

OPSAWG  
Internet-Draft  
Intended status: Standards Track  
Expires: January 4, 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  
July 3, 2013

Management Information Base for Virtual Machines Controlled by a  
Hypervisor  
draft-asai-vmm-mib-04

Abstract

This document defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, this specifies objects for managing virtual machines controlled by a hypervisor (a.k.a. virtual machine manager).

Status of this Memo

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

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

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

This Internet-Draft will expire on January 4, 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 . . . . .	42
5. Security Considerations . . . . .	43
6. Acknowledgements . . . . .	45
7. References . . . . .	46
7.1. Normative References . . . . .	46
7.2. Informative References . . . . .	47
Authors' Addresses . . . . .	48

## 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 managers). 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.

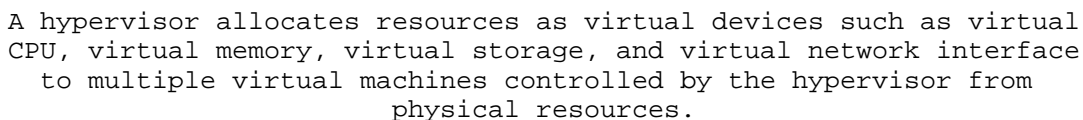
### 1.1. Requirements Language

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].

### 3.1. Managed Objects on Virtualization Environment

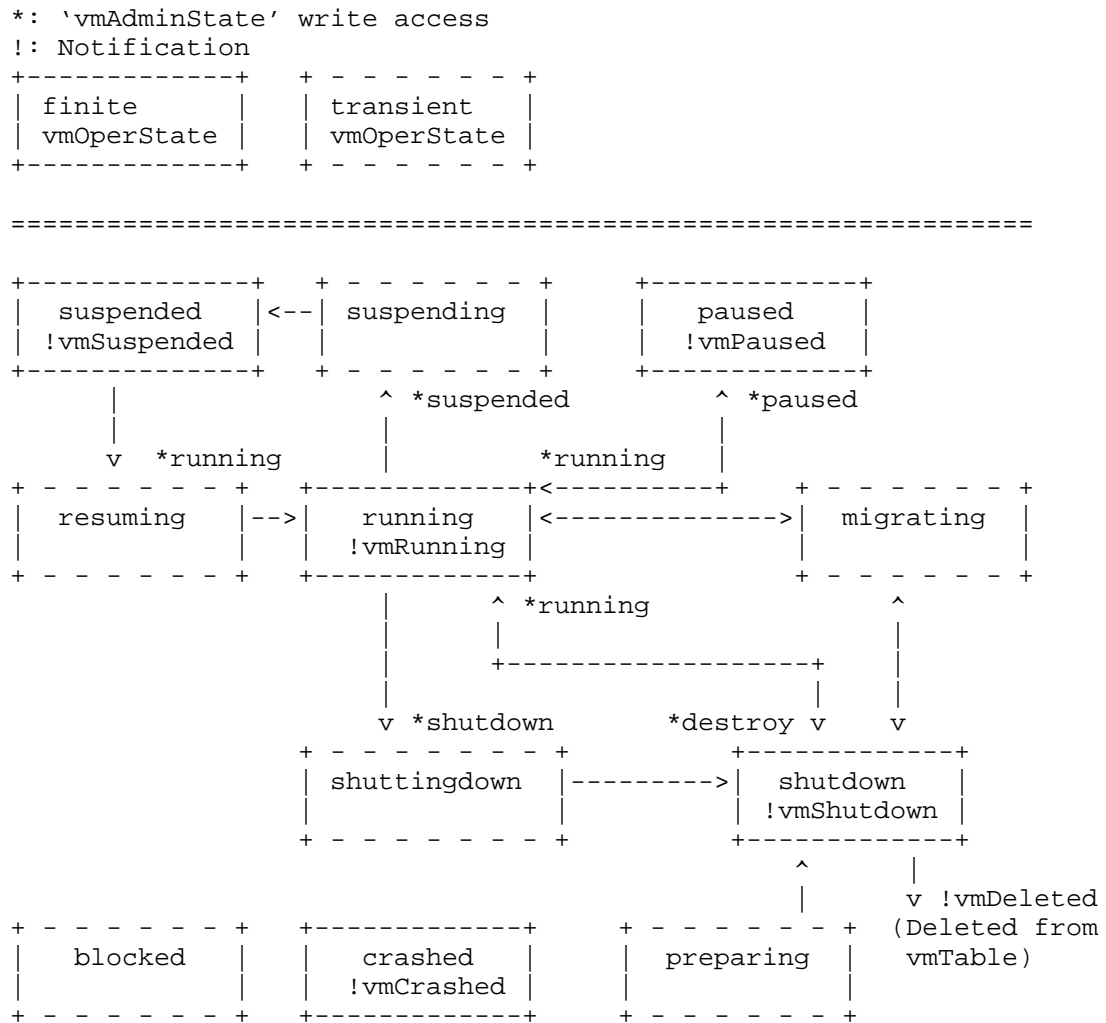


On the common implementations of hypervisor softwares, a hypervisor allocates resources as virtual devices such as virtual CPUs, virtual memory, virtual storage, and virtual network interface to multiple virtual machines controlled by the hypervisor from physical resources. This document defines objects 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. As shown in Figure 1, the virtual resource objects are defined as virtual devices. Consequently, this document specifies four specific types of virtual devices; CPUs (processors), memory, network interfaces, and storage devices. Note that physical resources are managed in HOST-RESOURCES-MIB [RFC2790]. In case that each virtual resource device object has a corresponding parent physical device managed in HOST-RESOURCES-MIB, the object of the virtual resource device contains a pointer to the physical device. The objects related to virtual network interfaces are mapped to the objects managed in IF-MIB [RFC2863].

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` must not take different values when the `vmStorageSourceType` is 'block' or 'raw'.

### 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' and 'vmCpuAffinityTable' provide the mapping of virtual CPUs and their affinity to virtual machines. The 'vmStorageTable' and the 'vmNetworkTable' provide the mapping of logical storage areas and network interfaces to virtual machines.



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-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]
|   +--vmShutdown(2) [vmName, vmUUID, vmOperState]
|   +--vmPaused(3) [vmName, vmUUID, vmOperState]
|   +--vmSuspended(4) [vmName, vmUUID, vmOperState]
|   +--vmCrashed(5) [vmName, vmUUID, vmOperState]
|   +--vmDeleted(6) [vmName, vmUUID, vmOperState, vmPersistent]
|   +--vmBulkRunning(7) [vmAffectedVMs]
|   +--vmBulkShutdown(8) [vmAffectedVMs]
|   +--vmBulkPaused(9) [vmAffectedVMs]
|   +--vmBulkSuspended(10) [vmAffectedVMs]
|   +--vmBulkCrashed(11) [vmAffectedVMs]
|   +--vmBulkDeleted(12) [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)
|   |   +-- rwn VirtualMachineAutoStart
|   |   |   vmAutoStart(7)
|   |   +-- r-n VirtualMachinePersistent
|   |   |   vmPersistent(8)
|   |   +-- r-n Integer32                vmCurCpuNumber(9)
|   |   +-- rwn Integer32                vmMinCpuNumber(10)
|   |   +-- rwn Integer32                vmMaxCpuNumber(11)
|   |   +-- r-n Integer32                vmMemUnit(12)
|   |   +-- r-n 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

```

```

|
|         +--- r-n VirtualMachineStorageMediaType          vmStorageAccess(7)
|         |
|         +--- r-n SnmpAdminString          vmStorageMediaType(8)
|         +--- r-n Integer32                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
|   |   |
|   |   +--- r-n InterfaceIndexOrZero      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 "201307020000Z" -- 2 July 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 "201307020000Z" -- 2 July 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
                        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
                        should be taken shuttingdown."
```

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."

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 state is unknown, e.g., because the implementation failed to obtain the state from the hypervisor.

other(2)        The state has been obtained but it is not a known state.

preparing(3)    The virtual machine is currently in the process of preparation, e.g., allocating and initializing virtual storage are after creating (defining) virtual machine.

running(4)      The virtual machine is currently running.

blocked(5)      The virtual machine is currently blocked.

suspending(6)   The virtual machine is currently in the process of suspending.

suspended(7)    The virtual machine is currently suspended.

resuming(8)     The virtual machine is currently in the process of resuming. This is a transient state from suspended state to running state.

paused(9)       The virtual machine is currently paused.

```
migrating(10)  The virtual machine is currently
                migrating.

shuttingdown(11)
                The virtual machine is currently in the
                process of shutting down.

shutdown(12)   The virtual machine is down.

crashed(13)    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. (read-only)

enable(2)      The autostart configuration of the
                virtual machine is enabled.

disable(3)     The autostart configuration of the
                virtual machine is disabled."
```

```
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.  
  
transient(3) The virtual machine is transient, i.e., the virtual machine does not exist after its power-off."  
SYNTAX INTEGER {  
unknown(1),  
persistent(2),  
transient(3)  
}

VirtualMachineCpuIndex ::= TEXTUAL-CONVENTION  
DISPLAY-HINT "d"  
STATUS current  
DESCRIPTION  
"A unique value, 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 virtual machine to the next re-initialization."  
SYNTAX Integer32 (1..2147483647)

VirtualMachineStorageIndex ::= TEXTUAL-CONVENTION  
DISPLAY-HINT "d"  
STATUS current  
DESCRIPTION  
"A unique value, 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 virtual machine 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.

```

SYNTAX      opticalDisk(4) The media type is optical disk."
            INTEGER {
                other(1),
                unknown(2),
                hardDisk(3),
                opticalDisk(4)
            }

```

VirtualMachineNetworkIndex ::= TEXTUAL-CONVENTION

DISPLAY-HINT "d"

STATUS      current

DESCRIPTION

"A unique value, greater than zero, identifying a virtual network interface allocated to a virtual machine. The value for each virtual network interface must remain constant at least from one re-initialization of the virtual machine to the next re-initialization."

SYNTAX      Integer32 (1..2147483647)

VirtualMachineList ::= TEXTUAL-CONVENTION

DISPLAY-HINT "1x"

STATUS      current

DESCRIPTION

"Each octet within this value specifies a set of eight Virtual Machine vmIndex, 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 a reboot. 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 VM from all other Virtual
        Machines in an administrative region. (*mrm -note-
        explain case when this value may be empty."
    ::= { 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  
    pause(3) to on(1)."  
 ::= { vmEntry 5 }

## vmOperState OBJECT-TYPE

SYNTAX VirtualMachineOperState  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
    "The current operational state of the virtual machine."  
 ::= { vmEntry 6 }

## vmAutoStart OBJECT-TYPE

SYNTAX VirtualMachineAutoStart  
MAX-ACCESS read-write  
STATUS current  
DESCRIPTION  
    "The autostart configuration of the virtual machine."  
 ::= { 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 shutting down."  
 ::= { vmEntry 8 }

vmCurCpuNumber OBJECT-TYPE  
SYNTAX Integer32 (0..2147483647)  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
"The number of virtual CPUs currently assigned to the  
virtual machine."  
 ::= { vmEntry 9 }

vmMinCpuNumber OBJECT-TYPE  
SYNTAX Integer32 (-1|0..2147483647)  
MAX-ACCESS read-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 may not persist across restarts 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 may not persist across restarts 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-only

```
STATUS          current
DESCRIPTION
    "The current memory size currently allocated to the
    virtual memory module in the unit designated by
    vmMemUnit."
 ::= { vmEntry 13 }

vmMinMem OBJECT-TYPE
SYNTAX          Integer32 (-1|0..2147483647)
MAX-ACCESS      read-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 may not persist
    across the restart 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 may not persist across the restart 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 to power
    on."
 ::= { 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."
 ::= { 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."
    ::= { 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."
    ::= { vmStorageEntry 13 }

vmStorageWriteIOs OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "The number of write I/O requests."
    ::= { 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 storage device 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 VM."
    ::= { vmObjects 9 }

vmBulkNotificationsEnabled OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "Indicates if notification generator will send
        notifications per set of VMs."
    ::= { 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' 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' from some other state.  The other state is
    indicated by the included value of vmOperState."
    ::= { vmNotifications 2 }

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' from some other state.  The other state is
    indicated by the included value of vmOperState."
    ::= { vmNotifications 3 }

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' from some other state. The other state is
    indicated by the included value of vmOperState."
 ::= { vmNotifications 4 }

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 5 }

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 6 }

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' from a all prior states except for 'running.'
    Management stations are encouraged to subsequently
    poll the subset of VMs of interest for vmOperState."
```

```
 ::= { vmNotifications 7 }

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' from a state other than 'shutdown'.
    Management stations are encouraged to subsequently poll
    the subset of VMs of interest for vmOperState."
 ::= { vmNotifications 8 }

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' from a state other than 'paused.'
    Management stations are encouraged to subsequently poll
    the subset of VMs of interest for vmOperState."
 ::= { vmNotifications 9 }

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' from a state other than 'suspended.'
    Management stations are encouraged to subsequently poll
    the subset of VMs of interest for vmOperState."
 ::= { vmNotifications 10 }

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 VMs of
        interest for vmOperState."
 ::= { vmNotifications 11 }

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 VMs of
    interest for vmOperState."
 ::= { vmNotifications 12 }

-- 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 vmAutoStart
```

```
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 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,
        vmShutdown,
        vmPaused,
        vmSuspended,
        vmCrashed,
        vmDeleted
    }
    STATUS          current
    DESCRIPTION
        "A collection of notifications for per-VM notification
        of changes to virtual machine state (vmOperState) as
        reported by a hypervisor."
    ::= { vmGroups 7 }

vmBulkNotificationsVariablesGroup OBJECT-GROUP
    OBJECTS {
        vmAffectedVMs
    }
    STATUS          current
    DESCRIPTION
        "The variables used in vmBulkNotificationOptionalGroup
        virtual network interfaces controlled by a hypervisor."
```



```
 ::= { vmGroups 8 }

vmBulkNotificationOptionalGroup NOTIFICATION-GROUP
  NOTIFICATIONS {
    vmBulkRunning,
    vmBulkShutdown,
    vmBulkPaused,
    vmBulkSuspended,
    vmBulkCrashed,
    vmBulkDeleted
  }
  STATUS          current
  DESCRIPTION
    "A collection of notifications for bulk notification of
    changes to virtual machine state (vmOperState) as
    reported by a given hypervisor."
  ::= { vmGroups 9 }

END
```

#### 4. IANA Considerations

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

Descriptor -----	OBJECT IDENTIFIER value -----
vmMIB	{ mib-2 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`, `vmAutoStart`, `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 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.

