

OPSAWG  
Internet-Draft  
Intended status: Standards Track  
Expires: April 16, 2014

H. Asai  
Univ. of Tokyo  
M. MacFaden  
VMware Inc.  
J. Schoenwaelder  
Jacobs University  
Y. Sekiya  
Univ. of Tokyo  
K. Shima  
IIJ Innovation Institute Inc.  
T. Tsou  
Huawei Technologies (USA)  
C. Zhou  
Huawei Technologies  
H. Esaki  
Univ. of Tokyo  
October 13, 2013

Management Information Base for Virtual Machines Controlled by a  
Hypervisor  
draft-asai-vmm-mib-05

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 April 16, 2014.

## Copyright Notice

Copyright (c) 2013 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. Requirements Language . . . . .   | 3  |
| 2. The Internet-Standard Management Framework . . . . .                      | 4  |
| 3. Managed Objects for Virtual Machines Controlled by a Hypervisor . . . . . | 5  |
| 3.1. Managed Objects on Virtualization Environment . . . . .                 | 5  |
| 3.2. Overview of the MIB Module . . . . .                                    | 6  |
| 3.3. Definitions . . . . .   | 10 |
| 4. IANA Considerations . . . . .   | 47 |
| 5. Security Considerations . . . . .   | 48 |
| 6. Acknowledgements . . . . .  | 50 |
| 7. References . . . . .  | 51 |
| 7.1. Normative References . . . . .  | 51 |
| 7.2. Informative References . . . . .  | 52 |
| Appendix A. State Transition Table . . . . .                                 | 53 |
| Authors' Addresses . . . . .   | 55 |

## 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 enterprise 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 hypervisors.

### 1.1. 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 RFC 2119 [RFC2119].

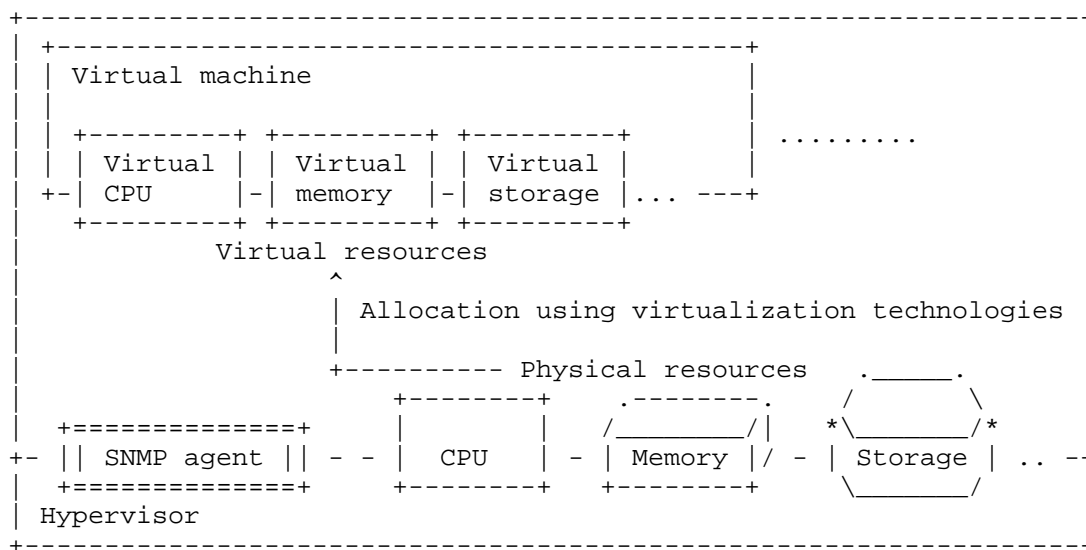
## 2. The Internet-Standard Management Framework

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. Managed Objects for Virtual Machines Controlled by a Hypervisor

#### 3.1. Managed Objects on Virtualization Environment

On the common implementations of hypervisor softwares, a hypervisor allocates virtual resources such as virtual CPUs, virtual memory, virtual storage devices, and virtual network interfaces to virtual machines from physical resources. This document defines objects related to system and software information of a hypervisor, the list of virtual machines controlled by the hypervisor, and virtual resources allocated by the hypervisor to virtual machines. This document specifies four specific types of virtual resources that are common to general hypervisors; CPUs (processors), memory, network interfaces, and storage devices.



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

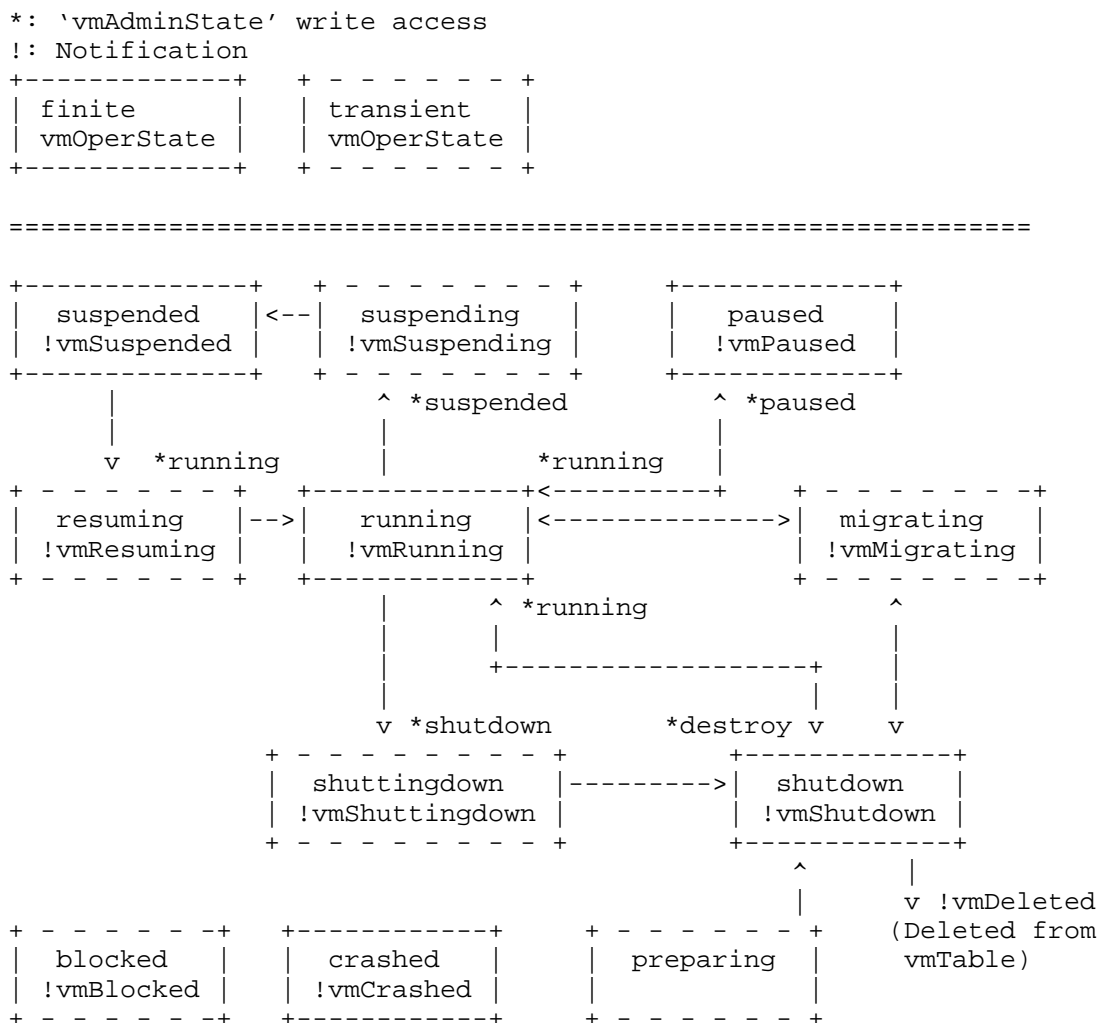
As shown in Figure 1, the objects defined in this document are managed at a hypervisor and an SNMP agent is launched at the hypervisor to provide access to the objects. The objects are managed from the viewpoint of the operators of hypervisors, but not the operators of virtual machines; i.e., the objects do not take into account the actual resource utilization on each virtual machine but the resource allocation from the physical resources. For example,

vmNetworIfIndex 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 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 [RFC4133]. The objects related to virtual switches are not also included in this MIB module though virtual switches shall 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, BRIDGE-MIB [RFC4188] and Q-BRIDGE-MIB [RFC4363] could be used.

### 3.2. Overview of the MIB Module

The MIB module is organized into a group of scalars and tables. The scalars below 'hypervisor' 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 'hrStorageTable' 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 'vmNetworIfIndex' 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 'ifTable' of IF-MIB, the entry contains a pointer 'vmNetworkParent' to the physical network interface.



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 transition of 'vmOperState' by the write access to 'vmAdminState' and the notifications generated by the operational state changes are summarized in Figure 2. 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 shall be a problem. An implementation shall support both, either of, or none of per-VM notifications and bulk notifications. The notification filtering mechanism described in section 6 of RFC 3413 [RFC3413] is used by the management applications to control the notifications.

The MIB module provides a few writable objects that can be used to make non-persistent changes, e.g., changing the memory allocation or the CPU allocation. It is not the goal of this MIB module to provide a configuration interface for virtual machines since other protocols and data modeling languages are more suitable for this task.

The OID tree structure of the MIB module is shown below.

```
--vmMIB (1.3.6.1.2.1.yyy)
+---vmNotifications(0)
|   +---vmRunning(1) [vmName, vmUUID, vmOperState]
|   +---vmShuttingdown(2) [vmName, vmUUID, vmOperState]
|   +---vmShutdown(3) [vmName, vmUUID, vmOperState]
|   +---vmPaused(4) [vmName, vmUUID, vmOperState]
|   +---vmSuspending(5) [vmName, vmUUID, vmOperState]
|   +---vmSuspended(6) [vmName, vmUUID, vmOperState]
|   +---vmResuming(7) [vmName, vmUUID, vmOperState]
|   +---vmMigrating(8) [vmName, vmUUID, vmOperState]
|   +---vmCrashed(9) [vmName, vmUUID, vmOperState]
|   +---vmBlocked(10) [vmName, vmUUID, vmOperState]
|   +---vmDeleted(11) [vmName, vmUUID, vmOperState, vmPersistent]
|   +---vmBulkRunning(12) [vmAffectedVMs]
|   +---vmBulkShutdown(13) [vmAffectedVMs]
|   +---vmBulkShuttingdown(14) [vmAffectedVMs]
|   +---vmBulkPaused(15) [vmAffectedVMs]
|   +---vmBulkSuspending(16) [vmAffectedVMs]
|   +---vmBulkSuspended(17) [vmAffectedVMs]
```



```

|   +---vmBulkResuming(18) [vmName, vmUUID, vmOperState]
|   +---vmBulkMigrating(19) [vmAffectedVMs]
|   +---vmBulkCrashed(20) [vmAffectedVMs]
|   +---vmBulkBlocked(21) [vmAffectedVMs]
|   +---vmBulkDeleted(22) [vmAffectedVMs]
+---vmObjects(1)
|   +---vmHypervisor(1)
|   |   +--- r-n SnmpAdminString      vmHvSoftware(1)
|   |   +--- r-n SnmpAdminString      vmHvVersion(2)
|   |   +--- r-n OBJECT IDENTIFIER    vmHvObjectID(3)
|   |   +--- r-n TimeTicks            vmHvUpTime(4)
|   +--- r-n Integer32      vmNumber(2)
|   +--- r-n TimeTicks      vmTableLastChange(3)
+---vmTable(4)
|   +---vmEntry(1) [vmIndex]
|   |   +--- --- VirtualMachineIndex  vmIndex(1)
|   |   +--- r-n SnmpAdminString      vmName(2)
|   |   +--- r-n UUIDorZero           vmUUID(3)
|   |   +--- r-n SnmpAdminString      vmOSType(4)
|   |   +--- rwn VirtualMachineAdminState
|   |   |   vmAdminState(5)
|   |   +--- r-n VirtualMachineOperState
|   |   |   vmOperState(6)
|   |   +--- r-n VirtualMachineAutoStart
|   |   |   vmAutoStart(7)
|   |   +--- r-n VirtualMachinePersistent
|   |   |   vmPersistent(8)
|   |   +--- rwn Integer32            vmCurCpuNumber(9)
|   |   +--- rwn Integer32            vmMinCpuNumber(10)
|   |   +--- rwn Integer32            vmMaxCpuNumber(11)
|   |   +--- r-n Integer32            vmMemUnit(12)
|   |   +--- rwn Integer32            vmCurMem(13)
|   |   +--- rwn Integer32            vmMinMem(14)
|   |   +--- rwn Integer32            vmMaxMem(15)
|   |   +--- r-n TimeTicks            vmUpTime(16)
|   |   +--- r-n Counter64            vmCpuTime(17)
+---vmCpuTable(5)
|   +---vmCpuEntry(1) [vmIndex, vmCpuIndex]
|   |   +--- --- VirtualMachineCpuIndex
|   |   |   vmCpuIndex(1)
|   |   +--- r-n Counter64            vmCpuCoreTime(2)
+---vmCpuAffinityTable(6)
|   +---vmCpuAffinityEntry(1) [vmIndex,
|   |   vmCpuIndex,
|   |   vmCpuPhysIndex]
|   |   +--- --- Integer32            vmCpuPhysIndex(1)
|   |   +--- rwn Integer32            vmCpuAffinity(2)
+---vmStorageTable(7)

```

```

+---vmStorageEntry(1) [vmStorageVmIndex, vmStorageIndex]
+--- --- VirtualMachineIndexOrZero
|
|           vmStorageVmIndex(1)
+--- --- VirtualMachineStorageIndex
|
|           vmStorageIndex(2)
+--- r-n Integer32           vmStorageParent(3)
+--- r-n VirtualMachineStorageSourceType
|
|           vmStorageSourceType(4)
+--- r-n SnmpAdminString     vmStorageSourceTypeString(5)
+--- r-n SnmpAdminString     vmStorageResourceID(6)
+--- r-n VirtualMachineStorageAccess
|
|           vmStorageAccess(7)
+--- r-n VirtualMachineStorageMediaType
|
|           vmStorageMediaType(8)
+--- r-n SnmpAdminString     vmStorageMediaTypeString(9)
+--- r-n Integer32           vmStorageSizeUnit(10)
+--- r-n Integer32           vmStorageDefinedSize(11)
+--- r-n Integer32           vmStorageAllocatedSize(12)
+--- r-n Counter64           vmStorageReadIOs(13)
+--- r-n Counter64           vmStorageWriteIOs(14)
+---vmNetworkTable(8)
+---vmNetworkEntry(1) [vmIndex, vmNetworkIndex]
+--- --- VirtualMachineNetworkIndex
|
|           vmNetworkIndex(1)
+--- r-n InterfaceIndexOrZero vmNetworkIfIndex(2)
+--- r-n InterfaceIndexOrZero vmNetworkParent(3)
+--- r-n SnmpAdminString     vmNetworkModel(4)
+--- r-n PhysAddress         vmNetworkPhysAddress(5)
+--- rwn TruthValue          vmPerVMNotificationsEnabled(9)
+--- rwn TruthValue          vmBulkNotificationsEnabled(10)
+--- --n VirtualMachineList  vmAffectedVMs(11)
+---vmConformance(2)
+---vmCompliances(1)
|
|   +---vmFullCompliances(1)
|   +---vmReadOnlyCompliances(2)
+---vmGroups(2)
+---vmHypervisorGroup(1)
+---vmVirtualMachineGroup(2)
+---vmCpuGroup(3)
+---vmCpuAffinityGroup(4)
+---vmStorageGroup(5)
+---vmNetworkGroup(6)
+---vmPerVMNotificationOptionalGroup(7)
+---vmBulkNotificationsVariablesGroup(8)
+---vmBulkNotificationOptionalGroup(9)

```

### 3.3. Definitions

```
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;
```

```
vmMIB MODULE-IDENTITY
```

```
    LAST-UPDATED "201310130000Z"           -- 13 October 2013  
    ORGANIZATION "IETF Operations and Management Area Working Group"  
    CONTACT-INFO
```

```
        "  
        WG E-mail: (To be added after approved by WG)  
        Mailing list subscription info:  
        http:// (To be added after approved by WG)
```

```
        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
```

```
        Yuji Sekiya  
        The University of Tokyo  
        2-11-16 Yayoi
```

Bunkyo-ku, Tokyo 113-8658  
JP  
Email: sekiya@wide.ad.jp

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

Cathy Zhou  
Huawei Technologies  
Bantian, Longgang District  
Shenzhen 518129  
P.R. China  
Email: cathyzhou@huawei.com

Hiroshi Esaki  
The University of Tokyo  
7-3-1 Hongo  
Bunkyo-ku, Tokyo 113-8656  
JP  
Email: hiroshi@wide.ad.jp  
"

#### DESCRIPTION

"This MIB module is for use in managing a hypervisor and virtual machines controlled by the hypervisor. The OID 'yyy' is temporary one, and it must be assigned by IANA when this becomes an official document.

Copyright (c) 2013 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 "201310130000Z"      -- 13 October 2013
DESCRIPTION
    "The original version of this MIB, published as
    RFCXXXX."
 ::= { mib-2 yyy }

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.
- destroy(5) The administrative state of the virtual machine indicating the virtual machine should be forcibly shutdown. After the destroy operation, the administrative state should be automatically changed to shutdown(4)."

```
SYNTAX      INTEGER {
                running(1),
                suspend(2),
                pause(3),
                shutdown(4),
                destroy(5)
            }
```

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(6), resuming(8), migrating(10), and shuttingdown(11).
- blocked(5)      The operational state of the virtual machine indicating the execution of the virtual machine is currently blocked, e.g., waiting for some action of the hypervisor to finish. This is a transient state from/to other states.
- suspending(6)   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(7).
- suspended(7)    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(8)     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(7) to running(4).
- paused(9)       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(10) The operational state of the virtual machine indicating the virtual machine is currently in the process of migration from/to another hypervisor.

shuttingdown(11)  
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(12).

shutdown(12) 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(13) The operational state of the virtual machine indicating the virtual machine has crashed."

```
SYNTAX      INTEGER {
                unknown(1),
                other(2),
                preparing(3),
                running(4),
                blocked(5),
                suspending(6),
                suspended(7),
                resuming(8),
                paused(9),
                migrating(10),
                shuttingdown(11),
                shutdown(12),
                crashed(13)
            }
```

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.

enable(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.

disable(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),  
    enable(2),  
    disable(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:

```

        readwrite(1)    The virtual storage is a read-write
                        device.

        readonly(2)     The virtual storage is a read-only
                        device."
SYNTAX      INTEGER {
                    readwrite(1),
                    readonly(2)
                }

VirtualMachineStorageMediaType ::= TEXTUAL-CONVENTION
STATUS      current
DESCRIPTION
    "The media type of a virtual 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."
SYNTAX      INTEGER {
                    other(1),
                    unknown(2),
                    hardDisk(3),
                    opticalDisk(4)
                }

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 "lx"
STATUS      current
DESCRIPTION

```

"Each octet within this value specifies a set of eight virtual machine vmIndex, 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, and 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 region. 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-write  
STATUS current  
DESCRIPTION  
"The administrative power state of the virtual machine. Note that a virtual machine is supposed to be resumed when vmAdminState of the virtual machine is changed from suspended(2) or paused(3) to running(1)."  
::= { 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-write
STATUS          current
DESCRIPTION
    "The number of virtual CPUs currently assigned to the
     virtual machine.  Changes to this object MUST NOT
     persist across re-initialization of the hypervisor."
 ::= { vmEntry 9 }

vmMinCpuNumber OBJECT-TYPE
SYNTAX          Integer32 (-1|0..2147483647)
MAX-ACCESS      read-write
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.  Changes to this
     object MUST NOT persist across re-initialization of the
     hypervisor."
```



```
::= { vmEntry 10 }

vmMaxCpuNumber OBJECT-TYPE
    SYNTAX      Integer32 (-1|0..2147483647)
    MAX-ACCESS   read-write
    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.  Changes to
        this object MUST NOT persist across re-initialization of
        the hypervisor."
    ::= { vmEntry 11 }

vmMemUnit OBJECT-TYPE
    SYNTAX      Integer32 (1..2147483647)
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The multiplication unit 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-write
    STATUS       current
    DESCRIPTION
        "The current memory size currently allocated to the
        virtual memory module in the unit designated by
        vmMemUnit.  Changes to this object MUST NOT persist
        across re-initialization of the hypervisor."
    ::= { vmEntry 13 }

vmMinMem OBJECT-TYPE
    SYNTAX      Integer32 (-1|0..2147483647)
    MAX-ACCESS   read-write
    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.  Changes to this object MUST NOT persist
        across re-initialization of the hypervisor."
    ::= { vmEntry 14 }
```

## vmMaxMem OBJECT-TYPE

SYNTAX Integer32 (-1|0..2147483647)  
MAX-ACCESS read-write  
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. Changes to this  
    object MUST NOT persist across re-initialization of the  
    hypervisor."  
 ::= { 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-write
    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 shall be
-- 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      VirtualMachineStorageMediaType,
        vmStorageMediaTypeString
                               SnmpAdminString,
        vmStorageSizeUnit      Integer32,
        vmStorageDefinedSize    Integer32,
        vmStorageAllocatedSize  Integer32,
        vmStorageReadIOs        Counter64,
        vmStorageWriteIOs       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      VirtualMachineStorageMediaType
    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 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 shall 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 shall 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 }

-- 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 shall 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."
 ::= { 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."
 ::= { 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 }

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(12) from some other state. The other state is
    indicated by the included value of vmOperState."
 ::= { vmNotifications 2 }

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(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(9) 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(6) 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(7) 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(8) 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(10) 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 previos state of the virtual
    machine is indicated by the included value of
    vmOperState."
 ::= { vmNotifications 9 }

vmBlocked 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
    blocked(5). The previos state of the virtual machine is
    indicated by the included value of vmOperState."
 ::= { vmNotifications 10 }
```

```
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 11 }

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 a all prior states except for
    running(4). Management stations are encouraged to
    subsequently poll the subset of virtual machines of
    interest for vmOperState."
  ::= { vmNotifications 12 }

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(11) from a state other than
    shuttingdown(11). Management stations are encouraged to
    subsequently poll the subset of virtual machines of
    interest for vmOperState."
  ::= { vmNotifications 13 }

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(12) from a state other than shutdown(12). Management stations are encouraged to subsequently poll the subset of virtual machines of interest for vmOperState."

::= { vmNotifications 14 }

## 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(9) from a state other than paused(9). Management stations are encouraged to subsequently poll the subset of virtual machines of interest for vmOperState."

::= { vmNotifications 15 }

## 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(6) from a state other than suspending(6). Management stations are encouraged to subsequently poll the subset of virtual machines of interest for vmOperState."

::= { vmNotifications 16 }

## 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(7) from a state other than suspended(7). Management stations are encouraged to subsequently poll

```

        the subset of virtual machines of interest for
        vmOperState."
 ::= { vmNotifications 17 }

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(8) from a state other than resuming(8).
    Management stations are encouraged to subsequently poll
    the subset of virtual machines of interest for
    vmOperState."
 ::= { vmNotifications 18 }

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(10) from a state other than migrating(10).
    Management stations are encouraged to subsequently poll
    the subset of virtual machines of interest for
    vmOperState."
 ::= { vmNotifications 19 }

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 20 }

vmBulkBlocked 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 blocked(5) from a state other than blocked(5).
    Management stations are encouraged to subsequently poll
    the subset of virtual machines of interest for
    vmOperState."
 ::= { vmNotifications 21 }

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 22 }

-- Compliance definitions:
vmGroups          OBJECT IDENTIFIER ::= { vmConformance 1 }
vmCompliances     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 vmAdminState
MIN-ACCESS    read-only
DESCRIPTION
    "Write access is not required."

