Authentication Protocol (EAP)", RFC 3748, June 2004.

- [RFC5216] Simon, D., Aboba, B., and R. Hurst, "The EAP-TLS Authentication Protocol", RFC 5216, March 2008.
- [RFC5281] Funk, P. and S. Blake-Wilson, "Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0)", RFC 5281, August 2008.
- [RFC5931] Harkins, D. and G. Zorn, "Extensible Authentication Protocol (EAP) Authentication Using Only a Password", RFC 5931, August 2010.
- [RFC6982] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", RFC 6982, July 2013.
- [I-D.wierenga-ietf-eduroam]  
Wierenga, K., Winter, S., and T. Wolniewicz, "The eduroam architecture for network roaming", draft-wierenga-ietf-eduroam-05 (work in progress), March 2015.
- [HS20] Wi-Fi Alliance, "Hotspot 2.0 Technical Specification", 2012, <<https://www.wi-fi.org/hotspot-20-technical-specification-v100>>.



## Appendix A. Appendix A: MIME Type Registration Template

The following values will be used for the online MIME type registration at <https://www.iana.org/form/media-types>

Your Name: Stefan Winter

Your Email Address: stefan.winter@restena.lu

Media Type Name: Application

Subtype name: (Standards tree) eap-config-xml

Required parameters: (none)

Optional parameters: (none)

Encoding Considerations: 8-Bit text

Security Considerations: This file type carries configuration information for consumer devices. It has the potential to substantially alter the consumer's device; particularly to install a new trusted Certification Authority. Applications consuming files of this type need to be cautious to explain to the end user what is being altered, so that they understand the consequences. For further explanations, see Section 8 of draft-winter-opsawg-eap-metadata. (Note to RFC Editor: replace this reference with the RFC number of this document once known)

Interoperability Considerations: The file content is XML version 1.0 or later. The encoding SHOULD be UTF-8, but implementations consuming the file SHOULD be prepared to encounter different encodings.

Published Specification: draft-winter-opsawg-eap-metadata (Note to RFC Editor: replace this reference with the RFC number of this document once known)

Applications which use this media type: files of this type are intended for consumption by software on edge devices; they consume the information therein to configure authentication parameters (EAP protocol and EAP method payload configurations) which are then applied to network or application authentication scenarios.

Fragment Identifier Considerations: files of this type are expected to be transmitted in their entirety. If a reference to a specific part of the content is to be made, XML XPath expressions

are to be used. I.e. fragment identifier formats are not expected to be used.

Restrictions on Usage: none

Provisional registration: initial submission of this form will be executed after adoption in the IETF; it will be a provisional registration. Final registration will be done after IESG review.

Additional information:

Deprecated alias types for this name: none

Magic numbers: none

File extensions: eap-config-xml

Macintosh File Type Codes: TBD

Object Identifiers or OIDs: none

Intended Usage: Common (no further provisions)

Other Information/General Comment: none

Person to contact for further information:

Name: Stefan Winter

E-Mail: stefan.winter@restena.lu

Author/Change controller: IETF

DATA

Author's Address

Stefan Winter  
Fondation RESTENA  
6, rue Richard Coudenhove-Kalergi  
Luxembourg 1359  
LUXEMBOURG

Phone: +352 424409 1  
Fax: +352 422473  
EMail: stefan.winter@restena.lu  
URI: <http://www.restena.lu>.

Operations and Management Area Working Group  
Internet-Draft  
Intended status: Informational  
Expires: January 5, 2015

Q. Wu  
M. Wexler  
Huawei  
M. Boucadair  
France Telecom  
S. Aldrin  
Huawei USA  
G. Mirsky  
Ericsson  
P. Jain  
Nuage Networks  
July 4, 2014

Problem Statement and Architecture for Transport-Independent Multiple  
Layer OAM  
draft-ww-opsawg-multi-layer-oam-02.txt

Abstract

Operations, Administration, and Maintenance (OAM) mechanisms are critical building blocks in network operations that are used for service assurance, fulfillment, or service diagnosis, troubleshooting, and repair. The current practice is that many technologies rely on their own OAM protocols that are exclusive to a given layer. There is little consolidation of OAM in either data plane or management plane nor well-documented inter-layer OAM operations. Vendors and Operators dedicate significant resources and effort through the whole OAM life-cycle each time when a new technology is (to be) introduced. This is even exacerbated when dealing with integration of OAM across multiple technologies.

This document describes the problem space and defines an architecture for the generic and integrated OAM with a focus of multi-layer and cross-layer considerations.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any

time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 5, 2015.

#### Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

#### Table of Contents

1. Introduction . . . . .	3
2. Terminology . . . . .	4
2.1. Acronyms and Abbreviations . . . . .	6
3. Problem Statement . . . . .	6
3.1. Use of Existing Protocols . . . . .	7
3.2. Strong Technology dependency . . . . .	8
3.3. Weakness of Cross-Layer OAM . . . . .	8
3.4. Lack of OAM above Layer 3 . . . . .	9
3.5. Issues of Abstraction . . . . .	9
3.6. Issue of OAM Information Gathering from Layers Covering Heterogeneous Network Technologies . . . . .	10
3.6.1. Focus on Service Function Chaining . . . . .	10
4. Architecture Overview . . . . .	11
5. Existing Work . . . . .	13
6. Architectural Consideration . . . . .	14
6.1. Basic Components . . . . .	14
6.1.1. Overlay OAM . . . . .	14
6.1.2. OAM at the top of Layer 3 . . . . .	14
6.2. OAM Functions in the Data Plane . . . . .	14
6.2.1. Continuity Check . . . . .	14
6.2.2. Connectivity Verification . . . . .	14
6.2.3. Path Discovery . . . . .	14
6.2.4. Performance Measurement . . . . .	14
6.2.5. Protection Switching Coordination . . . . .	15
6.2.6. Alarm/defect Indication . . . . .	15
6.2.7. Maintenance Commands . . . . .	15

6.3. OAM in Management Plane . . . . .	15
7. Building on Existing Protocols . . . . .	16
8. Scoping Future Work . . . . .	16
9. Manageability Considerations . . . . .	17
10. Security Considerations . . . . .	17
11. Acknowledgements . . . . .	17
12. References . . . . .	17
12.1. Normative References . . . . .	17
12.2. Informative References . . . . .	17
Authors' Addresses . . . . .	19

## 1. Introduction

Operations, Administration, and Maintenance (OAM) mechanisms being understood and used in context of RFC 6291 [RFC6291] are critical building blocks in network operations that are used for service assurance, fulfillment, or service diagnosis, troubleshooting, and repair. The key foundations of OAM and its functional roles in monitoring and diagnosing the behavior of networks have been studied at OSI layers 1, 2 and 3 since a while. As a reminder, OAM functions are used in many management applications for various objectives such as (i) failure detection, (ii) reporting the defect/ failure information, (iii) defect/failure localization, (iv) performance monitoring, and (v) service recovery.

The current practice that consists in enabling OAM techniques for each layer has shown its limits; this is a need for cross-layer and inter-layer OAM considerations [RFC7276]. This need for inter-layer OAM is motivated by the need to achieve: network optimization, efficient enforcement of TE (Traffic Engineering) techniques including ensuring path diversity at distinct layers or computing completely disjoint paths at several layers, fine-grain tweaking, ease of root cause analysis, ability to maintain a network-wise visibility in addition to layer-specific one, etc.

It is worth to mention also that there are two restrictions for multi-layer structure as discussed in [RFC7276]:

- o Each layer has its own OAM protocol, OAM should not cross layer boundaries.
- o Each layer OAM used at different level of hierarchy in the network.

Moreover, there is little consolidation of OAM in either data plane or management plane. Vendors and operators dedicate a lot resources and effort through the whole OAM life-cycle each time a new technology is (to be) introduced. Integration of OAM across multiple

technologies in either data plane or management plane is extremely difficult to achieve.

When operating networks with more than one technology, maintenance and troubleshooting are achieved per technology and per layer, operation process can be very cumbersome since OAM is not defined to cross layer boundaries. Another challenge is presented by use of different technologies and corresponding OAM on the same layer of adjacent network domains. Interworking between different OAM often not defined and are left to proprietary solutions. In many cases when keeping network complexity down and simplifying OAM is needed, it is desirable to have a generic and integrated OAM to cover heterogeneous networking technologies.

This document defines the problem space and describes an architecture for the generic and integrated OAM in the multi-layer and multi-domain networks. In particular, it outlines the problems encountered with existing OAM protocols and their impact on introduction of new technologies (see Section 3).