## 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.

## 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

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

Phone: +81 3 5841 6748  
Email: hiroshi@wide.ad.jp



Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: January 06, 2014

P. Fan  
L. Huang  
China Mobile  
M. Chen  
Huawei Technologies  
N. Kumar  
Cisco Systems  
July 05, 2013

IP Packet Loss Rate Measurement Testing and Problem Statement  
draft-fan-opsawg-packet-loss-01

Abstract

This document describes common methods for measuring packet loss rate and their effectiveness. Issues encountered when using the methods and necessary considerations are also discussed and recommended.

Status of This Memo

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

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

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

This Internet-Draft will expire on January 06, 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 . . . . .	2
2. Methods for Packet Loss Rate Measurement . . . . .	3
2.1. Active Approach . . . . .	3
2.1.1. Ping . . . . .	3
2.1.2. OWAMP and TWAMP . . . . .	3
2.1.3. Proprietary Tools . . . . .	4
2.2. Passive Approach . . . . .	4
2.2.1. Interface Statistics Report . . . . .	4
2.2.2. Coloring Based Performance Measurement . . . . .	5
3. Test on Packet Loss Rate Measurement . . . . .	5
3.1. Basic Test Information . . . . .	5
3.2. Ping with CLI vs. SNMP . . . . .	6
3.3. Ping Behaviors of Routers . . . . .	6
3.4. Statistics Report of Routers . . . . .	10
4. Measurement Issues . . . . .	10
4.1. Issues with Ping . . . . .	10
4.2. Issues with OWAMP and TWAMP . . . . .	11
4.3. Issues with Proprietary Tools . . . . .	11
4.4. Issues with Interface Statistics Report . . . . .	12
4.5. Issues with Coloring Based Performance Measurement . . . . .	12
5. Considerations and Recommendations . . . . .	12
6. Security Considerations . . . . .	14
7. IANA Considerations . . . . .	14
8. Acknowledgements . . . . .	14
9. References . . . . .	14
9.1. Normative References . . . . .	14
9.2. Informative References . . . . .	14
Authors' Addresses . . . . .	15

## 1. Introduction

IP packet loss rate is one of the important metrics that are frequently used to measure IP performance of a data path or link. A general framework of IP performance metrics is provided in [RFC2330], including fundamental concepts definition and issues related to defining sound metrics and methodologies. [RFC2680] and [RFC6673] further define metrics for one-way and round-trip packet loss.

In practical network operation, a number of methods are used by network engineers to calculate packet loss rate, and one of the common ways is to use ping. By checking ping statistics, people expect to get the idea of traffic transmission condition on the link. This document gives an overview of the frequently used methods for

measuring IP packet loss rate, and describes a test on packet loss rate measurement with multiple methods using routers from different vendors. Issues that should be taken into consideration during the measurement using different methods are discussed. Causes analysis and processing mechanisms of routers are also covered. It is expected that an operable measurement scheme with consistent testing results and equal treatment of network components can be reached.

## 2. Methods for Packet Loss Rate Measurement

This section describes common methods for measuring packet loss rate.

### 2.1. Active Approach

#### 2.1.1. Ping

Ping (ICMP echo request/reply) is a useful tool to examine the connectivity and performance of a path between two nodes in the network. The source node generates echo request packets with configured size, interval, count and other settings, and the destination node sends back an echo reply packet once it receives a request. Then we count the packets sent out and received and get the round-trip packet loss rate on the link between source and destination. This approach is clear and convenient, and is frequently used by engineers when packet loss rate is needed.

In practical network operation, the ping testing can be initiated manually and directly on the node by engineers, for example through the command line interface (CLI) of a router, or activated indirectly by instructions, for example through SNMP messages sent from network management system.

No matter through CLI or SNMP, ping testing can be conducted directly on the endpoint devices of the link to be tested, or other nodes as long as the request/reply packets pass through the link. Those nodes are often referred to as probes, which can be a router or a PC server, directly connected or indirectly reachable to the endpoints. Usually the probes and paths to the endpoints are not supposed to be congested to avoid affecting the ping testing result.

#### 2.1.2. OWAMP and TWAMP

The One-way Active Measurement Protocol (OWAMP, [RFC4656]) and Two-Way Active Measurement Protocol (TWAMP, [RFC5357]) are defined by the IP Performance Metrics (IPPM) working group. They provide a method and protocol for measuring delay and packet loss of IP flows, and are designed for wide scale deployment in the network to provide ubiquitous performance data. Both OWAMP and TWAMP use control

protocol and test protocol. The control protocol is used to negotiate test session between test endpoints, start and stop the test, and fetch the test result for OWAMP. The test protocol runs over UDP and conducts the test.

OWAMP can be used to perform one-way packet loss measurement, and requires synchronized time defined by GPS. The test results are collected at the receiving endpoints and returned using the control protocol. TWAMP is more simplified, and used for two-way packet loss measurement. The opposite endpoint is regarded as a reflector, and the test results are collected at the sender.

### 2.1.3. Proprietary Tools

There are some other proprietary performance measurement tools incorporating embedded and external probes. The probes generate and inject extra packets into the network to mimic the service flows that are intended to be tested. The performance of the target service flows can be evaluated by measuring the performance of the injected packets. Compared with Ping, these proprietary tools normally support more services, which include not only ICMP, but TCP, UDP, HTTP, etc.

The embedded proprietary tools have been widely implemented by routers to provide automatic detection of IP performance. Examples of this kind of tools include RPM (Juniper), IPSLA (Cisco), NQA (Huawei/H3C), SAA (ALU), etc. By necessary configurations on the router, the embedded tools support multi-service testing of multiple queues on an interface. Packet loss rate can be measured with ICMP ping function of the tool. Routers send out ICMP packets automatically according to the configured parameters, so the embedded tool is working in a similar way as ping method described above.

## 2.2. Passive Approach

### 2.2.1. Interface Statistics Report

Forwarding devices maintain statistics report of every interface. The report shows the detailed status of the interface as well as traffic information, including inbound and outbound speed and packet count. For a typical router, traffic statistics show number of packets transmitted and discarded by an interface, and even on the basis of QoS queue, so the entire packet loss rate of a link or packet loss rates regarding different queues can be calculated. Traffic data on the report can be displayed through CLI or obtained using SNMP which allows automatic packet loss sampling.

### 2.2.2. Coloring Based Performance Measurement

The concept of coloring based performance measurement is introduced in [I-D.tempia-opsawg-p3m], and [I-D.chen-coloring-based-ipfpm-framework] defines a framework for coloring based IP Flow Performance Measurement (IPFPM). By periodically setting/changing one or more bits of the IP header of the packets that belong to an IP flow to "color" the packets into different colors, the IP flow is split into different consecutive blocks. Packets in the same block have the same color and packets in consecutive blocks have different colors. This method gives a way to a measurement node to count and calculate, without inserting any extra auxiliary OAM packets, packet loss based on each color block. Since the measurement is based on the real traffic data, the measurement results will reflect the real performance of the tested flow.

## 3. Test on Packet Loss Rate Measurement

This section describes test result on packet loss rate measurement using different methods. Test equipment covers routers from several vendors. Results show the diverse outcome of the methods used, and the diverse responding mechanism of routers.

### 3.1. Basic Test Information

The basic topology of testing can be depicted as follows.

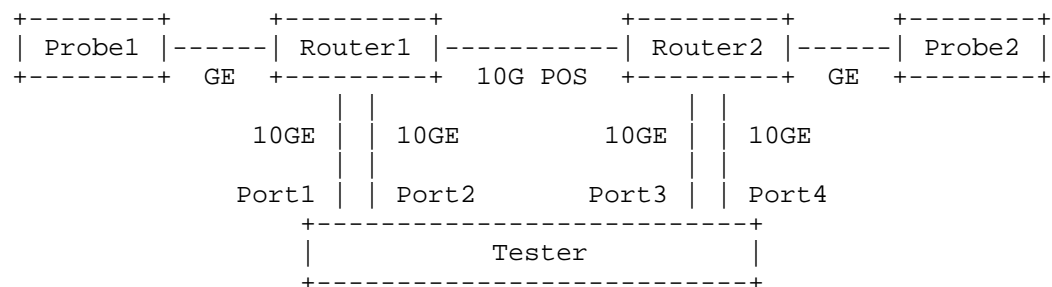


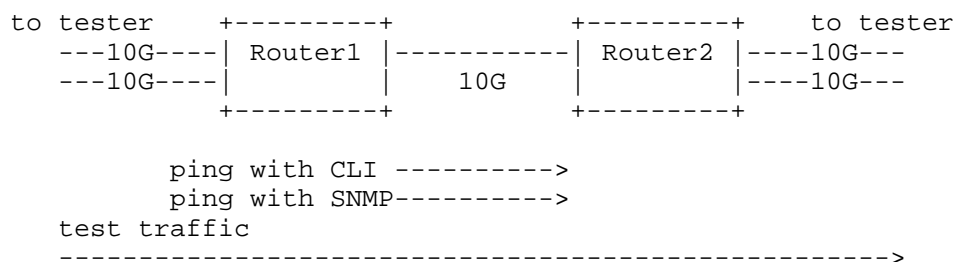
Figure 1: Basic topology for packet loss rate test

Two routers are connected by a 10G POS link, and each router is connected to the tester by two 10GE links. The tester generates unidirectional/bidirectional traffic between port 1 and port 3, and between port 2 and port 4, with frame length of 400 bytes. The total volume of traffic injected into a router by the tester is more than 10G, leading to congestion when the traffic passes through the 10G POS link between the two routers. Routers and probes generate ping packets for testing, with frame length of 400 and DSCP field of 0.

We tested routers from 3 vendors, indicated as A, B, and C in the following parts of discussion. The tester generated different levels of congestion, and we tested packet loss rates on the 10G POS interconnection link on those congestion levels with CLI, SNMP, and interface statistics report.

### 3.2. Ping with CLI vs. SNMP

Some routing boxes by default treat ping packets generated with CLI and SNMP in different ways. The following is a test on this issue.



The tester generates test traffic at 20 Gbps, and sends the traffic into a router of vendor A. The traffic goes through the 10G interconnecting link and past the router of vendor B on the other end. We use ping with CLI and SNMP on router A to test packet loss rate on the interconnecting link. The DSCP fields of test traffic and ping packets are all left to be 0..

By default, router A forwards the test traffic with the basic priority, like BE class. The ping packets with CLI are also treated as of best effort class, but ping packets with SNMP are given a higher priority, some class like network control. So the two kinds of ping are actually testing packet loss of streams in different classes. The test result verifies the issue. Ping with SNMP shows no packet loss, and ping with CLI shows a packet loss rate of around 50%.

The forwarding class of ICMP packets can be configured on router A. In the following tests we put all traffic in the same basic class.

### 3.3. Ping Behaviors of Routers

We considered the following test cases (TCs) when investigating packet loss rate with ping on the link between two different routers.