OBJECT vmCurCpuNumber
MIN-ACCESS    read-only
DESCRIPTION
    "Write access is not required."

OBJECT vmMinCpuNumber
MIN-ACCESS    read-only
DESCRIPTION
    "Write access is not required."

OBJECT vmMaxCpuNumber
MIN-ACCESS    read-only
DESCRIPTION
    "Write access is not required."

OBJECT vmCurMem
MIN-ACCESS    read-only
DESCRIPTION
    "Write access is not required."
```

```
OBJECT vmMinMem
MIN-ACCESS    read-only
DESCRIPTION
    "Write access is not required."

OBJECT vmMaxMem
MIN-ACCESS    read-only
DESCRIPTION
    "Write access is not required."

OBJECT vmCpuAffinity
MIN-ACCESS    read-only
DESCRIPTION
    "Write access is not required."

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  
    }  
    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,  
        vmBlocked,  
        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,
    vmBulkBlocked,
    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
```

#### 4. IANA Considerations

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

| Descriptor<br>----- | OBJECT IDENTIFIER value<br>----- |
|---------------------|----------------------------------|
| vmMIB               | { mib-2 TBD }                    |

## 5. Security Considerations

There are a number of management objects defined in this MIB that have a MAX-ACCESS clause of read-write and/or read-create. Such objects may be considered sensitive or vulnerable in some network environments. The support for SET operations in a non-secure environment without proper protection can have a negative effect on hypervisor and virtual machine operations.

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.

It is recommended that attention be specifically given to implementing the MAX-ACCESS clause in a number of objects, including `vmAdminState`, `vmMinCpuNumber`, `vmMaxCpuNumber`, `vmMinMem`, `vmMaxMem`, and `vmCpuAffinity` in scenarios that DO NOT use SNMPv3 strong security (i.e. authentication and encryption). Extreme caution must be used to minimize the risk of cascading security vulnerabilities when SNMPv3 strong security is not used. When SNMPv3 strong security is not used, these objects should have access of read-only, not read-create.

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 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.

## 6. Acknowledgements

The authors like to thank Joe Marcus Clarke, Randy Presuhn, and David Black 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.

## 7. References

### 7.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.
- [RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.
- [RFC2580] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.
- [RFC2790] Waldbusser, S. and P. Grillo, "Host Resources MIB", RFC 2790, March 2000.
- [RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, June 2000.
- [RFC3413] Levi, D., Meyer, P., and B. Stewart, "Simple Network Management Protocol (SNMP) Applications", STD 62, RFC 3413, December 2002.
- [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, December 2002.
- [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, December 2002.
- [RFC3418] Presuhn, R., "Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3418, December 2002.
- [RFC4122] Leach, P., Mealling, M., and R. Salz, "A Universally Unique IDentifier (UUID) URN Namespace", RFC 4122, July 2005.
- [RFC4133] Bierman, A. and K. McCloghrie, "Entity MIB (Version 3)", RFC 4133, August 2005.

- [RFC4188] Norseth, K. and E. Bell, "Definitions of Managed Objects for Bridges", RFC 4188, September 2005.
- [RFC4363] Levi, D. and D. Harrington, "Definitions of Managed Objects for Bridges with Traffic Classes, Multicast Filtering, and Virtual LAN Extensions", RFC 4363, January 2006.

## 7.2. Informative References

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

## Appendix A. State Transition Table

| State        | Action or<br>(Event)  | Next state   | Notification                           |
|--------------|---|--------------|--|
| suspended    | running   | resuming     | vmResuming  <br>vmBulkResuming         |
| suspending   | (suspend<br>operation<br>completed)                                       | suspended    | vmSuspended  <br>vmBulkSuspended       |
| running      | suspended   | suspending   | vmSuspending  <br>vmBulkSuspending     |
|              | shutdown  | shuttingdown | vmShuttingdown  <br>vmBulkShuttingdown |
|              | destroy   | shutdown     | vmShutdown  <br>vmBulkShutdown         |
|              | (migration to<br>other<br>hypervisor<br>initiated)                        | migrating    | vmMigrating  <br>vmBulkMigrating       |
| resuming     | (resume<br>operation<br>completed)  | running      | vmRunning  <br>vmBulkRunning           |
| paused       | running   | running      | vmRunning  <br>vmBulkRunning           |
| shuttingdown | (shutdown<br>operation<br>completed)                                      | shutdown     | vmShutdown  <br>vmBulkShutdown         |
| shutdown     | running   | running      | vmRunning  <br>vmBulkRunning           |
|              | (if this state<br>entry is<br>created by a<br>migration<br>operation (*)) | migrating    | vmMigrating  <br>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 |
| blocked    | (blocking operation completed)              | (previous state) | -                           |
| crashed    | -   | -                | -                           |
| (any)      | (blocking operation initiated)              | blocked          | vmBlocked   vmBulkBlocked   |
|            | (crashed)                                   | crashed          | vmCrashed   vmBulkCrashed   |
| (no state) | (preparation initiated)                     | preparing        | -                           |
|            | (migrate from other hypervisor initiated)   | shutdown (*)     | vmShutdown   vmBulkShutdown |

State transition table

## 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

Yuji Sekiya  
The University of Tokyo  
2-11-16 Yayoi  
Bunkyo-ku, Tokyo 113-8658  
JP

Email: sekiya@wide.ad.jp

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](mailto:tina.tsou.zouting@huawei.com)

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  
JP

Phone: +81 3 5841 6748  
Email: [hiroshi@wide.ad.jp](mailto:hiroshi@wide.ad.jp)





Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: August 5, 2013

Y. Chen  
D. Liu  
H. Deng  
China Mobile  
Lei. Zhu  
Huawei  
Feb 2013

CAPWAP Extension for 802.11n and Power/channel Reconfiguration  
draft-chen-opsawg-capwap-extension-00

Abstract

CAPWAP binding for 802.11 is specified by RFC5416 and it was based on IEEE 802-11.2007 standard. After RFC5416 was published in 2009, there was several new amendent of 802.11 has been published. 802.11n is one of those amendent and it has been widely used in real deployment. This document extends the CAPWAP binding for 802.11 to support 802.11n.

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 5, 2013.

Copyright Notice

Copyright (c) 2013 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. Conventions used in this document . . . . .      | 3  |
| 3. CAPWAP 802.11n support . . . . .                 | 3  |
| 4. CAPWAP extension for 802.11n support . . . . .   | 4  |
| 5. Power and Channel auto reconfiguration . . . . . | 8  |
| 6. Security Considerations . . . . .                | 13 |
| 7. IANA Considerations . . . . .                    | 13 |
| 8. Contributors . . . . .                           | 14 |
| 9. Acknowledgements . . . . .                       | 14 |
| 10. Normative References . . . . .                  | 14 |
| Authors' Addresses . . . . .                        | 14 |

## 1. Introduction

IEEE 802.11n standard was published in 2009 and it is an amendment to the IEEE 802.11-2007 standard to improve network throughput. The maximum data rate increases to 600Mbit/s physical throughput rate. In the physical layer, 802.11n use OFDM and MIMO to achieve the high throughput. 802.11n use multiple antennas to form antenna array which can be dynamically adjusted to improve the signal strength and extend the coverage.

There are couple of capabilities of 802.11n need to be supported by CAPWAP control message such as radio capability, radio configuration and station information.

## 2. 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].

## 3. CAPWAP 802.11n support

IEEE 802.11n standard was published in 2009 and it is an amendment to the IEEE 802.11-2007 standard to improve network throughput. The maximum data rate increases to 600Mbit/s physical throughput rate. In the physical layer, 802.11n use OFDM and MIMO to achieve the high throughput. 802.11n use multiple antennas to form antenna array which can be dynamically adjusted to improve the signal strength and extend the coverage.

802.11n support 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 2.4GHz. In the large scale deployment scenario, operator need to use 20MHz channel configuration in 2.4GHz to allow more non-overlapping channels.

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

Another feature in 802.11 MAC layer is Block ACK. 802.11n can use one ACK frame to acknowledge several MPDU receiving event.

CAPWAP need to be extended to support the above new 802.11n features. For example, CAPWAP should allow the access controller to know the supported 802.11n features and the access controller should be able to configure the differe channel binding modes. One possible solution is to extend the CAPWAP information element for 802.11n.

#### 4. CAPWAP extension for 802.11n support

There are couple of capabilities of 802.11n need to be supported by CAPWAP control message such as radio capability, radio configuration and station information. This section defines the extension of current CAPWAP 802.11 information element to support 802.11n.

1. 802.11n Radio Capability Information Element. The information element need to be extended to include 802.11n radio capability. Below is an example of the 802.11n radio capability information element.

| 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 |  |  |  |  |  |  |  |  |
| Element ID     |   |   |   |   |   |   |   |   |   | Length         |   |   |   |   |   |   |   |   |   |              |   |   |   |   |   |   |   |   |   |              |   |  |  |  |  |  |  |  |  |
| Radio ID       |   |   |   |   |   |   |   |   |   | SupChanl width |   |   |   |   |   |   |   |   |   | Power Save   |   |   |   |   |   |   |   |   |   | ShortGi20    |   |  |  |  |  |  |  |  |  |
| ShortGi40      |   |   |   |   |   |   |   |   |   | HtDelyBlkack   |   |   |   |   |   |   |   |   |   | Max Amsdu    |   |   |   |   |   |   |   |   |   | Max RxFactor |   |  |  |  |  |  |  |  |  |
| Min StaSpacing |   |   |   |   |   |   |   |   |   | HiSuppDataRate |   |   |   |   |   |   |   |   |   | AMPDUBufSize |   |   |   |   |   |   |   |   |   | HtcSupp      |   |  |  |  |  |  |  |  |  |
|                |   |   |   |   |   |   |   |   |   | 20MHZ 11gMCS   |   |   |   |   |   |   |   |   |   |              |   |   |   |   |   |   |   |   |   |              |   |  |  |  |  |  |  |  |  |
|                |   |   |   |   |   |   |   |   |   | 20MHZ 11gMCS   |   |   |   |   |   |   |   |   |   |              |   |   |   |   |   |   |   |   |   |              |   |  |  |  |  |  |  |  |  |
|                |   |   |   |   |   |   |   |   |   | 20MHZ 11gMCS   |   |   |   |   |   |   |   |   |   |              |   |   |   |   |   |   |   |   |   |              |   |  |  |  |  |  |  |  |  |
|                |   |   |   |   |   |   |   |   |   | 20MHZ 11gMCS   |   |   |   |   |   |   |   |   |   |              |   |   |   |   |   |   |   |   |   |              |   |  |  |  |  |  |  |  |  |
|                |   |   |   |   |   |   |   |   |   | 20MHZ 11aMCS   |   |   |   |   |   |   |   |   |   |              |   |   |   |   |   |   |   |   |   |              |   |  |  |  |  |  |  |  |  |
|                |   |   |   |   |   |   |   |   |   | 20MHZ 11aMCS   |   |   |   |   |   |   |   |   |   |              |   |   |   |   |   |   |   |   |   |              |   |  |  |  |  |  |  |  |  |
|                |   |   |   |   |   |   |   |   |   | 20MHZ 11aMCS   |   |   |   |   |   |   |   |   |   |              |   |   |   |   |   |   |   |   |   |              |   |  |  |  |  |  |  |  |  |
|                |   |   |   |   |   |   |   |   |   | 20MHZ 11aMCS   |   |   |   |   |   |   |   |   |   |              |   |   |   |   |   |   |   |   |   |              |   |  |  |  |  |  |  |  |  |
|                |   |   |   |   |   |   |   |   |   | 40MHZ 11gMCS   |   |   |   |   |   |   |   |   |   |              |   |   |   |   |   |   |   |   |   |              |   |  |  |  |  |  |  |  |  |
|                |   |   |   |   |   |   |   |   |   | 40MHZ 11gMCS   |   |   |   |   |   |   |   |   |   |              |   |   |   |   |   |   |   |   |   |              |   |  |  |  |  |  |  |  |  |
|                |   |   |   |   |   |   |   |   |   | 40MHZ 11gMCS   |   |   |   |   |   |   |   |   |   |              |   |   |   |   |   |   |   |   |   |              |   |  |  |  |  |  |  |  |  |
|                |   |   |   |   |   |   |   |   |   | 40MHZ 11aMCS   |   |   |   |   |   |   |   |   |   |              |   |   |   |   |   |   |   |   |   |              |   |  |  |  |  |  |  |  |  |
|                |   |   |   |   |   |   |   |   |   | 40MHZ 11aMCS   |   |   |   |   |   |   |   |   |   |              |   |   |   |   |   |   |   |   |   |              |   |  |  |  |  |  |  |  |  |
|                |   |   |   |   |   |   |   |   |   | 40MHZ 11aMCS   |   |   |   |   |   |   |   |   |   |              |   |   |   |   |   |   |   |   |   |              |   |  |  |  |  |  |  |  |  |
|                |   |   |   |   |   |   |   |   |   | 40MHZ 11aMCS   |   |   |   |   |   |   |   |   |   |              |   |   |   |   |   |   |   |   |   |              |   |  |  |  |  |  |  |  |  |

1. SupChanl width: The supported bandwith mode. 0x01: 20MHz bandwidth binding mode. 0x02: 40MHz bandwidth binding mode.

2. Power Save: 0x00: Static power saving mode. 0x01: Dynamic power saving mode. 0x03: Do not support power saving mode.
3. ShortGi20: Whether support short GI. 0x00: Do not support short GI. 0x01: Support short GI.
4. HtDelyBlkack: Whether block Ack support delay mode. 0x00: Do not support delay mode. 0x01: Support delay mode.
5. Max Amsdu: The maximal AMSDU length. 0: 3839 bytes. 1: 7935 bytes.
6. Max RxFactor: The maximal receiving AMPDU factor. Default value: 3.
7. Min StaSpacing: Minimum MPDU Start Spacing.
8. HiSuppDataRate: Maximal transmission speed.
9. AMPDUBufSize: AMPDU buffer size.
10. HtcSupp: Whether the packet have HT header.
11. 20MHZ 11gMCS: 128 bitmap.If support should be all zero, otherwise all one.
12. 20MHZ 11aMCS: 128 bitmap.If support should be all zero, otherwise all one.
13. 40MHZ 11gMCS: 128 bitmap.If support should be all zero, otherwise all one.
14. 40MHZ 11aMCS: 128 bitmap.If support should be all zero, otherwise all one.
15. 2. 802.11n Raido Configuration TLV. Following figure is an example of 802.11n radio configuration TLV.

| 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 |  |  |  |  |  |  |  |  |
| Element ID  |   |   |   |   |   |   |   |   |   | Length        |   |   |   |   |   |   |   |   |   |             |   |   |   |   |   |   |   |   |   |             |   |  |  |  |  |  |  |  |  |
| Radio ID    |   |   |   |   |   |   |   |   |   | Amsdu Cfg     |   |   |   |   |   |   |   |   |   | Ampdu Cfg   |   |   |   |   |   |   |   |   |   | 11nOnly Cfg |   |  |  |  |  |  |  |  |  |
| ShortGi Cfg |   |   |   |   |   |   |   |   |   | BandWidth Cfg |   |   |   |   |   |   |   |   |   | MaxSupp MCS |   |   |   |   |   |   |   |   |   | Max MandMCS |   |  |  |  |  |  |  |  |  |
| TxAntenna   |   |   |   |   |   |   |   |   |   | RxAntenna     |   |   |   |   |   |   |   |   |   | Reserved    |   |   |   |   |   |   |   |   |   |             |   |  |  |  |  |  |  |  |  |
| Reserved    |   |   |   |   |   |   |   |   |   |               |   |   |   |   |   |   |   |   |   |             |   |   |   |   |   |   |   |   |   |             |   |  |  |  |  |  |  |  |  |

1. A-MSDU CFG: 0x00: Disable 0x01: Enalbe
2. A-MPDU CFG: 0x00: Disable 0x01: Enalbe
3. 11N Only CFG: Whether allow only 11n user access. 0x00: Allow non-802.11n user access. 0x01: Do not allow non-802.11n user access.
4. Short GI CFG: 0x00: Disable 0x01: Enable
5. Bandwidth CFG: Bandwidth binding mode. 0x00: 40MHz 0x01: 20MHz

6. Max Support MCS: Maximal MCS.
7. Max Mandantory MCS: Maximal mandantory MCS.
8. TxAntenna: Transmitting antenna configuration.
9. RxAntenna: Receiving antenna configuration.
10. Each TxAntenna and RxAntenna bit represent one antenna, 1 means enable, 0 means disable.

3. 802.11n Station Information. Following figure is an example of 802.11n station information information element.

| 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 |  |  |  |  |  |  |  |  |
| Element ID     |   |   |   |   |   |   |   |   |   | Length         |   |   |   |   |   |   |   |   |   |                |   |   |   |   |   |   |   |   |   |           |   |  |  |  |  |  |  |  |  |
| MAC Address    |   |   |   |   |   |   |   |   |   |                |   |   |   |   |   |   |   |   |   |                |   |   |   |   |   |   |   |   |   |           |   |  |  |  |  |  |  |  |  |
| SupChanl width |   |   |   |   |   |   |   |   |   | Power Save     |   |   |   |   |   |   |   |   |   |                |   |   |   |   |   |   |   |   |   |           |   |  |  |  |  |  |  |  |  |
| ShortGi20      |   |   |   |   |   |   |   |   |   | ShortGi40      |   |   |   |   |   |   |   |   |   | HtDelyBlkack   |   |   |   |   |   |   |   |   |   | Max Amsdu |   |  |  |  |  |  |  |  |  |
| Max RxFactor   |   |   |   |   |   |   |   |   |   | Min StaSpacing |   |   |   |   |   |   |   |   |   | HiSuppDataRate |   |   |   |   |   |   |   |   |   |           |   |  |  |  |  |  |  |  |  |
| AMPDUBufSize   |   |   |   |   |   |   |   |   |   | HtcSupp        |   |   |   |   |   |   |   |   |   | MCS Set        |   |   |   |   |   |   |   |   |   |           |   |  |  |  |  |  |  |  |  |
| MCS Set        |   |   |   |   |   |   |   |   |   |                |   |   |   |   |   |   |   |   |   |                |   |   |   |   |   |   |   |   |   |           |   |  |  |  |  |  |  |  |  |
| MCS Set        |   |   |   |   |   |   |   |   |   |                |   |   |   |   |   |   |   |   |   |                |   |   |   |   |   |   |   |   |   |           |   |  |  |  |  |  |  |  |  |
| MCS Set        |   |   |   |   |   |   |   |   |   |                |   |   |   |   |   |   |   |   |   |                |   |   |   |   |   |   |   |   |   |           |   |  |  |  |  |  |  |  |  |

1. SupChanl width: Supporting bandwidth mode. 0x01: 20MHz bandwidth mode. 0x02: 40MHz bandwidth binding mode.
2. Power Save: 0x00: Static power saving mode. 0x01: Dynamic power saving mode. 0x03: Do not support power saving mode.
3. ShortGi20: Whether support short GI in 20MHz bandwidth mode. 0x00: Do not support short GI. 0x01: Support short GI.
4. ShortGi40: Whether support short GI in 40MHz bandwidth mode. 0x00: Do not support short GI. 0x01: Support short GI.
5. HtDelyBlkack: Whether block Ack support delay mode. 0x00: Do not support delay mode. 0x01: Support delay mode.
6. Max Amsdu: The maximal AMSDU length. 0x00: 3839 bytes. 0x01: 7935 bytes.
7. Max RxFactor: The maximal receiving AMPDU factor.
8. Min StaSpacing: Minimum MPDU Start Spacing.



9. HiSuppDataRate: Maximal transmission speed.
10. AMPDUBufSize: AMPDU buffer size.
11. HtcSupp: Whether the packet have HT header.
12. MCS Set: The MCS bitmap that the station supports.

## 5. Power and Channel auto reconfiguration

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

When the AP is power-on, it is of necessity to configure a proper channel to the AP in order to achieve best status of radio links. IEEE 802.11 Direct Sequence Control elements or IEEE 802.11 OFDM Control element defined in RFC5416 should be carried to offer AP a channel at this stage. Those element should be carried in the Configure Status Response message. If those information element is zero, the AP will determine its channel by itself, otherwise the AP should be configured according to the provided information element.

When the AP 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. The channel quality report will be sent to the AC using WTP Event Request message by the AP. The AC can use IEEE 802.11 Direct Sequence Control or IEEE 802.11 OFDM Control information element carried by the configure Update Request message to configure a new channel for the AP.

IEEE 802.11 Tx Power information element is used by the AC to control the transmission power of the AP. The 802.11 Tx Power information element is carried in the Configure Status Response message during the power on phase or in the Configure Update Request message during the running phase.

Channel Scan Procedure.

The Channel Scan Procedure is illustrated by the following figure.