This document covers the following:

- o Data plane OAM consolidation by looking at the common active OAM functions (including, Connectivity Verification (CV), Path Verification and Continuity Checks (CC), Path Discovery, Performance Measurement) necessary to monitor and diagnose a network;
- o Management plane consolidation by interacting with data plane OAM and abstracting OAM information common to different layer via uniformed interface.

## 2. Terminology

This document defines the following terms:

Transport Independent Multi-Layer OAM:

In an multi-layer network, transport independent OAM is OAM that can be deployed independent of media, data protocols, and routing protocols It denotes the ability to exchange OAM information across layers and domains between nodes along forwarding path, and gather OAM information that are common to different layers and expose it to the management application through a unified interface. These aspects are not specific to a given transport technology.

OAM function:

Refers to the atomic building blocks of OAM; an OAM function defines an OAM capability (See section 2.2.3 of [RFC7276]).

OAM protocol:

Refers to a protocol used for implementing one or more OAM functions (See section 2.2.3 of [RFC7276]).

OAM tool:

Denotes a specific means of applying one or more OAM functions. An OAM protocol can be an OAM tool. An OAM tool can use a set of OAM protocols or a set of protocols that are not strictly OAM related (See section 2.2.3 of [RFC7276]).

OAM packet:

Refers to a packet generated at Maintenance Point using an OAM protocol. An OAM packet, which carries OAM information, is usually forwarded through the same route/path as the data traffic and receive the same (forwarding) treatment.

Maintenance Domain (MD):

Refers to the part of a network where OAM function is performed (initiated).

Maintenance Point (MP):

Is a generic functional entity that is associated with a particular MD, defined at a specific layer of a network and can initiate and/or react to OAM packets.

Maintenance Endpoint (MEP):

Is an endpoint MP that initiates OAM packets and responds to them.

Maintenance Intermediary Point(MIP):

In between MEPs, there are zero or more intermediate points, called Maintenance Intermediary Point. A Maintenance Intermediary Point (MIP) is an intermediate MP that does not generally initiate OAM packets but is able to respond to OAM packets that are destined to it.

Maintenance Association (MA):



The relationship between a set of MEPs to which maintenance and monitoring operations apply.

Network Element (NE):

Denotes a physical or virtual network device/function that connects directly to the network. NE can host MPs and provide network connectivity to one or many MPs.

## 2.1. Acronyms and Abbreviations

CC - Continuity Check

CV - Connectivity Verification

SNMP - Simple Network Management Protocol

NETCONF - Network Configuration

ETH - Ethernet

APS - Automatic Protection Switching

LT - LinkTrace

RDI - Remote Defect Indication

AIS - Alarm indication Signal

OWAMP - One Way Active Measurement Protocol

TWAMP - Two Way Active Measurement Protocol

CFM - Connectivity Fault Management

## 3. Problem Statement

OAM mechanisms are usually oriented toward a single network technology or a single layer. Each technology or layer has its best suited OAM tools. Some of them providing rich functionality rely on the capabilities of one protocol, while the others provide each function with a different protocol; In the current situation, there is little, or no re-use, of software and hardware for each OAM protocol.

Integration of OAM across multiple technologies is extremely difficult. Vendors and operators waste a lot through the whole OAM life-cycle when a new technology is introduced:

(1) Design and development: For every new protocol there is a need to invest in complete life-cycle (i.e., the design and development of data, control and management planes). In some cases, even adding a single OAM function requires the above complete life-cycle.

(2) Operation and Maintenance: There is a need to re-train operation people for almost every newly introduced technology or feature. The above causes a slow time-to-market and a waste of time and effort for any new technology and/or OAM function.

Specifically, in Service Function Chaining environment, every Service Function may operate at a different layer and may use different encapsulation and tunneling techniques. When taking into account virtualization related technologies, the number of encapsulation and tunneling options increase even more. Still, end-to-end service OAM mechanisms and information exchanges between Service Functions should be provided to operate and maintain the network as a whole. This requires a generic toolkit that can provide all necessary tools in context of multi-technology, multi-layer, physical and virtual environments.

A particular problem is how OAM information at different layer is made available to a management application for use and learnt via the unified management interface. For example, in the case of a multi-layer network, OAM information needs to be imposed to the packet and injected into the network and at last abstracted from various layers and expose them to the management application.

### 3.1. Use of Existing Protocols

OAM information resides at each layer and may currently be exchanged at each network layer in a domain by using various encapsulation technologies at the Layer 2 & Layer 3 levels. OAM information may be gathered and exported from a domain (for example, northbound) using SNMP [RFC3411] or NETCONF/YANG [RFC6241].

It is desirable that a solution to the problem described in this document does not require the implementation of a new, network-wide protocol or introduce a shim layer to carry OAM information. Instead, it would be advantageous to make use of an existing protocols or functionalities that are commonly implemented and are currently deployed in operational networks. This has many benefits in network stability, time to deployment, and operator training.

It is recognized, however, that existing protocols or functionalities are unlikely to be immediately suitable to this problem space without some protocol extensions. Extending protocols must be done with care

and with consideration for the stability of existing deployments. In extreme cases, when there is a lack of functionality, although similar mechanisms exist in other technologies, a new protocol can be preferable to a "messy" hack of an existing protocol.

### 3.2. Strong Technology dependency

OAM protocols are relying heavily on the specific network technology they are associated with. For example, ICMP, LSP Ping are using different network technologies but provide the same OAM functionality, i.e., Path Discovery. Another example is BFD, LSP Ping are using different network technologies but provide the same functionality, i.e., Continuity Verification. Figure 1 shows common OAM functionalities shared by various existing IETF OAM protocols.

	Continuity Check	Connectivity Verification	Path Discovery	Performance Measurement
ICMP	Echo(Ping)		Traceroute	-Delay -Loss rough measurement
BFD	BFD Control /Echo	BFD Control		
LSP Ping		Ping	Traceroute	- Delay - Packet Loss
IPPM				-OWAMP -TWAMP
MPLS-TP OAM	CC (use of BFD)	CV (use of BFD) or LSP Ping)	Traceroute	-Delay -Packet Loss

Figure 1: Examples of IETF OAM tools

### 3.3. Weakness of Cross-Layer OAM

Troubleshooting is cumbersome due to protocol variety and lack of multi-layer OAM. Usually OAM messages should not cross layer boundaries. Each of the service, network and transport layers

possesses its well-discernible and native OAM stream. In addition, OAM messages should not be leaked outside of a management domain within a layer, where a management domain is governed by a single business organization. When having networks with more than one technology, maintenance and troubleshooting are done per technology and layer.

These rules could in some cases ease the understanding in which technology the operation is done or fault is located. In some cases, when one layer OAM fails, it may be desirable to drop down to the another layer OAM and issue the corresponding OAM command, using the same APIs, if OAM in multiple layers can be supported. However, in most cases switching tools and layers in the same operation process is cumbersome and not serving the main idea - to find the root cause location. It would be very helpful to have a generic mechanisms that is end to end basis, allow management application interact with data plane OAM and can ping IPv4 host by an IPv6 source or having one tool to troubleshoot combined IP, MPLS, Ethernet, GRE and VXLAN network.

In Service Function Chaining environment, it is necessary to provide end-to-end OAM across certain or all entities and involving many layers. Inter-layer OAM considerations are key in an SFC context because problems may occur at the network layer or at the service chaining layer.

#### 3.4. Lack of OAM above Layer 3

The Layer 2/3 OAM protocols are quite rich in their functionality, well defined, standardized and heavily used. In the last years a lot of work was conducted to consider maintenance domains and levels in order to better handle the issues of technology re-use, smooth interoperability and interworking between domains.

The above mechanisms are not defined for the technologies above Layer 3. Therefore, in the SFC environment where a Service Function Chaining is composed by a set of Service Functions, but providing an end-to-end chain or path from a source to destination in a given order [I.D-ietf-sfc-problem-statement], no standard exists as a reference for OAM since when the service packets is steered through a set of service nodes distributed in the network, each service node may act at different layers above layer 3.

#### 3.5. Issues of Abstraction

In multi-layer network, OAM functions are enabled at different layers and various OAM information needs to be gathered from various layers. Without multi-layer OAM in place, it is hard for management applications to understand what information at different layers

stands for. One possible solution to these issues is to abstract the OAM information shared across layers, i.e., using the same tool or API to activate the OAM functions at different layers and retrieve the results.

The challenge is to abstract in a way that retains as much useful information as possible while filtering the data that is not needed to be leaked to other layers. An important part of this effort is a clear understanding of what information is actually needed.

### 3.6. Issue of OAM Information Gathering from Layers Covering Heterogeneous Network Technologies

In SFC, the service packets are steered through a set of service nodes (virtual or physical) hosting the service function distributed in the network. In the NVO3 network, the data packet may also traverse a set of overlay nodes distributed in the network. Overlay technologies or other tunneling technologies can be used to stitch these service nodes or overlay node in order to form end to end path.