TC 1: Router sends ICMP echo request packets with SNMP instruction to the peering router.



```

+-----+           +-----+
| Router1 |-----| Router2 |
+-----+           +-----+

ping with SNMP----->

```

TC 2: Router sends ICMP echo request packets with CLI to the peering router.

```

+-----+           +-----+
| Router1 |-----| Router2 |
+-----+           +-----+

ping with CLI ----->

```

TC 3: Router sends ICMP echo request packets with SNMP instruction to the probe behind the peering router.

```

+-----+           +-----+           +-----+
| Router1 |-----| Router2 |-----| Probe2 |
+-----+           +-----+           +-----+

ping with SNMP----->

```

TC 4: Router sends ICMP echo request packets with CLI to the probe behind the peering router.

```

+-----+           +-----+           +-----+
| Router1 |-----| Router2 |-----| Probe2 |
+-----+           +-----+           +-----+

ping with CLI ----->

```

TC 5: Probe behind router sends ICMP echo request packets to the probe behind the peering router.

```

+-----+           +-----+           +-----+           +-----+
| Probe1 |-----| Router1 |-----| Router2 |-----| Probe2 |
+-----+           +-----+           +-----+           +-----+

ping with CLI----->

```

The link between the two routers is injected bidirectional or unidirectional test traffic to cause congestion. The packet loss rate of test traffic is calculated with the Rx and Tx rate on the tester. We use router A, B and C in pairs and get the ICMP packet loss rate in each test case. The comparison of the packet loss rate of ICMP and test traffic shows diverse behaviors of ping process on routers. The following tables show the test results

Pkt loss rate of test traffic		ICMP pkt loss rate (echo req drct: A->B)					ICMP pkt loss rate (echo req drct: B->A)				
A->B	B->A	TC1	TC2	TC3	TC4	TC5	TC1	TC2	TC3	TC4	TC5
48.60%	48.60%	54%	56%	80%	76%	73%	54%	54%	58%	58%	77%
28%	28%	27%	30%	61%	58%	47%	32%	32%	27%	21%	53%
7.60%	7.60%	9%	12%	15%	18%	21%	13%	15%	11%	11%	21%
48.60%	No traffic	54%	56%	57%	54%	54%	62%	56%	54%	48%	56%
28%	No traffic	31%	33%	32%	33%	33%	36%	34%	34%	35%	35%
7.60%	No traffic	14%	13%	12%	9%	14%	14%	13%	11%	12%	14%
No traffic	48.60%	1%	0%	54%	50%	47%	1%	1%	0%	1%	50%
No traffic	28%	0%	0%	26%	31%	28%	0%	0%	0%	0%	28%
No traffic	7.60%	0%	0%	10%	9%	9%	0%	0%	0%	0%	8%

Table 1: Test result when interconnecting router A and router B

Pkt loss rate of test traffic		ICMP pkt loss rate (echo req drct: A->B)					ICMP pkt loss rate (echo req drct: C->A)				
A->C	C->A	TC1	TC2	TC3	TC4	TC5	TC1	TC2	TC3	TC4	TC5
48.70%	44.70%	58%	54%	57%	58%	53%	57%	55%	48%	57%	56%
28%	22.40%	38%	31%	37%	33%	35%	30%	33%	33%	37%	35%
7.70%	7.30%	14%	13%	13%	13%	12%	16%	13%	15%	16%	14%
48.80%	No traffic	50%	54%	51%	53%	55%	54%	56%	55%	59%	57%
28%	No traffic	27%	29%	32%	32%	33%	35%	30%	35%	33%	33%
7.60%	No traffic	11%	10%	15%	15%	13%	11%	11%	15%	15%	13%
No traffic	44.50%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
No traffic	22.60%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
No traffic	7.74%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%

Table 2: Test result when interconnecting router A and router C

Pkt loss rate of	ICMP pkt loss rate	ICMP pkt loss rate
------------------	--------------------	--------------------

test traffic		(echo req drct: C->B)			(echo req drct: B->C)		
C->B	B->C	TC1	TC2	TC5	TC1	TC2	TC5
48.76%	44.69%	1%	0%	54%	0%	1%	50%
28.04%	22.29%	0%	0%	40%	0%	1%	29%
7.62%	7.62%	0%	0%	11%	0%	0%	8%
48.69%	No traffic	0%	0%	0%	0%	0%	0%
28.03%	No traffic	0%	0%	0%	0%	0%	0%
7.62%	No traffic	0%	0%	0%	0%	0%	0%
No traffic	44.50%	1%	0%	51%	0%	1%	51%
No traffic	22.29%	0%	0%	29%	0%	0%	29%
No traffic	7.74%	0%	0%	9%	0%	0%	10%

Table 3: Test result when interconnecting router C and router B

The behaviors of the three vendors' routers are summarized here, and we leave the discussion on reasons for the behaviors to the next section.

Router A: Ping by router A with SNMP, CLI and by the probe behind router A lead to similar usable results. However, all the methods encounter larger errors when the test traffic is less congested.

Router B: Ping by router B with SNMP and CLI will not report correctly the packet loss rate of test traffic. Ping by the probe behind router B gives usable result of packet loss rate, but also with certain errors.

Router C: Ping by router C with SNMP, CLI and by the probe behind router C will not report correctly the packet loss rate of test traffic.

We can further highlight the outcomes when testing the packet loss rate on the interconnection link between each pair of routers.

Router A - router B: If one wants to get relatively accurate value of packet loss rate in all congestion scenarios, he is advised to use ping between probes (test case 5), or have A generate ping to the probe behind B.

Router A - router C: All the test methods will only reflect the outbound packet loss rate of A.

Router B - router C: Packet loss rate is difficult to measure with this combination- only using ping between probes (test case 5) can reflect the outbound packet loss rate of B.

### 3.4. Statistics Report of Routers

We also checked the interface statistics reports given with CLI on the 3 routers, and we confirmed that the outbound packet loss rate of an interface obtained from the statistics report was in accordance with the actual packet loss rate of test traffic. The following table shows the test result.

Router	Outbound pkt loss rate of test traffic	Outbound pkt loss rate shown on statistics report
A	48.52%	48.52%
B	48.52%	48.52%
C	44.60%	44.60%

Table 4: Test result when referring to the statistics report on routers

## 4. Measurement Issues

This section describes issues encountered when measuring the packet loss rate of a link using different testing methods.

### 4.1. Issues with Ping

Routers from every vendor have their unique processing procedure when sending and receiving ICMP packets, thus resulting in diverse ping packet loss rates, as described in the section above. Errors exist using the ping method, and in some cases ping no longer reflects the actual packet loss rate correctly. Relevant issues that have to be taken into account include:

**Forwarding class:** When sending ping packets locally, routers are likely to put the packets into a certain QoS queue/class although the DSCP field of ICMP packets is kept zero. QoS queue of ping may be different than that of the traffic to be measured, and even ping packets sent by CLI commands and SNMP are in different queues by default. Usually forwarding class can be adjusted by CLI or SNMP commands.

**Inner priority:** For some routers, although ping traffic and service traffic will not be treated differently by QoS, packets sent out by the router itself, for example ping packets, are put into an inner high priority while other forwarding service traffic into low priority. These kinds of inner priority are valid within the interior of routers and do not rewrite the packets. One of the

purposes of using the priorities is to get the protocol packets (ping included) processed in prior. These priorities are set by vendor and may not be able to adjust, so in this case ping will not give the correct packet loss rate as ping packets are not processed and discarded together with service traffic.

Ingress line card: If the ping testing is conducted on a probe which is connected or IP reachable to the router, then the ping packets will be treated by the router as forwarding traffic, eliminating the queue and priority issues. However, the location of interfaces through which ingress traffic is received matters when using some types of routers. In this case, the router employs a polling schedule which allows traffic from different line cards or modules to get forwarding chance. For a card with small volume of traffic, the chance will be little but not none. So if ping packets come through a card different from the high-volume service traffic, the packets would probably get enough forwarding resources as ping traffic itself requires little bandwidth. As a result, ping will suffer little from congestion and shows disaccord in packet loss rate.

Internal rate limitation: Routers normally have rate limitation towards CPU, which is considered a kind of protection to the control plane of routers. So if a packet is sent to CPU for processing rather than line card ASIC (e.g. in many routers, an ICMP echo reply packet received in response to an earlier echo request packet sent by the router will be sent to the CPU), it might be influenced by the rate limiter. Typical rate limitation of ICMP packets would be 1000 pps, though the value is highly dependent on vendor implementation and can be configured. In practical deployment, if there is a large number of ICMP packets sending to a router, the ping test packets may be dropped, causing test errors. This problem did not arise in our test in section 3 as the ICMP traffic is rather small.

#### 4.2. Issues with OWAMP and TWAMP

OWAMP and TWAMP fall into the category of active measurement, so the general issues of active measurement apply to them. When using the two methods, one is advised to make sure that the measurement traffic will have the same drop probability as non-measurement traffic. However, it is usually difficult to guarantee this, as too many factors effect the behavior of traffic.

#### 4.3. Issues with Proprietary Tools

Since the proprietary tools are implemented by vendors independently, interoperability is one of the major issues when using the tools,

especially for one-way measurement. Besides, these tools also share the common issues of active measurement. The accuracy of results depends on the rate, numbers and interval of the injected packets. It also needs to guarantee that the injected packets follows the same path as the tested packets, otherwise the results cannot reflect the real performance.

Although these tools provide automatic testing method, the basic principle is still to ping from the router itself. So it is believed toolset method will experience the same issues about class and priority as local ping from router does. However, we did not test diagnosis toolsets, and the discussion is left to be further continued.

#### 4.4. Issues with Interface Statistics Report

Interface statistic is the most direct and accurate way to get performance of an interface. Packet loss rate calculated from traffic statistics is in accordance with the expected value. By referring to statistics collected from the endpoint routers, bidirectional packet loss rate can easily be obtained.

However, this approach requires access to routers, while in some scenarios it is difficult to do that. For example, if we would like to know the inbound packet loss rate of the interconnection link to another service operator, we may have to rely on statistics provided by the peering router. Normally, this information is not easily shared by interworking operators.

#### 4.5. Issues with Coloring Based Performance Measurement

The challenge for coloring based performance measurement is that there are not so many bits in the IP header that can be used for IP packet coloring. Operators have to carefully think of the color bits selection to make sure that the setting and changing of the color bits will not affect the normal packet forwarding and process.

### 5. Considerations and Recommendations

We summarize the above analysis here and come to the following considerations:

- a. The ping method to measure packet loss rate is easy to be influenced by the diverse processing mechanism of ICMP packet within routers. If this method is to be used on a router, one is advised to make sure that the ICMP packets experience the same forwarding and discarding courses as the service traffic (of which the packet loss rate is to be measured) does, otherwise the

measurement will not make sense. When measuring with ping, the following points are also worth reminding:

- \* Packet loss rate given by measurement with ping is a value related to a certain forwarding class in which the ICMP packets are forwarded. So it is not a scientific way to say what the packet loss rate is on a link if traffic is transmitted in more than one class on the link.
  - \* Measurement with ping is enough if one only wants to get a general, qualitative picture of packet loss. But if one is to measure precisely and quantitatively, possible errors (sometimes very large errors) should be taken into account.
  - \* If configured in the right way on router, ping with CLI and SNMP lead to similar results.
- b. It is more likely to get good results if a probe is used to perform ping measurement (though not 100% guaranteed), but following issues also need to be considered.
- \* If the probe is directly connected to a router, then a router port is occupied. This will be a problem for routers with limited or expensive port resources, as the probing traffic is usually extremely small.
  - \* If the probe is more than one hop away from a router, load of the path to the router is supposed to be under the congestion level.
- c. Interface statistics report gives us the most accurate value of packet loss rate, and the value is irrelevant to router platforms. From the report we can find numbers of packets being received, transmitted, and discarded in different classes within a period of time, thus we get packet loss rate. Actually this is indeed how packet loss rate is defined.
- \* Referring to report requires access to routers, which may be easier if routers are within a single administrative area. However it gets annoying if more routers are evolved, for instance measurement on a long path with a number of routers.
  - \* Router interface report only gives the outbound packet loss rate. If we want to see if traffic in the other direction is congested, we'll have to check the upstream routers in that direction. This will be difficult on certain links, say, interconnection link to another provider.

## 6. Security Considerations

TBD.

## 7. IANA Considerations

This memo includes no request to IANA.

## 8. Acknowledgements

The authors would like to thank Brian Trammell for the kind comments.

## 9. References

### 9.1. Normative References

- [RFC2680] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way Packet Loss Metric for IPPM", RFC 2680, September 1999.
- [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.
- [RFC4656] Shalunov, S. and B. Teitelbaum, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, September 2006.
- [RFC5357] Hedayat, K. and R. Krzanowski, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, October 2008.
- [RFC6673] Morton, A., "Round-Trip Packet Loss Metrics", RFC 6673, August 2012.
- [RFC792] Postel, J., "Internet Control Message Protocol", RFC 792, September 1981.

### 9.2. Informative References

- [I-D.chen-coloring-based-ipfpm-framework]  
Chen, M., Liu, H., and Y. Yin, "Coloring based IP Flow Performance Measurement Framework", draft-chen-coloring-based-ipfpm-framework-01 (Work in Progress), February 2013.
- [I-D.tempia-opsawg-p3m]  
Capello, A., Cociglio, M., Castaldelli, L., and A. Bonda, "A packet based method for passive performance monitoring", draft-tempia-opsawg-p3m-03 (Work in Progress), February 2013.



[RFC2330] Paxson, V., Almes, G., Mahdavi, J., and M. Mathis,  
"Framework for IP Performance Metrics", RFC 2330, May  
1998.