```

WTP                                Configure Status Req                                AC
----->
  Configure Status Res(Scan Para TLV, Chl Bind TLV)
<-----
or
WTP Configure Update Req(Scan Para, Bind TLV)                                AC
<-----
  Configure Update Res
----->

```

The definition of the Scan Para TLV is as follows:

|                     |   |      |   |   |              |   |   |   |   |        |   |   |   |   |                      |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |  |  |
|---------------------|---|------|---|---|--------------|---|---|---|---|--------|---|---|---|---|----------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|--|--|
| 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 |  |  |  |  |  |  |  |  |
| Element ID          |   |      |   |   |              |   |   |   |   | Length |   |   |   |   |                      |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |  |  |
| Radio ID            |   |      |   |   | AP oper mode |   |   |   |   |        |   |   |   |   | Scan Type            |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |  |  |
| Reserved            |   |      |   |   |              |   |   |   |   |        |   |   |   |   |                      |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |  |  |
| Report Time         |   |      |   |   |              |   |   |   |   |        |   |   |   |   | PrimeChlSrvTime      |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |  |  |
| On Channel ScanTime |   |      |   |   |              |   |   |   |   |        |   |   |   |   | Off Channel ScanTime |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |  |  |
| L D                 |   | Flag |   |   |              |   |   |   |   |        |   |   |   |   |                      |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |  |  |

Element ID: TBD; Length:18

AP oper mode: the work mode of the AP. 0x01:normal mode. 0x02: monitor only mode.

Scan Type: 0x01: active scan; 0x02: passive scan.

Report Time: Channel quality report time.

PrimeChlSrvTime: Service time on the working scan channel. This segment is invalid(set to 0) when AP oper mode is set to 2. The maximum value of this segment is 10000, the minimum value of this

segment is 5000, the default value is 5000.

On Channle ScanTime: The scan time of the working channel. When the AP oper mode is set to 2, this segment is invalid(set to 0). The maximum value of thi segment is 120, the minimum value of this segment is 60, the default value is 60.

L=1: Open Load Balance Scan. D=1: Open Rogue AP detection scan.  
Flag: Bitmap, resered for furture use.

The definition of the Channel Bind TLV is as follows:

| 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 |               |  |  |  |  |  |  |  |  |  |
| -----    |   |   |   |   |   |   |   |   |   | -----               |   |   |   |   |   |   |   |   |   | ----- |          |   |   |   |   |   |   |   |   | ----- |   |               |  |  |  |  |  |  |  |  |  |
|          |   |   |   |   |   |   |   |   |   | Element ID          |   |   |   |   |   |   |   |   |   |       | Length   |   |   |   |   |   |   |   |   |       |   |               |  |  |  |  |  |  |  |  |  |
| Radio ID |   |   |   |   |   |   |   |   |   | Flag                |   |   |   |   |   |   |   |   |   |       |          |   |   |   |   |   |   |   |   |       |   |               |  |  |  |  |  |  |  |  |  |
|          |   |   |   |   |   |   |   |   |   | Max Cycles          |   |   |   |   |   |   |   |   |   |       | Reserved |   |   |   |   |   |   |   |   |       |   | Channel Count |  |  |  |  |  |  |  |  |  |
|          |   |   |   |   |   |   |   |   |   | Scan Channel Set... |   |   |   |   |   |   |   |   |   |       |          |   |   |   |   |   |   |   |   |       |   |               |  |  |  |  |  |  |  |  |  |

Element ID: TBD. Length>=12

Flag: bitmap, reserved.

Scan Src: the trigger of the scan event. not defined in this version of the document. set to 0.

Device Type: the scope of the scan. not defined in this version of the document. set to 0.

Max Cycles: Scan repeat times. 255 means continuous scan.

Scan Channel Set: the channle information. the format is as follows:

|  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |  |  |
|--|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|--|--|
| 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 |  |  |  |  |  |  |  |  |
| +----- |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |  |  |

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

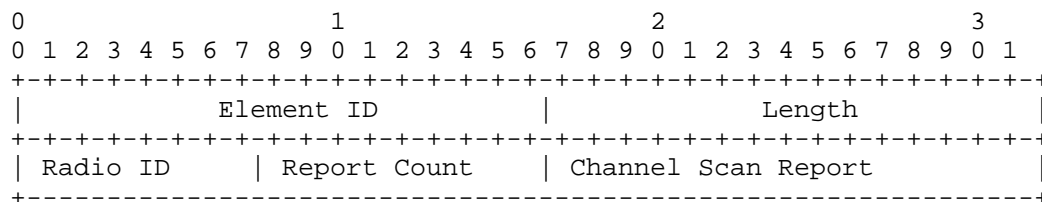
Flag: bitmap, reserved for future use.

The channle scan procedure:

The AP has two work mode: the first one is normal mode. In this mode, the AP can provide service for the STA access and scan the channel at the same time. Whether the AP will scan the channel is determined by the Max Cycles segment in the Channle Bind TLV. When this segment is set to 0, the AP will not scan the channle. If this segment is set to 255, the AP will continuous scan the channel. The type of the scan is determined by the Sacn Type segment. In the passive scan type, the AP monitor the airinterface, based on the received beacon frame to determine the nearby APs. In the active scan type, the AP will send probe message and receive the probe response message. In the normal scan mode, the AP will use 3 parameters: PrimeChlSrvTime, OnChannelScanTime, OffChannelScnTime. The AP will provide access service for the STAs for PrimeChlSrvTime duration and then start to scan the channel for On Channel ScnTime duration. Back to the working channel, provide STA access service for PrimeChlSrvTime, then leave the working channel, start to scan the next channel for Off Channel ScanTime duration. This process will be repeated until all the channel is scanned.

When the AP work in the scan only mode, there is no difference between the working channel and scan channel. Every channel's scan duration will be OffChannelScnTime and the PrimeChlSrvTime and OnChannelScanTime is set to 0.

Scan Report. The AP 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 TLV and Neighbor AP Report TLV. The definition of the Channel Scan Report TLV is as follows:



Element ID: 133; Length: >= 20.

Report Count: the channle number will be reported. The definition of

the channel scan report is as follows:

| 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 |  |  |  |  |  |  |  |  |
| Channel Number |   |   |   |   |   |   |   |   |   | Radar Statistics |   |   |   |   |   |   |   |   |   | Mean                |   |   |   |   |   |   |   |   |   |                 |   |  |  |  |  |  |  |  |  |
| Time           |   |   |   |   |   |   |   |   |   | Mean RSSI        |   |   |   |   |   |   |   |   |   | Screen Packet Count |   |   |   |   |   |   |   |   |   |                 |   |  |  |  |  |  |  |  |  |
| NeighborCount  |   |   |   |   |   |   |   |   |   | Mean Noise       |   |   |   |   |   |   |   |   |   | Interference        |   |   |   |   |   |   |   |   |   | Self Tx Occp    |   |  |  |  |  |  |  |  |  |
| SelfStaOccp    |   |   |   |   |   |   |   |   |   | Unknown Occp     |   |   |   |   |   |   |   |   |   | CRC Err Cnt         |   |   |   |   |   |   |   |   |   | Decrypt Err Cnt |   |  |  |  |  |  |  |  |  |
| Phy Err Cnt    |   |   |   |   |   |   |   |   |   | Retrans Cnt      |   |   |   |   |   |   |   |   |   |                     |   |   |   |   |   |   |   |   |   |                 |   |  |  |  |  |  |  |  |  |

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.

Mean RSSI: The signal strength of the scanned channel.

Screen Packet Count: Received packet number.

Neighbor Count: The neighbor number of this channel.

Mean Noise: the average noise on this channel.

Interference: The interference of the channel.

Self Tx Occp: The time duration for transmission.

Unknown Occp: TBD.

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.

The definition of neighbor AP report TLV is as follows:

|  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |  |  |
|--|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|--|--|
|  |   |   |   |   |   |   |   |   |   | 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 |  |  |  |  |  |  |  |  |
| +----- |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |  |  |

Element ID: 134; Length: >=16

The definition of Neighbor info is as follows:

|  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |  |  |
|--|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|--|--|
|  |   |   |   |   |   |   |   |   |   | 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 |  |  |  |  |  |  |  |  |
| +----- |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |  |  |

BSSID: The BSSID of this neighbor channel.

Channel Number: The channel number of this neighbor channel.

2rd channel offset: TBD.

Mean RSSI: The average signal strength of the channel.

Sta Intf: TBD.

AP Intf: TBD.

## 6. Security Considerations

TBD

## 7. IANA Considerations

None

## 8. 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

## 9. Acknowledgements

The authors would like to thanks Ronald Bonica and Benoit Claise for their usefull suggestions.

## 10. Normative References

- [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.

Authors' Addresses

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

Email: chen yifan@chinamobile.com

Dapeng Liu  
China Mobile  
No.32 Xuanwumen West Street  
Beijing 100053  
China

Email: liudapeng@chinamobile.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





IEEE RAC  
Internet Draft  
Intended status: Informational  
Expires: April 2014

G. Parsons  
Ericsson  
September 6, 2013

OUI Registry Restructuring  
draft-ieee-rac-oui-restructuring-01.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, and it may not be published except as an Internet-Draft.

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 9, 2014.

Copyright Notice

Copyright (c) 2013 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.

## Abstract

The IEEE Registration Authority Committee, which has oversight for the OUI based registries, is seeking IETF community input as it finalizes restructuring the OUI registries. This document provides background on the RAC as well as explaining the proposed restructuring and the rationale.

## Table of Contents

|   |    |
|---|----|
| 1. Introduction.....                        | 3  |
| 1.1. History of the IEEE RA and RAC.....    | 3  |
| 1.2. Mission Statement of the IEEE RAC..... | 4  |
| 2. Existing OUI based registries.....       | 4  |
| 2.1. OUI.....                               | 6  |
| 2.2. OUI-36.....                            | 7  |
| 2.3. IAB.....                               | 7  |
| 3. Common identifiers.....                  | 8  |
| 3.1. EUI-48.....                            | 8  |
| 3.2. EUI-64.....                            | 8  |
| 3.3. Company ID / Protocol identifier.....  | 8  |
| 4. Preventing exhaustion.....               | 9  |
| 4.1. IEEE RAC Prime Directive.....          | 9  |
| 4.2. New devices.....                       | 10 |
| 4.3. Assignment efficiencies.....           | 10 |
| 4.3.1. MAC (EUI48) Addressing.....          | 10 |
| 4.3.2. Company ID.....                      | 10 |
| 4.4. Virtualization.....                    | 10 |
| 4.4.1. Reusing addresses.....               | 11 |
| 4.4.2. EUI-128 addresses.....               | 11 |
| 5. Proposed new OUI-based registries.....   | 12 |
| 5.1. OUI-24: MAC Addresses - Large.....     | 14 |
| 5.2. OUI-28: MAC Addresses - Medium.....    | 14 |
| 5.3. OUI-36: MAC Addresses - Small.....     | 15 |
| 5.4. CompanyID.....                         | 15 |
| 5.4.1. Application Note.....                | 16 |
| 6. Protocol Considerations.....             | 16 |
| 7. Security Considerations.....             | 16 |
| 8. IANA Considerations.....                 | 16 |
| 9. Conclusions.....                         | 17 |
| 10. References.....                         | 17 |
| 10.1. Normative References.....             | 17 |

|                                   |    |
|-----------------------------------|----|
| 10.2. Informative References..... | 17 |
| 11. Acknowledgments.....          | 17 |

## 1. Introduction

The IEEE Registration Authority (RA) operates under the direction of the IEEE-SA Board of Governors. IEEE is recognized by ISO/IEC as the authorized Registration Authority to provide this service world-wide. The IEEE Registration Authority Committee (RAC) provides technical oversight for the IEEE Registration Authority Activities.

The IEEE RA administers the assignment of 24-bit identifiers, formally known as an "Organizationally Unique Identifier" (OUI). It can be used alone as an identifier, or used to create MAC Addresses, Bluetooth Device Addresses or Ethernet Addresses.

Given the possibility of consuming all the MAC addresses, the IEEE RAC places restrictions on their use. While the number space is large, it is not inexhaustible, and the IEEE-RAC reviews trends to determine if a new strategy is required to prevent exhaustion. Current usage trends and new applications have convinced the RAC that measures are needed to more efficiently use the MAC address space. This document presents the background as well as the proposed changes to the OUI registries.

### 1.1. History of the IEEE RA and RAC

The IEEE Registration Authority (RA) was formed by the IEEE Standards Board in 1986 at the initiative of the IEEE P802 (LAN/MAN) standards group in order to register Organizationally Unique Identifiers (OUI). Since that time, the activities of the Registration Authority have continued to expand.

The IEEE Registration Authority Committee (IEEE RAC) was formed in 1991 as a volunteer oversight of the IEEE staff operated RA. In 1998, the IEEE RAC became a committee of the IEEE Standards Association Board of Governors, (IEEE SA BoG).

In 1997, the IEEE Registration Authority assumed responsibility for the registration of EtherType Fields, as defined in the current edition of IEEE Std 802.3, and in 1998 began administering Individual Address Block assignments in an effort to preserve the OUI assignments and offer the option of obtaining a smaller amount of addresses.

In 2003, it assumed responsibility for administering, allocating and managing the Logical Link Control (LLC) and Standard Group MAC addresses. IEEE has become the single point of contact with respect to all information associated with LAN addresses.

In 2004, IEEE established three registration authorities associated with IEEE 1451.4-2004. They are:

- o Unique Registration Numbers (URNS)
- o IEEE Templates and TDL Items
- o Manufacturer\_ID

On 27 April 2007, three additional registries were launched. Unlike the registries launched in 2004, each registry represents a different IEEE standard.

- o OUI-36
- o IEEE 802.16 Operator ID
- o Provider Service Identifier (PSID)

The IEEE Registration Authority formerly had administrative responsibility for the IEEE POSIX Certification.

## 1.2. Mission Statement of the IEEE RAC

The IEEE Registration Authority Committee (IEEE RAC) is the oversight committee for the IEEE Registration Authority.

The IEEE RAC is international in scope, assisting standard developing organizations in their establishment of unambiguous, sustainable registration authorities.

The IEEE RAC considers the long-term interests of the ultimate users of these standards, while pragmatically addressing the needs of the affected organizations, industries, and the IEEE.

## 2. Existing OUI based registries

The OUI ("Organizationally Unique Identifier") is defined in IEEE Std 802-2001 [1] and its structure is shown in Figure 1 below

with an example for use as a protocol identifier shown in Figure 2.

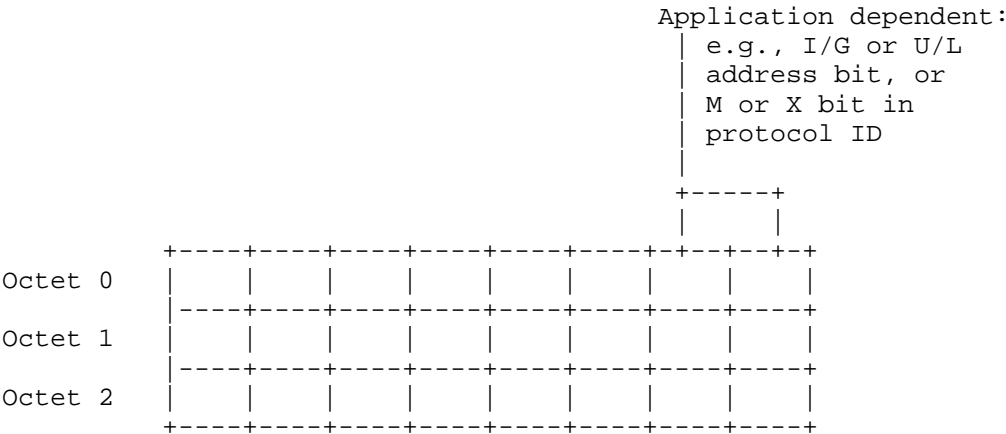


Figure 1 - Structure of an OUI

Of note is that only 22 bits are actually assigned as there are specific uses for the first two bits transmitted (the two least significant bits of octet 0). As a MAC address, the first bit transmitted indicates either an individual or group address (I/G), and the second bit transmitted indicates universal or local administration of the address (U/L). When used as a protocol identifier (Figure 2), these bits are the M and X bits. As a result of these uses, all previous OUI assignments have set these two bits to 0.

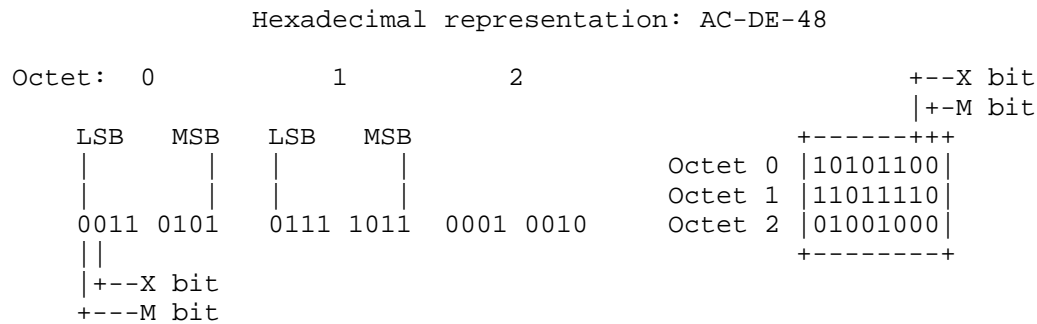


Figure 2 - Format of an OUI used as protocol identifier

While the majority of customers purchase the OUI, there are currently three OUI based registries:

1. OUI
2. OUI-36
3. IAB

The latter two use an IEEE reserved OUI from the first registry as their root.

These registries support the standards of IEEE 802 as well as ISO/IEC 8802 and other standards that use unique LAN addresses. IEEE has been authorized by the ISO Council to act as the exclusive registration authority for the implementation of International Standards in the ISO/IEC 8802 series.

### 2.1. OUI

An OUI or 'company\_id' is a 24-bit globally unique assigned number referenced by various standards. The OUI is usually concatenated with 24 or 40 bits by an Organization to create a 48-bit or 64-bit number that is unique to a particular piece of hardware. It can be used to create MAC Addresses, Bluetooth Device Addresses or Ethernet Addresses.

There are other uses of the OUI as well, such as its use as a company identifier in the SNAP protocol.

The OUI or 'company\_id' can be used in conjunction with a number of standards. It does not limit your right to use your assignment for both OUI and 'company\_id' purposes.

Additional information can be found on the IEEE RA website:  
<http://standards.ieee.org/develop/regauth/oui/index.html>

## 2.2. OUI-36

OUI-36 is a 36-bit identifier that can be used as an Individual Address Block (IAB) or as an extended OUI. The OUI-36 may be appended with four organization-supplied bits to form a 40-bit Context Dependent Identifier (CDI-40), with twelve organization-supplied bits to form an EUI-48, or with organization-supplied 28 bits to form an EUI-64. Applications making use of an OUI-36 should make no assumptions about the bit pattern that will be present in the (24-bit most-significant) OUI portion of the assigned OUI-36.

Additional information can be found on the IEEE RA website:  
<http://standards.ieee.org/develop/regauth/oui36/index.html>

## 2.3. IAB

An IAB is for organizations that need less than 4097 unique 48-bit numbers (EUI-48) and thus find it hard to justify buying their own OUI. It is a particular OUI concatenated with 12 additional IEEE-provided bits, leaving only 12 bits for the owners to assign to their (up to 4096) individual devices.

Unlike an OUI, which allows the assignee to assign values in various different number spaces (for example, EUI-48, EUI-64, and the various CDI number spaces), the IAB can only be used to assign EUI-48 identifiers.

The Individual Address Block (IAB) can be used in conjunction with a number of standards. It does not limit your right to use your assignment for multiple purposes.

Additional information can be found on the IEEE RA website:  
<http://standards.ieee.org/develop/regauth/iab/index.html>



### 3. Common identifiers

The OUI defined in IEEE Std 802-2001 [1] can be used to generate 48-bit Universal LAN MAC addresses to uniquely identify Local and Metropolitan Area Networks stations, and Protocol Identifiers to identify public and private protocols. A revision [3] of this standard is underway (expecting to complete in late 2013) that will, among other updates, also describe the 64-bit address.

#### 3.1. EUI-48

The IEEE defined 48-bit extended unique identifier (EUI-48) is a concatenation of either a 24-bit Organizationally Unique Identifier (OUI) value administered by the IEEE Registration Authority (IEEE-RA) and a 24-bit extension identifier assigned by the organization with that OUI assignment, or the concatenation of a 36-bit Individual Address Block (IAB) identifier /or 36-bit Organizationally Unique Identifier (OUI-36)/ and a 12-bit extension identifier assigned by the organization with that IAB assignment.

Additional information can be found on the IEEE RA website:  
<http://standards.ieee.org/develop/regauth/tut/eui48.pdf>

#### 3.2. EUI-64

The IEEE-defined 64-bit extended unique identifier (EUI-64) is a concatenation of the Organizationally Unique Identifier (OUI) value assigned by the IEEE Registration Authority (IEEE RA) and the extension identifier assigned by the organization with that OUI assignment resulting in a 64-bit unique identifier. The extension identifiers shall be 40 bits for the 24-bit OUI-24 and 28 bits for the 36-bit OUI-36. Other OUI lengths will have extension identifiers making up the difference between each assigned OUI length and the 64-bit EUI-64.

Additional information can be found on the IEEE RA website:  
<http://standards.ieee.org/develop/regauth/tut/eui48.pdf>

#### 3.3. Company ID / Protocol identifier

IEEE Std 802 provides for the use of Protocol Identifiers in conjunction with the SNAP/SAP reserved LLC address. A Protocol Identifier is defined as a sequence of five octets. The first

three octets take the values of the three octets of the OUI in order; the following two octets are administered by the OUI assignee. The hexadecimal representation of the Protocol Identifier consists of the hexadecimal values of the five octets in order, separated by hyphens, in the order transmitted by the network application, left to right.

Additional information can be found on the IEEE RA website:  
<http://standards.ieee.org/develop/regauth/tut/lanman.pdf>

#### 4. Preventing exhaustion

Given the possibility of consuming all the MAC addresses, the IEEE RAC places restrictions on their use. For new applications, EUI-48 identifiers are restricted to use in low volume applications, such as the identification of software interface standards or hardware model numbers.

While the number of EUI-48 identifiers is large, it is not inexhaustible, and the IEEE-RAC reviews trends to determine if a new strategy is required to prevent exhaustion. Current usage trends and new applications have convinced the RAC that measures are needed to more efficiently use the EUI-48 address space.

##### 4.1. IEEE RAC Prime Directive

A "prime directive" of the IEEE RAC is to not run out of global EUI-48 addresses (previously called MAC or MAC-48 addresses) for 100 years. The clock started in 1980 when this space was created by Xerox (and was called Block ID at the time).