When any overlay Segment or segment of service chain in the network fails to deliver user traffic, there is a need to provide a tool that would enable users to detect such failures at different layer using various encapsulation protocols and locate faults in the specific part of the network, and a mechanism to isolate these faults. It may also be desirable to test the data path before mapping user traffic to the Overlay Segment or segment of service chain. When multiple layer OAMs are used in the different parts of the network; how these layers OAM interwork at the boundary of each part of network is also a serious issue.

#### 3.6.1. Focus on Service Function Chaining

When the service packets are steered through a set of Service Nodes (virtual or physical) hosting the Service Function distributed in the network, each Service Node may work at different layer above layer 3 and may embed several SFs. When OAM mechanism is applied, it is necessary to allow OAM packets to be exchanged:

- o between Service Functions/Service Nodes and the SFC Management System,
- o between these Service Nodes,
- o between Service Functions at different layers,
- o or between Service Nodes and ingress node of the SFC-enabled domain.

When Service Functions that are part of the SFC-enabled domain do support the OAM capability (e.g., an SFC-unaware Service Function) and Service Node has OAM capability, Service Nodes may be responsible for monitoring and diagnosing and reporting service availability of these Service Functions. It is more desirable to allow Service Functions register with a Service Node. Either Service Functions report status to the Service Node or the Service Node performs liveness check of the Service Function.

In addition, some Service Functions may not have Layer 2-3 switching/routing capability and therefore are not aware of any OAM function at Layer 2-3. Also when there are no OAM functions at service Layers above layer 3, it is hard to identify the layer that can be used to gather OAM information when it comes to a fault situation or degradation of performance. For example, when a data packet is transmitted from SFC ingress node (i.e., Classifier) and traverse a set of Service Nodes that host Service Function, the data packet may be discarded either at the SFC ingress node, one specific Service Node or one specific Service Function. Also the data packet may be lost between SFC ingress and one Service Node, or between two Service Nodes, or between one Service Node and one Service Function, how to detect the fault between them and how to isolate problem to that layer?

Editor's Note: Section 3.6.1 is too specific. This text can be presented as an example to illustrate a problem not a problem per se or moved to a use case draft.

#### 4. Architecture Overview

Figure 2 shows the reference architecture for Layering OAM. This reference architecture assumes that

- o Any network element can use different technologies and corresponding OAM on the same layer at the boundary of two adjacent domains
- o Any two network element may provide service delivery at different layer
- o Management entity can manage network devices in more than one maintenance domains.

In this architecture, three layers are defined:

M1: "Data Plane layer"

M2: "Management Plane layer"

## M3: "Service Plane layer"

In the M1 layer, a typical network can be partitioned into several domains. Each domain has at least two MEPs and none or several MIPs. One domain can contain one or more maintenance associations (MAs). MEP is a maintenance functional entity that is implemented into a Network Element at the maintenance domain boundary and can send and receive OAM packets. MIP is a maintenance functional entity that is implemented into a Network Element in the maintenance domain and can forward OAM packets and respond OAM packets only when triggered by a specific OAM function (e.g., Path Discovery or Connectivity Verification). MEPs and MIPs can exist in the same maintenance domain and belong to different MAs. They can also exist at different layers and use various encapsulating protocols.

The M2 contains the interface which management entity uses to manage individual network devices. In this document, we further require management entities to use this interface as uniform interface (API and or UI) to gather OAM information from MEP and MIP in the network devices (either physical or virtual entity) and execute transactions or operations on MEP and MIP across domains, layers and vendors. Protocols that can be used to manipulate the configuration of a network device include SNMP [RFC1157], Command Line Interfaces, NETCONF [RFC6241], and other protocols.

On the M3 layer, there is a uniform interface (API and/or UI) that covers all the managed devices and can execute network-wide transactions. This layer allows applications and operators to execute configuration, monitoring and action tasks across multiple network devices, from a mix of domains, layers, vendors. Still the abstraction level is that of the network elements themselves, so whatever configuration, status, actions and notifications they provide, that is what you get here, but without having to worry about the location and the protocol to reach the device.

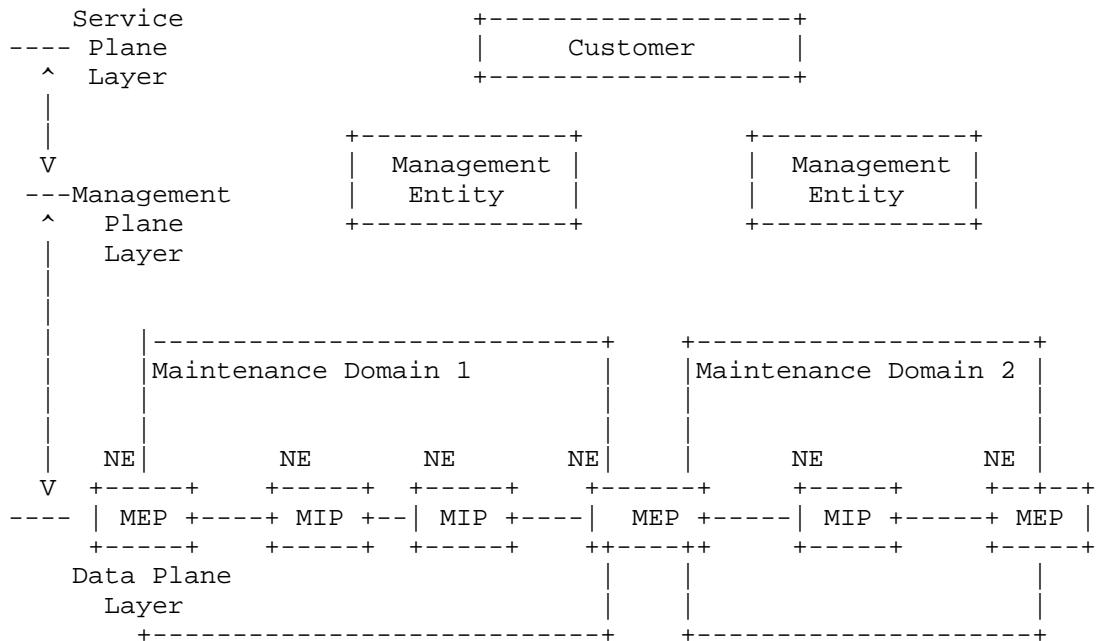


Figure 2: Architecture for Layering OAM in the management plane

An example of service-specific that depicts OAM layers can be found in [RFC4176] (L3VPN case).

## 5. Existing Work

The following discuss related IETF work and is provided for reference. This section is not exhaustive, rather it provides an overview of few initiatives focusing on the pain-points of OAM:

1. [I-D.tissa-netmod-oam] is an important work that creates a YANG unified data model for OAM that is based on IEEE CFM model. This model may be used also for IP OAM functionality. This effort is focused on the management plane of OAM and should be complemented by an accompanying data-plane and/or control-plane work. It may require also some extensions to address wider variety of functions and technologies.
2. Several contributions conducted in the past years, had tried to address new technologies using existing mechanisms. [I-D.jain-nvo3-overlay-oam] and MPLS-TP OAM documents are only examples for such efforts.



## 6. Architectural Consideration

### 6.1. Basic Components

#### 6.1.1. Overlay OAM

#### 6.1.2. OAM at the top of Layer 3

### 6.2. OAM Functions in the Data Plane

Many OAM functions may require protocol extensions or new protocol development to meet the transport requirements. In the existing OAM tools, Some of them providing rich functionality in one protocol, the other providing each function with a different protocol and each technology is developed independently.

To consolidate OAM in the data plane, the OAM in multi-layer Environment is expect to support the following common OAM functions used in OAM-related standards. These functions are used as building blocks in the data plane OAM standards described in this document.

#### 6.2.1. Continuity Check

This type of mechanisms check that the monitored layer and/or entity are alive and providing path from specific point(s) to other point(s). Some examples are IP Ping, BFD [RFC5880] and ETH CC.

#### 6.2.2. Connectivity Verification

Verifying that the actual connection is consistent with the required connection and no mis-connection occurred. Some examples are IP Ping, and ETH loopback.

#### 6.2.3. Path Discovery

Used to discover the path that specific service traverses in the network. Some examples are LSP Traceroute, IP Traceroute and ETH-LT/linktrace.

#### 6.2.4. Performance Measurement

A function that monitors the performance parameters of a network entity. Such parameters could be Delay, Delay-variation, loss, availability of services and class of services. Examples are TWAMP[RFC5357]/ OWAMP[RFC4656] and Y.1731, MPLS Loss and Delay Measurement [RFC6374].