Authors' Addresses

Peng Fan  
China Mobile  
32 Xuanwumen West Street, Xicheng District  
Beijing 100053  
P.R. China

Email: fanpeng@chinamobile.com

Lu Huang  
China Mobile  
32 Xuanwumen West Street, Xicheng District  
Beijing 100053  
P.R. China

Email: huanglu@chinamobile.com

Mach(Guoyi) Chen  
Huawei Technologies

Email: mach.chen@huawei.com

Nagendra Kumar  
Cisco Systems

Email: naikumar@cisco.com

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

Y. Chen  
China Mobile  
D. Liu

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

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

Abstract

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

Status of This Memo

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

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

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

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

Copyright Notice

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

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

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

## Table of Contents

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

## 1. Introduction

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

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

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

## 2. Terminology

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

This document uses the following abbreviations:

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

## 3. CAPWAP 802.11n Support

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

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

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

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

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

### 3.1. CAPWAP Extension for 802.11n Support

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

#### 3.1.1. 802.11n Radio Capability Information

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

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

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

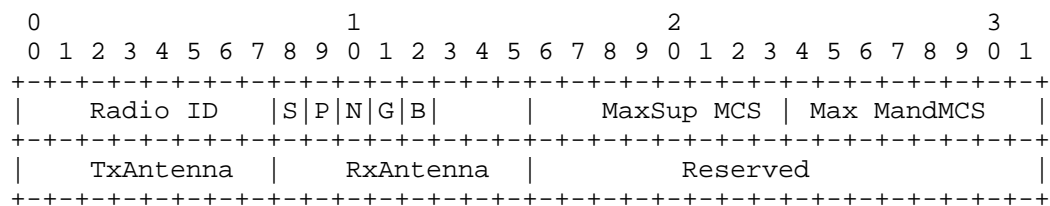


Figure 1: 802.11n Radio Configuration Message Element

Type: TBD1 for 802.11n Radio Configuration Message Element.

Length: 16.

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

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

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

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

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

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

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

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

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

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

The detail definition of TxAntenna/RxAntenna is as follows:

```

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

```

Figure 2: Definition of TxAntenna/RxAntenna

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

### 3.1.3. 802.11n Station Information

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

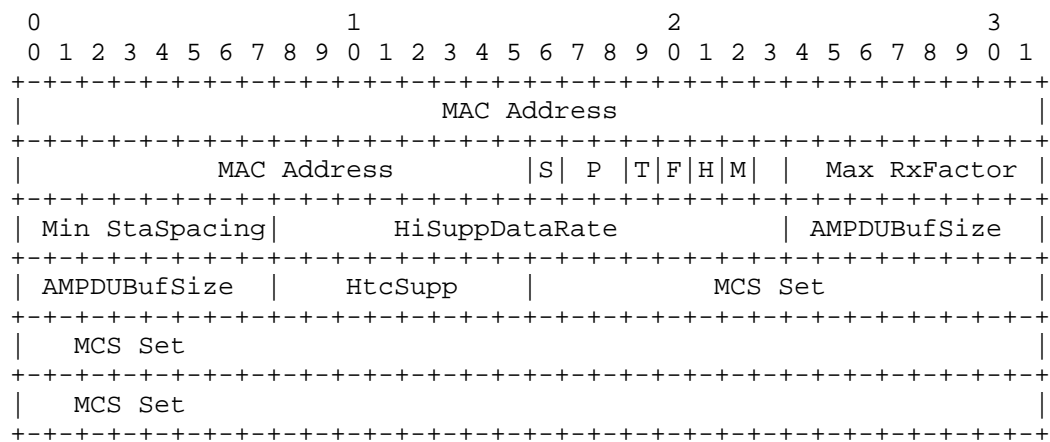


Figure 3: 802.11n Station Information

MAC Address: The station's MAC Address.

Type: TBD2 for 802.11 Station Information.

Length: 24.

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

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

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

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

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

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

Max RxFactor: The maximal receiving A-MPDU factor.

Min StaSpacing: Minimum MPDU Start Spacing.

HiSuppDataRate: Maximal transmission speed (Mbps).

AMPDUBufSize: A-MPDU buffer size (Byte).

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

MCS Set: The MCS bitmap that the station supports.

#### 4. Power and Channel Autoconfiguration

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

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

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

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



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

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

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

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

#### 4.3. Channel/Power Auto Adjustment

The Channel Scan Procedure is illustrated by the figure 4.

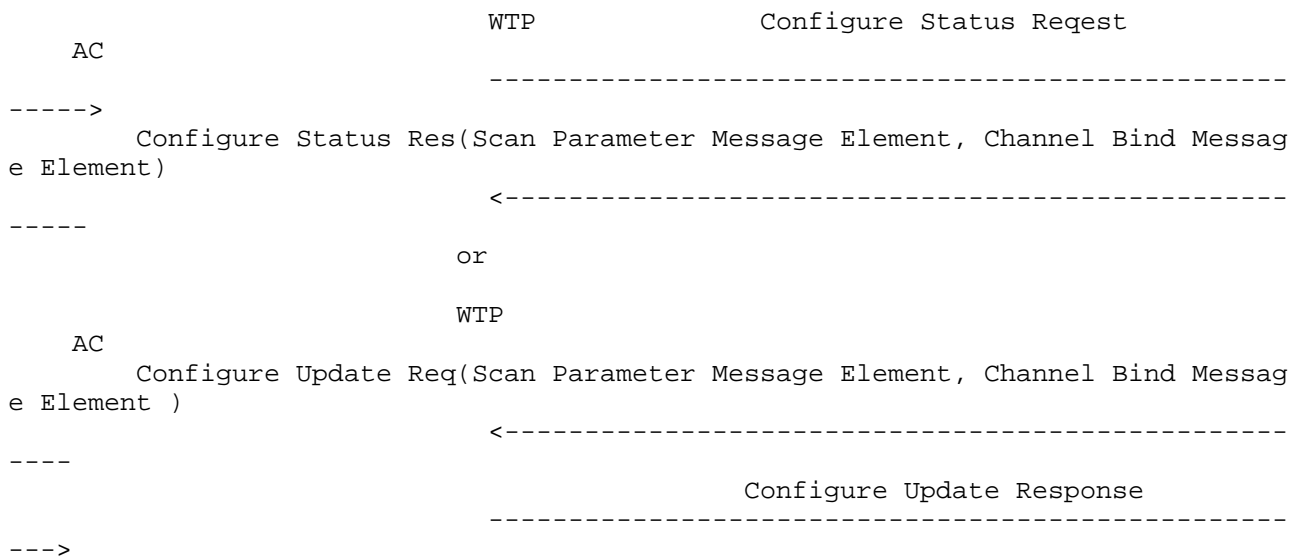


Figure 4: Channel Scan Procedure

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

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

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

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

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

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

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

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

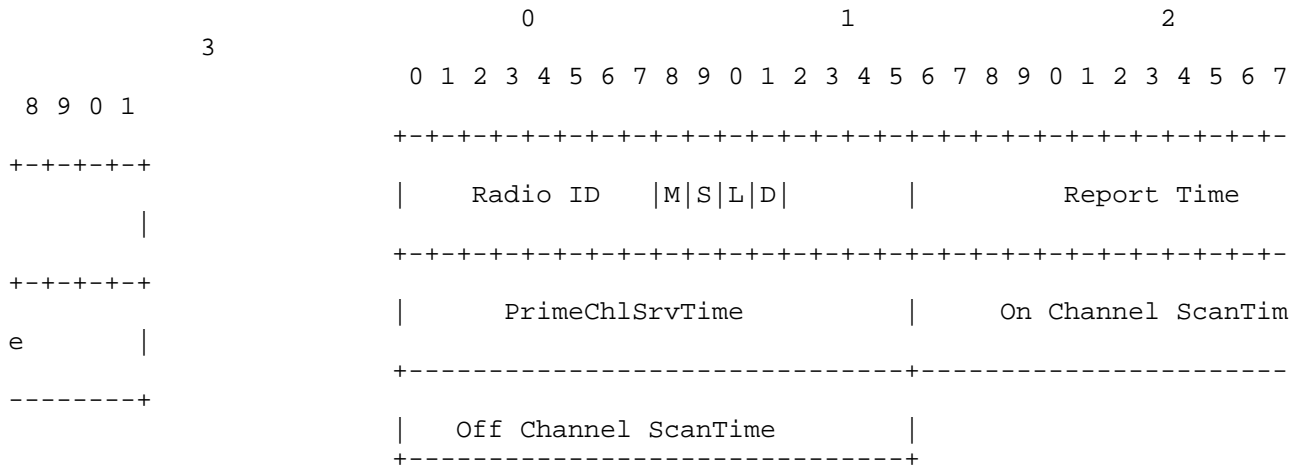


Figure 5: IEEE 802.11 Scan Parameters Message Element

Type: TBD3 for IEEE 802.11 Scan Parameters Message Element.

Length: 10.

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

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

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

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

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

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

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

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

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

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

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

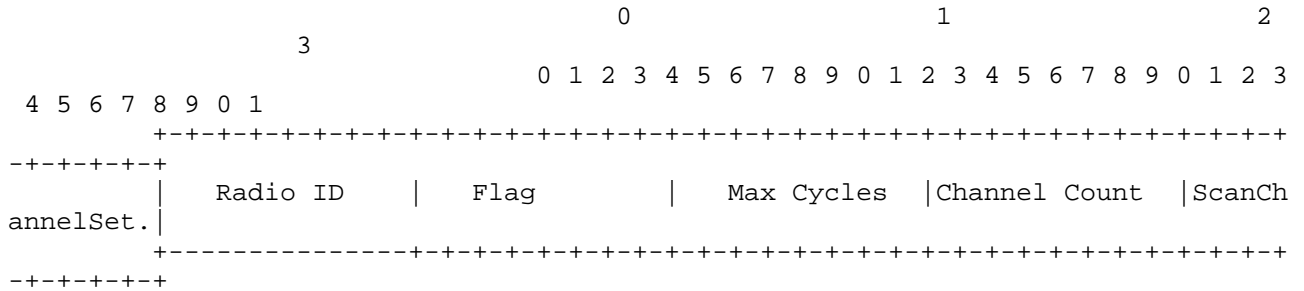


Figure 6: IEEE 802.11 Scan Channel Bind Message Element

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

Length: variable.

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

Flag: reserved.

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

Channel Count: The number of channels will be scanned.

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

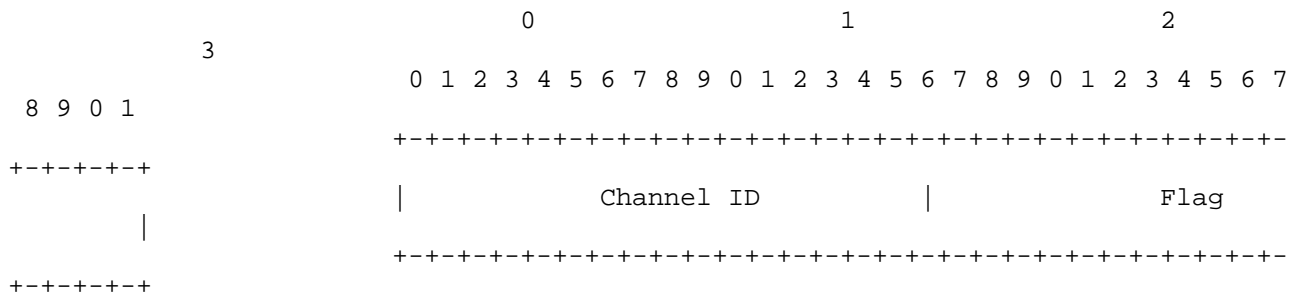


Figure 7: Channel Information Format

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

Flag: Bitmap, reserved for future use.

#### 4.3.3. IEEE 802.11 Channel Scan Report

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

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

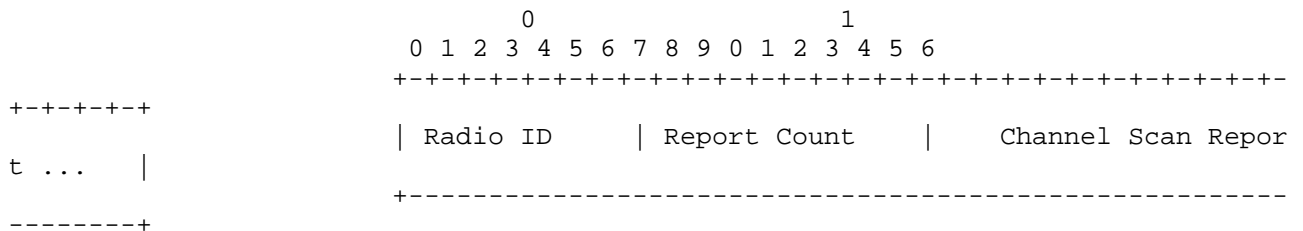


Figure 8: IEEE 802.11 Channel Scan Report Message Element

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

Length: >=29.

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

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

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

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

Figure 9: Channel Scan Report

Channel Number: The channel number.

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

Mean Time: Channel measurement duration (ms).

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

Screen Packet Count: Received packet number.

Neighbor Count: The neighbor number of this channel.

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

Interference: The interference of the channel.

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

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

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

CRC Err Cnt: CRC err packet number.