In about 30 years, less than 20,000 OUIs have been assigned. So if the growth is linear, there is more than 99% of the space left, giving the world a 4000 year supply. However, the growth trend from last few years is not linear. If that trend continues, then there is only 26 years left before exhaustion of OUIs and global address space they are used to create. The IEEE RAC is studying these trends and has considered several possible causes.

#### 4.2. New devices

There has been an increase in new device categories in the last several years - including smart phones, tablets and various sensors - all that have more than one network interface (e.g., WiFi, Bluetooth, Ethernet) that requires a MAC address.

In addition, there are a few manufacturers that are volume users. That is, they are using more than 32 million MAC (EUI-48) addresses per month.

#### 4.3. Assignment efficiencies

Most manufacturers, however, use far less MAC (EUI-48) addresses per month. They either have a smaller production volume or are just starting. And actually, most OUI customers have only bought one OUI. If they need only MAC addresses, then they could benefit from options that would offer them fewer.

This would reduce the many "lost" or "unused" MAC addresses from OUIs that were assigned but the manufacturer did not use the full 16 million.

##### 4.3.1. MAC (EUI48) Addressing

~260 billion EUI-48 (of ~70 trillion possible) addresses have been assigned. While the RAC knows these have not all be used in devices, there is no way to confirm this. The RAC does however, require that repeat customers confirm that they have used 95% of the addresses before they are assigned another OUI block.

The RAC requires that only one (or at most a few) global EUI-48 addresses be assigned to a single hardware device. This is to avoid stockpiling of addresses in devices. However, this may be problematic for some applications like virtualization

##### 4.3.2. Company ID

In order to get a Protocol identifier or company ID, an OUI must be assigned. If the manufacturer does not intend to use it for addressing, then those addresses are lost.

#### 4.4. Virtualization

Virtualization from the IEEE RAC perspective is essentially the usage of global MAC (EUI-48) addresses by software - instead of

by a hardware device (i.e., "burned in") as was originally intended.

Traditionally the RAC limited manufacturers to only a few addresses per hardware device to prevent stockpiling addresses in devices. This would invalidate virtualization solutions. As a result, the RAC is now allowing assignment of an OUI (16M EUI-48 addresses) for virtualization use until a further policy is clarified.

One requirement for virtual machines is that they are mobile and can be moved around on a rack, within a data center or even across data centers. Such movement in a multi-vendor environment requires a globally unique MAC (EUI48) address to be scalable.

#### 4.4.1. Reusing addresses

However, another inherent nature of virtualization is the creation and destruction of the virtual machine. Hundreds, thousands or millions can be created or destroyed per second in a data center. If kept in a closed environment, this requires a local or reusable MAC (EUI48) address. If a global address is used, then they could be used at an alarming rate as they are not defined as reusable.

Unfortunately, there appears to be violation of the IEEE RAC policy in the virtualization sector. That is, some are using global MAC (EUI-48) addresses per rack / cluster / data center and then reusing them in an adjacent rack / cluster / data center.

Clearly this is not permitted and the RAC has been studying what guidance should be given to virtualization vendors such that the global MAC address space is not tainted.

It has been suggested that a DHCP-like mechanism or a standard allocation should be developed for reusable MAC addresses such that there is some order to assignment in an environment where addresses are created and destroyed.

#### 4.4.2. EUI-128 addresses

Given the potential for using a large number of addresses, the RAC is also exploring the feasibility of defining a new "EUI-128" identifier (i.e., 128 bits) specifically for future virtualization applications.

## 5. Proposed new OUI-based registries

The IEEE RAC has been studying options to restructure the OUI-based registries and products for over a year and is now reviewing a final proposal. This proposal provides a refinement of the OUI-based registries improves efficiency of assignment allocations and attempts to address virtualization issues.

While there was some desire for the OUI registries to fully separate the semantic of protocol identifier (e.g., the 24 bits assigned) and addresses (e.g., a 48-bit address created based on the 24 bits assigned), the concern raised was that this was not enforceable by definition.

The IEEE RAC conducted a survey of its customers and it quickly became clear that there were first-time customers (and in most cases they never made another purchase) and repeat customers (many of who were volume users). It was also very clear that the dominant use was to create global MAC (EUI48) addresses. As a result, the assignment decisions could be separated as proposed in Figure 3 below.

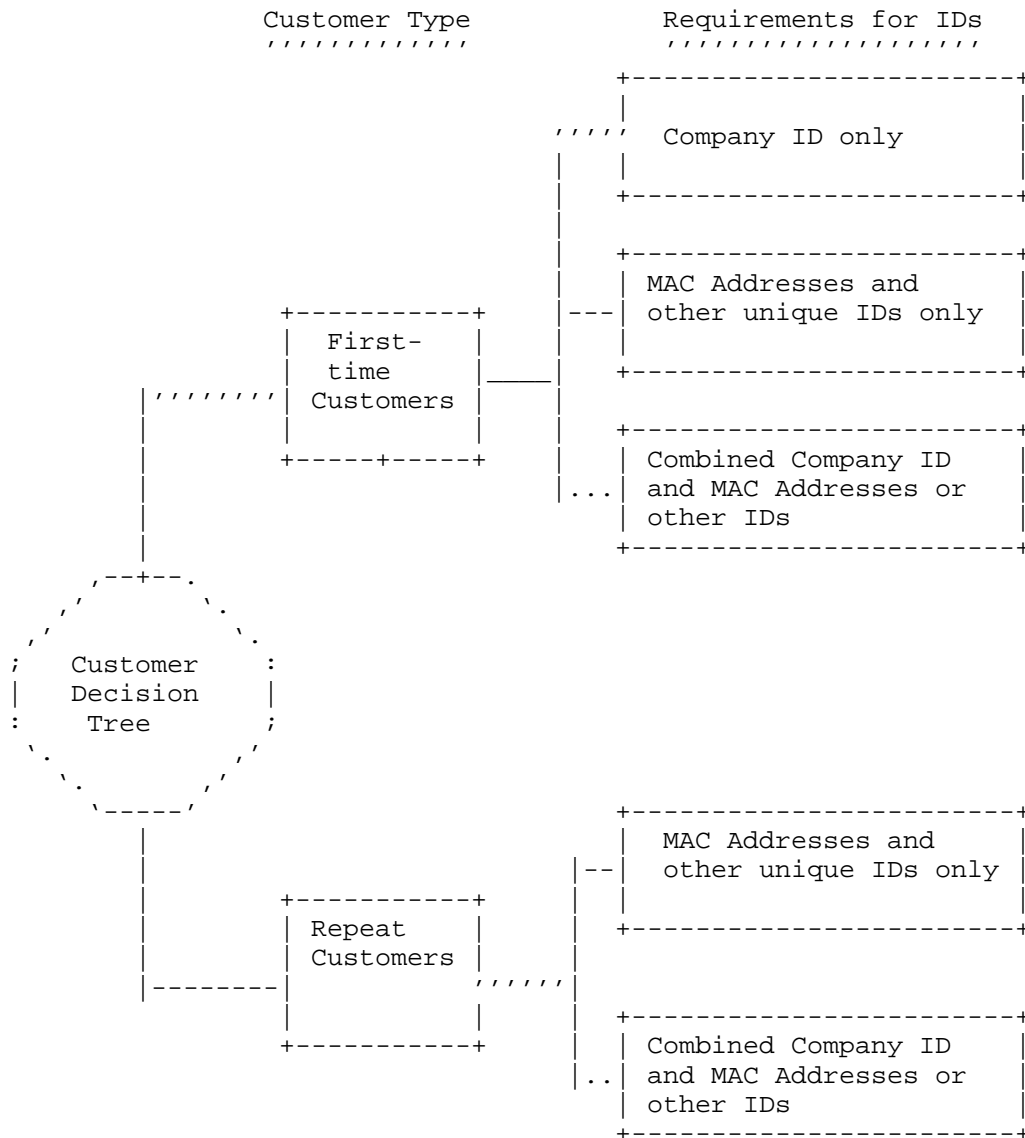


Figure 3: Decision Tree for assignment of Unique IDs/MAC addresses

The proposal that the IEEE RAC is considering is to add an additional size option for creating MAC (EUI48) addresses -- 1 million -- as well as creating a new CompanyID registry. This is

shown in Table 1 below and described in the following sub-sections.

Table 1: New Proposed OUI-based Product Registries

| Manufacturer field | Product     | EUI48(MAC)addresses |
|--------------------|-------------|---------------------|
| 24-bit identifier  | OUI-24/MA-L | 16777216            |
| -                  | OUI-28/MA-M | 1048576             |
| 36-bit identifier  | OUI-36/MA-S | 4096                |
| 24-bit identifier  | CompanyID   | -                   |

#### 5.1. OUI-24: MAC Addresses - Large

The OUI-24 is a 24-bit globally unique assigned number.

This is the base OUI registry. It is simply a renaming of the existing OUI registry.

An assignment from this registry includes the ability to create:

- o 24-bit company ID / protocol identifiers
- o 48-bit EUI48 addresses
- o 64-bit EUI64 addresses

#### 5.2. OUI-28: MAC Addresses - Medium

The OUI-28 is a 28-bit globally unique assigned number.

This new OUI-28 is created by the IEEE RA by assigning an additional 4 bits from an OUI-24 (that would be listed as IEEE reserved).

An assignment from this registry includes the ability to create:

- o 48-bit EUI48 addresses

- o 64-bit EUI64 addresses

Note that the IEEE RAC does not intend to define the usage of a 28-bit company ID / protocol identifier at this time.

### 5.3. OUI-36: MAC Addresses - Small

The OUI-36 is a 36-bit globally unique assigned number.

The OUI-36 is created by the IEEE RA by assigning an additional 12 bits from an OUI-24 (that is listed as IEEE reserved).

This is the existing OUI-36 registry, and it is proposed to merge the IAB registry with this as well.

An assignment from this registry includes the ability to create:

- o 36-bit company ID / protocol identifiers
- o 48-bit EUI48 addresses
- o 64-bit EUI64 addresses

### 5.4. CompanyID

The CompanyID is a 24-bit globally unique assigned number. However, any MAC addresses created with this Company ID would only be locally significant (i.e., the U/L bit is set to 1)

This new CompanyID is created by the IEEE RA assigning an OUI with the X bit set to 1 (this bit becomes the U/L bit when used to create a MAC address). Traditionally, this use has been reserved to separate the local and global address spaces but no use had been defined for protocol identifiers. It is proposed that only a segment (e.g., the bottom half) of the potential 22-bit space be made available for allocation.

An assignment from this registry includes the ability to create:

- o 24-bit company ID / protocol identifiers

NOTE: This requires that legacy uses of the OUI in protocols do not try to define the M and X bits for other uses. The RAC is not aware of any standard uses of the M and X bits that would prevent defining this new registry.



#### 5.4.1. Application Note

It is further proposed that virtualization manufacturers apply for assignments of these CompanyIDs. These could then be used to create MAC (EUI48) addresses in the local space that could be reused. Additionally, it would also provide some order and allow for multi-vendor usage of a subset of the local space for the virtualization application (or any application that could benefit from reusable addresses).

### 6. Protocol Considerations

There may be unintended consequences of these additions to the OUI-based registries for existing protocols. A study and review of many protocols was conducted and there were no apparent issues identified.

IETF community input is requested, especially as it relates to the embedded use or carriage of addresses or protocol identifiers in other protocols. For protocol identifiers, the IEEE RAC would be interested if any protocol defines the M and X bits for other uses.

### 7. Security Considerations

There may be unintended consequences of these additions to the OUI-based registries, though none are apparent.

IETF community input is requested.

### 8. IANA Considerations

There may be some affect on the existing IANA registries based on the restructuring of the OUI based registries.

However, this has not yet been studied.

IETF community input is requested.

## 9. Conclusions

While the background presented in this document is representative of the current situation, the proposals in this document have not yet been implemented, and therefore may change.

The IEEE-SA Board of Governors has made a decision, based on the recommendation of the IEEE RAC, on the implementation of the OUI registry restructuring. A summary has been provided in this document, but full details are under development by the IEEE RAC. It is expected that implementation will start in 2014.

IETF community input is requested to identify any issues with the restructuring proposal, especially as it affects IETF protocols. Please provide your comments to the RAC public list with "IETF community comment" as the start of the subject field:

STDS-RAC-PUBLIC@LISTSERV.IEEE.ORG

## 10. References

### 10.1. Normative References

[1] IEEE Std 802-2001, "IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture"  
<https://standards.ieee.org/about/get/802/802.html>

### 10.2. Informative References

[2] IEEE Registration Authority website  
<http://standards.ieee.org/develop/regauth/>

[3] IEEE P802 - Overview & Architecture revision project  
<http://www.ieee802.org/1/pages/802-rev.html>

## 11. Acknowledgments

The IEEE RAC appreciates the cooperation of IETF in publicizing these proposals to the IETF community including at its meetings.

Some of the background material in this document is based on information previously available on the IEEE RA website [2].

Authors Addresses

Glenn Parsons  
Ericsson

Phone: +1-613-963-8141  
Email: glenn.parsons@ericsson.com

Operations Area Working Group  
Internet-Draft  
Intended status: BCP  
Expires: April 21, 2013

F. Baker  
Cisco Systems  
P. Hoffman  
VPN Consortium  
October 18, 2012

On Firewalls in Internet Security  
draft-ietf-opsawg-firewalls-01

Abstract

This document discusses the most important operational and security implications of using modern firewalls in networks. It makes recommendations for operators of firewalls, as well as for firewall vendors.

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 21, 2013.

Copyright Notice

Copyright (c) 2012 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. Modern Firewall Features That Should Not Be Confused<br>with Firewalling . . . . . | 4  |
| 1.2. Terminology . . . . .  | 4  |
| 2. High-Level Firewall Concepts . . . . .   | 4  |
| 2.1. The End-to-End Principle . . . . .   | 4  |
| 2.2. Building a Communication . . . . .   | 5  |
| 3. Firewalling Strategies . . . . .   | 6  |
| 3.1. Blocking Traffic Unless It Is Explicitly Allowed . . . . .                         | 7  |
| 3.2. Typical Firewall Categories . . . . .  | 7  |
| 3.3. Newer categories of firewalling . . . . .  | 8  |
| 4. Recommendations for Operators . . . . .  | 8  |
| 5. Recommendations for Firewall Vendors . . . . .                                       | 8  |
| 6. IANA Considerations . . . . .  | 9  |
| 7. Security Considerations . . . . .  | 9  |
| 8. Acknowledgements . . . . .   | 9  |
| 9. References . . . . .   | 9  |
| 9.1. Normative References . . . . .   | 9  |
| 9.2. Informative References . . . . .   | 9  |
| Appendix A. IPv4 NATs Are Not Security Devices . . . . .                                | 10 |
| Appendix B. Origin Reputation and Firewalls . . . . .                                   | 10 |
| Authors' Addresses . . . . .  | 10 |

## 1. Introduction

In this document, a firewall is defined as a device or software that imposes a policy whose effect is "a stated type of packets may or may not pass from A to B". All modern firewalls allow an administrator to change the policies in the firewall, although the ease of administration for making those changes, and the granularity of the policies, vary widely between firewalls and vendors.

Given this definition, it is easy to see that there is a perimeter (the position between A and B) in which the specific security policy applies. In typical deployed networks, there are usually some easy-to-define perimeters. If two or more networks that are connected by a single device, the perimeter is inside the device. If that device is a firewall, it can impose a security policy at the shared perimeters of those networks.

Many firewalls also employ some perimeters that are not as easy to define. Some of these perimeters in modern firewalls include:

- o An application-layer gateway (ALG) in front of a server creates a perimeter between that server and the network it is connected to. The ALG blocks some of the flows in the application protocol based on policies such as "do not allow traffic from this network" and "do not allow the client to send a message of this type".
- o Routing domains that are controlled with role-based administration create perimeters in a routed network. Role-based administration makes rules such as "Domain X cannot see Domain Y in its routing table"; this prevents any host in Domain X from sending traffic to any host in Domain Y.
- o [[[ MORE HERE with other interesting perimeters ]]]

Modern firewalls apply perimeters at three layers:

Layer 3: Most firewalls can filter based on source and destination IPv4 addresses. Many (but, frustratingly, not all) firewalls can filter based on IPv6 addresses.

Layer 4: Most firewalls can filter based on TCP and UDP ports. Many (but, frustratingly, not all) firewalls can also filter based on transports other than TCP and UDP.

Layer 7: Modern firewalls can filter based on the application protocol contents, such as to allow or block certain types of protocol-defined messages, or based on the contents of those messages.

Note that many firewall devices can only create policies at one or two of the layers.

Hardware-based firewalls by their nature inspect traffic flowing through them, sometimes using proprietary mechanisms to make traffic analysis as fast as possible on the given hardware. Some firewalls use network visibility protocols such as NetFlow and sFlow to help capture and analyze traffic. [[ References needed ]]

### 1.1. Modern Firewall Features That Should Not Be Confused with Firewalling

There are a few features that appear in any firewall devices that have become associated with firewalls but in fact are not used for firewalling. Those non-firewalling features include:

Network Address Translation (NAT) [RFC2993], which is not used for security policy

IPsec [RFC4301], which is used for virtual private networks (VPNs). Although the core IPsec protocol has firewalling in it, when IPsec appears in a firewall device, it is normally only associated with the application of authenticated encryption and integrity protection of traffic.

"SSL VPN" is a set of technologies that rely on tunneling traffic through the TLS [RFC5246] protocol running on port 443. Some firewalls offer SSL VPNs as an alternative to IPsec.

Traffic prioritization is a feature common in firewalls, but does not meet the definition of firewalling at all.

### 1.2. 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].

Some terms which have specific meanings in this document (such as "firewall") are defined earlier in this section.

## 2. High-Level Firewall Concepts

### 2.1. The End-to-End Principle

One common complaint about firewalls in general is that they violate the End-to-End Principle [EndToEnd]. The End-to-End Principle is

often incorrectly stated as requiring that "application specific functions ought to reside in the end hosts of a network rather than in intermediary nodes, provided they can be implemented 'completely and correctly' in the end hosts" or that "there should be no state in the network."

What it actually says is heavily nuanced, and is a line of reasoning applicable when considering any two communication layers. The document says that it "presents a design principle that helps guide placement of functions among the modules of a distributed computer system. The principle, called the end-to-end argument, suggests that functions placed at low levels of a system may be redundant or of little value when compared with the cost of providing them at that low level."

In other words, the End-to-End Argument is not a prohibition against lower layer retries of transmissions, which can be important in certain LAN technologies, nor of the maintenance of state, nor of consistent policies imposed for security reasons. It is, however, a plea for simplicity. Any behavior of a lower communication layer, whether found in the same system as the higher layer (and especially application) functionality or in a different one, that from the perspective of a higher layer introduces inconsistency, complexity, or coupling extracts a cost. That cost may be in user satisfaction, difficulty of management or fault diagnosis, difficulty of future innovation, reduced performance, or other forms. Such costs need to be clearly and honestly weighed against the benefits expected, and used only if the benefit outweighs the cost.

From that perspective, introduction of a policy that prevents communication under an understood set of circumstances, whether it is to prevent access to pornographic sites or prevents traffic that can be characterized as an attack, does not fail the end to end argument; there are any number of possible sites on the network that are inaccessible at any given time, and the presence of such a policy is easily explained and understood.

What does fail the end-to-end argument is behavior that is intermittent, difficult to explain, or unpredictable. If I can sometimes reach a site and not at other times, or reach it using this host or application but not another, I wonder why that is true, and may not even know where to look for the issue.

## 2.2. Building a Communication

Any communication requires at least three components:



- o a sender, someone or some thing that sends a message,
- o a receiver, someone or some thing that receives the message, and
- o a channel, which is a medium by which the message is communicated.

In the Internet, the IP network is the channel; it may traverse something as simple as a directly connected cable or as complex as a sequence of ISPs, but it is the means of communication. In normal communications, a sender sends a message via the channel to the receiver, who is willing to receive and operate on it. In contrast, attacks are a form of harassment. A receiver exists, but is unwilling to receive the message, has no application to operate on it, or is by policy unwilling to. Attacks on infrastructure occur when message volume overwhelms infrastructure or uses infrastructure but has no obvious receiver.

By that line of reasoning, a firewall primarily protects infrastructure, by preventing traffic that would attack it from it. The best prophylactic might use a procedure for the dissemination of flow specification rules from [RFC5575] to drop traffic sent by an unauthorized or inappropriate sender or which has no host or application willing to receive it as close as possible to the sender.

In other words, as discussed in Section 1, a firewall compares to the human skin, and has as its primary purpose the prophylactic defense of a network. By extension, the firewall also protects a set of hosts and applications, and the bandwidth that serves them, as part of a strategy of defense in depth. A firewall is not itself a security strategy; the analogy to the skin would say that a body protected only by the skin has an immune system deficiency and cannot be expected to long survive. That said, every security solution has a set of vulnerabilities; the vulnerabilities of a layered defense is the intersection of the vulnerabilities of the various layers (e.g., a successful attack has to thread each layer of defense).

### 3. Firewalling Strategies

There is a great deal of tension in firewall policies between two primary goals of networking: the security goal of "block traffic unless it is explicitly allowed" and the networking goal of "trust hosts with new protocols". The two inherently cannot coexist easily in a set of policies for a firewall.

### 3.1. Blocking Traffic Unless It Is Explicitly Allowed

The security goal of "block traffic unless it is explicitly allowed" prevents useful new applications. This problem has been seen repeatedly over the past decade: a new and useful application protocol is deployed, but it cannot get wide adoption because it is blocked by firewalls. The result has been a tendency to try to run new protocols over established applications, particularly over HTTP [RFC3205]. The result is protocols that do not work as well they might if they were designed from scratch.