#### 6.2.5. Protection Switching Coordination

A function that is used to signal protection switching states and commands. Examples are ETH APS messages and MPLS-TP Protection Switching Coordination OAM [RFC6378].

#### 6.2.6. Alarm/defect Indication

A function that is used to indicate that a failure occurred downstream or upstream within a connection/service. Used also to trigger fast protection or to suppress alarms. Examples are ETH AIS and ETH RDI, MPLS-TP RDI [RFC6428].

#### 6.2.7. Maintenance Commands

A function that is used to signal a maintenance state or command within a connection/service. Examples can be ETH Lockout.

### 6.3. OAM in Management Plane

Management systems play an important role in configuring or provisioning OAM functionality consistently across all devices in the network, and for automating the monitoring and troubleshooting of network faults. However OAM is not provisioned. In general, provisioning is used to configure the network to provide new services, whereas OAM is used to keep the network in a state that it can support already existing services.

As we know each layer has its own OAM protocols. OAM can be used at different levels of hierarchy in the network to form a multi-layer OAM solution [RFC7276]. To support multi-layer OAM covering various heterogeneous transport technologies, the OAM in the management needs to be consolidated as follows:

- o OAM information needs to be abstracted that are common to different layer and different domain.
- o Support customized OAM service, e.g., customized service diagnose.
- o OAM information is provided to management entity from managed device via a uniform interface (API and/or UI)
- o Sets up MD MEP and MIP in the network provision phase
- o Enables basic OAM functionality(e.g., enable the origin of ping and trace packets or configure Connectivity Fault Management (CFM)) on the managed devices in the service activation phase.

The different OAM tools may be used in one of two basic types of activation:

- o Proactive activation - indicates that the tool is activated on a continual basis, where messages are sent periodically, and errors are detected when a certain number of expected messages are not received.
- o On-demand activation - indicates that the tool is activated "manually" to detect a specific anomaly.

#### 7. Building on Existing Protocols

#### 8. Scoping Future Work

This section includes a set of candidate items for activities to be conducted within IETF.

These objectives are not frozen; further discussion is required to target key issues and scope the work to be conducted within IETF accordingly.

Candidate investigation items are listed below:

- o Understand and discuss situations where an OAM protocol can be tuned and optimized for a specific data plane.
- o OAM consolidation in the data plane:
  - \* Exchange OAM information at the service layer atop of layer 3.
  - \* Deployed over various encapsulating protocols, and in various medium types
- o OAM consolidation in the management plane:
  - \* Abstract OAM information common to different layers.
  - \* Expose OAM information via unified interface to management entities, independently of the layer they belong to.
  - \* Discuss how information gathered from various layers can be correlated for the sake of network operations optimization purposes.
  - \* Propose means to help during service diagnosis; these means may rely on filtering information to be leaked to other layers so that time recovery can be optimized. A typical example would

be efficient root cause analysis that is fed with input from various layers.

- \* Propose means that would help to optimize a network as a whole instead of the monolithic approach that is specific to a given layer. For example, investigate means that would help in computing diverse and completely disjoint paths, not only at layer 3 but also at the physical layer.

## 9. Manageability Considerations

## 10. Security Considerations

Security considerations are not addressed in this problem statement only document. Given the scope of OAM, and the implications on data and control planes, security considerations are clearly important and will be addressed in the specific protocol and deployment documents.

## 11. Acknowledgements

The authors would like to thank Romascanu, Dan, Tom Taylor, Tissa Senevirathne, Huub van Helvoort, Yuji Tochio for their valuable reviews and suggestions.

## 12. References

### 12.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", March 1997.
- [RFC6291] Andersson, L., Helvoort, H., Bonica, R., Romascanu, D., and S. Mansfield, "Guidelines for the Use of the "OAM" Acronym in the IETF", RFC 6291, June 2011.
- [RFC7276] Mizrahi, T. and N. Sprecher, "An Overview of Operations, Administration, and Maintenance (OAM) Tools", RFC 7276, June 2014.

### 12.2. Informative References

- [I-D.jain-nvo3-overlay-oam]  
Jain, P., "Generic Overlay OAM and Datapath Failure Detection", ID draft-jain-nvo3-overlay-oam-01, February 2014.

- [I-D.tissa-netmod-oam]  
Senevirathne , T., Finn, N., Kumar , D., and S. Salam ,  
"YANG Data Model for Operations Administration and  
Maintenance (OAM)", ID draft-tissa-netmod-oam-00, March  
2014.
- [I-D.ietf-sfc-problem-statement]  
Quinn, P., Guichard, J., and S. Surendra, "Network Service  
Chaining Problem Statement", ID draft-ietf-sfc-problem-  
statement, August 2013.
- [RFC3411] Harrington, D. and R. Presuhn, "An Architecture for  
Describing Simple Network Management Protocol (SNMP)  
Management Frameworks", RFC 3411, December 2002.
- [RFC4176] El Mghazli, Y., Nadeau, T., Boucadair, M., Chan, K., and  
A. Gonguet, "Framework for Layer 3 Virtual Private  
Networks (L3VPN) Operations and Management", RFC 4176,  
October 2005.
- [RFC4656] Shalunov, S., Karp, A., Boote, J., and M. Zekauskas, "A  
One-way Active Measurement Protocol (OWAMP)", RFC 4656,  
September 2006.
- [RFC5357] Hedeyat, K., Krzanowski, R., Morton, A., Yum, K., and J.  
Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)",  
RFC 5357, October 2008.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection  
(BFD)", RFC 5880, June 2010.
- [RFC6241] Enns, R., Bjorklund, M., Schoenwaelder, J., and A.  
Bierman, "Network Configuration Protocol (NETCONF)", RFC  
6241, June 2011.
- [RFC6374] Frost, D. and S. Bryant, "Packet Loss and Delay  
Measurement for MPLS Networks", RFC 6374, September 2011.
- [RFC6378] Weingarten, Y., Bryant, S., Osborne, E., Sprecher, N., and  
A. Fuligoli, "Packet Loss and Delay Measurement for MPLS  
Networks", RFC 6378, October 2011.
- [RFC6428] Allan, D., Swallow, G., and J. Drake, "Proactive  
Connectivity Verification, Continuity Check, and Remote  
Defect Indication for the MPLS Transport Profile", RFC  
6428, November 2011.

Authors' Addresses

Qin Wu  
Huawei  
101 Software Avenue, Yuhua District  
Nanjing, Jiangsu 210012  
China

Email: bill.wu@huawei.com

Mishaël Wexler  
Huawei  
Riesstr. 25  
Munich 80992  
Germany

Email: mishaël.wexler@huawei.com

Mohamed Boucadair  
France Telecom  
Rennes 35000  
France

Email: mohamed.boucadair@orange.com

Sam Aldrin  
Huawei Technologies USA  
2330 Central Expressway  
Santa Clara, CA 95051  
USA

Email: aldrin.ietf@gmail.com

Greg Mirsky  
Ericsson

Email: gregory.mirsky@ericsson.com

Pradeep Jain  
Nuage Networks  
755 Ravendale Drive  
Mountain View, CA 94043  
USA

Email: [pradeep@nuagenetworks.net](mailto:pradeep@nuagenetworks.net)

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: April 27, 2015

D. Liu  
China Mobile  
R. Zhang  
China Telecom  
L. Xue  
J. Kaippallimalil  
Huawei  
R. Pazhyannur  
S. Gundavelli  
Cisco  
October 24, 2014

Specification Alternate Tunnel Information for Data Frames in WLAN  
draft-xue-opsawg-capwap-alt-tunnel-information-01

Abstract

In IEEE 802.11 Wireless Local Area Network (WLAN) architecture, in order to satisfy the scalability requirement, customer data frames are desired to be distributed to an endpoint as Access Router (AR) different from the Access Controller (AC). For tunneling the data frames, there are many known alternate tunnel technologies can be used, such as IP-GRE, IP-in-IP, CAPWAP, L2TP/L2TPv3, etc. To assist a WTP to set up the alternate tunnels for data plane, this document extends the CAPWAP message elements.

Requirements Language

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 [RFC2119].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."



This Internet-Draft will expire on April 27, 2015.

#### Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

#### Table of Contents

1. Introduction . . . . .	3
2. Data Frame Alternate Tunnel in WLAN . . . . .	4
2.1. CAPWAP . . . . .	4
2.2. L2TP . . . . .	5
2.3. L2TPv3 . . . . .	6
2.4. IP-in-IP . . . . .	6
2.5. PMIPv6 . . . . .	7
2.6. GREv4/6 . . . . .	8
3. Alternate Tunnel Information Elements . . . . .	9
3.1. Access Router Information Sub-Elements . . . . .	9
3.1.1. AR IPv4 Address Sub-Element . . . . .	9
3.1.2. AR IPv4 Address for Load-balance Sub-Element . . . . .	10
3.1.3. AR IPv6 Address Sub-Element . . . . .	10
3.1.4. AR IPv6 Address for Load-balance Sub-Element . . . . .	11
3.1.5. AR FQDN Sub-Element . . . . .	12
3.1.6. AR FQDN for Load-balance Sub-Element . . . . .	12
3.2. Tunnel DTLS Policy Sub-Element . . . . .	13
3.3. IEEE 802.11 Tagging Mode Policy Sub-Element . . . . .	14
3.4. CAPWAP Transport Protocol Sub-Element . . . . .	14
3.5. GRE Key Sub-Element . . . . .	15
4. IANA Considerations . . . . .	15
5. Security Considerations . . . . .	15
6. References . . . . .	15
6.1. Normative References . . . . .	15
6.2. Informative References . . . . .	16
Authors' Addresses . . . . .	17

## 1. Introduction

Control and Provisioning of Wireless Access Points (CAPWAP) ([RFC5415], [RFC5416]) defines CAPWAP tunnel mode which can be used to encapsulate data frames and control/management frames of a station between the Wireless Transmission Point (WTP) and the Access Controller (AC). The customer data traffic on WTP can be either locally bridged or tunneled to the AC. In practice, operators who have deployed large numbers of WTPs desire to distribute the data traffic to a different entity (e.g., Access Router) rather than the AC for redundancy reasons. The architecture for tunneling WLAN user data frames to ARs is defined in [I-D.ietf-opsawg-capwap-alt-tunnel] and shown in Figure 1.

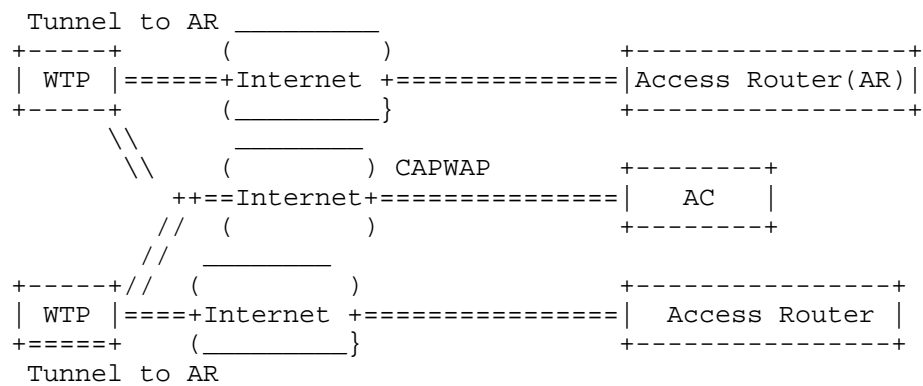


Figure 1: Centralized Control with Distributed Data

How the WTP can be configured with this alternate tunnel is already defined in [I-D.ietf-opsawg-capwap-alt-tunnel]. However, [I-D.ietf-opsawg-capwap-alt-tunnel] specifies only the generic container of the extension CAPWAP message elements used for this alternate tunnel (see Figure 2). The message elements information rely on a binding specification for a particular alternate tunnel protocol, such as GRE, IP-in-IP, CAPWAP, L2TP/L2TPv3 etc. This specification defines the binding specific CAPWAP message elements for using the different alternate tunnel protocols, one for each alternate tunnel protocol. Different Alternate Tunnel sub-message elements are defined.

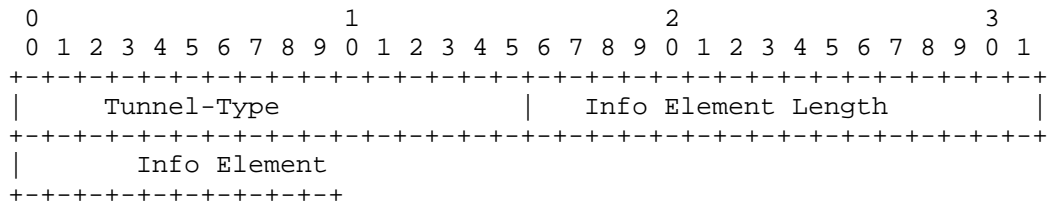


Figure 2: Alternate Tunnel Encapsulations Type

## 2. Data Frame Alternate Tunnel in WLAN

### 2.1. CAPWAP

When the WTP joins the AC, it should indicate its alternate tunnel encapsulation capability and the CAPWAP protocol should be one option. If the CAPWAP encapsulation is selected by the AC and configured by the AC to the WTP, the Info Element field of the generic encapsulation shown in Figure 2 should contain the following information:

- o Access Router Information: IPv4 address or IPv6 address or Fully Qualified Domain Name (FQDN), which includes the Access Router information with which the WTP can associated for tunneling the user traffic.
- o Tunnel DTLS Policy: The CAPWAP protocol allows optional protection of data packets using DTLS. Use of data packet protection on a WTP is determined by the associated AC policy. When the AC determines the DTLS is utilized, the D bit should be set. Otherwise, clear data packets will be encapsulated (see [RFC5415]).
- o IEEE 802.11 Tagging Mode Policy: It is used to specify how the CAPWAP data channel packet are to be tagged for QoS purposes (see [RFC5416]).
- o CAPWAP Transport Protocol: The CAPWAP protocol supports both UDP and UDP-Lite (see [RFC3828]). When run over IPv4, UDP is used for the CAPWAP data channels. When run over IPv6, the CAPWAP data channel may use either UDP or UDP-lite.

The message element structure for CAPWAP encapsulation is shown in Figure 3:

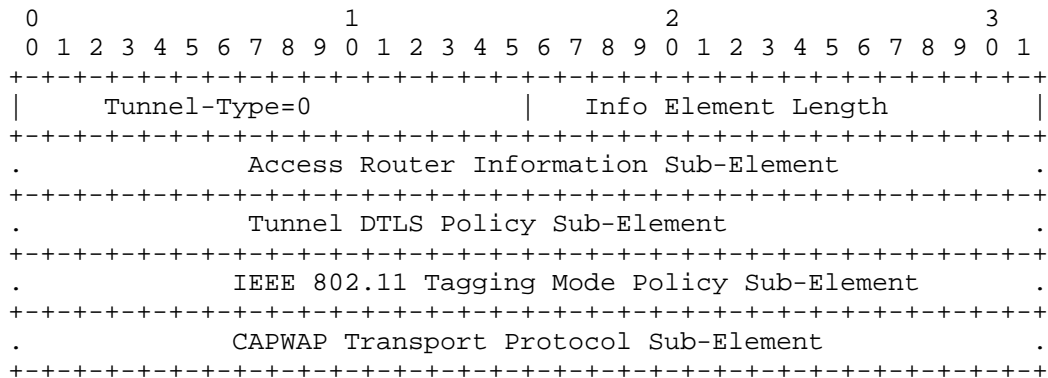


Figure 3: Alternate Tunnel Encapsulation - CAPWAP

## 2.2. L2TP

Layer Two Tunneling Protocol (L2TP) can pass PPP frames over an L2TP tunnel within a UDP datagram. When a AC selects the L2TP as the alternate tunnel encapsulation and reports the selection to the WTP, the WTP initiates the L2TP data tunnel establishment with the specific AR(s). The AR whose responsibility is to be a L2TP Network Server (LNS) (see [RFC2661]) should configure WTP during the calling request from hosts attaching to the WTP in IEEE 802.11 network. For L2TP, the Info Element field of the generic encapsulation shown in Figure 2 should contain the following information (not-exhaustive):

- o Access Router (acts as LNS) Information: IPv4 address or IPv6 address or Fully Qualified Domain Name (FQDN), which includes the Access Router information with which the WTP can associate for tunneling the user traffic.

The message element structure for L2TP encapsulation is shown in Figure 4:

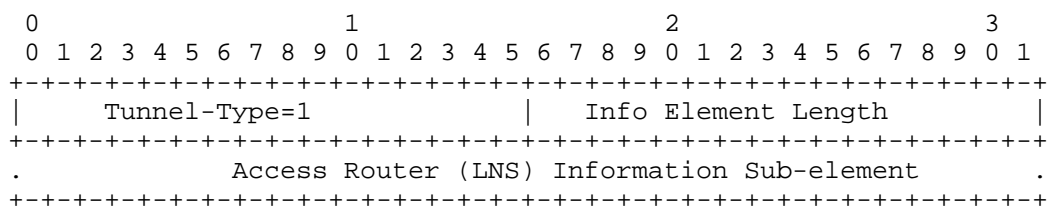


Figure 4: Alternate Tunnel Encapsulation - L2TP

### 2.3. L2TPv3

L2TPv3 (see [RFC3931]) borrows largely from L2TPv2. L2TPv3 tunnel can be used over multiple Packet-Switched Networks (PSN) such as IP, UDP, Frame Relay, ATM, MPLS, etc. L2TPv3 data tunnels may be utilized with or without the L2TP control channel, either via manual configuration or via other signaling methods to per-configure or distribute L2TP session information. In this document, L2TPv3 control channel is assumed to establish, manage and tear down the L2TPv3 data tunnels. For L2TPv3, the Info Element field of the generic encapsulation shown in Figure 2 should contain the following information:

- o Access Router (acts as LNS) Information: IPv4 address or IPv6 address or Fully Qualified Domain Name (FQDN), which includes the Access Router information with which the WTP can associate for tunneling the user traffic.