Decrypt Err Cnt: Decryption err packet number.

Phy Err Cnt: Physical err packet number.

Retrans Cnt: Retransmission packet number.

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

#### 4.3.4. IEEE 802.11 WTP Neighbor Report

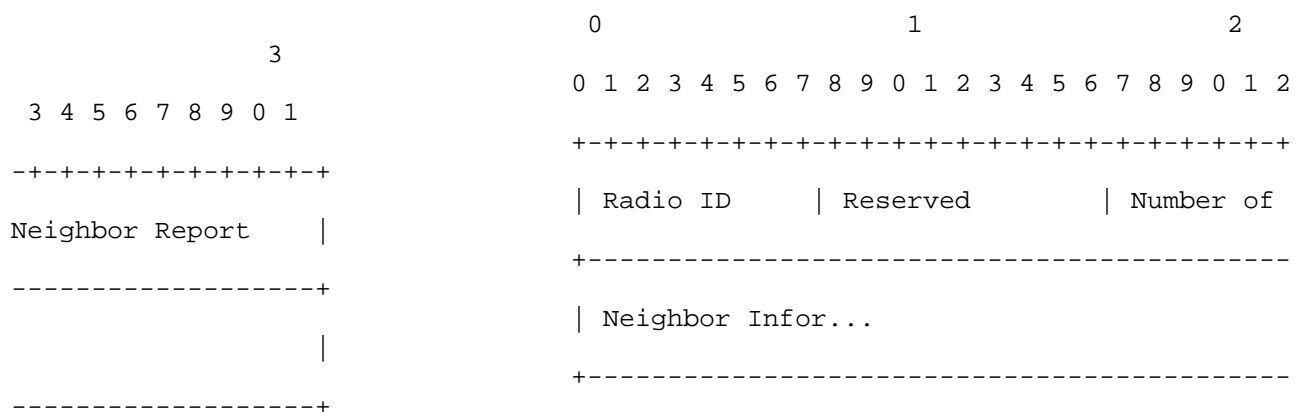


Figure 10: WTP Neighbor Report TLV

The definition of Neighbor info is as follows:

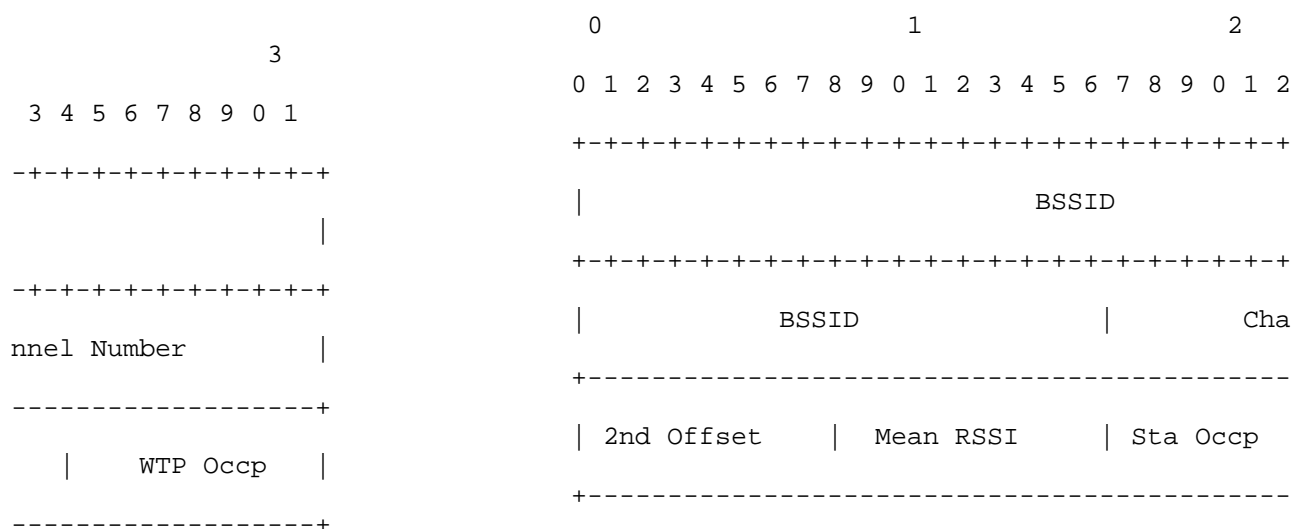


Figure 11: Neighbor info

BSSID: The BSSID of this neighbor WTP.

Channel Number: The channel number of this WTP neighbor.

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





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

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

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

## 5. Security Considerations

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

## 6. IANA Considerations

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

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

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

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

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

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

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

## 7. Contributors

This draft is a joint effort from the following contributors:

Gang Chen: China Mobile chengang@chinamobile.com

Naibao Zhou: China Mobile zhounaibao@chinamobile.com

Chunju Shao: China Mobile shaochunju@chinamobile.com

Hao Wang: Huawei3Come hwang@h3c.com

Yakun Liu: AUTELAN liuyk@autelan.com

Xiaobo Zhang: GBCOM

Xiaolong Yu: Ruijie Networks

Song zhao: ZhiDaKang Communications

Yiwen Mo: ZhongTai Networks

Dorothy Stanley: dstanley1389@gmail.com

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

## 8. Acknowledgements

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

## 9. Normative References

[IEEE-802.11.2009]

"IEEE Standard for Information technology - Telecommunications and information exchange between systems Local and metropolitan area networks - Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Enhancements for Higher Throughput (Amendment 5)", 2009.

[IEEE-802.11.2012]

"IEEE Standard for Information technology - Telecommunications and information exchange between systems Local and metropolitan area networks - Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", March 2012.

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

- [RFC4564] Govindan, S., Cheng, H., Yao, ZH., Zhou, WH., and L. Yang, "Objectives for Control and Provisioning of Wireless Access Points (CAPWAP)", RFC 4564, July 2006.
- [RFC5415] Calhoun, P., Montemurro, M., and D. Stanley, "Control And Provisioning of Wireless Access Points (CAPWAP) Protocol Specification", RFC 5415, March 2009.
- [RFC5416] Calhoun, P., Montemurro, M., and D. Stanley, "Control and Provisioning of Wireless Access Points (CAPWAP) Protocol Binding for IEEE 802.11", RFC 5416, March 2009.

## Authors' Addresses

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

Email: chen yifan@chinamobile.com

Dapeng Liu  
Beijing  
China

Email: maxpassion@gmail.com

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

Email: denghui@chinamobile.com

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

Email: lei.zhu@huawei.com

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

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

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

Abstract

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

Status of This Memo

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

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

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

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

#### Copyright Notice

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

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

#### Table of Contents

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

#### 1. Introduction

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

experience interoperability issues with WTPs and ACs from different vendors.

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

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

Figure 1: Functions in Local MAC and Split MAC

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

## 2. IEEE MAC Profile Descriptions

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

### 2.1. Split MAC with WTP encryption

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



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

Figure 2: Functions in Split MAC with WTP Encryption

## 2.2. Split MAC with AC encryption

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

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

Figure 3: Functions in Split MAC with AC encryption

### 2.3. IEEE 802.11 MAC Profile Frame Exchange

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

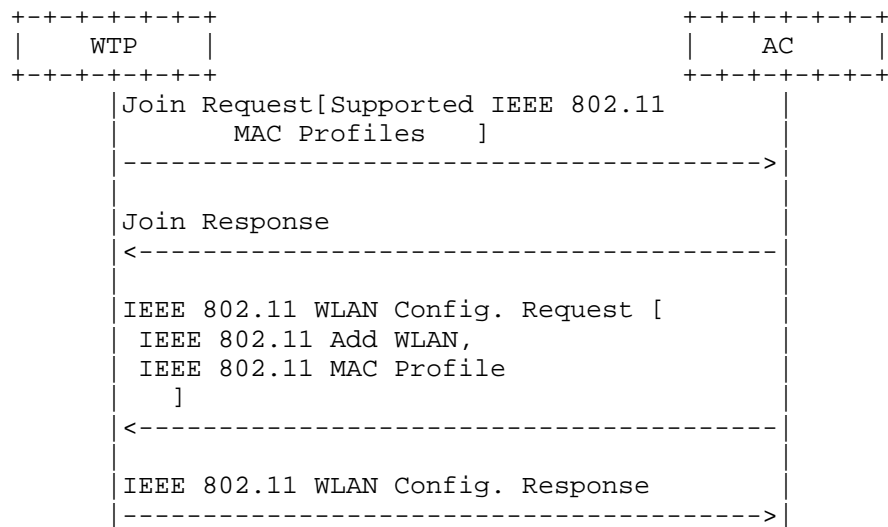


Figure 4: Message Exchange For Negotiating MAC Profile

### 3. MAC Profile Message Element Definitions

#### 3.1. IEEE 802.11 Supported MAC Profiles

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

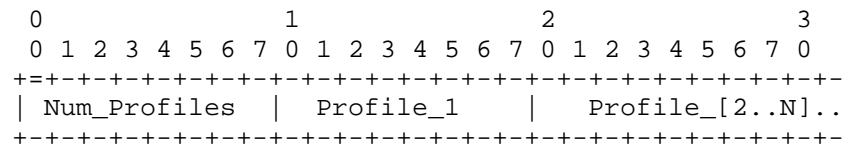


Figure 5: IEEE 802.11 Supported MAC Profiles

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

### 3.2. IEEE 802.11 MAC Profile

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

```

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

```

Figure 6: IEEE 802.11 MAC Profile

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

### 4. Security Considerations

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

### 5. IANA Considerations

This document requires the following IANA actions:

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

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

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

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

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

## 6. Contributors

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

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

## 7. Acknowledgments

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

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

## 8. Normative References

- [RFC5415] Calhoun, P., Montemurro, M., and D. Stanley, "Control And Provisioning of Wireless Access Points (CAPWAP) Protocol Specification", RFC 5415, March 2009.
- [RFC5416] Calhoun, P., Montemurro, M., and D. Stanley, "Control and Provisioning of Wireless Access Points (CAPWAP) Protocol Binding for IEEE 802.11", RFC 5416, March 2009.

## Authors' Addresses

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

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

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

Email: denghui@chinamobile.com

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

Email: rpazhyan@cisco.com

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

Email: farooq.bari@att.com

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

Email: zhangr@gsta.com

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

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

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

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

Mechanisms for Optimizing LAG/ECMP Component Link Utilization in  
Networks

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

Status of this Memo

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

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

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

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

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

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

Copyright Notice

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

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

## Abstract

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

## Table of Contents

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



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

## 1. Introduction

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

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

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

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

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

### 1.1. Acronyms

DOS: Denial of Service

ECMP: Equal Cost Multi-path

GRE: Generic Routing Encapsulation

LAG: Link Aggregation Group

MPLS: Multiprotocol Label Switching

NVGRE: Network Virtualization using Generic Routing Encapsulation

PBR: Policy Based Routing

QoS: Quality of Service

STT: Stateless Transport Tunneling

TCAM: Ternary Content Addressable Memory

VXLAN: Virtual Extensible LAN

### 1.2. Terminology

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

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

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

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

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

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

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

## 2. Flow Categorization

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

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

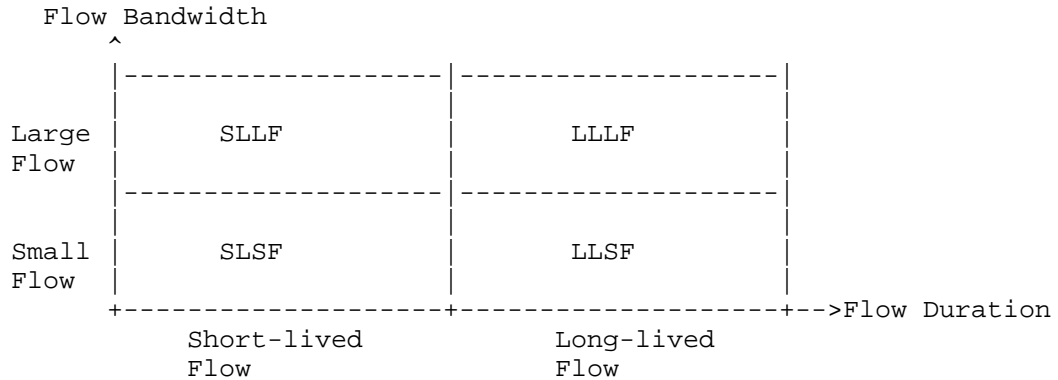


Figure 1: Flow Categorization

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

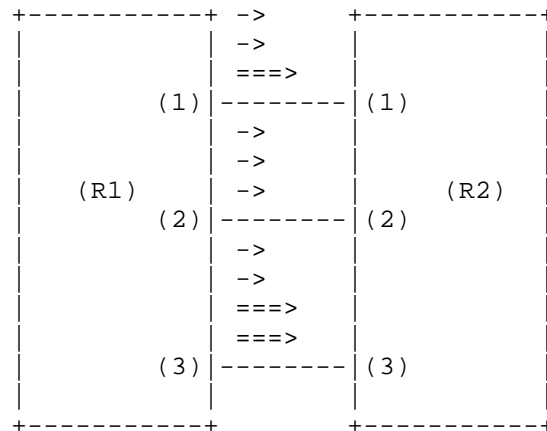
### 3. Hash-based Load Distribution in LAG/ECMP

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

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

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

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



Where: ->    small flow  
      ==>    large flow

Figure 2: Unevenly Utilized Component Links

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

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

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

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

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

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

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

#### 4.1. Differences in LAG vs ECMP

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

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

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

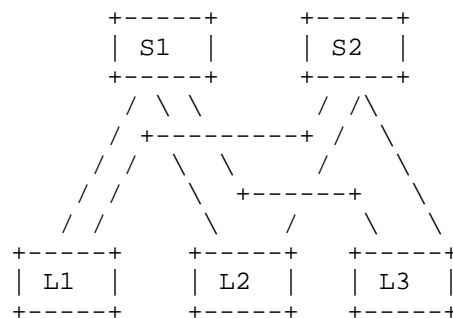


Figure 3: Two-level Clos Network

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

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

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

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

#### 4.2. Operational Overview

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

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

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

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

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

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

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

#### 4.3. Large Flow Recognition

##### 4.3.1. Flow Identification

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

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

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



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

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

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

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

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

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

Various techniques to recognize a large flow are described below.