Worse, the same goal prevents the deployment of useful transports other than TCP, UDP, and ICMP. A conservative firewall that only knows those three transports will block new transports such as SCTP [RFC4960]; this in turn causes the Internet to not be able to grow in a healthy fashion. Many firewalls will also block TCP and UDP options they don't understand, and this has the same unfortunate result.

[[[ MORE HERE about forcing more costly and error-prone layer 7 inspection ]]]

### 3.2. Typical Firewall Categories

Most IPv4 firewalls have pre-configured security policies that fall into one of the following categories:

I: Block all outside-initiated traffic, allow all inside-initiated traffic

II: Same as I, but allow outside-initiated traffic to some specific inside hosts. The specified hosts are often added by IP address (or sometimes by DNS host name), and the host may be limited to particular transport and application protocols. For example, a rule might allow traffic destined to 203.0.113.226 on TCP ports 80 and 443.

III: Same as I or II, but allow some outside-initiated traffic over some protocols to all hosts. For example, a firewall protecting a farm of web servers might want to allow traffic using TCP ports 80 and 443 to all addresses protected by the firewall so that new servers can be deployed without having to update the firewall rules.

Firewalls that understand IPv6 may have a fourth category:

IV: Allow nearly all outside-initiated traffic. [[[ MORE HERE about why this is considered a good idea by some and a bad idea by

others ]]]]

### 3.3. Newer categories of firewalling

[[[ MORE HERE on blocking traffic based on dynamic origin reputation such as the long-expired vyncke-advanced-ipv6-security ]]]

## 4. Recommendations for Operators

[[[ MORE HERE with the following outline ]]]

Firewalling strategies

None. This is really the operator's choice.

Be aware that deep packet inspection causes varying amounts of delay in firewalls, particularly for long-lived flows

Don't enforce protocol semantics in the firewall

Applications are easier to change than firewalls

Avoid using application-layer gateways for firewalling

Use the security in the applications servers instead

Servers are easier to change than firewalls

However, ALGs are useful for IPv4-IPv6 conversion and proxying in some protocols

Allow fragments

Except in specific protocols where layer 7 content filtering is deemed crucial

Document your intended firewall strategy and settings

Be sure that other operators of the firewall are able to see it

Don't rely on a NAT for security (see Appendix A)

If using IPsec or SSL VPN, test whether the filtering rules for the rest of the firewall apply

## 5. Recommendations for Firewall Vendors

[[[ MORE HERE with the following outline ]]]

Make a set of NAT-like rules for IPv6 easily choosable

Interface for pinholing of IPv4 NATs needs clearly identify security issues

Follow the BEHAVE RFC rules for binding timeouts on NATs

Keep a summary log of non-normal events to aid reviewing

Make leaving notes about the firewalling rules easy and useful

Implement draft-ietf-pcp-base and probably the follow-on protocols from that WG

## 6. IANA Considerations

None.

## 7. Security Considerations

This document is all about security considerations. It introduces no new ones.

## 8. Acknowledgements

Warren Kumari commented on this document.

## 9. References

### 9.1. Normative References

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

### 9.2. Informative References

[EndToEnd]

Saltzer, JH., Reed, DP., and DD. Clark, "End-to-end arguments in system design", ACM Transactions on Computer Systems (TOCS) v.2 n.4, p277-288, Nov 1984.

[RFC2993] Hain, T., "Architectural Implications of NAT", RFC 2993, November 2000.

[RFC3205] Moore, K., "On the use of HTTP as a Substrate", BCP 56, RFC 3205, February 2002.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC4960] Stewart, R., "Stream Control Transmission Protocol", RFC 4960, September 2007.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC5575] Marques, P., Sheth, N., Raszuk, R., Greene, B., Mauch, J., and D. McPherson, "Dissemination of Flow Specification Rules", RFC 5575, August 2009.

## Appendix A. IPv4 NATs Are Not Security Devices

Their security is a side-effect of their design. [[[ MORE HERE about the history and why some operators mistake the security policy of NATs with firewalls. ]]]

[[[ MORE HERE about how pinholes mess badly that security policy. ]]]

[[[ MORE HERE about PCP and how to integrate it with a firewall security policy. ]]]

Recommendations for deploying NATs in firewalls include:

- o NATs should only be used when more IPv4 addresses are needed
- o Operators should not pinhole to addresses that are unpredictably assigned by DHCP

## Appendix B. Origin Reputation and Firewalls

[[[ MORE HERE with the following outline ]]]

Letting someone else curate your security policy  
Different types of reputation for different layers  
draft-ietf-repute-model  
draft-vyncke-advanced-ipv6-security  
draft-hallambaker-omnibroker  
Recommendations

- Check logs to be sure updates are happening
- Check vendors' policies

## Authors' Addresses

Fred Baker  
Cisco Systems

Email: fred@cisco.com

Paul Hoffman  
VPN Consortium

Email: paul.hoffman@vpnc.org



OPSAWG  
Internet-Draft  
Intended status: Informational  
Expires: October 15, 2014

V. Kuarsingh, Ed.  
J. Cianfarani  
Rogers Communications  
April 13, 2014

CGN Deployment with BGP/MPLS IP VPNs  
draft-ietf-opsawg-lsn-deployment-06

Abstract

This document specifies a framework to integrate a Network Address Translation layer into an operator's network to function as a Carrier Grade NAT (also known as CGN or Large Scale NAT). The CGN infrastructure will often form a NAT444 environment as the subscriber home network will likely also maintain a subscriber side NAT function. Exhaustion of the IPv4 address pool is a major driver compelling some operators to implement CGN. Although operators may wish to deploy IPv6 to strategically overcome IPv4 exhaustion, near term needs may not be satisfied with an IPv6 deployment alone. This document provides a practical integration model which allows the CGN platform to be integrated into the network, meeting the connectivity needs of the subscriber while being mindful of not disrupting existing services and meeting the technical challenges that CGN brings. The model included in this document utilizes BGP/MPLS IP VPNs which allow for virtual routing separation helping ease the CGNs impact on the network. This document does not intend to defend the merits of CGN.

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 October 15, 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. 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. Terms . . . . .  | 3  |
| 2. Existing Network Considerations . . . . .  | 4  |
| 3. CGN Network Deployment Requirements . . . . .  | 4  |
| 3.1. Centralized versus Distributed Deployment . . . . .                                | 5  |
| 3.2. CGN and Traditional IPv4 Service Co-existence . . . . .                            | 6  |
| 3.3. CGN By-Pass . . . . .  | 6  |
| 3.4. Routing Plane Separation . . . . .   | 7  |
| 3.5. Flexible Deployment Options . . . . .  | 7  |
| 3.6. IPv4 Overlap Space . . . . .   | 7  |
| 3.7. Transactional Logging for CGN Systems . . . . .                                    | 8  |
| 3.8. Base CGN Requirements . . . . .  | 8  |
| 4. BGP/MPLS IP VPN based CGN Framework . . . . .  | 8  |
| 4.1. Service Separation . . . . .   | 10 |
| 4.2. Internal Service Delivery . . . . .  | 11 |
| 4.2.1. Dual Stack Operation . . . . .   | 13 |
| 4.3. Deployment Flexibility . . . . .   | 15 |
| 4.4. Comparison of BGP/MPLS IP VPN Option versus other CGN Attachment Options . . . . . | 15 |
| 4.4.1. Policy Based Routing . . . . .   | 15 |
| 4.4.2. Traffic Engineering . . . . .  | 16 |
| 4.4.3. Multiple Routing Topologies . . . . .  | 16 |
| 4.5. Multicast Considerations . . . . .   | 16 |
| 5. Experiences . . . . .  | 16 |
| 5.1. Basic Integration and Requirements Support . . . . .                               | 16 |
| 5.2. Performance . . . . .  | 17 |
| 6. IANA Considerations . . . . .  | 17 |
| 7. Security Considerations . . . . .  | 17 |
| 8. BGP/MPLS IP VPN CGN Framework Discussion . . . . .                                   | 17 |
| 9. Acknowledgements . . . . .   | 18 |
| 10. References . . . . .  | 18 |



|  |    |
|--|----|
| 10.1. Normative References . . . . .   | 18 |
| 10.2. Informative References . . . . . | 18 |
| Authors' Addresses . . . . .           | 19 |

## 1. Introduction

Operators are faced with near term IPv4 address exhaustion challenges. Many operators may not have a sufficient amount of IPv4 addresses in the future to satisfy the needs of their growing subscriber base. This challenge may also be present before or during an active transition to IPv6 somewhat complicating the overall problem space.

To face this challenge, operators may need to deploy CGN (Carrier Grade NAT) as described in [RFC6888] to help extend the connectivity matrix once IPv4 address caches run out on the local local operator. CGN deployments will most often be added into operator networks which already have active IPv4 and/or IPv6 services.

The addition of the CGN introduces an operator controlled and administered translation layer which should be added in a manner which minimizes disruption to existing services. The CGN system addition may also include interworking in a dual stack environment where the IPv4 path requires translation.

This document shows how BGP/MPLS IP VPNs as described in [RFC4364] can be used to integrate the CGN infrastructure solving key integration challenges faced by the operator. This model has also been tested and validated in real production network models and allows fluid operation with existing IPv4 and IPv6 services.

### 1.1. Terms

A list of acronyms used throughout this document are defined in list below.

CGN - Carrier Grade NAT

DOCSIS - Data Over Cable Service Interface Specification

CMTS - Cable Modem Termination System

DSL -Digital subscriber line

BRAS - Broadband Remote Access Server

GGSN - Gateway GPRS Support Node

GPRS - General Packet Radio Service

ASN-GW - Access Service Network Gateway

GRT - Global Routing Table

Internal Realm - Addressing and/or network zone between the CPE and CGN as specified in [RFC6888]

External Realm - Public side network zone and addressing on the Internet facing side of the CGN as specified in [RFC6888]

## 2. Existing Network Considerations

The selection of CGN may be made by an operator based on a number of factors. The overall driver to use CGN may be the depletion of IPv4 address pools which leaves little to no addresses for a growing IPv4 service or connection demand growth. IPv6 is considered the strategic answer for IPv4 address depletion; however, the operator may independently decide that CGN is needed to supplement IPv6 and address their particular IPv4 service deployment needs.

If the operator has chosen to deploy CGN, they should do this in a manner as not to negatively impact the existing IPv4 or IPv6 subscriber base. This will include solving a number of challenges since subscribers whose connections require translation will have network routing and flow needs which are different from legacy IPv4 connections.

## 3. CGN Network Deployment Requirements

If a service provider is considering a CGN deployment with a provider NAT44 function, there are a number of basic architectural requirements which are of importance. Preliminary architectural requirements may require all or some of those captured in the list below. Each of the architectural requirement items listed are expanded upon in the following subsections. It should be noted that architectural CGN requirements add additive to base CGN functional requirements in [RFC6888]. The assessed architectural requirements for deployment are:

- Support distributed (sparse) and centralized (dense) deployment models;
- Allow co-existence with traditional IPv4 based deployments, which provide global scoped IPv4 addresses to CPEs;

- Provide a framework for CGN by-pass supporting non-translated flows between endpoints within a provider's network;
- Provide a routing framework which allows the segmentation of routing control and forwarding paths between CGN and non-CGN mediated flows;
- Provide flexibility for operators to modify their deployments over time as translation demands change (connections, bandwidth, translation realms/zones and other vectors);
- Flexibility should include integration options for common access technologies such as DSL (BRAS), DOCSIS (CMTS), Mobile (GGSN/PGW/ASN-GW), and direct Ethernet;
- Support deployment modes that allow for IPv4 address overlap within the operator's network (between various translation realms or zones);
- Allow for evolution to future dual-stack and IPv4/IPv6 transition deployment modes;
- Transactional logging and export capabilities to support auxiliary functions including abuse mitigation;
- Support for stateful connection synchronization between translation instances/elements (redundancy);
- Support for CGN Shared Space [RFC6598] deployment modes if applicable;
- Allows for the enablement of CGN functionality (if required) while still minimizing costs and subscriber impact to the best extend possible;

Other requirements may be assessed on a operator-by-operator basis, but those listed above may be considered for any given deployment architecture.

### 3.1. Centralized versus Distributed Deployment

Centralized deployments of CGN (longer proximity to end user and/or higher densities of subscribers/connections to CGN instances) differ from distributed deployments of CGN (closer proximity to end user and/or lower densities of subscribers/connections to CGN instances). Service providers may likely deploy CGN translation points more centrally during initial phases if the early system demand is low. Early deployments may see light loading on these new systems since

legacy IPv4 services will continue to operate with most endpoints using globally unique IPv4 addresses. Exceptional cases which may drive heavy usage in initial stages may include operators who already translate a significant portion of their IPv4 traffic; may transition to a CGN implementation from legacy translation mechanisms (i.e. traditional firewalls); or build a green field deployment which may see quick growth in the number of new IPv4 endpoints which require Internet connectivity.

Over time, some providers may need to expand and possibly distribute the translation points if demand for the CGN system increases. The extent of the expansion of the CGN infrastructure will depend on factors such as growth in the number of IPv4 endpoints, status of IPv6 content on the Internet and the overall progress globally to an IPv6-dominate Internet (reducing the demand for IPv4 connectivity). The overall demand for CGN resources will probably follow a bell-like curve with a growth, peak and decline period.

### 3.2. CGN and Traditional IPv4 Service Co-existence

Newer CGN serviced endpoints will exist alongside endpoints served by traditional IPv4 globally routed IPv4 addresses. Operators will need to rationalize these environments since both have distinct forwarding needs. Traditional IPv4 services will likely require (or be best served) direct forwarding towards Internet peering points while CGN mediated flows require access to a translator. CGN and non-CGN mediated flows pose two fundamentally different forwarding needs.

The new CGN environments should not negatively impact the existing IPv4 service base by forcing all traffic to translation enabled network points since many flows do not require translation and this would reduce performance of the existing flows. This would also require massive scaling of the CGN which is a cost and efficiency concern as well.

Traffic flow and forwarding efficiency is considered important since networks are under considerable demand to deliver more and more bandwidth without the luxury of needless inefficiencies which can be introduced with CGN.

### 3.3. CGN By-Pass

The CGN environment is only needed for flows with translation requirements. Many flows which remain within the operator's network, do not require translation. Such services include operator offered DNS Services, DHCP Services, NTP Services, Web Caching, E-Mail, News and other services which are local to the operator's network.

The operator may want to leverage opportunities to offer third parties a platform to also provide services without translation. CGN by-pass can be accomplished in many ways, but a simplistic, deterministic and scalable model is preferred.

### 3.4. Routing Plane Separation

Many operators will want to engineer traffic separately for CGN flows versus flows which are part of the more traditional IPv4 environment. Many times the routing of these two major flow types differ, therefore route separation may be required.

Routing plane separation also allows the operator to utilize other addressing techniques, which may not be feasible on a single routing plane. Such examples include the use of overlapping private address space [RFC1918], Shared Address Space [RFC6598] or use of other IPv4 space which may overlap globally within the operator's network.

### 3.5. Flexible Deployment Options

Service providers operate complex routing environments and offer a variety of IPv4 based services. Many operator environments utilize distributed peering infrastructures for transit and peering and these may span large geographical areas and regions. A CGN solution should offer the operator an ability to place CGN translation points at various points within their network.

The CGN deployment should also be flexible enough to change over time as demand for translation services increase or change as noted in [RFC6264]. In turn, the deployment will need to then adapt as translation demand decreases caused by the transition of flows to IPv6. Translation points should be able to be placed and moved with as little re-engineering effort as possible minimizing the risks to the subscriber base.

Depending on hardware capabilities, security practices and IPv4 address availability, the translation environments may need to be segmented and/or scaled over time to meet organic IPv4 demand growth. Operators may also want to choose models that support transition to other translation environments such as DS-Lite [RFC6333] and/or NAT64 [RFC6146]. Operators will want to seek deployment models which are conducive to meeting these goals as well.

### 3.6. IPv4 Overlap Space

IPv4 address overlap for CGN translation realms may be required if insufficient IPv4 addresses are available within the operator environment to assign internally unique IPv4 addresses to the CGN

subscriber base . The CGN deployment should provide mechanisms to manage IPv4 overlap if required.

### 3.7. Transactional Logging for CGN Systems

CGNs may require transactional logging since the source IP and related transport protocol information is not easily visible to external hosts and system.

If needed, the CGN systems should be able to generate logs which identify internal realm host parameters (i.e. IP/Port) and associated them to external realm parameters imposed by the translator. The logged information should be stored on the CGN hardware and/or exported to another system for processing. The operator may choose to also enable mechanisms to help reduce logging such as block allocation of UDP and TCP ports or deterministic translation options such as [I-D.donley-behave-deterministic-cgn].

Operators may be legally obligated to keep track of translation information. The operator may need to utilize their standard practices in handling sensitive customer data when storing and/or transporting such data. Further information can be found in [RFC6888] with respect to CGN logging requirements (Logging section).

### 3.8. Base CGN Requirements

Whereas the requirements above represent assessed architectural requirements, the CGN platform will also need to meet the need to meet the base CGN requirements of a CGN function. Base requirements include such functions as Bulk Port Allocation and other CGN device specific functions. These base CGN platform requirements are captured within [RFC6888].

## 4. BGP/MPLS IP VPN based CGN Framework

The BGP/MPLS IP VPN [RFC4364] framework for CGN segregates the internal realms within the service provider space into Layer-3 MPLS based VPNs. The operator can deploy a single realm for all CGN based flows, or can deploy multiple realms based on translation demand and other factors such as geographical proximity. A realm in this model refers to a 'VPN' which shares a unique Route Distinguisher/Route Target (RD/RT) combination, routing plane and forwarding behaviours.

The BGP/MPLS IP VPN infrastructure provides control plane and forwarding separation for the traditional IPv4 service environment and CGN environment(s). The separation allows for routing information (such as default routes) to be propagated separately for CGN and non-CGN based subscriber flows. Traffic can be efficiently

routed to the Internet for normal flows, and routed directly to translators for CGN mediated flows. Although many operators may run a "default-route-free" core, IPv4 flows which require translation must obviously be routed first to a translator, so a default route is acceptable for the internal realms.

The physical location of the Virtual Routing and Forwarding (VRF) Termination point for a BGP/MPLS IP VPN enabled CGN can vary and be located anywhere within the operator's network. This model fully virtualizes the translation service from the base IPv4 forwarding environment which will likely be carrying Internet bound traffic. The base IPv4 environment can continue to service traditional IPv4 subscriber flows plus post translated CGN flows.

Figure 1 provides a view of the basic model. The Access node provides CPE access to either the CGN VRF or the Global Routing Table, depending on whether the subscriber receives a private or public IP. Translator mediated traffic follows an MPLS Label-switched Path (LSP) which can be setup dynamically and can span one hop, or many hops (with no need for complex routing policies). Traffic is then forwarded to the translator (shown below) which can be an external appliance or integrated into the VRF Termination (Provider Edge) router. Once traffic is translated, it is forwarded to the global routing table for general Internet forwarding. The Global Routing table can also be a separate VRF (Internet Access VPN/VRF) should the provider choose to implement their Internet based services in that fashion. The translation services are effectively overlaid onto the network, but are maintained within a separate forwarding and control plane.