The message element structure for L2TPv3 encapsulation is shown in Figure 5:

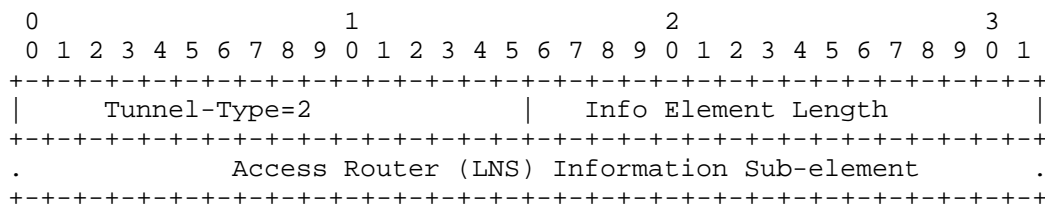


Figure 5: Alternate Tunnel Encapsulation - L2TPv3

## 2.4. IP-in-IP

If IP-in-IP encapsulation (see [RFC2003]) is selected by AC, the user traffic that arrives to a WTP is encapsulated within IP datagrams and delivered to an intermediate destination which is the Access Router. Once the encapsulated datagram arrives the AR, it is decapsulated. In the general case, the encapsulator WTP should obtain the AR as the decapsulator. If IP-in-IP encapsulation is selected by AC and configured by AC to WTP, the Info Element field of the generic encapsulation shown in Figure 2 should contain the following information:

- o Access Router Information: IPv4 address or IPv6 address or Fully Qualified Domain Name (FQDN), which includes the Access Router information with which the WTP can associate for tunneling the user traffic.

The message element structure for IP-in-IP encapsulation is shown in Figure 6:

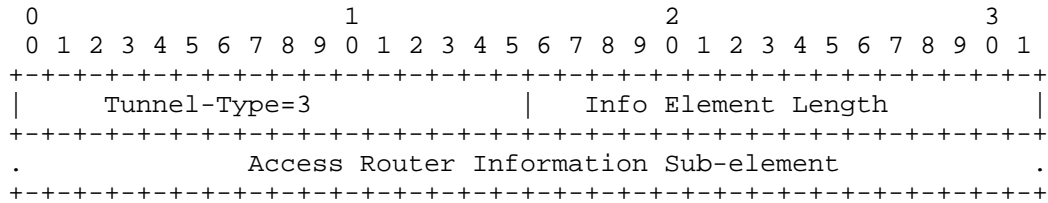


Figure 6: Alternate Tunnel Encapsulation - IP-in-IP

## 2.5. PMIPv6

Proxy Mobile IPv6 (PMIPv6, see [RFC5213]) is one option for alternate tunnel encapsulation between the WTP and the AR. In this scenario, a WTP should act as the Mobile Access Gateway (MAG) function that manages the mobility-related signaling for a station that is attached to the WTP IEEE 802.11 radio access. The Local Mobility Anchor (LMA) function should be located at the AR. In Proxy Mobile IPv6, the address of the LMA should be discovered by the MAG. If PMIPv6 encapsulation is selected by the AC and configured by the AC to a WTP, the Info Element field of the generic encapsulation shown in Figure 2 should contain the following information:

- o Access Router (acts as LMA) Information: IPv6 address or Fully Qualified Domain Name (FQDN), which includes the Access Router information with which the WTP can associate for tunneling the user traffic.

The message element structure for PMIPv6 encapsulation is shown in Figure 7:

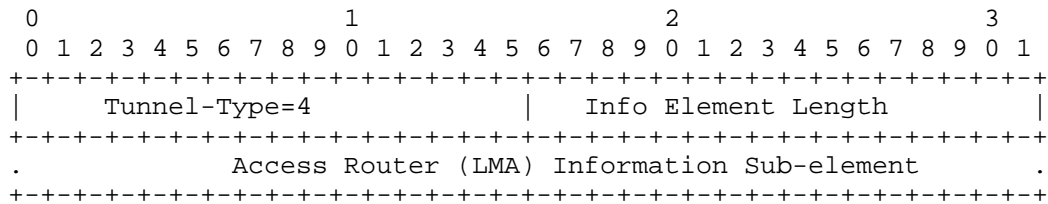


Figure 7: Alternate Tunnel Encapsulation - PMIPv6

## 2.6. GREv4/6

In order to encapsulate data traffic using GREv4/6 (see and [RFC1701][RFC2784]), the WTP needs to obtain the destination node IP address of a GRE tunnel (e.g., the AR address). Optionally, GRE Key Sub-element (see [RFC2784] and [RFC2890]) is needed for WTP to configure the complementary tunnel information. If WTP obtains the GRE Key Sub-element, the key MUST be inserted into the GRE encapsulation header. The Key is used for identifying extra context information about the received payload on AR. If the WTP obtains the Key information from the AC, the payload packets without the correspondent GRE Key or with an unmatched GRE Key will be silently dropped on the AR. For GRE, the Info Element field of the generic encapsulation shown in Figure 2 should contain the following information (not-exhaustive):

- o Access Router Information: IPv4 address (for GREv4) or IPv6 address (for GREv6) or Fully Qualified Domain Name (FQDN) (For both GREv4 and GREv6), which includes the Access Router information with which the WTP can associate for tunneling the user traffic.
- o GRE Key: The Key field contains a four octet number which is inserted by the WTP as defined in [RFC2890].

The message element structure for GREv4 encapsulation is shown in Figure 8:

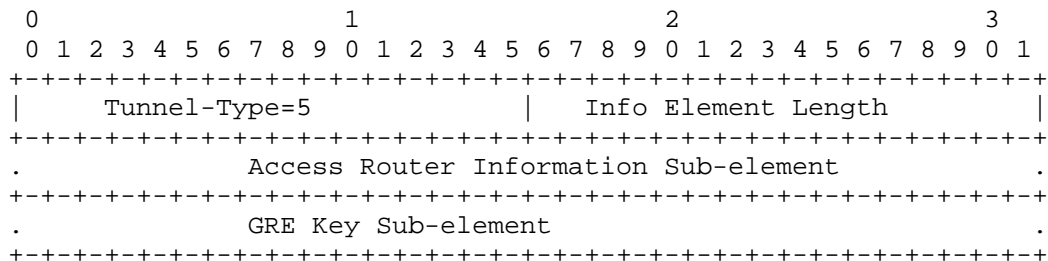


Figure 8: Alternate Tunnel Encapsulation - GREv4

The message element structure for GREv6 encapsulation is shown in Figure 9:

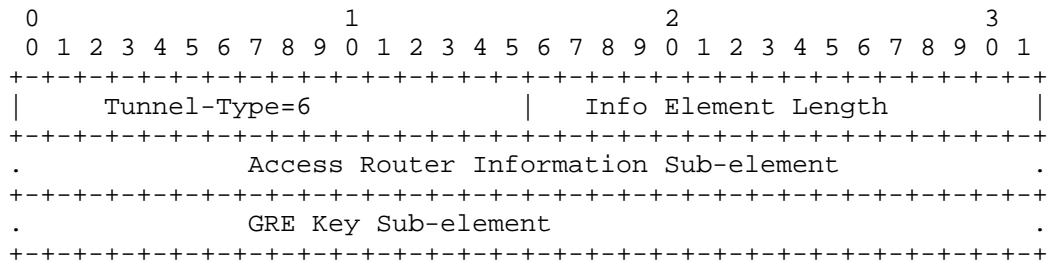


Figure 9: Alternate Tunnel Encapsulation - GREv6

### 3. Alternate Tunnel Information Elements

#### 3.1. Access Router Information Sub-Elements

The Access Router Information Sub-Elements allow the AC to notify a WTP of which AR(s) are available for establishing a data tunnel. The AR information may be IPv4 address, IPv6 address, or AR domain name. If a WTP obtains the correct AR FQDN, the Name-to-IP address mapping is handled in the WTP (see [RFC2782]).