#### 4.3.3. Sampling Techniques

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

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

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

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

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

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

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

The advantages and disadvantages of sampling techniques are as follows.

Advantages:

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

Disadvantages:

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

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

#### 4.3.4. Inline Data Path Measurement

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

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

The advantages and disadvantages of inline data path measurement are:

##### Advantages:

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

##### Disadvantages:

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

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

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

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

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

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

#### 4.4. Load Rebalancing Options

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

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

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

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

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

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

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

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

#### 4.4.2. Redistributing Small Flows

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

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

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

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

#### 4.4.3. Component Link Protection Considerations

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

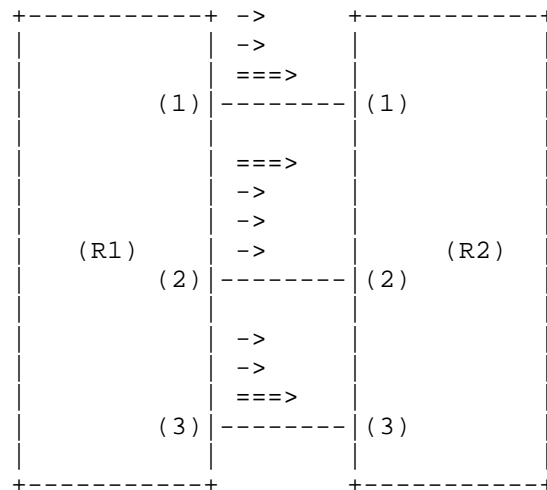
#### 4.4.4. Load Rebalancing Algorithms

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

#### 4.4.5. Load Rebalancing Example

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

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



Where: -> small flow  
==> large flow

Figure 4: Evenly Utilized Composite Links

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

## 5. Information Model for Flow Rebalancing

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

### 5.1. Configuration Parameters for Flow Rebalancing

The following parameters are required the configuration of this feature:

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

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

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

## 5.2. System Configuration and Identification Parameters

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

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

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



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

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

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

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

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

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

### 5.4. Information for Redistribution of Small Flows

Redistribution of small flows is done using the following:

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

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

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

### 5.5. Export of Flow Information

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

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

### 5.6. Monitoring information

#### 5.6.1. Interface (link) utilization

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

The link utilization can then be computed as follows:

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

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

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

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

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

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

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

#### 5.6.2. Other monitoring information

Additional monitoring information that is useful includes:

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

### 6. Operational Considerations

#### 6.1. Rebalancing Frequency

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

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

#### 6.2. Handling Route Changes

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

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

### 6.3. Forwarding Resources

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

### 7. IANA Considerations

This memo includes no request to IANA.

### 8. Security Considerations

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

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

### 9. Contributing Authors

Sanjay Khanna  
Cisco Systems  
Email: sanjakha@gmail.com

## 10. Acknowledgements

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

## 11. References

### 11.1. Normative References

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

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

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

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

### 11.2. Informative References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

#### Authors' Addresses

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

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

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

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

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



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

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



OPSAWG  
Internet-Draft  
Intended status: 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.

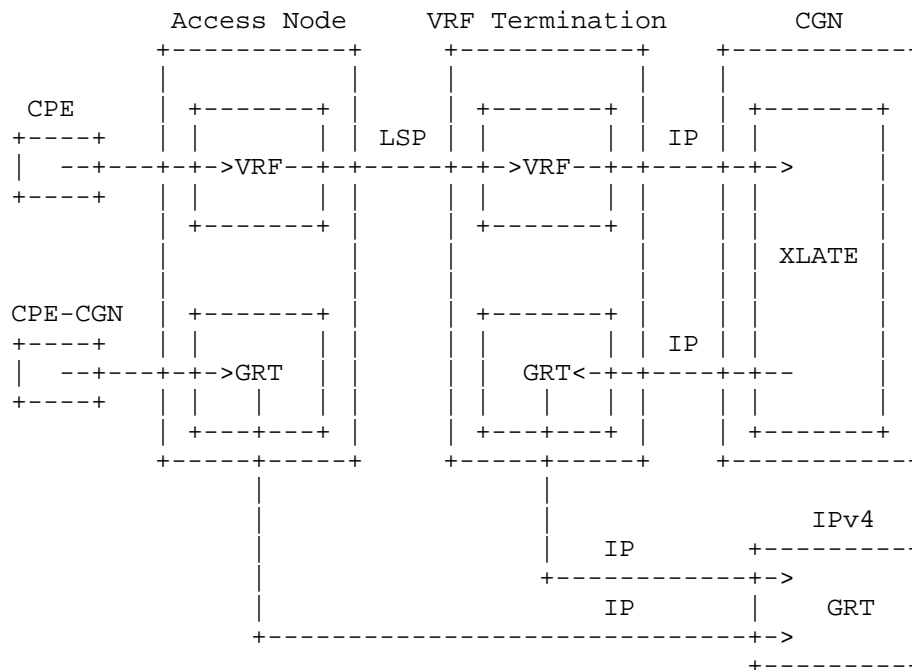


Figure 1: Basic BGP/MPLS IP VPN CGN Model

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.

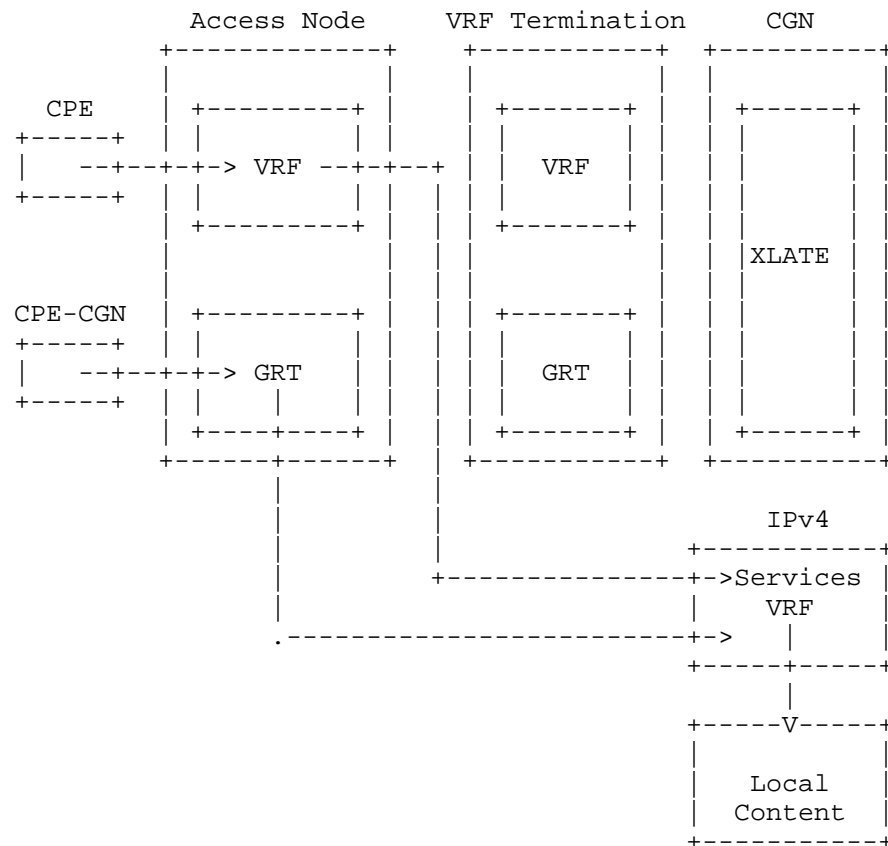
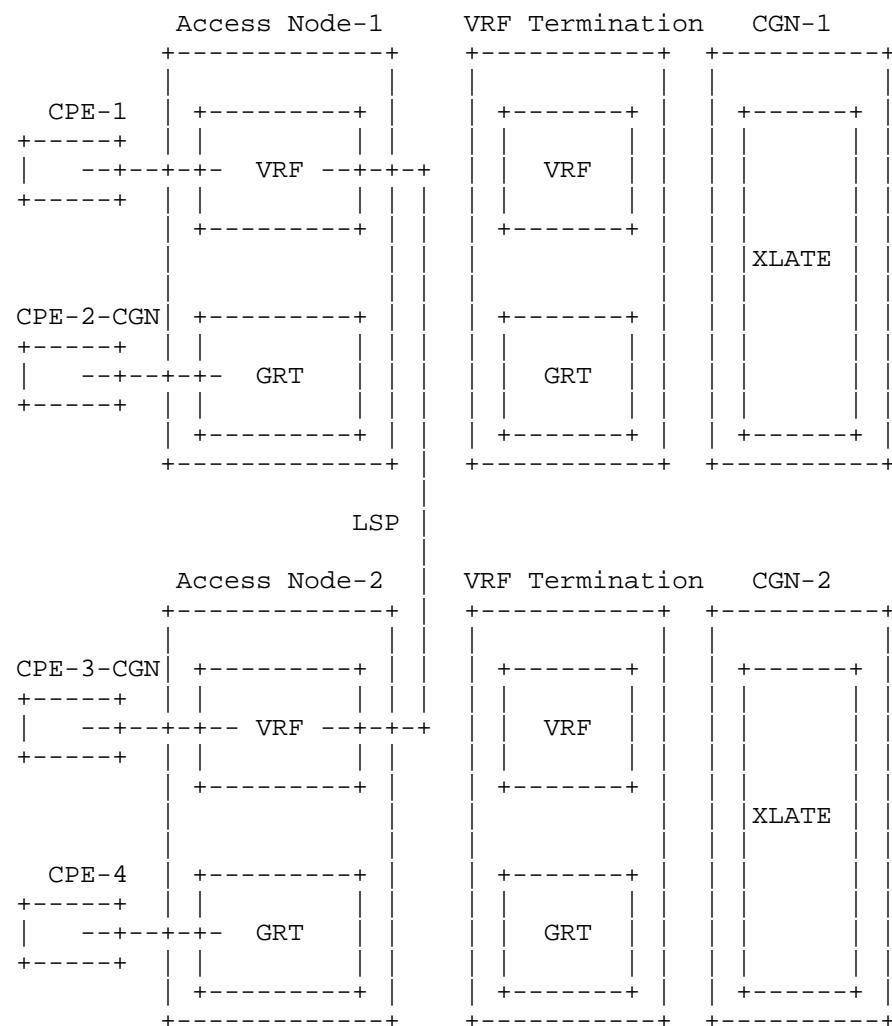


Figure 2: Internal Services and CGN By-Pass

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.