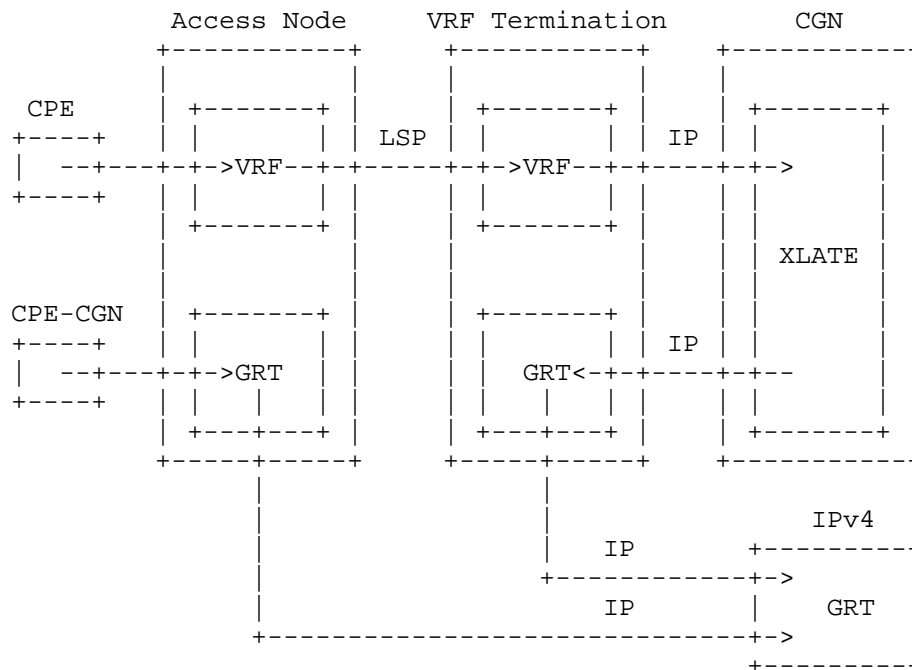


Figure 1: Basic BGP/MPLS IP VPN CGN Model

If more than one VRF (translation realm) is used within the operator's network, each VPN instance can manage CGN flows independently for the respective realm. The described architecture does not prescribe a single redundancy model that ensures network availability as a result of CGN failure. Deployments are able to select a redundancy model that fits best with their network design. If state information needs to be passed or maintained between hardware instances, the vendor would need to enable this feature in a suitable manner.

#### 4.1. Service Separation

The MPLS/VPN CGN framework supports route separation. The traditional IPv4 flows can be separated at the access node (Initial Layer 3 service point) from those which require translation. This type of service separation is possible on common technologies used for Internet access within many operator networks. Service separation can be accomplished on common access technology including those used for DOCSIS (CMTS), Ethernet Access, DSL (BRAS), and Mobile Access (GGSN/ASN-GW) architectures.



#### 4.2. Internal Service Delivery

Internal services can be delivered directly to the privately addressed endpoint within the CGN domain without translation. This can be accomplished in one of two methods. The first method may include reducing the overall number of VRFs in the system and exposing services in the GRT along with a method of exchanging routes between the CGN VRF and GRT called route leaking. The second method, which is described in detail within this section is the use of a Services VRF. The second model is a more traditional extranet services model, but requires more system resources to implement.

Using direct route exchange (import/export) between the CGN VRFs and the Services VRFs creates reachability using the aforementioned extranet model available in the BGP/MPLS IP VPN structure. This model allows the provider to maintain separate forwarding rules for translated flows, which require a pass through the translator to reach external network entities, versus those flows which need to access internal services. This operational detail can be advantageous for a number of reasons such as service access policies and endpoint identification.

First, the provider can reduce the load on the translator since internal services do not need to be factored into the scaling of the CGN hardware (which may be quite large). Secondly, more direct forwarding paths can be maintained providing better network efficiency. Thirdly, geographic locations of the translators and the services infrastructure can be deployed in locations in an independent manner. Additionally, the operator can allow CGN subject endpoints to be accessible via an untranslated path reducing the complexities of provider initiated management flows. This last point is of key interest since NAT removes transparency to the end device in normal cases.

Figure 2 below shows how internal services are provided untranslated since flows are sent directly from the access node to the services node/VRF via an MPLS LSP. This traffic is not forwarded to the CGN translator and therefore is not subject to problematic behaviours related to NAT. The services VRF contains routing information which can be "imported" into the access node VRF and the CGN VRF routing information can be "imported" into the Services VRF.

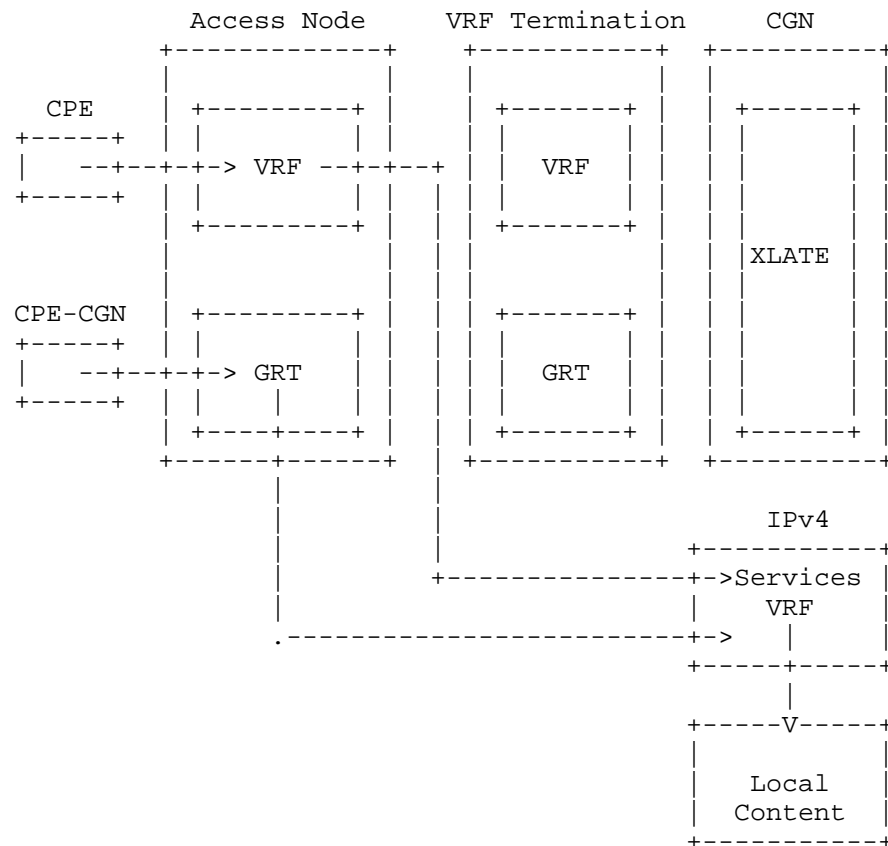
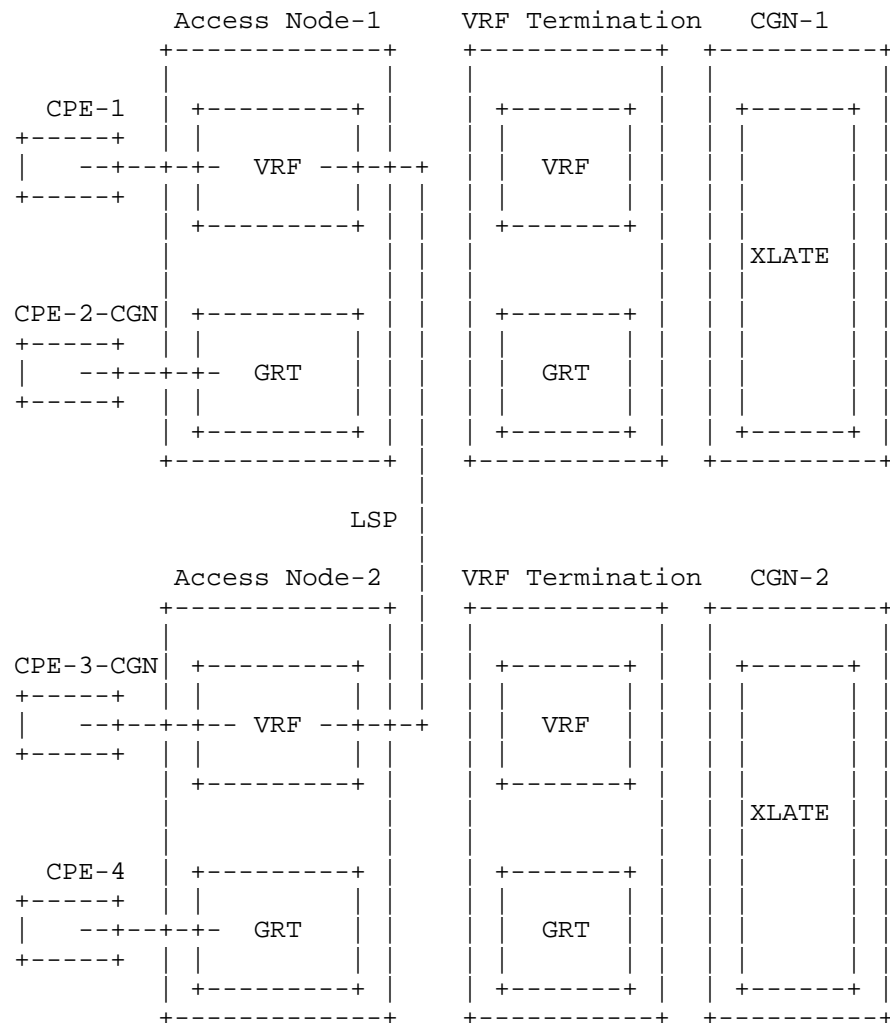


Figure 2: Internal Services and CGN By-Pass

An extension to the services delivery LSP is the ability to also provide direct subscriber to subscriber traffic flows between CGN zones. Each zone or realm may be fitted with separate CGN resources, but the subtending subscribers don't necessarily need to be mediated (translated) by the CGN translators. This option, as shown in Figure 3 below, is easy to implement and can only be enabled if no IPv4 address overlap is used between communicating CGN zones.



The inherent capabilities of the BGP/MPLS IP VPN model demonstrates the ability to offer CGN By-Pass in a standard and deterministic manner without the need of policy based routing or traffic engineering.

#### 4.2.1. Dual Stack Operation

The BGP/MPLS IP VPN CGN model can also be used in conjunction with IPv4/IPv6 dual stack service modes. Since many providers will use CGNs on an interim basis while IPv6 matures within the global Internet or due to technical constraints, a dual stack option is of strategic importance. Operators can offer this dual stack service

for both traditional IPv4 (global IP) endpoints and CGN mediated endpoints.

Operators can separate the IP flows for IPv4 and IPv6 traffic, or use other routing techniques to move IPv6 based flows towards the GRT (Global Routing Table or Instance) while allowing IPv4 flows to remain within the IPv4 CGN VRF for translator services.

The Figure 4 below shows how IPv4 translation services can be provided alongside IPv6 based services. The model shown allows the provider to enable CGN to manage IPv4 flows (translated) and IPv6 flows are routed without translation efficiently towards the Internet. Once again, forwarding of flows to the translator does not impact IPv6 flows which do not require this service.

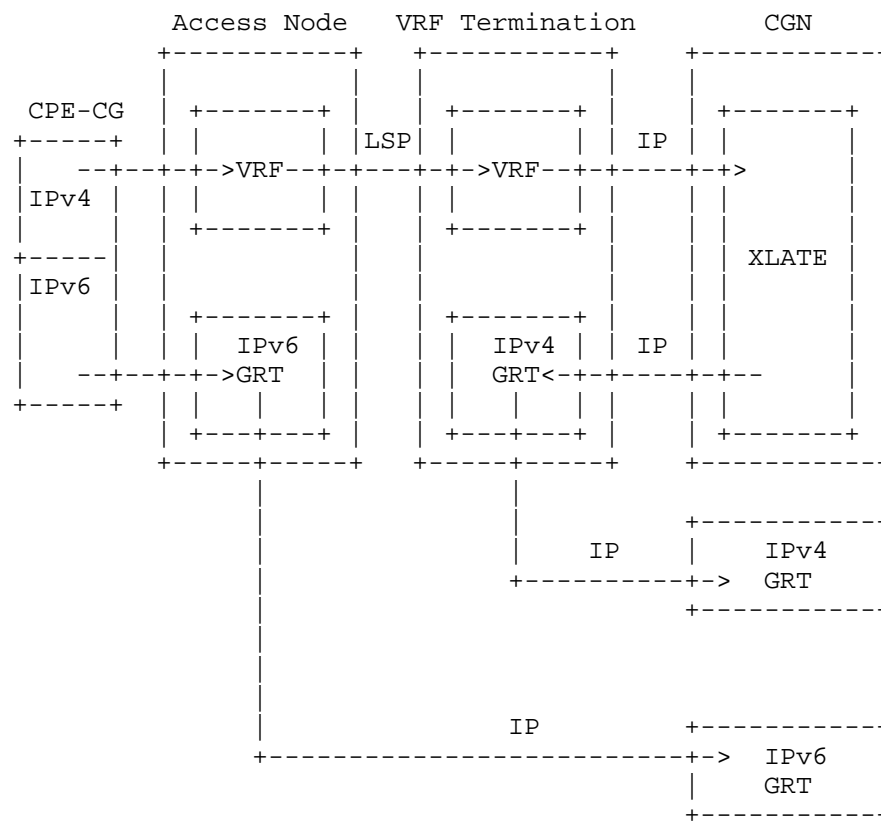


Figure 4: CGN with IPv6 Dual Stack Operation

#### 4.3. Deployment Flexibility

The CGN translator services can be moved, separated or segmented (new translation realms) without the need to change the overall translation design. Since dynamic LSPs are used to forward traffic from the access nodes to the translation points, the physical location of the VRF termination points can vary and be changed easily.

This type of flexibility allows the service provider to initially deploy more centralized translation services based on relatively low loading factors, and distribute the translation points over time to improve network traffic efficiencies and support higher translation load.

Although traffic engineered paths are not required within the MPLS/VPN deployment model, nothing precludes an operator from using technologies like MPLS with Traffic Engineering [RFC3031]. Additional routing mechanisms can be used as desired by the provider and can be seen as independent. There is no specific need to diversify the existing infrastructure in most cases.

#### 4.4. Comparison of BGP/MPLS IP VPN Option versus other CGN Attachment Options

Other integration architecture options exist which can attach CGN based service flows to a translator instance. Alternate options which can be used to attach such services include:

- Policy Based Routing (Static) to direct translation bound traffic to a network based translator;
- Traffic Engineering or;
- Multiple Routing Topologies

##### 4.4.1. Policy Based Routing

Policy Based Routing (PBR) provides another option to direct CGN mediated flows to a translator. PBR options, although possible, are difficult to maintain (static policy) and must be configured throughout the network with considerable maintenance overhead.

More centralized deployments may be difficult or too onerous to deploy using Policy Based Routing methods. Policy Based Routing would not achieve route separation (unless used with other options), and may add complexities to the providers' routing environment.

#### 4.4.2. Traffic Engineering

Traffic Engineering can also be used to direct traffic from an access node towards a translator. Traffic Engineering, like MPLS-TE, may be difficult to setup and maintain. Traffic Engineering provides additional benefits if used with MPLS by adding potentials for faster path re-convergence. Traffic Engineering paths would need to be updated and redefined overtime as CGN translation points are augmented or moved.

#### 4.4.3. Multiple Routing Topologies

Multiple routing topologies can be used to direct CGN based flows to translators. This option would achieve the same basic goal as the MPLS/VPN option but with additional implementation overhead and platform configuration complexity. Since operator based translation is expected to have an unknown lifecycle, and may see various degrees of demand (dependant on operator IPv4 Global space availability and shift of traffic to IPv6), it may be too large of an undertaking for the provider to enabled this as their primary option for CGN.

#### 4.5. Multicast Considerations

When deploying BGP/MPLS IP VPN's as an service method for user plane traffic to access CGN, one needs to be cognizant of current or future IP multicast requirements. User plane IP Multicast which may originate outside of the VRF requires more consideration specific consideration. Adding the requirement for user plane IP multicast can potentially cause additional complexity related to import and exporting the IP multicast routes in addition to sub optimal scaling, and bandwidth utilization.

It is recommended to reference best practice and designs from [RFC6037], [RFC6513], and [RFC5332]

### 5. Experiences

#### 5.1. Basic Integration and Requirements Support

The MPLS/VPN CGN environment has been successfully integrated into real network environments utilizing existing network service delivery mechanisms. It solves many issues related to provider based translation environments, while still subject to problematic behaviours inherent within NAT.

Key issues which are solved or managed with the MPLS/VPN option include:

- Centralized and Distributed Deployment model support
- Routing Plane Separation for CGN flows versus traditional IPv4 flows
- Flexible Translation Point Design (can relocate translators and split translation zones easily)
- Low maintenance overhead (dynamic routing environment with little maintenance of separate routing infrastructure other than management of MPLS/VPNs)
- CGN By-pass options (for internal and third party services which exist within the provider domain)
- IPv4 Translation Realm overlap support (can reuse IP addresses between zones with some impact to extranet service model)
- Simple failover techniques can be implemented with redundant translators, such as using a second default route

## 5.2. Performance

The MPLS/VPN CGN model was observed to support basic functions which are typically used by subscribers within an operator environment. A full review of the observed impacts related to CGN (NAT444) are covered in [RFC7021].

## 6. IANA Considerations

This document has no IANA actions.

## 7. Security Considerations

An operator implementing CGN using BGP/MPLS IP VPNs should refer to [RFC6888] section 7 for security considerations related to CGN deployments. The operator should continue to employ standard security methods in place for their standard MPLS deployment and can also refer to the security considerations section in [RFC4364] which discusses both control plane and data plane security.

## 8. BGP/MPLS IP VPN CGN Framework Discussion

The MPLS/VPN delivery method for a CGN deployment is an effective and scalable way to deliver mass translation services. The architecture avoids the complex requirements of traffic engineering and policy based routing when combining these new service flows to existing IPv4 operation. This is advantageous since the NAT44/CGN environments

should be introduced with as little impact as possible and these environments are expected to change over time.

The MPLS/VPN based CGN architecture solves many of this issues related to deploying this technology in existing operator networks.

## 9. Acknowledgements

Thanks to the following people for their comments and feedback: Dan Wing, Chris Metz, Chris Donley, Tina TSOU, Christophoer Liljenstolpe and Tom Taylor.

Thanks to the following people for their participating in integrating and testing the CGN environment and for their IPv6 transition guidance: Syd Alam, Richard Lawson, John E Spence, John Jason Brzozowski, Chris Donley, Jason Weil, Lee Howard, Jean-Francois Tremblay

## 10. References

### 10.1. Normative References

- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, February 2006.

### 10.2. Informative References

- [I-D.donley-behave-deterministic-cgn]  
Donley, C., Grundemann, C., Sarawat, V., Sundaresan, K., and O. Vautrin, "Deterministic Address Mapping to Reduce Logging in Carrier Grade NAT Deployments", draft-donley-behave-deterministic-cgn-07 (work in progress), January 2014.
- [RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.
- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001.
- [RFC5332] Eckert, T., Rosen, E., Aggarwal, R., and Y. Rekhter, "MPLS Multicast Encapsulations", RFC 5332, August 2008.
- [RFC5969] Townsley, W. and O. Troan, "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) -- Protocol Specification", RFC 5969, August 2010.



- [RFC6037] Rosen, E., Cai, Y., and IJ. Wijnands, "Cisco Systems' Solution for Multicast in BGP/MPLS IP VPNs", RFC 6037, October 2010.
- [RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, April 2011.
- [RFC6264] Jiang, S., Guo, D., and B. Carpenter, "An Incremental Carrier-Grade NAT (CGN) for IPv6 Transition", RFC 6264, June 2011.
- [RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, August 2011.
- [RFC6513] Rosen, E. and R. Aggarwal, "Multicast in MPLS/BGP IP VPNs", RFC 6513, February 2012.
- [RFC6598] Weil, J., Kuarsingh, V., Donley, C., Liljenstolpe, C., and M. Azinger, "IANA-Reserved IPv4 Prefix for Shared Address Space", BCP 153, RFC 6598, April 2012.
- [RFC6888] Perreault, S., Yamagata, I., Miyakawa, S., Nakagawa, A., and H. Ashida, "Common Requirements for Carrier-Grade NATs (CGNs)", BCP 127, RFC 6888, April 2013.
- [RFC7021] Donley, C., Howard, L., Kuarsingh, V., Berg, J., and J. Doshi, "Assessing the Impact of Carrier-Grade NAT on Network Applications", RFC 7021, September 2013.

#### Authors' Addresses

Victor Kuarsingh (editor)  
Rogers Communications  
8200 Dixie Road  
Brampton, Ontario L6T 0C1  
Canada

Email: [victor@jvknet.com](mailto:victor@jvknet.com)  
URI: <http://www.rogers.com>

John Cianfarani  
Rogers Communications  
8200 Dixie Road  
Brampton, Ontario L6T 0C1  
Canada

Email: [john.cianfarani@rci.rogers.com](mailto:john.cianfarani@rci.rogers.com)  
URI: <http://www.rogers.com>

Network Working Group  
Internet-Draft  
Updates: 5066 (if approved)  
Intended status: Standards Track  
Expires: June 13, 2014

E. Beili  
Actelis Networks  
December 10, 2013

Ethernet in the First Mile Copper (EFMCu) Interfaces MIB  
draft-ietf-opsawg-rfc5066bis-07.txt

Abstract

This document updates RFC 5066. It amends that specification by informing the internet community about the transition of the EFM-CU-MIB module from the concluded IETF Ethernet Interfaces and Hub MIB Working Group to the Institute of Electrical and Electronics Engineers (IEEE) 802.3 working group.

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 13, 2014.

Copyright Notice

Copyright (c) 2013 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. The Internet-Standard Management Framework . . . . .         | 3 |
| 3. Mapping between EFM-CU-MIB and IEEE8023-EFM-CU-MIB . . . . . | 3 |
| 4. Updating the MIB Modules . . . . .                           | 4 |
| 5. Security Considerations . . . . .                            | 4 |
| 6. IANA Considerations . . . . .                                | 5 |
| 7. Acknowledgments . . . . .                                    | 5 |
| 8. References . . . . .   | 5 |
| 8.1. Normative References . . . . .                             | 5 |
| 8.2. Informative References . . . . .                           | 6 |

## 1. Introduction

RFC 5066 [RFC5066] defines two MIB modules:

EFM-CU-MIB, with a set of objects for managing 10PASS-TS and 2BASE-TL Ethernet in the First Mile Copper (EFMCu) interfaces;

IF-CAP-STACK-MIB, with a set of objects describing cross-connect capability of a managed device with multi-layer (stacked) interfaces, extending the stack management objects in the Interfaces Group MIB and the Inverted Stack Table MIB modules.

With the conclusion of the [HUBMIB] working group, the responsibility for the maintenance and further development of a MIB module for managing 2BASE-TL and 10PASS-TS interfaces, has been transferred to the Institute of Electrical and Electronics Engineers (IEEE) 802.3 [IEEE802.3] working group. In 2011, the IEEE developed IEEE8023-EFM-CU-MIB module, based on the original EFM-CU-MIB module [RFC5066]. The current revision of IEEE8023-EFM-CU-MIB is defined in IEEE Std 802.3.1-2013 [IEEE802.3.1].

The IEEE8023-EFM-CU-MIB and EFM-CU-MIB MIB modules can coexist. Existing deployments of the EFM-CU-MIB need not be upgraded, but operators using the MIB should expect that new equipment will use the IEEE8023-EFM-CU-MIB.

Please note that IF-CAP-STACK-MIB module was not transferred to IEEE and remains as defined in RFC 5066. This memo provides an updated security considerations section for that module, since the original RFC did not list any security consideration for IF-CAP-STACK-MIB.

## 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].

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

## 3. Mapping between EFM-CU-MIB and IEEE8023-EFM-CU-MIB

The current version of IEEE8023-EFM-CU-MIB, defined in IEEE Std 802.3.1-2013, has MODULE-IDENTITY of ieee8023efmCuMIB with an object identifier allocated under the { org ieee standards-association-numbers-series-standards lan-man-stds ieee802dot3 ieee802dot3dot1mibs

} sub-tree.

The EFM-CU-MIB has MODULE-IDENTITY of efmCuMIB with an object identifier allocated under the mib-2 sub-tree.

The names of the objects in the first version of the IEEE8023-EFM-CU-MIB are identical to those in the EFM-CU-MIB. However, since both MIB modules have different OID values, they can coexist, allowing the management of the newer IEEE MIB-based devices, alongside the legacy IETF MIB-based devices.

#### 4. Updating the MIB Modules

With the transfer of the responsibility for maintenance and further development of the EFM-CU-MIB module to the IEEE 802.3 working group, the EFM-CU-MIB defined in RFC 5066 becomes the last version of that MIB module.

All further development of the EFM Copper Interfaces MIB will be done by the IEEE 802.3 working group in the IEEE8023-EFM-CU-MIB module. Requests and comments pertaining to EFM Copper Interfaces MIB should be sent to the IEEE 802.3.1 task force, currently chartered with MIB development, via its mailing list [LIST802.3.1].

The IF-CAP-STACK-MIB remains under IETF control and is currently maintained by the [OPSAWG] working group.

#### 5. Security Considerations

There are no managed objects defined in IF-CAP-STACK-MIB module with a MAX-ACCESS clause of read-write and/or read-create.

Some of the readable objects in this MIB module (i.e., those with MAX-ACCESS other than not-accessible) may be considered sensitive or vulnerable in some network environments since they can reveal some configuration aspects of the network interfaces.

In particular, ifCapStackStatus and ifInvCapStackStatus can identify cross-connect capability of multi-layer (stacked) network interfaces, potentially revealing the underlying hardware architecture of the managed device.

It is thus important to control even GET access to these objects and possibly even encrypt the values of these objects when sending them over the network via SNMP.

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPsec),

there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB module.

Implementations MUST provide the security features described by the SNMPv3 framework (see [RFC3410]), including full support for authentication and privacy via the User-based Security Model (USM) [RFC3414] with the AES cipher algorithm [RFC3826]. Implementations MAY also provide support for the Transport Security Model (TSM) [RFC5591] in combination with a secure transport such as SSH [RFC5592] or TLS/DTLS [RFC6353].

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

## 6. IANA Considerations

No action is required from IANA.

## 7. Acknowledgments

This document was produced by the OPSAWG working group, whose efforts were advanced by the contributions of the following people (in alphabetical order):

Dan Romascanu

David Harrington

Michael MacFaden

Tom Petch

This document updates RFC 5066, authored by Edward Beili of Actelis Networks, and produced by the, now concluded, HUBMIB working group.

## 8. References

### 8.1. Normative References

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

- [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, December 2002.
- [RFC3826] Blumenthal, U., Maino, F., and K. McCloghrie, "The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model", RFC 3826, June 2004.
- [RFC5066] Beili, E., "Ethernet in the First Mile Copper (EFMCu) Interfaces MIB", RFC 5066, November 2007.

## 8.2. Informative References

- [HUBMIB] IETF, "Ethernet Interfaces and Hub MIB (hubmib) Charter",  
<<http://datatracker.ietf.org/wg/hubmib/charter/>>.
- [IEEE802.3] IEEE, "802.3 Ethernet Working Group",  
<<http://www.ieee802.org/3>>.
- [IEEE802.3.1] IEEE, "IEEE Standard for Management Information Base (MIB) Definitions for Ethernet", IEEE Std 802.3.1-2013, June 2013.
- [LIST802.3.1] IEEE, "802.3 MIB Email Reflector",  
<<http://www.ieee802.org/3/be/reflector.html>>.
- [OPSAWG] IETF, "Operations and Management Area Working Group (opswg) Charter",  
<<http://datatracker.ietf.org/wg/opswag/charter/>>.
- [RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.
- [RFC5591] Harrington, D. and W. Hardaker, "Transport Security Model for the Simple Network Management Protocol (SNMP)", RFC 5591, June 2009.
- [RFC5592] Harrington, D., Salowey, J., and W. Hardaker, "Secure Shell Transport Model for the Simple Network Management Protocol (SNMP)", RFC 5592, June 2009.
- [RFC6353] Hardaker, W., "Transport Layer Security (TLS) Transport Model for the Simple Network Management



Protocol (SNMP)", RFC 6353, July 2011.

Author's Address

Edward Beili  
Actelis Networks  
Bazel 25  
Petach-Tikva  
Israel