The following are the Access Router Information Sub-Elements defined in this specification. The AC can use one of them to notify the destination information of the data tunnel to the WTP. The Sub-Elements containing the AR IPv4 address MUST NOT be used if an IPv6 data channel such as PMIPv6 or GREv6 is used.

##### 3.1.1. AR IPv4 Address Sub-Element

This Sub-Element (see Figure 10) is used by the AC to configure a WTP with the AR IPv4 address available for the WTP to establish the data tunnel for user traffic.

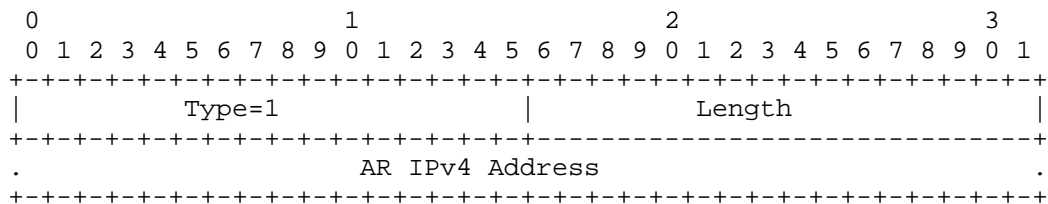


Figure 10: AR IPv4 Address Sub-Element

Type: 1 for AR IPv4 Address

Length: 4



AR IPv4 Address: 32-bit integer containing AR IPv4 Address.

### 3.1.2. AR IPv4 Address for Load-balance Sub-Element

This Sub-Element (see Figure 11) is used to satisfy load-balance and reliability requirements. There may be multiple AR addresses available for a WTP and provided by an AC. The WTP can use the AR information to send user traffic.

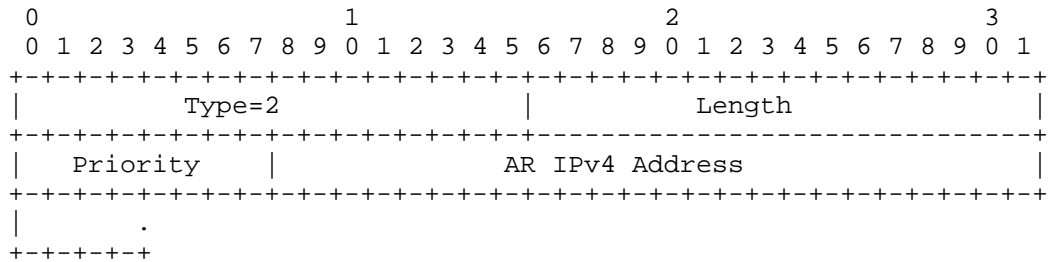


Figure 11: AR IPv4 Address for Load-balance Sub-Element

Type: 2 for AR IPv4 Address for Load-balance

Length: >=5

Priority: A value between 1 and 255 specifying the priority order for the preferred AR. For instance, the value of one (1) is used to set the primary AR, the value of two (2) is used to set the secondary; two instances with the same value are used for load-balance, etc.

AR IPv4 Address: 32-bit integer containing AR IPv4 Address binding with the specific priority. There may be an array of pairs binding priority and AR IPv4 address.

### 3.1.3. AR IPv6 Address Sub-Element

This Sub-Element (see Figure 12) is used by the AC to configure a WTP with the AR IPv6 address available for the WTP to establish the data tunnel for user traffic.

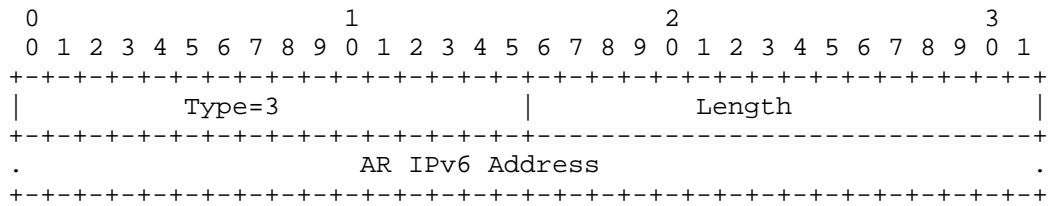


Figure 12: AR IPv6 Address Sub-Element

Type: 3 for AR IPv6 Address

Length: 16

AR IPv6 Address: 128-bit integer containing AR IPv6 Address

#### 3.1.4. AR IPv6 Address for Load-balance Sub-Element

This Sub-Element (see Figure 13) is used to satisfy load-balance and reliability requirements. There may be multiple AR addresses available for a WTP and provided by an AC. A WTP can use the AR information to send user traffic.

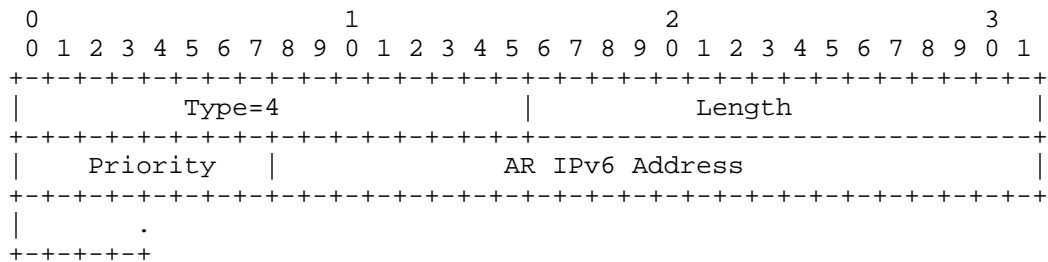


Figure 13: AR IPv6 Address for Load-balance Sub-Element

Type: 4 for AR IPv6 Address for Load-balance

Length: >= 17

Priority: A value between 1 and 255 specifying the priority order of the preferred AR. For instance, the value of one (1) is used to set the primary AR, the value of two (2) is used to set the secondary; two instances with the same value are used for load-balance, etc.

AR IPv6 Address: 128-bit integer containing AR IPv6 Address binding with the specific priority. There may be an array of pairs binding priority and AR IPv6 address.

## 3.1.5. AR FQDN Sub-Element

This Sub-Element (see Figure 14) is used by the AC to configure a WTP with AR FQDN available to establish the data tunnel for user traffic. Based on the FQDN, a WTP can acquire the AR IP address via DNS.

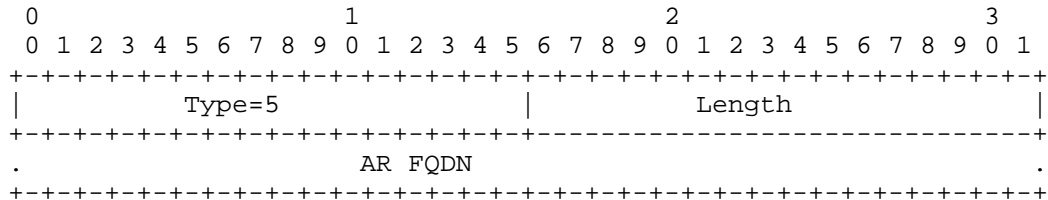


Figure 14: AR FQDN Sub-Element

Type: 5 for AR FQDN

Length: >=1

AR FQDN: A variable-length string containing the AR FQDN.

## 3.1.6. AR FQDN for Load-balance Sub-Element

This Sub-Element (see Figure 15) is used to satisfy load-balance and reliability requirements. There may be multiple AR FQDNs available for a WTP and provided by an AC. A WTP can use the AR information to send user traffic.

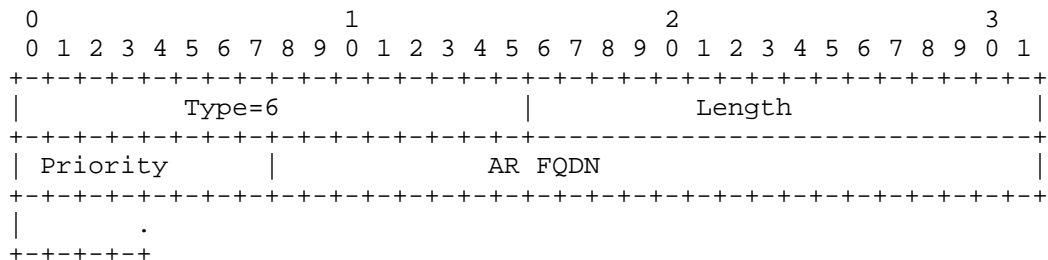


Figure 15: AR FQDN for Load-balance Sub-Element

Type: 6 for AR FQDN for Load-balance

Prefer: A value between 1 and 255 specifying the priority order of the preferred AR. For instance, the value of one (1) is used to set the primary AR, the value of two (2) is used to set the secondary; two instances with the same value are used for load-balance, etc.

AR FQDN: Variable-length string containing AR FQDN binding with the specific priority. There may be an array of pairs binding priority and AR FQDN.

### 3.2. Tunnel DTLS Policy Sub-Element

The AC distributes its DTLS usage policy for the CAPWAP data tunnel between a WTP and the AR. There are multiple supported options, represented by the bit field below as defined in AC Descriptor message elements. The WTP MUST abide by one of the options for tunneling user traffic with AR. The Tunnel DTLS Policy Sub-Element obey the definition in [RFC5415]. If there are more than one ARs information provided by the AC for reliability reasons, the same Tunnel DTLS Policy (see Figure 16) is generally applied for all tunnels associated with the ARs. Otherwise, Tunnel DTLS Policy MUST be bonding together with each of the ARs, then WTP will enforce the independent tunnel DTLS policy for each tunnel with a specific AR.

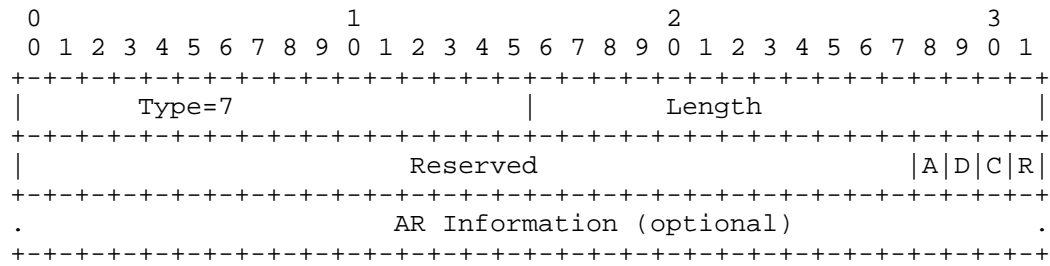


Figure 16: Tunnel DTLS Policy Sub-Element

Type: 7 for Tunnel DTLS Policy

Length: >=6

Reserved: A set of reserved bits for future use. All implementations complying with this protocol MUST set to zero any bits that are reserved in the version of the protocol supported by that implementation. Receivers MUST ignore all bits not defined for the version of the protocol they support.

A: If A bit is set, there is an AR information associated with the DTLS policy. There may be an array of pairs binding DTLS policy information and AR information contained in the Tunnel DTLS Policy Sub-Element. Otherwise, the same Tunnel DTLS Policy (see Figure 16) is generally applied for all tunnels associated with the ARs configured by the AC.

D: DTLS-Enabled Data Channel Supported (see [RFC5415]).

C: Clear Text Data Channel Supported (see [RFC5415]).

R: A reserved bit for future use abide (see [RFC5415]).

### 3.3. IEEE 802.11 Tagging Mode Policy Sub-Element

In 802.11 networks, IEEE 802.11 Tagging Mode Policy Sub-Element is used to specify how the WTP apply the QoS tagging policy when receiving the packets from stations on a particular radio. When the WTP sends out the packet to data channel to the AR(s), the packets have to be tagged for QoS purposes (see [RFC5416]).

The IEEE 802.11 Tagging Mode Policy abides the IEEE 802.11 WTP Quality of Service defined in Section 6.22 of [RFC5416].

### 3.4. CAPWAP Transport Protocol Sub-Element

The CAPWAP data tunnel supports both UDP and UDP-Lite (see [RFC3828]). When run over IPv4, UDP is used for the CAPWAP data channels. When run over IPv6, the CAPWAP data channel may use either UDP or UDP-lite. The AC specifies and configure the WTP for which transport protocol is to be used for the CAPWAP data tunnel.

The CAPWAP Transport Protocol Sub-Element abides the definition in Section 4.6.14 of [RFC5415].

```

      0               1               2               3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|           Type=51           |           Length           |
+-----+-----+-----+-----+-----+-----+-----+-----+
|           Transport         |
+-----+-----+-----+-----+-----+-----+-----+

```

#### CAPWAP Transport Protocol Sub-Element

Type: 51 for CAPWAP Transport Protocol [RFC5415].

Length: 1

Transport: The transport to use for the CAPWAP Data channel. The following enumerated values are supported:

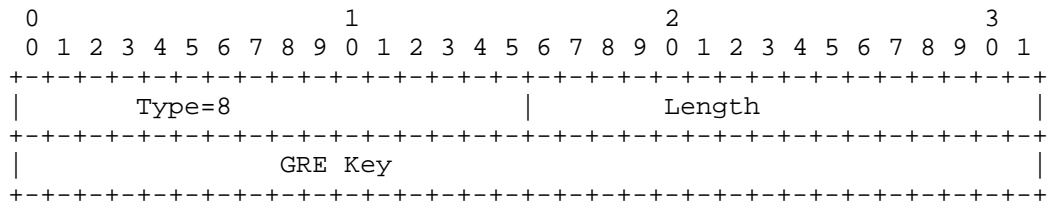
1 - UDP-Lite: The UDP-Lite transport protocol is to be used for the CAPWAP Data channel. Note that this option MUST NOT be used if the CAPWAP Control channel is being used over IPv4 and AR address is IPv4 contained in the AR Information Sub-Element.

2 - UDP: The UDP transport protocol is to be used for the CAPWAP Data channel.

### 3.5. GRE Key Sub-Element

If a WTP receives the GRE Key Sub-Element in the Alternate Tunnel Encapsulation message element for GREv4 or GREv6 selection, the WTP must insert the GRE Key to the encapsulation packet (see [RFC2890]). An AR acting as decapsulating tunnel endpoint identifies packets belonging to a traffic flow based on the Key value.

The GRE Key Sub-Element field contains a four octet number defined in [RFC2890].



GRE Key Sub-Element

Type: 8 for GRE Key Sub-Element

Length: 4

GRE Key: The Key field contains a four octet number which is inserted by the WTP according to [RFC2890].

### 4. IANA Considerations

To be specified in later versions.

### 5. Security Considerations

To be specified in later versions.

### 6. References

#### 6.1. Normative References

- [RFC1701] Hanks, S., Li, T., Farinacci, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 1701, October 1994.
- [RFC2003] Perkins, C., "IP Encapsulation within IP", RFC 2003, October 1996.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2401] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.
- [RFC2661] Townsley, W., Valencia, A., Rubens, A., Pall, G., Zorn, G., and B. Palter, "Layer Two Tunneling Protocol "L2TP"", RFC 2661, August 1999.
- [RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000.
- [RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, March 2000.
- [RFC2890] Dommety, G., "Key and Sequence Number Extensions to GRE", RFC 2890, September 2000.
- [RFC3828] Larzon, L-A., Degermark, M., Pink, S., Jonsson, L-E., and G. Fairhurst, "The Lightweight User Datagram Protocol (UDP-Lite)", RFC 3828, July 2004.
- [RFC3931] Lau, J., Townsley, M., and I. Goyret, "Layer Two Tunneling Protocol - Version 3 (L2TPv3)", RFC 3931, March 2005.
- [RFC4347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security", RFC 4347, April 2006.
- [RFC5213] Gundavelli, S., Leung, K., Devarapalli, V., Chowdhury, K., and B. Patil, "Proxy Mobile IPv6", RFC 5213, August 2008.
- [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.

## 6.2. Informative References

[I-D.ietf-opsawg-capwap-alt-tunnel]

Zhang, R., Cao, Z., Deng, H., Pazhyannur, R., Gundavelli, S., and L. Xue, "Alternate Tunnel Encapsulation for Data Frames in CAPWAP", draft-ietf-opsawg-capwap-alt-tunnel-03 (work in progress), September 2014.

Authors' Addresses

Dapeng Liu  
China Mobile  
Unit 2, 28 Xuanwumenxi Ave, Xuanwu District  
Beijing 100053  
China

Email: liudapeng@chinamobile.com

Rong Zhang  
China Telecom  
No. 109 Zhongshandadao avenue  
Guangzhou 510630  
China

Email: zhangr@gsta.com

Li Xue  
Huawei  
No. 156 Beiqing Rd. Z-park, Shi-Chuang-Ke-Ji-Shi-Fan-Yuan  
Beijing, Haidian District 100095  
China

Email: xueli@huawei.com

John Kaippallimalil  
Huawei  
5430 Legacy Drive, Suite 175  
Plano, TX 75024

Email: john.kaippallimalil@huawei.com



Rajesh S. Pazhyannur  
Cisco  
170 West Tasman Drive  
San Jose, CA 95134  
USA

Email: [rpazhyan@cisco.com](mailto:rpazhyan@cisco.com)

Sri Gundavelli  
Cisco  
170 West Tasman Drive  
San Jose, CA 95134  
USA

Email: [sgundave@cisco.com](mailto:sgundave@cisco.com)