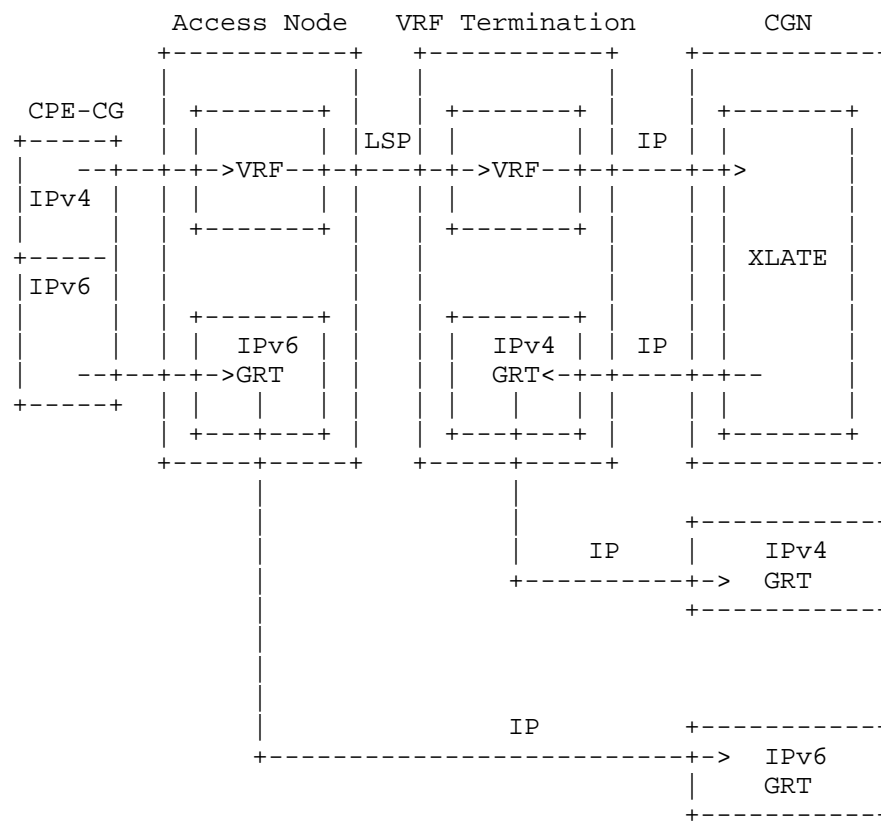


Figure 4: CGN with IPv6 Dual Stack Operation



#### 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>

Internet Engineering Task Force  
Internet-Draft  
Intended status: Standards Track  
Expires: December 07, 2013

J. Schoenwaelder  
Jacobs University Bremen  
T. Tsou  
C. Zhou  
T. Taylor  
Huawei Technologies  
June 05, 2013

Dynamic Host Configuration Protocol (DHCPv4 and DHCPv6) Options for  
Network Management Protocols  
draft-schoenw-opsawg-nm-dhc-04

## Abstract

This document defines new Dynamic Host Configuration Protocol (DHCPv4 and DHCPv6) options providing lists of IP addresses that can be used to locate network management services, specifically for the collection of logs and of Simple Network Management Protocol (SNMP) notifications.

## 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 December 07, 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 . . . . .	2
2. DHC Options for SYSLOG . . . . .	3
2.1. SYSLOG Collector Address Option for DHCPv4 . . . . .	3
2.2. SYSLOG Collector Address Option for DHCPv6 . . . . .	4
3. DHC Options for SNMP . . . . .	5
3.1. SNMP Notification Receiver Address Option for DHCPv4 . . . . .	5
3.2. SNMP Notification Receiver Address Option for DHCPv6 . . . . .	6
4. Security Considerations . . . . .	6
5. IANA Considerations . . . . .	7
6. Acknowledgements . . . . .	7
7. References . . . . .	7
7.1. Normative References . . . . .	7
7.2. Informational References . . . . .	9
Appendix A. Relationship to the SNMP Configuration MIB Modules . . . . .	9
Authors' Addresses . . . . .	10

## 1. Introduction

New networks such as 3GPP Long Term Evolution (LTE) are being deployed today with tens of thousands of network nodes. All of these nodes have to be configured and monitored for the network to operate correctly. Any steps to automate this process will be helpful to reduce the cost of deployment.

The Dynamic Host Configuration Protocol (DHCPv4 [RFC2131] and DHCPv6 [RFC3315]) is a relevant tool for this purpose. It provides a number of existing options to allow a node to acquire its configuration file and to locate key servers in the network. However, the existing options have been defined more with end hosts than with network nodes in mind. Network nodes require active management, and therefore need to acquire the addresses of their management servers.

This document is specifically focussed on the requirement for event reporting. To that end, it defines new DHCP options capable of listing:

- o one or more addresses of SYSLOG [RFC5424] collectors;
- o one or more addresses of SNMP [RFC3410] entities hosting Notification Receiver applications.



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. DHC Options for SYSLOG

The SYSLOG protocol [RFC5424] supports several transport mappings. According to RFC 5424, implementations MUST support the TLS/TCP-based transport defined in [RFC5425] and they SHOULD also support the UDP-based transport defined in [RFC5426] for compatibility with traditional SYSLOG. An optional transport of SYSLOG messages over DTLS/DCCP and DTLS/UDP is defined in [RFC6012].

The DHC options described below provide a list of IPv4 or IPv6 addresses of SYSLOG collectors in order of preference. The client SHOULD use the addresses sequentially but may be configured to try secure and/or congestion aware transports before falling back to transports that are not congestion aware or insecure. As such, the client may prefer to select an address providing a secure congestion aware transport even if it is listed with lower preference.

### 2.1. SYSLOG Collector Address Option for DHCPv4

This section describes the SYSLOG IPv4 Address Option for DHCPv4. The SYSLOG IPv4 Address Option begins with an option code followed by a length octet. The value of the length octet does not include itself or the option code. The option layout is depicted below:

Code	Len	IPv4 Address 1				IPv4 Address 2		
TBD1	n	a1	a2	a3	a4	a1	a2	...

The code for the SYSLOG DHCPv4 option is [IANA: TBD1]. The minimum length of the option is 4 octets, and the length MUST always be a multiple of 4.

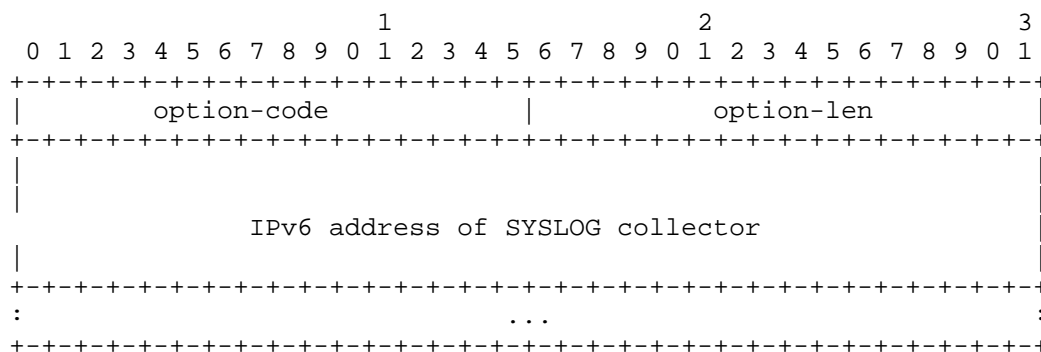
The option MUST NOT be specified by the DHCPv4 client, as it is intended only to be returned from the DHCPv4 server. If the DHCPv4 client wants to receive this information from the server, it needs to include the number [IANA: TBD1] in the "DHCP Parameter Request List" option (55).

Server addresses SHOULD be listed in order of preference, and the client SHOULD use the addresses sequentially but may be configured to use addresses in a different order according to some local policy (e.g., the client prefers secure and/or congestion aware transports as described above).

Historical note: DHCPv4 option 7 was originally defined (Section 3.9 of [RFC2132]) to provide the addresses of one or more log servers. However, the logging technology concerned was specified to be "MIT-LCS UDP log servers". It seems preferable to define a new option for SYSLOG collectors.

## 2.2. SYSLOG Collector Address Option for DHCPv6

This section describes the SYSLOG IPv6 Address Option for DHCPv6. The SYSLOG IPv6 Address Option begins with an option-code followed by the option-len. The value of the option-len does not include itself or the option-code. The option layout is depicted below:



The option-code of the SYSLOG DHCPv6 option `OPTION_SYSLOG_COLLECTOR` is [IANA: TBD2]. The minimum option-len is 16 octets, and the length MUST always be a multiple of 16.

The option MUST NOT appear in other than the following messages: Solicit, Advertise, Request, Renew, Rebind, Information-Request and Reply. The option number for these options MAY appear in the Option Request Option (6) in the following messages: Solicit, Request, Renew, Rebind, Information-Request and Reconfigure.

The addresses SHOULD be listed in order of preference, and the client SHOULD use the addresses sequentially but may be configured to use addresses in a different order according to some local policy (e.g., the client prefers secure and/or congestion aware transports as described above).

### 3. DHC Options for SNMP

The SNMP protocol [RFC3410] supports several transport mappings. The preferred IP-based transport is SNMP over UDP [RFC3417]. An experimental transport of SNMP over TCP is defined in [RFC3430]. An optional standards-track transport of SNMP over SSH is defined in [RFC5592] while optional standards-track transports over TLS and DTLS are defined in [RFC6353]

The DHC options described below provide a list of IPv4 or IPv6 addresses of SNMP entities hosting Notification Receiver applications in order of preference. The client SHOULD use the addresses sequentially but may be configured to try secure and/or congestion aware transports before falling back to transports that are not congestion aware or insecure. As such, the client may prefer to select an address providing a secure congestion aware transport even if it is listed with lower preference.

#### 3.1. SNMP Notification Receiver Address Option for DHCPv4

This section describes the SNMP IPv4 Address Option for DHCPv4. The SNMP IPv4 Address Option begins with an option code followed by a length octet. The value of the length octet does not include itself or the option code. The option layout is depicted below:

Code	Len	IPv4 Address 1						IPv4 Address 2	
TBD3	n	a1	a2	a3	a4	a1	a2	...	

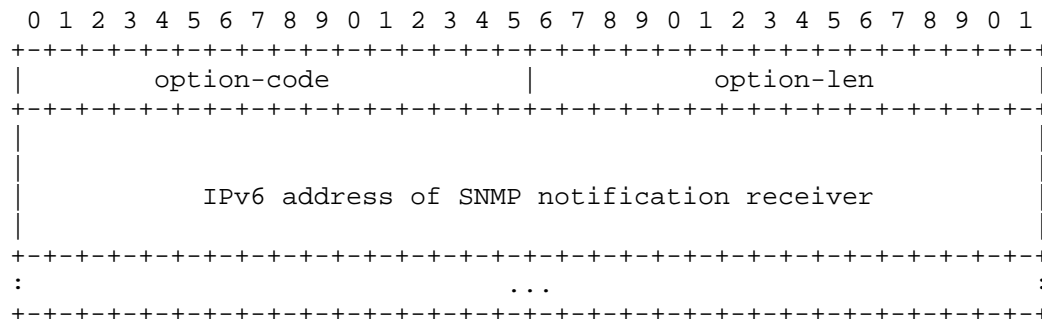
The code for the SNMP notification receiver DHCPv4 option is [IANA: TBD3]. The minimum length of the option is 4 octets, and the length MUST always be a multiple of 4.

The option MUST NOT be specified by the DHCPv4 client, as it is intended only to be returned from the DHCPv4 server. If the DHCPv4 client wants to receive this information from the server, it needs to include the number [IANA: TBD3] in the "DHCP Parameter Request List" option (55).

The addresses SHOULD be listed in order of preference, and the client SHOULD use the addresses sequentially but may be configured to use addresses in a different order according to some local policy (e.g., the client prefers secure and/or congestion aware transports as described above).

### 3.2. SNMP Notification Receiver Address Option for DHCPv6

This section describes the SNMP IPv6 Address Option for DHCPv6. The SNMP IPv6 Address Option begins with an option-code followed by the option-len. The value of the option-len does not include itself or the option-code. The option layout is depicted below:



The option-code of the SNMP notification receiver DHCPv6 option `OPTION_SNMP_NOT_RECEIVER` is [IANA: TBD4]. The minimum option-len is 16 octets, and the length MUST always be a multiple of 16.

The option MUST NOT appear in other than the following messages: Solicit, Advertise, Request, Renew, Rebind, Information-Request and Reply. The option number for these options MAY appear in the Option Request Option (6) in the following messages: Solicit, Request, Renew, Rebind, Information-Request and Reconfigure.

Server addresses SHOULD be listed in order of preference, and the client SHOULD use the addresses sequentially but may be configured to use addresses in a different order according to some local policy (e.g., the client prefers secure and/or congestion aware transports as described above).

## 4. Security Considerations

The security considerations in [RFC2131] and [RFC3315] apply. If an adversary manages to modify the response from a DHCPv4 or DHCPv6 server or insert its own response, a node could be led to contact a rogue network management server.

It is recommended to use the DHCPv4 authentication option described in [RFC3118] where available. This will also protect against denial-of-service attacks to DHCP servers. [RFC3118] provides mechanisms for both entity authentication and message authentication.

In IPv6 networks using DHCPv6, it is recommended that clients use authentication of DHCPv6 messages as described in Section 21 of [RFC3315].

In deployments where DHCPv4 or DHCPv6 authentication is not available, lower-layer security services may be sufficient to protect DHCPv4 and DHCPv6 messages.

## 5. IANA Considerations

IANA is requested to assign [IANA: TBD1] and [IANA: TBD3] as option codes in the "DHCP Option Codes" registry. The desired entries are shown in Table 1.

Tag	Name	Data Length	Meaning	Reference
TBD1	SYSLOG Collector Address	N	N/4 SYSLOG collector addresses	RFCxxxx
TBD3	SNMP Notification Receiver Address	N	N/4 SNMP Notification Receiver addresses	RFCxxxx

Table 1: DHCPv4 Option Codes For Network Management Servers

IANA is requested to assign [IANA: TBD2] as an option code from the "DHCPv6 Options Codes" registry for OPTION\_SYSLOG\_COLLECTOR, with reference RFCxxxx.

IANA is requested to assign [IANA: TBD4] as an option code from the "DHCPv6 Options Codes" registry for OPTION\_SNMP\_NOT\_RECEIVER, with reference RFCxxxx.

RFC Editor's Note: RFCxxxx denotes the present document.