Phone: +972-73-237-6852  
EMail: [edward.beili@actelis.com](mailto:edward.beili@actelis.com)



OPSAWG  
Internet Draft  
Intended status: Informational  
Expires: October 2013  
April 24, 2013

R. Krishnan  
S. Khanna  
Brocade Communications  
L. Yong  
Huawei USA  
A. Ghanwani  
Dell  
Ning So  
Tata Communications  
B. Khasnabish  
ZTE Corporation

Mechanisms for Optimal LAG/ECMP Component Link Utilization in  
Networks

draft-krishnan-opsawg-large-flow-load-balancing-08.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 October 24, 2013.

Copyright Notice

Copyright (c) 2013 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.

## Abstract

Demands on networking infrastructure are growing exponentially; the drivers are bandwidth hungry rich media applications, inter-data center communications, etc. In this context, it is important to optimally use the bandwidth in wired networks that extensively use LAG/ECMP techniques for bandwidth scaling. This draft explores some of the mechanisms useful for achieving this.

## Table of Contents

|  |    |
|--|----|
| 1. Introduction.....   | 3  |
| 1.1. Acronyms.....   | 3  |
| 1.2. Terminology.....  | 4  |
| 2. Hash-based Load Distribution in LAG/ECMP.....                   | 4  |
| 3. Mechanisms for Optimal LAG/ECMP Component Link Utilization..... | 5  |
| 3.1. Large Flow Recognition.....                                   | 7  |
| 3.1.1. Flow Identification.....                                    | 7  |
| 3.1.2. Criteria for Identifying a Large Flow.....                  | 8  |
| 3.1.3. Sampling Techniques.....                                    | 8  |
| 3.1.4. Automatic Hardware Recognition.....                         | 9  |
| 3.2. Load Re-balancing Options.....                                | 10 |
| 3.2.1. Alternative Placement of Large Flows.....                   | 10 |
| 3.2.2. Redistributing Small Flows.....                             | 11 |
| 3.2.3. Component Link Protection Considerations.....               | 11 |
| 3.2.4. Load Re-Balancing Example.....                              | 12 |
| 4. Information Model for Flow Re-balancing.....                    | 13 |
| 4.1. Configuration Parameters.....                                 | 13 |
| 4.2. Import of Flow Information.....                               | 13 |
| 5. Operational Considerations.....                                 | 14 |
| 6. IANA Considerations.....  | 14 |
| 7. Security Considerations.....                                    | 15 |
| 8. Acknowledgements.....   | 15 |
| 9. References.....   | 15 |
| 9.1. Normative References.....                                     | 15 |
| 9.2. Informative References.....                                   | 15 |

## 1. Introduction

Networks extensively use LAG/ECMP techniques for capacity scaling. Network traffic can be predominantly categorized into two traffic types: long-lived large flows and other flows (which include long-lived small flows, short-lived small/large flows). Stateless hash-based techniques [ITCOM, RFC 2991, RFC 2992, RFC 6790] are often used to distribute both long-lived large flows and other 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 best practices for optimal LAG/ECMP component link utilization while using hash-based techniques. These best practices comprise the following steps -- recognizing long-lived large flows in a router; and assigning the long-lived large flows to specific LAG/ECMP component links or redistributing other flows when a component link on the router is congested.

It is useful to keep in mind that the typical use case is where the long-lived 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 link. In other words, the number of long-lived large flows is NOT expected to be on the order of millions of flows. Examples of such long-lived large flows would be IPSec tunnels in service provider backbones or storage backup traffic in data center networks.

### 1.1. Acronyms

COTS: Commercial Off-the-shelf

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

Large flow(s): long-lived large flow(s)

Small flow(s): long-lived small flow(s) and short-lived small/large flow(s)

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

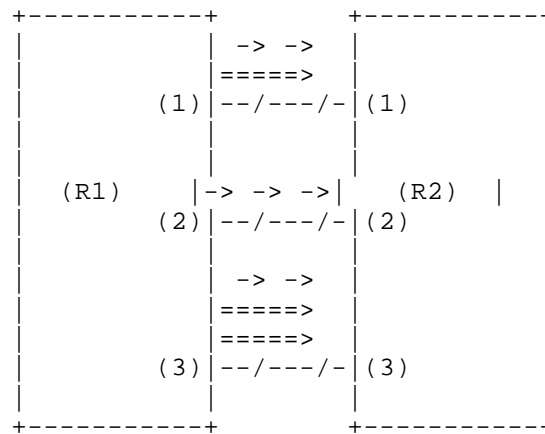
Hashing techniques are often used for traffic load balancing to select among multiple available paths with LAG/ECMP. The advantages of hash-based load distribution are the preservation of the packet sequence in a flow and the real-time distribution without maintaining per-flow state in the router. Hash-based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

If the traffic load 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, the hash-based algorithm produces 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 unbalanced loads on individual component links.

One example is illustrated in Figure 1. In the figure, 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 hashing 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 flows  
 ==> large flow

Figure 1: Unevenly Utilized Component Links

This document presents improved load distribution techniques based on the large flow awareness. The techniques compensate for unbalanced load distribution resulting from hashing as demonstrated in the above example.

### 3. Mechanisms for Optimal LAG/ECMP Component Link Utilization

The suggested techniques 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 globally optimal 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.

The various steps in achieving optimal 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 3.1.

Step 2) The egress component links are periodically scanned for link utilization. If the egress component link utilization exceeds a pre-programmed threshold, an operator alert is generated. The large flows mapped to the congested egress component link are exported to a central management entity.

Step 3) On receiving the alert about the congested component link, the operator, through a central management entity, finds the large flows mapped to that component link and the LAG/ECMP group to which the component link belongs.



Step 4) The operator can choose to rebalance the large flows on lightly loaded component links of the LAG/ECMP group or redistribute the small flows on the congested link to other component links of the group. The operator, through a central management entity, can choose one of the following actions:

- 1) Indicate specific large flows to rebalance;
- 2) Have the router decide the best large flows to rebalance;
- 3) Have the router redistribute all the small flows on the congested link to other component links in the group.

The central management entity conveys the above information to the router. The load re-balancing options are explained in Section 3.2.

Steps 2) to 4) could be automated if desired.

Providing large flow information to a central management entity provides the capability to further optimize flow distribution at with multi-node visibility. 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 P1 may be congested, while P2 and P3 may be under-utilized, which 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 P3 resulting in a more optimized flow of traffic.

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

### 3.1. Large Flow Recognition

#### 3.1.1. 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 from the following list:

- . Layer 2: source MAC address, destination MAC address, VLAN ID.
- . IP header: IP Protocol, IP source address, IP destination address, flow label (IPv6 only), TCP/UDP source port, TCP/UDP destination port.

.   MPLS Labels.

For tunneling protocols like GRE, VXLAN, NVGRE, STT, etc., flow identification is possible based on inner and/or outer headers. The above list is not exhaustive. The mechanisms described in this document are agnostic to the fields that are used for flow identification.

### 3.1.2. Criteria for Identifying a Large Flow

From a bandwidth and time duration perspective, in order to identify 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 in a router 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 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 identify a large flow are described below.

### 3.1.3. 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 identification, sampling must be enabled on all of the egress ports in the router where such measurements are desired.

Using sflow as an example, processing in an 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. This 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 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.

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].

#### 3.1.4. Automatic Hardware Recognition

Implementations may perform automatic recognition of large flows in hardware on a router. Since this is done in hardware, it is an inline solution and would be expected to operate at line rate.

Using automatic hardware recognition of large flows, a faster indication of large flows mapped to each of the component links in a LAG/ECMP group is available (as compared to the sampling approach described above).

The advantages and disadvantages of automatic hardware recognition are:

Advantages:

- . Large flow detection is offloaded to hardware freeing up software resources and possible dependence on an external management station.
- . 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 automatic hardware recognition, large flows can be detected in shorter windows on higher link speeds since every packet is accounted for in hardware [NDTM]

Disadvantages:

- . Not supported in many routers.

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 of automatic hardware recognition of large flows is vendor dependent and beyond the scope of this document.

### 3.2. Load Re-balancing Options

Below are suggested techniques for load re-balancing. Equipment vendors should implement all of these techniques and allow the operator to choose one or more techniques based on their applications.

Note that regardless of the method used, perfect re-balancing 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.

#### 3.2.1. 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 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 unbalanced utilization

once again across the link components. 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 4 10 Gbps component links and there are 4 large flows each of 1 Gbps. These flows are each placed on one of the component links. Subsequent, a 5-th 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.

### 3.2.2. 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 3.2.1) 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.

### 3.2.3. Component Link Protection Considerations

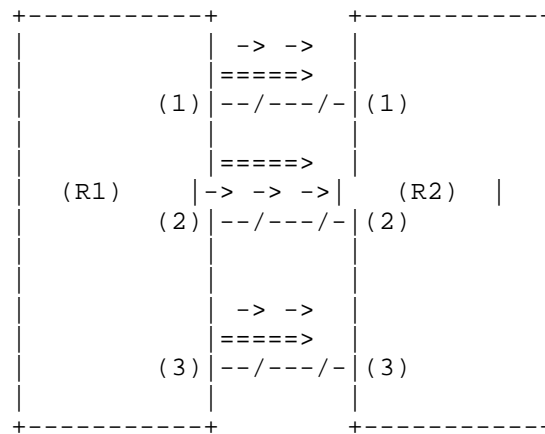
If desired, certain component links may be reserved for link protection. These reserved component links are not used for any flows in the absence of any failures.. In the case when the component link(s) fail, all the flows on the failed component link(s) are moved to the reserved component link(s). The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

reserved link. Likewise, the LAG/ECMP hash table replaces the failed component link with the reserved link.

#### 3.2.4. Load Re-Balancing Example

Optimal LAG/ECMP component utilization for the use case in Figure 1 is depicted below in Figure 2. The large flow rebalancing explained in Section 3.2.1 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 flows  
 ==> large flow

Figure 2: Evenly utilized Composite Links

Basically, the use of the mechanisms described in Section 3.2.1 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.

#### 4. Information Model for Flow Re-balancing

##### 4.1. Configuration Parameters

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: the difference between the utilization of the least utilized and most utilized component links. Expressed as a percentage of link speed.

##### 4.2. Import of Flow Information

In cases where large flow recognition is handled by an external management station (see Section 3.1.3), an information model for flows is required to allow the import of large flow information to the router.

The following are some of the elements of information model for importing of flows:

- . Layer 2: source MAC address, destination MAC address, VLAN ID.
- . Layer 3 IP: IP Protocol, IP source address, IP destination address, flow label (IPv6 only), TCP/UDP source port, TCP/UDP destination port.
- . MPLS Labels.

This list is not exhaustive. For example, with overlay protocols such as VXLAN and NVGRE, fields from the outer and/or inner headers may be specified. In general, all fields in the packet that can be used by forwarding decisions should be available for use when importing flow information from an external management station.

## 5. Operational Considerations

Flows should be re-balanced only when the imbalance in the utilization across component links exceeds a certain threshold. Frequent re-balancing 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 re-distributed.

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. IANA Considerations

This memo includes no request to IANA.



## 7. 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.

## 8. Acknowledgements

The authors would like to thank the following individuals for their review and valuable feedback on earlier versions of this document: Shane Amante, Curtis Villamizar, Fred Baker, Wes George, Brian Carpenter, George Yum, Michael Fargano, Michael Bugenhagen, Jianrong Wong, Peter Phaal, Roman Krzanowski and Weifeng Zhang.

## 9. References

### 9.1. Normative References

### 9.2. Informative References

[I-D.ietf-rtgwg-cl-requirement] Villamizar, C. et al., "Requirements for MPLS over a Composite Link", June 2012.

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

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

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

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

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

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

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

[sFlow-v5] Phaal, P. and M. Lavine, "sFlow version 5", July 2004.

[sFlow-LAG] Phaal, P. and A. Ghanwani, "sFlow LAG counters structure", September 2012.

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

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

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

#### 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: ramk@brocade.com

Sanjay Khanna  
Brocade Communications  
San Jose, 95134, USA  
Phone: +1-408-333-4850  
Email: skhanna@brocade.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  
Email: bhumip.khasnabish@zteusa.com



Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: August 22, 2013

C. Shao  
H. Deng  
China Mobile  
F. Bari  
AT&T  
R. Zhang  
China Telecom  
S. Matsushima  
SoftBank Telecom  
February 18, 2013

Hybrid-MAC Model for CAPWAP  
draft-shao-opsawg-capwap-hybridmac-00

Abstract

The CAPWAP protocol supports two modes of operation: Split and Local MAC (medium access control), which has been described in [RFC5415]. There are many functions in IEEE 802.11 MAC layer that have not yet been clearly defined whether they belong to either the AP (Access Point) or the AC (Access Controller) in the Split and Local modes. Because different vendors have their own definition of these two models, depending upon the vendor many MAC layer functions continue to be mapped differently to either the AP or AC. If there is no clear definition of split MAC and local MAC, then operators will not only need to perform vendor specific configurations in their network but will continue to experience difficulty in interoperating APs and ACs from different vendors.

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 22, 2013.

Copyright Notice

Copyright (c) 2013 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. Conventions used in this document . . . . .              | 3 |
| 3. The difference between Local MAC and Split MAC . . . . . | 3 |
| 4. Functions in Local MAC and Split MAC . . . . .           | 4 |
| 5. Hybrid-MAC model recommendation . . . . .                | 5 |
| 6. Hybrid-MAC model Frames Exchange . . . . .               | 6 |
| 7. Security Considerations . . . . .                        | 7 |
| 8. IANA Considerations . . . . .                            | 7 |
| 9. Contributors . . . . .                                   | 8 |
| 10. Normative References . . . . .                          | 8 |
| Authors' Addresses . . . . .                                | 8 |

## 1. Introduction

The CAPWAP protocol supports two modes of operation: Split and Local MAC (medium access control), which has been described in [RFC5415]. In Split MAC mode, all L2 wireless data and management frames are encapsulated via the CAPWAP protocol and exchanged between the AC and the AP. The Local MAC mode of operation allows for the data frames to be either locally bridged or tunneled as 802.3 frames. The latter implies that the AP performs the 802.11 Integration function. Unfortunately, there are many functions that have not yet been clearly defined whether they belong to either the AP or the AC in the Split and Local modes. Because different vendors have their own definition of the two models, many MAC layer functions are mapped differently to either the AP 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 definition of split MAC and local MAC, then operators will continue to experience difficulty in interoperating APs and ACs from different vendors.

## 2. 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].

## 3. The difference between Local MAC and Split MAC

The main difference between Local MAC and Split MAC lies in the processing of the wireless frames. This is shown in Figure 1 where depending upon the mode, either the AP or the AC performs the 802.11 Integration function. According to the 802.11 protocol definition, the 802.11 wireless frame is divided into three kinds of frames, including wireless control frames, wireless management frames, and wireless data frames.

WWireless control frames, such as TS, CTS, ACK, PS-POLL, etc., are processed locally by AP in both Local MAC and Split MAC. However, wireless management frames, including Beacon, Probe, Association, Authentication, are processed differently in the Local MAC and the Split MAC. In the Local MAC, depending upon the vendor wireless management frames can be processed in the AP or the AC. In the case of Split MAC, the real-time part of wireless frames are processed in AP, while the non-real-time frames are processed in the AC. This is shown in Figure 2. In Split MAC mode, the wireless data frames



received from a mobile device are directly encapsulated by the AP and forwarded to the AC. The Local MAC mode of operation allows data frames to be processed locally by the AP and then forwarded to the AC.

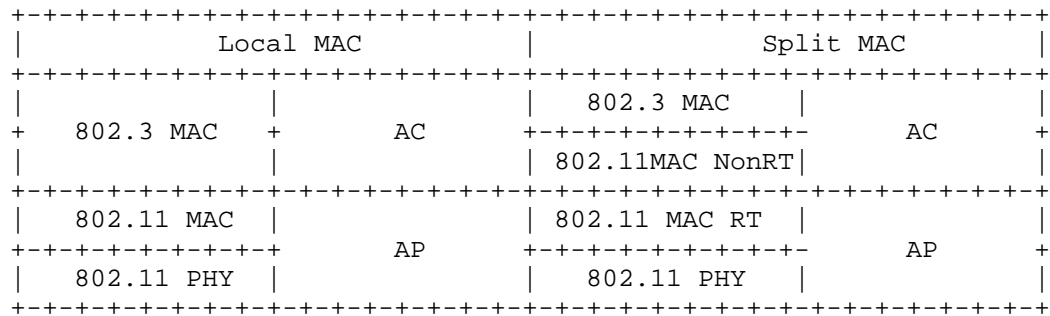


Figure 1: The comparison between Local MAC and Split MAC

#### 4. Functions in Local MAC and Split MAC

As shown in Figure 2, main functions are processed in different places in the Local MAC and Split MAC. In addition, for some functions (for example, the Frag. / Defrag. Assoc. / Disassoc / Reassoc., Etc.) the protocol does not explicitly map processing of such functions to the AP or the AC. Therefore the location of these features becomes vendor specific and this increases the difficulty of interoperability between APs and ACs from different vendors.

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

Figure 2: Functions in Local MAC and Split MAC

## 5. Hybrid-MAC model recommendation

As discussed above, if the functions have been clearly defined to be implemented in AP or AC, the interoperability will be much better between different vendors products. To achieve this goal a common Hybrid-MAC model, as shown in Figure 3, is proposed.

| Functions describe           |                                      | Hybrid-MAC |
|------------------------------|--------------------------------------|------------|
| Function                     | Distribution Service                 | AC         |
|                              | Integration Service                  | AC         |
|                              | Beacon Generation                    | AP         |
|                              | Probe Response Generation            | AP         |
|                              | Power Mgmt                           | AP         |
|                              | /Packet Buffering                    |            |
|                              | Fragmentation                        | AC         |
|                              | /Defragmentation                     |            |
|                              | Assoc/Disassoc/Reassoc               | AC         |
|                              | Classifying                          | AC         |
| IEEE<br>802.11 QoS           | Scheduling                           | AP         |
|                              | Queuing                              | AP         |
|                              | IEEE 802.1X/EAP                      | AC         |
| IEEE<br>802.11 RSN<br>(WPA2) | RSNA Key Management                  | AC         |
|                              | IEEE 802.11<br>Encryption/Decryption | AP         |

Figure 3: Functions in Hybrid MAC

## 6. Hybrid-MAC model Frames Exchange

An example of frame exchange using the proposed Hybrid-MAC Model shown in Figure 4.

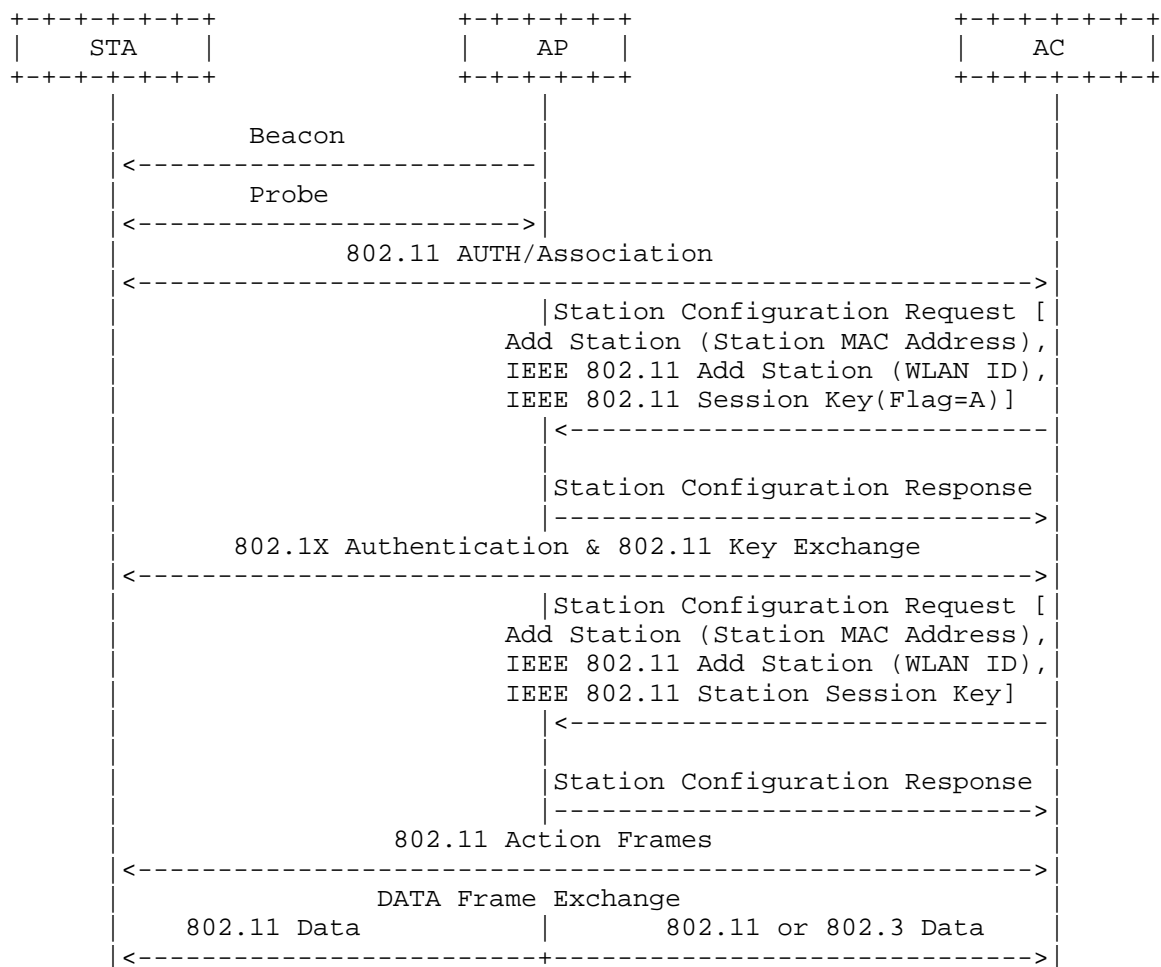


Figure 4: Hybrid-MAC model Frames Exchange

## 7. Security Considerations

TBD

## 8. IANA Considerations

None

## 9. Contributors

Naibao Zhou zhounaibao@chinamobile.com

## 10. Normative References

- [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.

## Authors' Addresses

Chunju Shao  
China Mobile  
No.32 Xuanwumen West Street  
Beijing 100053  
China  
  
Email: shaochunju@chinamobile.com

Hui Deng  
China Mobile  
No.32 Xuanwumen West Street  
Beijing 100053  
China  
  
Email: denghui@chinamobile.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  
Tianhe District,  
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



Network Working Group  
Internet-Draft  
Intended status: BCP  
Expires: August 7, 2014

M. Shore  
No Mountain Software  
C. Pignataro  