## 6. Acknowledgements

The authors like to thank Ralf Droms, Ted Lemon and Bernie Volz for their helpful comments.

## 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.
- [RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.
- [RFC2132] Alexander, S. and R. Droms, "DHCP Options and BOOTP Vendor Extensions", RFC 2132, March 1997.
- [RFC3118] Droms, R. and W. Arbaugh, "Authentication for DHCP Messages", RFC 3118, June 2001.
- [RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.
- [RFC3413] Levi, D., Meyer, P., and B. Stewart, "Simple Network Management Protocol (SNMP) Applications", STD 62, RFC 3413, December 2002.
- [RFC3417] Presuhn, R., "Transport Mappings for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3417, December 2002.
- [RFC3430] Schoenwaelder, J., "Simple Network Management Protocol Over Transmission Control Protocol Transport Mapping", RFC 3430, December 2002.
- [RFC5424] Gerhards, R., "The Syslog Protocol", RFC 5424, March 2009.
- [RFC5425] Miao, F., Ma, Y., and J. Salowey, "Transport Layer Security (TLS) Transport Mapping for Syslog", RFC 5425, March 2009.
- [RFC5426] Okmianski, A., "Transmission of Syslog Messages over UDP", RFC 5426, March 2009.
- [RFC5592] Harrington, D., Salowey, J., and W. Hardaker, "Secure Shell Transport Model for the Simple Network Management Protocol (SNMP)", RFC 5592, June 2009.
- [RFC6012] Salowey, J., Petch, T., Gerhards, R., and H. Feng, "Datagram Transport Layer Security (DTLS) Transport Mapping for Syslog", RFC 6012, October 2010.
- [RFC6353] Hardaker, W., "Transport Layer Security (TLS) Transport Model for the Simple Network Management Protocol (SNMP)", RFC 6353, July 2011.

## 7.2. Informational 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. Relationship to the SNMP Configuration MIB Modules

The SNMP notification receiver address DHCPv4 and DHCPv6 options defined in Section 3.1 and Section 3.2 provide the basic information to setup a target in the SNMP-TARGET-MIB and the SNMP-NOTIFICATION-MIB [RFC3413]. After selecting the transport (e.g., by probing the availability of possible SNMP transport endpoints according to some local policy, a volatile entry in the `snmpTargetTable` can be created as follows (assuming xyz is some suitable unique handle for the received DHCP option):

<code>snmpTargetAddrName</code>	= "dhcp-xyz"	(INDEX)
<code>snmpTargetAddrTDomain</code>	= <code>snmpUDPDomain</code>	
<code>snmpTargetAddrTAddress</code>	= "a.b.c.d"	
<code>snmpTargetAddrTimeout</code>	= 1500	(DEFVAL)
<code>snmpTargetAddrRetryCount</code>	= 3	(DEFVAL)
<code>snmpTargetAddrTagList</code>	= "dhcp-xyz-tag"	
<code>snmpTargetAddrParams</code>	= "dhcp-xyz-param"	
<code>snmpTargetAddrStorageType</code>	= <code>volatile(2)</code>	
<code>snmpTargetAddrRowStatus</code>	= <code>active(1)</code>	

A matching volatile entry in the `snmpNotifyTable` can also be easily created:

<code>snmpNotifyName</code>	= "dhcp-xyz"	(INDEX)
<code>snmpNotifyTag</code>	= "dhcp-xyz-tag"	
<code>snmpNotifyType</code>	= <code>trap(1)</code>	(DEFVAL)
<code>snmpNotifyStorageType</code>	= <code>volatile(2)</code>	
<code>snmpNotifyRowStatus</code>	= <code>active(1)</code>	

In addition, an entry in the `snmpTargetParamsTable` is needed. Its structure for SNMPv3/USM user "joe" is as follows:

```
snmpTargetParamsName      = "dhcp-xyz-param"    (INDEX)
snmpTargetParamsMPModel   = 3                  (SNMPv3)
snmpTargetParamsSecurityModel = 3              (USM)
snmpTargetParamsSecurityName = "joe"
snmpTargetParamsSecurityLevel = authNoPriv(2)
snmpTargetParamsStorageType = volatile(2)
snmpTargetParamsRowStatus  = active(1)
```

Creation of a suitable entry in the `snmpTargetParamsTable` requires local information. Depending on the security model, additional information will be necessary.

The creation of a suitable `snmpTargetParamsTable` entry may either be dynamic (i.e., the entry is created upon receipt of a DHC lease using some local policy information and deleted when the DHC lease expires) or suitable `snmpTargetParamsTable` entries may be pre-provisioned based on the expected naming of the target entries that are created dynamically. Implementations may also pre-provision `snmpTargetAddrTable` entries and only dynamically create suitable `snmpNotifyTable` entries.

#### Authors' Addresses

Juergen Schoenwaelder  
Jacobs University Bremen  
Campus Ring 1  
Bremen 28759  
Germany

Email: [j.schoenwaelder@jacobs-university.de](mailto:j.schoenwaelder@jacobs-university.de)

Tina Tsou  
Huawei Technologies  
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

Tom Taylor  
Huawei Technologies  
Ottawa  
Canada

Email: tom.taylor.stds@gmail.com

Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: January 15, 2014

R. Zhang  
China Telecom  
Z. Cao  
H. Deng  
China Mobile  
R. Pazhyannur  
S. Gundavelli  
Cisco  
July 14, 2013

Separation of CAPWAP Control and Data Plane: Scenarios, Requirements and  
Solutions  
draft-zhang-opsawg-capwap-cds-00

Abstract

This document describes the scenarios and requirements of separating CAPWAP Data and Control plane. This specification provides a CAPWAP extension to allow two distinct AC component: AC-DP (AC-Data Plane) and AC-CP (AC-Control Plane). AC-DP handles all user payload with the exception of layer 2 management frames between the AC and user such as IEEE 802.11 association, authentication, probe, Action Frame. AC-CP handles all control messages between the WTP and AC. In addition, the AC-CP will handle user payload related to layer-2 management frames such as those mentioned above.

Status of This Memo

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

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

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

This Internet-Draft will expire on January 15, 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 . . . . .	2
1.1. Conventions used in this document . . . . .	3
1.2. Terminology . . . . .	3
2. Scenario and Analysis . . . . .	3
3. Analysis of Local Bridging Model . . . . .	5
4. Multiple CAPWAP Data Tunnels . . . . .	5
5. IANA Considerations . . . . .	6
6. Security Considerations . . . . .	6
7. Contributors . . . . .	6
8. References . . . . .	6
8.1. Normative References . . . . .	6
8.2. Informative References . . . . .	7
Authors' Addresses . . . . .	7

## 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 specificaitons mentioned above mainly design the different control message types used by the AC to control multiple WTPs. CAPWAP specifies that all user payload is transported on the CAPWAP-DATA channel. As an example, EAP messages, as key protocol exchange elements in the WLAN architecture also need to be encapsulated in the CAPWAP-DATA. The CAPWAP protocol does not specify how to encapsulate EAP message in its control plane. As a result, the protocol does not allow for splitting the CAPWAP control and data plane where control messages

There are multiple ways of meeting the above requirements. This document first analyzes the capability of current CAPWAP solutions

and proposes ways to working around the problem without changing existing specifications.

### 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 WTP 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 (WTP), The physical or network entity that contains an RF antenna and wireless Physical Layer (PHY) to transmit and receive station traffic for wireless access networks.

**CAPWAP Control 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 WTP, the AC may not scale to meet the requirements.

```

CAPWAP-CTL +-----+
++=====+   AC   |

```

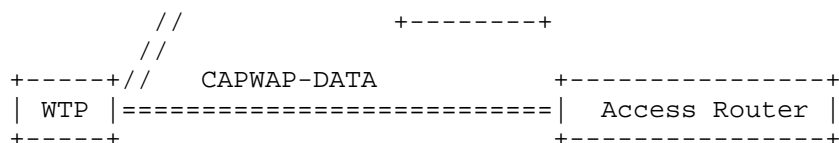


Figure 1: Split between CAPWAP-CTL and CAPWAP-DATA Plane

Because there are no explicit message types to support the encapsulation of EAP packets (and more generally layer 2 management frames) in the CAPWAP-CTL plane, the EAP messages are tunneled via the CAPWAP-DATA plane to the AR. AR would act as the authenticator in the EAP framework. After authentication, the AR receives the EAP keying message for the session. However, this mode of operation would undermine the main benefit of having the AC as the centralized entity for authentication and policy.

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.

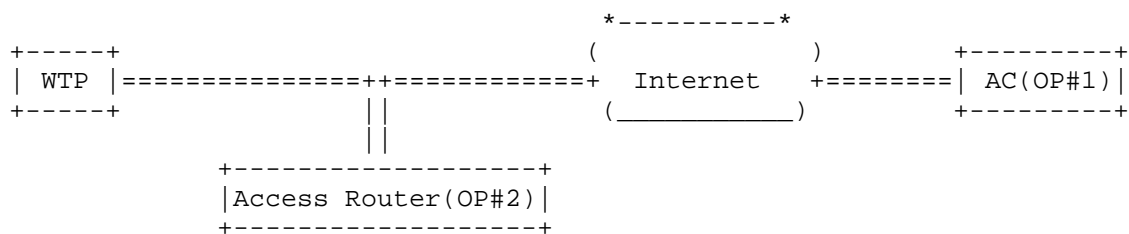


Figure 2: Access Service and Authentication Service Provided by different Operators

### 3. Analysis of Local Bridging Model

In the Local-MAC model defined in [RFC5416] Section 2.2.2, it says that:

"The WTP MAY locally bridge client data frames (and provide the necessary encryption and decryption services). The WTP MAY also tunnel client data frames to the AC, using 802.3 frame tunnel mode or 802.11 frame tunnel mode."

Some have rightly suggested that the Local-MAC model provides a way to separate Data and Control Plane. In this case where the WTP can locally bridge the user traffic (without any CAPWAP encapsulation). EAP and other management traffic can still be carried over the CAPWAP-DATA tunnel between the WTP and AC. The limitation of this behavior is two fold: This requires the Access Router (that will apply policy, etc) to be on the same Layer-2 network as the WTP. In many deployments, the traffic would need to be tunneled between the WTP and the Access Router that applies the policy. Second, without outer layer CAPWAP Data header, charging and controlling policies could not be applied to the data plane.

The Figure 3 shows this case where WTP encapsulates EAP messages into CAPWAP-DATA plane but locally bridges data frames.

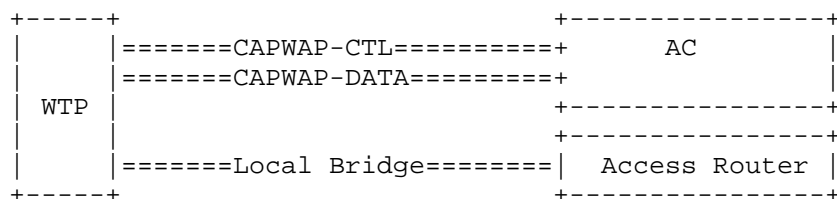


Figure 3: Local Bridging Model

### 4. Multiple CAPWAP Data Tunnels

A proposed solution is to create multiple CAPWAP-DATA tunnels. As shown in Figure 4, the WTP encapsulates all control messages between the WTP and AC in the CAPWAP-Control tunnel. In addition, all Layer 2 management frames (EAP, etc) are also transported in the CAPWAP-DATA tunnel between WTP and AC-CP. In addition, WTP encapsulates all non-management user payload into a secondary CAPWAP-DATA tunnel between WTP and AC-DP.

This brings up issues related to setting up of the secondary data tunnel, such as how does the WTP discover the IP address of AC-DP,

and what security credentials are used to setup the tunnel. We plan to address this in the next version of this draft.

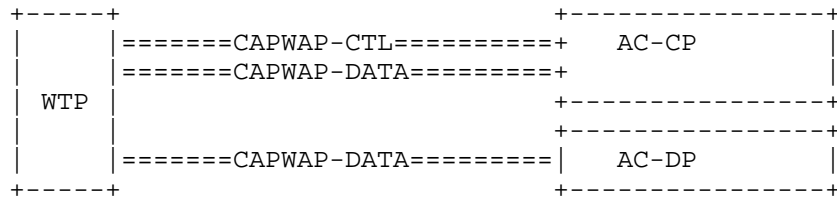


Figure 4: Multiple DATA tunnels Model

## 5. IANA Considerations

This document has no requests to the IANA.

## 6. Security Considerations

Security considerations for the CAPWAP protocol has been analyzed in Section 12 of [RFC5415]. This document does not introduce other security issues besides what has been analyzed in RFC5415.

## 7. Contributors

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

Thank Dorothy Stanley for reviewing the document and recommending ways to move forward with both technology and editorial parts of the document.

Thank all the contributors of this document.

## 8. References

### 8.1. Normative References

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

## 8.2. 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

Email: zhangr@gsta.com

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

Phone: +86-10-52686688

Email: zehn.cao@gmail.com, caozhen@chinamobile.com



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

Email: denghui@chinamobile.com

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

Email: rpazhyan@cisco.com

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

Email: sgundave@cisco.com