Cisco Systems, Inc.  
February 3, 2014

An Acceptable Use Policy for New ICMP Types and Codes  
draft-shore-icmp-aup-12

Abstract

In this document we provide a basic description of ICMP's role in the IP stack and some guidelines for future use.

This document is motivated by concerns about lack of clarity concerning when to add new Internet Control Message Protocol (ICMP) types and/or codes. These concerns have highlighted a need to describe policies for when adding new features to ICMP is desirable and when it is not.

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 August 7, 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. 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. Acceptable use policy . . . . .                      | 3 |
| 2.1. Classification of existing message types . . . . . | 3 |
| 2.1.1. ICMP Use as a Routing Protocol . . . . .         | 5 |
| 2.1.2. A few notes on RPL . . . . .                     | 6 |
| 2.2. Applications using ICMP . . . . .                  | 6 |
| 2.3. Extending ICMP . . . . .                           | 6 |
| 2.4. ICMPv4 vs. ICMPv6 . . . . .                        | 6 |
| 3. ICMP's role in the internet . . . . .                | 7 |
| 4. Security considerations . . . . .                    | 7 |
| 5. IANA considerations . . . . .                        | 8 |
| 6. Acknowledgments . . . . .                            | 8 |
| 7. References . . . . .                                 | 8 |
| 7.1. Normative references . . . . .                     | 8 |
| 7.2. Informative references . . . . .                   | 8 |
| Authors' Addresses . . . . .                            | 9 |

## 1. Introduction

There has been some recent concern expressed about a lack of clarity around when to add new message types and codes to ICMP (including ICMPv4 [RFC0792] and ICMPv6 [RFC4443]). We lay out a description of when (and when not) to move functionality into ICMP.

This document is the result of discussions among ICMP experts within the OPS area's IP Diagnostics Technical Interest Group [1] and concerns expressed by the OPS area leadership.

Note that this document does not supercede the IANA Allocation Guidelines for Values in the Internet Protocol and Related Headers, RFC 2780 [RFC2780], which specifies best practices and processes for the allocation of values in the IANA registries but does not describe the policies to be applied in the standards process.

## 2. Acceptable use policy

In this document we describe an acceptable use policy for new ICMP message types and codes, and provide some background behind the policy.

In summary, any future message types added to ICMP should be limited to two broad categories:

1. to inform a datagram's originator that a forwarding plane anomaly has been encountered downstream. The datagram originator must be able to determine whether or not the datagram was discarded by examining the ICMP message
2. to discover and convey dynamic information about a node (other than information usually carried in routing protocols), to discover and convey network-specific parameters, and to discover on-link routers and hosts.

Normally, ICMP SHOULD NOT be used to implement a general-purpose routing or network management protocol. However, ICMP does have a role to play in conveying dynamic information about a network, which would belong in category 2 above.

### 2.1. Classification of existing message types

This section provides a rough breakdown of existing message types according to the taxonomy described in Section 2 at the time of publication.

IPv4 forwarding plane anomaly reporting:

- 3: Destination unreachable
- 4: Source quench (deprecated)
- 6: Alternate host address (deprecated)
- 11: Time exceeded
- 12: Parameter problem
- 31: Datagram conversion error (deprecated)
- 41: ICMP messages utilized by experimental mobility protocols,  
such as Seamoby

IPv4 router or host discovery:

- 0: Echo reply
- 5: Redirect
- 8: Echo
- 9: Router advertisement
- 10: Router solicitation
- 13: Timestamp
- 14: Timestamp reply
- 15: Information request (deprecated)
- 16: Information reply (deprecated)
- 17: Address mask request (deprecated)
- 18: Address mask reply (deprecated)
- 30: Traceroute (deprecated)
- 32: Mobile host redirect (deprecated)
- 33: IPv6 Where-Are-You (deprecated)
- 34: IPv6 I-Am-Here (deprecated)
- 35: Mobile registration request (deprecated)
- 36: Mobile registration reply (deprecated)
- 37: Domain name request (deprecated)
- 38: Domain name reply (deprecated)
- 39: SKIP (deprecated)
- 40: Photuris
- 41: ICMP messages utilized by experimental mobility protocols,  
such as Seamoby

Please note that some ICMP message types were formally deprecated by [RFC6918].

IPv6 forwarding plane anomaly reporting:

- 1: Destination unreachable
- 2: Packet too big
- 3: Time exceeded

- 4: Parameter problem
- 150: ICMP messages utilized by experimental mobility protocols,  
such as Seamoby

IPv6 router or host discovery:

- 128: Echo request
- 129: Echo reply
- 130: Multicast listener query
- 131: Multicast listener report
- 132: Multicast listener done
- 133: Router solicitation
- 134: Router advertisement
- 135: Neighbor solicitation
- 136: Neighbor advertisement
- 137: Redirect message
- 138: Router renumbering
- 139: ICMP node information query
- 140: ICMP node information response
- 141: Inverse neighbor discovery solicitation message
- 142: Inverse neighbor discovery advertisement message
- 143: Version 2 multicast listener report
- 144: Home agent address discovery request message
- 145: Home agent address discovery reply message
- 146: Mobile prefix solicitation
- 147: Mobile prefix advertisement
- 148: Certification path solicitation message
- 149: Certification path advertisement message
- 150: ICMP messages utilized by experimental mobility protocols,  
such as Seamoby
- 151: Multicast router advertisement
- 152: Multicast router solicitation
- 153: Multicast router termination
- 154: FMIPv6 messages
- 155: RPL control message

#### 2.1.1. ICMP Use as a Routing Protocol

As mentioned in Section 2, using ICMP as a general-purpose routing or network management protocol is not advisable, and SHOULD NOT be used that way.

ICMP has a role in the Internet as an integral part of the IP layer. This is not as a routing protocol, or as a transport protocol for other layers including routing information. From a more pragmatic perspective, some of the key characteristics of ICMP make it a less than ideal choice for a routing protocol. Those include that ICMP is frequently filtered, is not authenticated, is easily spoofed, and

that specialist hardware processing of ICMP would disrupt the deployment of an ICMP-based routing or management protocol.

#### 2.1.2. A few notes on RPL

RPL, the IPv6 Routing protocol for low-power and lossy networks (see [RFC6550]) uses ICMP as a transport. In this regard, it is an exception among the ICMP message types. Note that, although RPL is an IP routing protocol, it is not deployed on the general Internet, but is limited to specific, contained networks.

This should be considered anomalous and is not a model for future ICMP message types. That is, ICMP is not intended as a transport for other protocols and SHOULD NOT be used in that way in future specifications. In particular, while it is adequate to use ICMP as a discovery protocol, this does not extend to full routing capabilities.

#### 2.2. Applications using ICMP

Some applications make use of ICMP error notifications, or even deliberately create anomalous conditions in order to elicit ICMP messages, to then use those ICMP messages to generate feedback to the higher layer. Some of these applications include most widespread examples such as PING, TRACEROUTE and Path MTU Discovery (PMTUD). These uses are considered acceptable as they use existing ICMP message types and do not change ICMP functionality.

#### 2.3. Extending ICMP

ICMP multi-part messages are specified in [RFC4884] by defining an extension mechanism for selected ICMP messages. This mechanism addresses a fundamental problem in ICMP extensibility. An ICMP multi-part message carries all of the information that ICMP messages carried previously, as well as additional information that applications may require.

Some currently defined ICMP extensions include ICMP extensions for Multiprotocol Label Switching [RFC4950] and ICMP extensions for interface and next-hop identification [RFC5837].

Extensions to ICMP SHOULD follow [RFC4884].

#### 2.4. ICMPv4 vs. ICMPv6

Because ICMPv6 is used for IPv6 Neighbor Discovery, deployed IPv6 routers, IPv6-capable security gateways, and IPv6-capable firewalls normally support administrator configuration of how specific ICMPv6

message types are handled. By contrast, deployed IPv4 routers, IPv4-capable security gateways, and IPv4-capable firewalls are less likely to allow an administrator to configure how specific ICMPv4 message types are handled. So, at present, ICMPv6 messages usually have a higher probability of travelling end-to-end than ICMPv4 messages.

### 3. ICMP's role in the internet

ICMP was originally intended to be a mechanism for gateways or destination hosts to report error conditions back to source hosts in ICMPv4 [RFC0792], and ICMPv6 [RFC4443] is modeled after it. ICMP is also used to perform IP-layer functions, such as diagnostics (e.g., "PING").

ICMP is defined to be an integral part of IP, and must be implemented by every IP module. This is true for ICMPv4 as an integral part of IPv4 (see the Introduction of [RFC0792]), and for ICMPv6 as an integral part of IPv6 (see Section 2 of [RFC4443]). When first defined, ICMP messages were thought of as IP messages that didn't carry any higher layer data. It could be conjectured that the term "control" was used given that ICMP messages were not "data" messages.

The word "control" in the protocol name did not describe ICMP's function (i.e. it did not "control" the internet), but rather that it was used to communicate about the control functions in the internet. For example, even though ICMP included a redirect message type that affects routing behavior in the context of a LAN segment, it was and is not used as a generic routing protocol.

### 4. Security considerations

This document describes a high-level policy for adding ICMP types and codes. While special attention must be paid to the security implications of any particular new ICMP type or code, this recommendation presents no new security considerations.

From a security perspective, ICMP plays a part in the Photuris [RFC2521] protocol. But more generally, ICMP is not a secure protocol, and does not include features to be used to discover network security parameters or to report on network security anomalies in the forwarding plane.

Additionally, new ICMP functionality (e.g., ICMP extensions, or new ICMP types or codes) needs to consider potential ways of how ICMP can be abused (e.g., Smurf IP DoS [CA-1998-01]).

## 5. IANA considerations

There are no actions required by IANA.

## 6. Acknowledgments

This document was originally proposed by, and received substantial review and suggestions from, Ron Bonica. Discussions with Pascal Thubert helped clarify the history of RPL's use of ICMP. We are very grateful for the review, feedback, and comments from Ran Atkinson, Tim Chown, Joe Clarke, Adrian Farrel, Ray Hunter, Hilarie Orman, Eric Rosen, JINMEI Tatuya, and Wen Zhang, which resulted in a much improved document.

## 7. References

### 7.1. Normative references

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, September 1981.
- [RFC4443] Conta, A., Deering, S., and M. Gupta, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 4443, March 2006.
- [RFC4884] Bonica, R., Gan, D., Tappan, D., and C. Pignataro, "Extended ICMP to Support Multi-Part Messages", RFC 4884, April 2007.

### 7.2. Informative references

- [RFC2780] Bradner, S. and V. Paxson, "IANA Allocation Guidelines For Values In the Internet Protocol and Related Headers", BCP 37, RFC 2780, March 2000.
- [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.
- [RFC6918] Gont, F. and C. Pignataro, "Formally Deprecating Some ICMPv4 Message Types", RFC 6918, April 2013.

- [RFC4950] Bonica, R., Gan, D., Tappan, D., and C. Pignataro, "ICMP Extensions for Multiprotocol Label Switching", RFC 4950, August 2007.
- [RFC5837] Atlas, A., Bonica, R., Pignataro, C., Shen, N., and JR. Rivers, "Extending ICMP for Interface and Next-Hop Identification", RFC 5837, April 2010.
- [RFC2521] Karn, P. and W. Simpson, "ICMP Security Failures Messages", RFC 2521, March 1999.
- [CA-1998-01] CERT, "Smurf IP Denial-of-Service Attacks", CERT Advisory CA-1998-01, January 1998, <<http://www.cert.org/advisories/CA-1998-01.html>>.

## URIs

- [1] <[https://svn.tools.ietf.org/area/ops/trac/wiki/TIG\\_DIAGNOSTICS](https://svn.tools.ietf.org/area/ops/trac/wiki/TIG_DIAGNOSTICS)>

## Authors' Addresses

Melinda Shore  
No Mountain Software  
PO Box 16271  
Two Rivers, AK 99716  
US

Phone: +1 907 322 9522  
Email: [melinda.shore@nomountain.net](mailto:melinda.shore@nomountain.net)

Carlos Pignataro  
Cisco Systems, Inc.  
7200-12 Kit Creek Road  
Research Triangle Park, NC 27709  
US

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





Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: August 22, 2013

R. Zhang  
China Telecom  
Z. Cao  
H. Luo  
H. Deng  
China Mobile  
S. Gundavelli  
Cisco  
February 18, 2013

Encapsulation of EAP Messages in CAPWAP Control Plane  
draft-zhang-opsawg-capwap-eap-00

Abstract

This document describes the scenario and requirement of encapsulating Extensible Authentication Protocol (EAP) in the CAPWAP control plane. After the analysis and description, this document proposes the design of the new message types to encapsulate EAP messages.

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 22, 2013.

Copyright Notice

Copyright (c) 2013 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. Conventions used in this document . . . . .      | 3 |
| 1.2. Terminology . . . . .                            | 3 |
| 2. Scenario and Analysis . . . . .                    | 4 |
| 3. Encapsulation of EAP in CAPWAP-CTL Plane . . . . . | 5 |
| 3.1. Control Message Type for EAP . . . . .           | 5 |
| 3.2. Message Element of the EAP . . . . .             | 6 |
| 4. IANA Considerations . . . . .                      | 7 |
| 5. Security Considerations . . . . .                  | 7 |
| 6. Contributors . . . . .                             | 7 |
| 7. References . . . . .                               | 7 |
| 7.1. Normative References . . . . .                   | 7 |
| 7.2. Informative References . . . . .                 | 7 |
| Authors' Addresses . . . . .                          | 8 |

## 1. Introduction

Control and Provisioning of Wireless Access Points (CAPWAP) was designed as an interoperable protocol between the wireless access point and the access controller. This architecture makes it possible for the access controller to manage a huge number of wireless access points. With the goals and requirements established in [RFC4564], CAPWAP protocols were specified in [RFC5415], [RFC5416] and [RFC5417].

The specifications mentioned above mainly design the different control message types used by the AC to control multiple APs. The EAP messages, as key protocol exchange elements in the WLAN architecture, also need to be encapsulated in the CAPWAP. However, the CAPWAP protocol does not specify how to encapsulate the EAP message in its control plane. This situation makes it default to encapsulate the EAP messages in the CAPWAP-DATA plane.

We found issues of encapsulating EAP in the CAPWAP-DATA plane in the scenario where there is a split between the CAPWAP-DATA and CAPWAP-CTL plane. This document describes such scenario and proposes a resolution to the problem.

### 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

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

**Access Point (AP):** the same with Wireless Termination Point, 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 Plane:** A bi-directional flow over which CAPWAP Control packets are sent and received.

**CAPWAP Data Plane:** A bi-directional flow over which CAPWAP Data packets are sent and received.

**EAP:** Extensible Authentication Protocol, the EAP framework is specified in [RFC3748].

## 2. Scenario and Analysis

The following figure shows where and how the problem arises. In many operators' network, the Access Controller is placed remotely at the central data center. In order to avoid the traffic aggregation at the AC, the data traffic from the AP is directed to the Access Router (AR). In this scenario, the CAPWAP-CTL plane and CAPWAP-DATA plane are separated from each other.

Note: a powerful AC that aggregates the data flows is not a long-term solution to the problem. Because operators always plan the network capacity at a certain level, but with the air interface bandwidth increasing (e.g., from 11g to 11n and 11ac), and the increasing number of access requests on each AP, the powerful AC could not always be "powerful" enough in the long run.

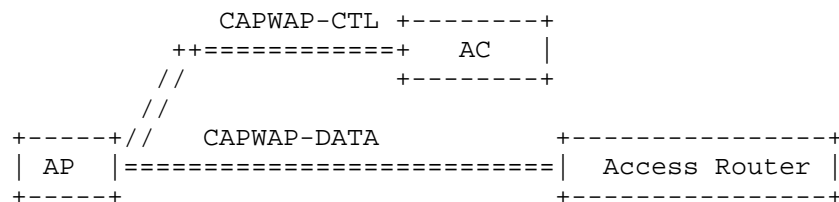


Figure 1: Split between CAPWAP-CTL and CAPWAP-DATA Plane

Because there are no explicit message types to support the encapsulation of EAP packets in the CAPWAP-CTL plane, the EAP messages are tunneled via the CAPWAP-DATA plane to the AR. AR acts as authenticator in the EAP framework. After authentication, the AR receives the EAP keying message for the session. But AC is supposed to deliver these keying messages to the AP, and AR has no standard interface to ship them to the AP or the AC. This is unacceptable in the scenario of EAP-based auto-authentication.

Another scenario is the third-party WLAN deployment scenario, in which the access network is a rental property from an broadband operator different from the one who provides authentication services. As shown in Figure 2, The AP is broadcasting a SSID of the Operator #1, say "Operator-1-WLAN", but broadband access network is provided by another Operator #2. To authenticate the users of operator one, the users should be authenticated by the AC in operator one. The data traffic can be routed locally with the access router of operator #2. In this case, there is also a need of separation between CAPWAP-CTL and CAPWAP-DATA traffics.

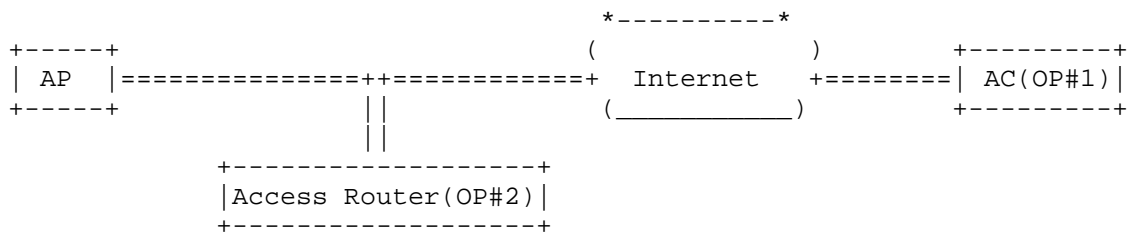


Figure 2: Access Service and Authentication Service Provided by different Operators

### 3. Encapsulation of EAP in CAPWAP-CTL Plane

In order to encapsulate EAP message in CAPWAP-CTL plane, we can reuse the control message header defined in RFC5415 and extend the message type to accommodate EAP messages.

The CAPWAP Control message header is shown in Figure 3. Only 26 message types have been defined in Section 4.5.5.1 of RFC5415. We can extend the message type here to encapsulate EAP messages.

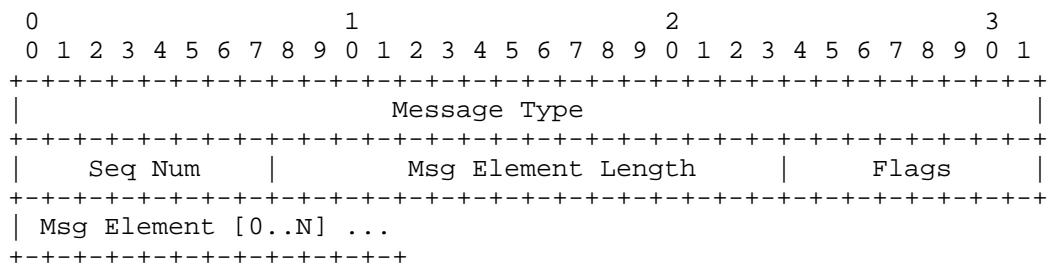


Figure 3: The CAPWAP Control Message Header

### 3.1. Control Message Type for EAP

This document defines a new control message type for EAP, i.e. "AUTHENTICATION CONTROL". The message type value is to be defined by IANA.

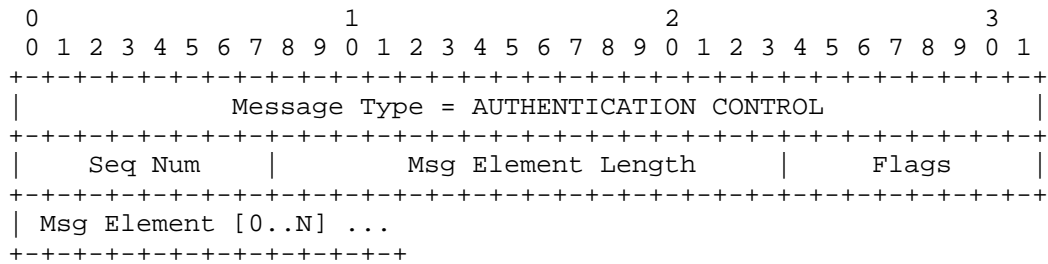


Figure 4: The CAPWAP-EAP Control Message Header

The Seq Num is design to match the response with the request for other control messages like "Discovery Request" and "Discovery Response". But this field is not useful for authentication control, because the EAP message encapsulated between the AP and AC is not handled in a request-response way. For AUTHENTICATION CONTROL messages, the AP and AC do not need to handle the 'Seq Num' field.

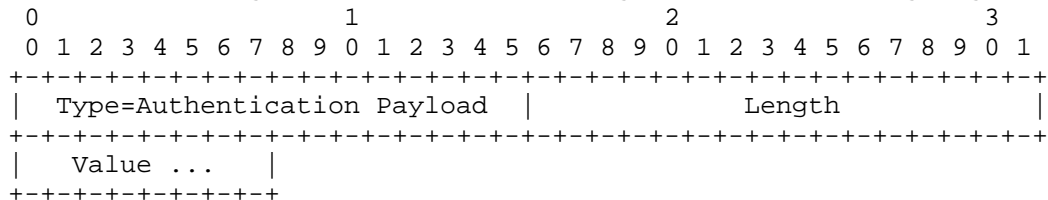
Msg Element Length field indicates the number of bytes following the Sequence Number field.

Flags field is left for future definition.

### 3.2. Message Element of the EAP

The message element(s) carry the information pertinent to each of the control message types. Every control message in this specification specifies which message elements are permitted.

We define the message element of EAP message in the following figure.



Message Element for EAP

Section 4.6 of [RFC5415] defines the semantics of Message Element Types. Type values from 1-49 have been used. An extended message element type is requested by this document to carry the EAP authentication payload.

#### 4. IANA Considerations

This document has the following requests to the IANA.

CAPWAP Control Message Type Value for the EAP-AUTHENTICATION-CONTROL, as defined in Section. 3.1 of this document.

CAPWAP Control Message Element Type Value for the EAP-AUTHENTICATION-PAYLOAD, as defined in Section. 3.2 of this document.

#### 5. Security Considerations

Security considerations for the CAPWAP protocol has been analyzed in Section 12 of [RFC5415]. This document extends the CAPWAP CONTROL Message Type and Control Message Element Type, and it does not introduce other security issues besides what has been analyzed in RFC5415.

#### 6. Contributors

This document stems from the joint work of Hong Liu, Yifan Chen, Chunju Shao from China Mobile Research. Thank all the contributors of this document.

#### 7. References

##### 7.1. Normative References

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

##### 7.2. Informative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3118] Droms, R. and W. Arbaugh, "Authentication for DHCP Messages", RFC 3118, June 2001.
- [RFC3748] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowitz, "Extensible Authentication Protocol (EAP)", RFC 3748, June 2004.
- [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.

[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.

[RFC5417] Calhoun, P., "Control And Provisioning of Wireless Access Points (CAPWAP) Access Controller DHCP Option", RFC 5417, March 2009.

#### Authors' Addresses

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

Phone:  
Fax:  
Email: zhangr@gsta.com  
URI:

Zhen Cao  
China Mobile  
Xuanwumenxi Ave. No. 32  
Beijing, 100871  
China

Phone: +86-10-52686688  
Email: zehn.cao@gmail.com, caozhen@chinamobile.com

Haiyun Luo  
China Mobile  
United States

Phone:  
Fax:  
Email: haiyunluo@chinamobile.com  
URI:

Hui Deng  
China Mobile  
Xuanwumenxi Ave. No. 32  
Beijing, 100053  
China

Phone:  
Fax:  
Email: denghui@chinamobile.com  
URI:

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

Phone:  
Fax:  
Email: sgundave@cisco.com  
URI:

