

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

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

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

Abstract

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

Status of this Memo

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

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

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

This Internet-Draft will expire on April 16, 2014.

## Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

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

## Table of Contents

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

## 1. Introduction

This document defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, this specifies objects for managing virtual machines controlled by a hypervisor (a.k.a. virtual machine monitor). A hypervisor controls multiple virtual machines on a single physical machine by allocating resources to each virtual machine using virtualization technologies. Therefore, this MIB module contains information on virtual machines and their resources controlled by a hypervisor as well as hypervisor's hardware and software information.

The design of this MIB module has been derived from enterprise specific MIB modules, namely a MIB module for managing guests of the Xen hypervisor, a MIB module for managing virtual machines controlled by the VMware hypervisor, and a MIB module using the libvirt programming interface to access different hypervisors. However, this MIB module attempts to generalize the managed objects to support other hypervisors.

### 1.1. Requirements Language

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

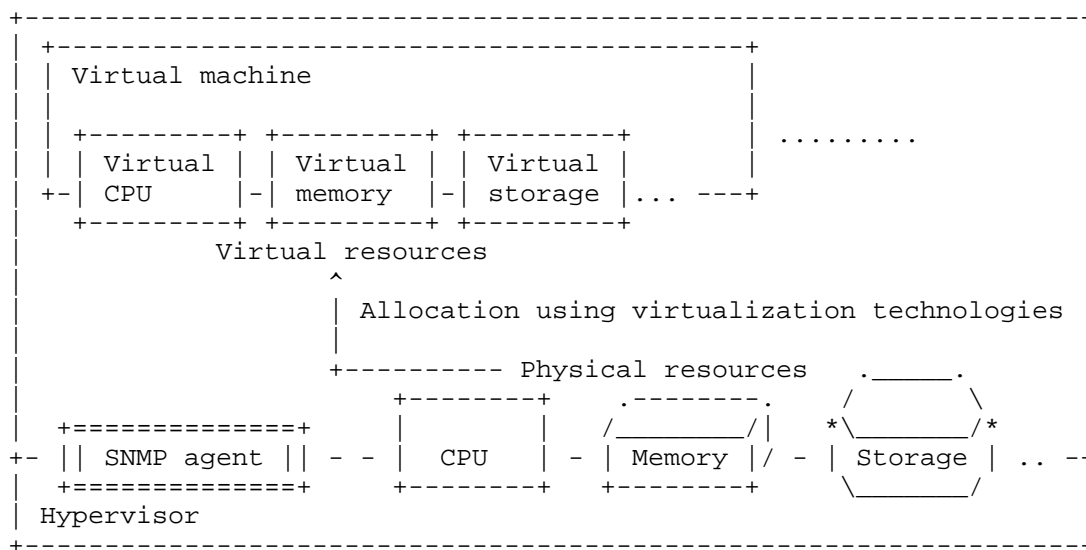
## 2. The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC 3410 [RFC3410]. Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This memo specifies a MIB module that is compliant to the SMIv2, which is described in STD 58, RFC 2578 [RFC2578], STD 58, RFC 2579 [RFC2579] and STD 58, RFC 2580 [RFC2580].

### 3. Managed Objects for Virtual Machines Controlled by a Hypervisor

#### 3.1. Managed Objects on Virtualization Environment

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



A hypervisor allocates virtual resources such as virtual CPUs, virtual memory, virtual storage devices, and virtual network interfaces to virtual machines from physical resources.

Figure 1: An example of a virtualization environment

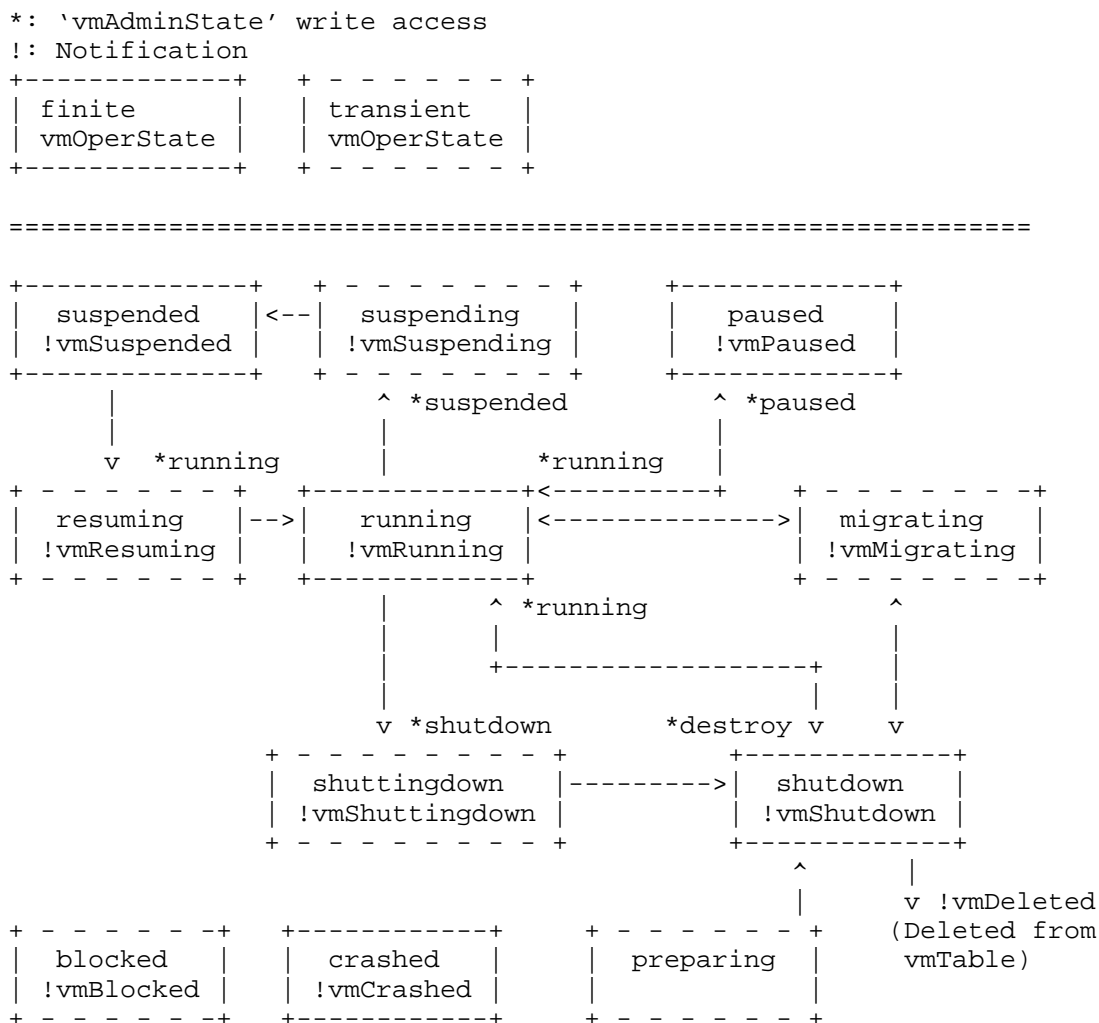
As shown in Figure 1, the objects defined in this document are managed at a hypervisor and an SNMP agent is launched at the hypervisor to provide access to the objects. The objects are managed from the viewpoint of the operators of hypervisors, but not the operators of virtual machines; i.e., the objects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

vmNetworIfIndex indicates the virtual interface associated with an interface of a virtual machine at the hypervisor, and consequently, the 'in' and 'out' directions denote 'from a virtual machine to the hypervisor' and 'from the hypervisor to a virtual machine', respectively. Moreover, vmStorageAllocatedSize denotes the size allocated by the hypervisor, but not the size actually used by the operating system on the virtual machine. This means that vmStorageDefinedSize and vmStorageAllocatedSize do not take different values when the vmStorageSourceType is 'block' or 'raw'.

The other objects related to virtual machines such as management IP addresses of a virtual machine are not included in this MIB module because this MIB module defines the objects common to general hypervisors but they are specific to some hypervisors. They may be included in the entLogicalTable of ENTITY-MIB [RFC4133]. The objects related to virtual switches are not also included in this MIB module though virtual switches shall be placed on a hypervisor. This is because the virtual network interfaces are the lowest abstraction of network resources allocated to a virtual machine. Instead of including the objects related to virtual switches, for example, BRIDGE-MIB [RFC4188] and Q-BRIDGE-MIB [RFC4363] could be used.

### 3.2. Overview of the MIB Module

The MIB module is organized into a group of scalars and tables. The scalars below 'hypervisor' provide basic information about the hypervisor. The 'vmTable' lists the virtual machines (guests) that are known to the hypervisor. The 'vmCpuTable' provides the mapping table of virtual CPUs to virtual machines, including CPU time used by each virtual CPU. The 'vmCpuAffinityTable' provides the affinity of each virtual CPU to a physical CPU. The 'vmStorageTable' provides the list of virtual storage devices and their mapping to virtual machines. In case that an entry in the 'vmStorageTable' has a corresponding parent physical storage device managed in 'hrStorageTable' of HOST-RESOURCES-MIB [RFC2790], the entry contains a pointer 'vmStorageParent' to the physical storage device. The 'vmNetworkTable' provides the list of virtual network interfaces and their mapping to virtual machines. Each entry in the 'vmNetworkTable' also provides a pointer 'vmNetworIfIndex' to the corresponding entry in the 'ifTable' of IF-MIB [RFC2863]. In case that an entry in the 'vmNetworkTable' has a corresponding parent physical network interface managed in 'ifTable' of IF-MIB, the entry contains a pointer 'vmNetworkParent' to the physical network interface.



The state transition of a virtual machine

Figure 2: State transition of a virtual machine

The 'vmAdminState' and 'vmOperState' textual conventions define an administrative state and an operational state model for virtual machines. Events causing transitions between major operational states will cause the generation of notifications. Per virtual machine (per-VM) notifications (vmRunning, vmShutdown, vmPaused, vmSuspended, vmCrashed, vmDeleted) are generated if vmPerVMNotificationsEnabled is true(1). Bulk notifications (vmBulkRunning, vmBulkShutdown, vmBulkPaused, vmBulkSuspended,

vmBulkCrashed, vmBulkDeleted) are generated if vmBulkNotificationsEnabled is true(1). The transition of 'vmOperState' by the write access to 'vmAdminState' and the notifications generated by the operational state changes are summarized in Figure 2. Note that the notifications shown in this figure are per-VM notifications. In the case of Bulk notifications, the prefix 'vm' is replaced with 'vmBulk'.

The bulk notification mechanism is designed to reduce the number of notifications that are trapped by an SNMP manager. This is because the number of virtual machines managed by a bunch of hypervisors in a datacenter possibly becomes several thousands or more, and consequently, many notifications could be trapped if these virtual machines frequently change their administrative state. The per-VM notifications carry more detailed information, but the scalability shall be a problem. An implementation shall support both, either of, or none of per-VM notifications and bulk notifications. The notification filtering mechanism described in section 6 of RFC 3413 [RFC3413] is used by the management applications to control the notifications.

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

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

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



```

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

```

```

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

```

### 3.3. Definitions

```
VM-MIB DEFINITIONS ::= BEGIN
```

```
IMPORTS
```

```
    MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE, TimeTicks,  
    Counter64, Integer32, mib-2  
        FROM SNMPv2-SMI  
    OBJECT-GROUP, MODULE-COMPLIANCE, NOTIFICATION-GROUP  
        FROM SNMPv2-CONF  
    TEXTUAL-CONVENTION, PhysAddress, TruthValue  
        FROM SNMPv2-TC  
    SnmpAdminString  
        FROM SNMP-FRAMEWORK-MIB  
    UUIDorZero  
        FROM UUID-TC-MIB  
    InterfaceIndexOrZero  
        FROM IF-MIB;
```

```
vmMIB MODULE-IDENTITY
```

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

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

```
        Hirochika Asai  
        The University of Tokyo  
        7-3-1 Hongo  
        Bunkyo-ku, Tokyo 113-8656  
        JP  
        Phone: +81 3 5841 6748  
        Email: panda@hongo.wide.ad.jp
```

```
        Michael MacFaden  
        VMware Inc.  
        Email: mrm@vmware.com
```

```
        Juergen Schoenwaelder  
        Jacobs University  
        Campus Ring 1  
        Bremen 28759  
        Germany  
        Email: j.schoenwaelder@jacobs-university.de
```

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

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

Keiichi Shima  
IIJ Innovation Institute Inc.  
3-13 Kanda-Nishikicho  
Chiyoda-ku, Tokyo 101-0054  
JP  
Email: keiichi@iijlab.net

Tina Tsou  
Huawei Technologies (USA)  
2330 Central Expressway  
Santa Clara CA 95050  
USA  
Email: tina.tsou.zouting@huawei.com

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

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

#### DESCRIPTION

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

Copyright (c) 2013 IETF Trust and the persons identified as authors of the code. All rights reserved.

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

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

vmNotifications OBJECT IDENTIFIER ::= { vmMIB 0 }
vmObjects        OBJECT IDENTIFIER ::= { vmMIB 1 }
vmConformance    OBJECT IDENTIFIER ::= { vmMIB 2 }

-- Textual conversion definitions
--
VirtualMachineIndex ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS      current
    DESCRIPTION
        "A unique value, greater than zero, identifying a
        virtual machine. The value for each virtual machine
        must remain constant at least from one re-initialization
        of the hypervisor to the next re-initialization."
    SYNTAX      Integer32 (1..2147483647)

VirtualMachineIndexOrZero ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS      current
    DESCRIPTION
        "This textual convention is an extension of the
        VirtualMachineIndex convention. This extension permits
        the additional value of zero. The meaning of the value
        zero is object-specific and must therefore be defined as
        part of the description of any object which uses this
        syntax. Examples of the usage of zero might include
        situations where a virtual machine is unknown, or when
        none or all virtual machines need to be referenced."
    SYNTAX      Integer32 (0..2147483647)

VirtualMachineAdminState ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "The administrative state of a virtual machine:

        running(1)    The administrative state of the virtual
                        machine indicating the virtual machine
                        is currently online or should be brought
                        online.
```

- suspended(2) The administrative state of the virtual machine where its memory and CPU execution state has been saved to persistent store and will be restored at next running(1).
- paused(3) The administrative state indicating the virtual machine is resident in memory but is no longer scheduled to execute by the hypervisor.
- shutdown(4) The administrative state of the virtual machine indicating the virtual machine is currently offline or should be taken shutting down.
- destroy(5) The administrative state of the virtual machine indicating the virtual machine should be forcibly shutdown. After the destroy operation, the administrative state should be automatically changed to shutdown(4)."

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

VirtualMachineOperState ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"The operational state of a virtual machine:

- unknown(1) The operational state of the virtual machine is unknown, e.g., because the implementation failed to obtain the state from the hypervisor.
- other(2) The operational state of the virtual machine indicating that an operational state is obtained from the hypervisor but it is not a state defined in this MIB module.
- preparing(3) The operational state of the virtual machine indicating the virtual machine is currently in the process of preparation,

e.g., allocating and initializing virtual storage after creating (defining) virtual machine.

- running(4)      The operational state of the virtual machine indicating the virtual machine is currently executed but it is not in the process of preparing(3), suspending(6), resuming(8), migrating(10), and shuttingdown(11).
- blocked(5)      The operational state of the virtual machine indicating the execution of the virtual machine is currently blocked, e.g., waiting for some action of the hypervisor to finish. This is a transient state from/to other states.
- suspending(6)   The operational state of the virtual machine indicating the virtual machine is currently in the process of suspending to save its memory and CPU execution state to persistent store. This is a transient state from running(4) to suspended(7).
- suspended(7)    The operational state of the virtual machine indicating the virtual machine is currently suspended, which means the memory and CPU execution state of the virtual machine are saved to persistent store. During this state, the virtual machine is not scheduled to execute by the hypervisor.
- resuming(8)     The operational state of the virtual machine indicating the virtual machine is currently in the process of resuming to restore its memory and CPU execution state from persistent store. This is a transient state from suspended(7) to running(4).
- paused(9)       The operational state of the virtual machine indicating the virtual machine is resident in memory but no longer scheduled to execute by the hypervisor.

migrating(10) The operational state of the virtual machine indicating the virtual machine is currently in the process of migration from/to another hypervisor.

shuttingdown(11)  
The operational state of the virtual machine indicating the virtual machine is currently in the process of shutting down. This is a transient state from running(4) to shutdown(12).

shutdown(12) The operational state of the virtual machine indicating the virtual machine is down, and CPU execution is no longer scheduled by the hypervisor and its memory is not resident in the hypervisor.

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

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

VirtualMachineAutoStart ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"The autostart configuration of a virtual machine:

unknown(1) The autostart configuration is unknown, e.g., because the implementation failed to obtain the autostart configuration from the hypervisor.

enable(2) The autostart configuration of the



virtual machine is enabled. The virtual machine should be automatically brought online at the next re-initialization of the hypervisor.

disable(3) The autostart configuration of the virtual machine is disabled. The virtual machine should not be automatically brought online at the next re-initialization of the hypervisor."

SYNTAX INTEGER {  
    unknown(1),  
    enable(2),  
    disable(3)  
}

VirtualMachinePersistent ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"This value indicates whether a virtual machine has a persistent configuration which means the virtual machine will still exist after shutting down:

unknown(1) The persistent configuration is unknown, e.g., because the implementation failed to obtain the persistent configuration from the hypervisor. (read-only)

persistent(2) The virtual machine is persistent, i.e., the virtual machine will exist after its shutting down.

transient(3) The virtual machine is transient, i.e., the virtual machine will not exist after its shutting down."

SYNTAX INTEGER {  
    unknown(1),  
    persistent(2),  
    transient(3)  
}

VirtualMachineCpuIndex ::= TEXTUAL-CONVENTION

DISPLAY-HINT "d"

STATUS current

DESCRIPTION

"A unique value for each virtual machine, greater than zero, identifying a virtual CPU assigned to a virtual machine. The value for each virtual CPU must remain

constant at least from one re-initialization of the  
hypervisor to the next re-initialization."

SYNTAX Integer32 (1..2147483647)

VirtualMachineStorageIndex ::= TEXTUAL-CONVENTION

DISPLAY-HINT "d"

STATUS current

DESCRIPTION

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

SYNTAX Integer32 (1..2147483647)

VirtualMachineStorageSourceType ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"The source type of a virtual storage device:

unknown(1) The source type is unknown, e.g., because  
the implementation failed to obtain the  
media type from the hypervisor.

other(2) The source type is other than those  
defined in this conversion.

block(3) The source type is a block device.

raw(4) The source type is a raw-formatted file.

sparse(5) The source type is a sparse file.

network(6) The source type is a network device."

SYNTAX INTEGER {  
    unknown(1),  
    other(2),  
    block(3),  
    raw(4),  
    sparse(5),  
    network(6)  
}

VirtualMachineStorageAccess ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"The access permission of a virtual storage:

```

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

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

VirtualMachineStorageMediaType ::= TEXTUAL-CONVENTION
STATUS      current
DESCRIPTION
    "The media type of a virtual storage device:

        unknown(1)      The media type is unknown, e.g., because
                        the implementation failed to obtain the
                        media type from the hypervisor.

        other(2)        The media type is other than those
                        defined in this conversion.

        hardDisk(3)     The media type is hard disk.

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

VirtualMachineNetworkIndex ::= TEXTUAL-CONVENTION
DISPLAY-HINT "d"
STATUS      current
DESCRIPTION
    "A unique value for each virtual machine, greater than
    zero, identifying a virtual network interface allocated
    to the virtual machine.  The value for each virtual
    network interface must remain constant at least from one
    re-initialization of the hypervisor to the next
    re-initialization."
SYNTAX      Integer32 (1..2147483647)

VirtualMachineList ::= TEXTUAL-CONVENTION
DISPLAY-HINT "lx"
STATUS      current
DESCRIPTION

```

"Each octet within this value specifies a set of eight virtual machine vmIndex, with the first octet specifying virtual machine 1 through 8, the second octet specifying virtual machine 9 through 16, etc. Within each octet, the most significant bit represents the lowest numbered vmIndex, and the least significant bit represents the highest numbered vmIndex. Thus, each virtual machine of the host is represented by a single bit within the value of this object. If that bit has a value of '1', then that virtual machine is included in the set of virtual machines; the virtual machine is not included if its bit has a value of '0'."

SYNTAX OCTET STRING

-- The hypervisor group

--

-- A collection of objects common to all hypervisors.

--

vmHypervisor OBJECT IDENTIFIER ::= { vmObjects 1 }

vmHvSoftware OBJECT-TYPE

SYNTAX SnmpAdminString (SIZE (0..255))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"A textual description of the hypervisor software. This value should not include its version, and it should be included in 'vmHvVersion'."

::= { vmHypervisor 1 }

vmHvVersion OBJECT-TYPE

SYNTAX SnmpAdminString (SIZE (0..255))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"A textual description of the version of the hypervisor software."

::= { vmHypervisor 2 }

vmHvObjectID OBJECT-TYPE

SYNTAX OBJECT IDENTIFIER

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The vendor's authoritative identification of the hypervisor software contained in the entity. This value is allocated within the SMI enterprises subtree (1.3.6.1.4.1). Note that this is different from

```

        sysObjectID in the SNMPv2-MIB [RFC3418] because
        sysObjectID is not the identification of the hypervisor
        software but the device, firmware, or management
        operating system."
 ::= { vmHypervisor 3 }

vmHvUpTime OBJECT-TYPE
    SYNTAX      TimeTicks
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The time (in centi-seconds) since the hypervisor was
        last re-initialized. Note that this is different from
        sysUpTime in the SNMPv2-MIB [RFC3418] and hrSystemUptime
        in the HOST-RESOURCES-MIB [RFC2790] because sysUpTime is
        the uptime of the network management portion of the
        system, and hrSystemUptime is the uptime of the
        management operating system but not the hypervisor
        software."
 ::= { vmHypervisor 4 }

-- The virtual machine information
--
-- A collection of objects common to all virtual machines.
--
vmNumber OBJECT-TYPE
    SYNTAX      Integer32 (0..2147483647)
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of virtual machines (regardless of their
        current state) present on this hypervisor."
 ::= { vmObjects 2 }

vmTableLastChange OBJECT-TYPE
    SYNTAX      TimeTicks
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The value of vmHvUpTime at the time of the last creation
        or deletion of an entry in the vmTable."
 ::= { vmObjects 3 }

vmTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF VmEntry
    MAX-ACCESS   not-accessible
    STATUS       current
```

## DESCRIPTION

"A list of virtual machine entries. The number of entries is given by the value of vmNumber."

::= { vmObjects 4 }

## vmEntry OBJECT-TYPE

SYNTAX VmEntry  
MAX-ACCESS not-accessible  
STATUS current

## DESCRIPTION

"An entry containing management information applicable to a particular virtual machine."

INDEX { vmIndex }

::= { vmTable 1 }

## VmEntry ::=

```
SEQUENCE {
    vmIndex          VirtualMachineIndex,
    vmName           SnmpAdminString,
    vmUUID           UUIDorZero,
    vmOSType         SnmpAdminString,
    vmAdminState     VirtualMachineAdminState,
    vmOperState      VirtualMachineOperState,
    vmAutoStart      VirtualMachineAutoStart,
    vmPersistent     VirtualMachinePersistent,
    vmCurCpuNumber  Integer32,
    vmMinCpuNumber   Integer32,
    vmMaxCpuNumber   Integer32,
    vmMemUnit        Integer32,
    vmCurMem        Integer32,
    vmMinMem         Integer32,
    vmMaxMem         Integer32,
    vmUpTime         TimeTicks,
    vmCpuTime        Counter64
}
```

## vmIndex OBJECT-TYPE

SYNTAX VirtualMachineIndex  
MAX-ACCESS not-accessible  
STATUS current

## DESCRIPTION

"A unique value, greater than zero, identifying the virtual machine. The value assigned to a given virtual machine may not persist across re-initialization of the hypervisor. A command generator must use the vmUUID to identify a given virtual machine of interest."

::= { vmEntry 1 }

## vmName OBJECT-TYPE

SYNTAX SnmpAdminString (SIZE (0..255))  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
"A textual name of the virtual machine."  
 ::= { vmEntry 2 }

## vmUUID OBJECT-TYPE

SYNTAX UUIDorZero  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
"The virtual machine's 128-bit UUID or the zero-length string when a UUID is not available. The UUID if set must uniquely identify a virtual machine from all other virtual machines in an administrative region. A zero-length octet string is returned if no UUID information is known."  
 ::= { vmEntry 3 }

## vmOSType OBJECT-TYPE

SYNTAX SnmpAdminString (SIZE (0..255))  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
"A textual description containing operating system information installed on the virtual machine. This value corresponds to the operating system the hypervisor assumes to be running when the virtual machine is started. This may differ from the actual operating system in case the virtual machine boots into a different operating system."  
 ::= { vmEntry 4 }

## vmAdminState OBJECT-TYPE

SYNTAX VirtualMachineAdminState  
MAX-ACCESS read-write  
STATUS current  
DESCRIPTION  
"The administrative power state of the virtual machine. Note that a virtual machine is supposed to be resumed when vmAdminState of the virtual machine is changed from suspended(2) or paused(3) to running(1)."  
 ::= { vmEntry 5 }

## vmOperState OBJECT-TYPE

SYNTAX VirtualMachineOperState

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

vmAutoStart OBJECT-TYPE
SYNTAX          VirtualMachineAutoStart
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "The autostart configuration of the virtual machine.  If
     this value is enable(2), the virtual machine
     automatically starts at the next initialization of the
     hypervisor."
 ::= { vmEntry 7 }

vmPersistent OBJECT-TYPE
SYNTAX          VirtualMachinePersistent
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "This value indicates whether the virtual machine has a
     persistent configuration which means the virtual machine
     will still exist after its shutdown."
 ::= { vmEntry 8 }

vmCurCpuNumber OBJECT-TYPE
SYNTAX          Integer32 (0..2147483647)
MAX-ACCESS      read-write
STATUS          current
DESCRIPTION
    "The number of virtual CPUs currently assigned to the
     virtual machine.  Changes to this object MUST NOT
     persist across re-initialization of the hypervisor."
 ::= { vmEntry 9 }

vmMinCpuNumber OBJECT-TYPE
SYNTAX          Integer32 (-1|0..2147483647)
MAX-ACCESS      read-write
STATUS          current
DESCRIPTION
    "The minimum number of virtual CPUs that are assigned to
     the virtual machine when it is in a power-on state.  The
     value -1 indicates that there is no hard boundary for
     the minimum number of virtual CPUs.  Changes to this
     object MUST NOT persist across re-initialization of the
     hypervisor."
```



```
::= { vmEntry 10 }

vmMaxCpuNumber OBJECT-TYPE
    SYNTAX      Integer32 (-1|0..2147483647)
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "The maximum number of virtual CPUs that are assigned to
        the virtual machine when it is in a power-on state. The
        value -1 indicates that there is no limit. Changes to
        this object MUST NOT persist across re-initialization of
        the hypervisor."
    ::= { vmEntry 11 }

vmMemUnit OBJECT-TYPE
    SYNTAX      Integer32 (1..2147483647)
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "The multiplication unit for vmCurMem, vmMinMem, and
        vmMaxMem. For example, when this value is 1024, the
        memory size unit for vmCurMem, vmMinMem, and vmMaxMem is
        KiB."
    ::= { vmEntry 12 }

vmCurMem OBJECT-TYPE
    SYNTAX      Integer32 (0..2147483647)
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "The current memory size currently allocated to the
        virtual memory module in the unit designated by
        vmMemUnit. Changes to this object MUST NOT persist
        across re-initialization of the hypervisor."
    ::= { vmEntry 13 }

vmMinMem OBJECT-TYPE
    SYNTAX      Integer32 (-1|0..2147483647)
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "The minimum memory size defined to the virtual machine
        in the unit designated by vmMemUnit. The value -1
        indicates that there is no hard boundary for the minimum
        memory size. Changes to this object MUST NOT persist
        across re-initialization of the hypervisor."
    ::= { vmEntry 14 }
```

## vmMaxMem OBJECT-TYPE

SYNTAX Integer32 (-1|0..2147483647)  
 MAX-ACCESS read-write  
 STATUS current  
 DESCRIPTION  
 "The maximum memory size defined to the virtual machine  
 in the unit designated by vmMemUnit. The value -1  
 indicates that there is no limit. Changes to this  
 object MUST NOT persist across re-initialization of the  
 hypervisor."  
 ::= { vmEntry 15 }

## vmUpTime OBJECT-TYPE

SYNTAX TimeTicks  
 MAX-ACCESS read-only  
 STATUS current  
 DESCRIPTION  
 "The time (in centi-seconds) since the administrative  
 state of the virtual machine was last changed from  
 shutdown(4) to running(1)."  
 ::= { vmEntry 16 }

## vmCpuTime OBJECT-TYPE

SYNTAX Counter64  
 UNITS "microsecond"  
 MAX-ACCESS read-only  
 STATUS current  
 DESCRIPTION  
 "The total CPU time used in microsecond. If the number  
 of virtual CPUs is larger than 1, vmCpuTime may exceed  
 real time.  
  
 Discontinuities in the value of this counter can occur  
 at re-initialization of the hypervisor, and  
 administrative state (vmAdminState) changes of the  
 virtual machine."  
 ::= { vmEntry 17 }

-- The virtual CPU on each virtual machines

## vmCpuTable OBJECT-TYPE

SYNTAX SEQUENCE OF VmCpuEntry  
 MAX-ACCESS not-accessible  
 STATUS current  
 DESCRIPTION  
 "The table of virtual CPUs provided by the hypervisor."  
 ::= { vmObjects 5 }

```

vmCpuEntry OBJECT-TYPE
    SYNTAX      VmCpuEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "An entry for one virtual processor assigned to a
        virtual machine."
    INDEX { vmIndex, vmCpuIndex }
    ::= { vmCpuTable 1 }

VmCpuEntry ::=
    SEQUENCE {
        vmCpuIndex          VirtualMachineCpuIndex,
        vmCpuCoreTime       Counter64
    }

vmCpuIndex OBJECT-TYPE
    SYNTAX      VirtualMachineCpuIndex
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "A unique value identifying a virtual CPU assigned to
        the virtual machine."
    ::= { vmCpuEntry 1 }

vmCpuCoreTime OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "microsecond"
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "The total CPU time used by this virtual CPU in
        microsecond.

        Discontinuities in the value of this counter can occur
        at re-initialization of the hypervisor, and
        administrative state (vmAdminState) changes of the
        virtual machine."
    ::= { vmCpuEntry 2 }

-- The virtual CPU affinity on each virtual machines
vmCpuAffinityTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF VmCpuAffinityEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "A list of CPU affinity entries of a virtual CPU."
    ::= { vmObjects 6 }

```

```

vmCpuAffinityEntry OBJECT-TYPE
    SYNTAX      VmCpuAffinityEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "An entry containing CPU affinity associated with a
        particular virtual machine."
    INDEX       { vmIndex, vmCpuIndex, vmCpuPhysIndex }
    ::= { vmCpuAffinityTable 1 }

VmCpuAffinityEntry ::=
    SEQUENCE {
        vmCpuPhysIndex      Integer32,
        vmCpuAffinity       Integer32
    }

vmCpuPhysIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..2147483647)
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "A value identifying a physical CPU on the hypervisor.
        On systems implementing the HOST-RESOURCES-MIB, the
        value must be the same value that is used as the index
        in the hrProcessorTable (hrDeviceIndex)."
    ::= { vmCpuAffinityEntry 2 }

vmCpuAffinity OBJECT-TYPE
    SYNTAX      INTEGER {
                    unknown(0),    -- unknown
                    enable(1),     -- enabled
                    disable(2)     -- disabled
                }
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "The CPU affinity of this virtual CPU to the physical
        CPU represented by 'vmCpuPhysIndex'."
    ::= { vmCpuAffinityEntry 3 }

-- The virtual storage devices on each virtual machine. This
-- document defines some overlapped objects with hrStorage in
-- HOST-RESOURCES-MIB [RFC2790], because virtual resources shall be
-- allocated from the hypervisor's resources, which is the 'host
-- resources'
vmStorageTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF VmStorageEntry

```

```

MAX-ACCESS    not-accessible
STATUS        current
DESCRIPTION
    "The conceptual table of virtual storage devices
    attached to the virtual machine."
 ::= { vmObjects 7 }

```

```

vmStorageEntry OBJECT-TYPE
    SYNTAX      VmStorageEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry for one virtual storage device attached to the
        virtual machine."
    INDEX { vmStorageVmIndex, vmStorageIndex }
    ::= { vmStorageTable 1 }

```

```

VmStorageEntry ::=
    SEQUENCE {
        vmStorageVmIndex      VirtualMachineIndexOrZero,
        vmStorageIndex        VirtualMachineStorageIndex,
        vmStorageParent        Integer32,
        vmStorageSourceType    VirtualMachineStorageSourceType,
        vmStorageSourceTypeString
                               SnmpAdminString,
        vmStorageResourceID    SnmpAdminString,
        vmStorageAccess        VirtualMachineStorageAccess,
        vmStorageMediaType     VirtualMachineStorageMediaType,
        vmStorageMediaTypeString
                               SnmpAdminString,
        vmStorageSizeUnit      Integer32,
        vmStorageDefinedSize    Integer32,
        vmStorageAllocatedSize Integer32,
        vmStorageReadIOs        Counter64,
        vmStorageWriteIOs       Counter64
    }

```

```

vmStorageVmIndex OBJECT-TYPE
    SYNTAX      VirtualMachineIndexOrZero
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This value identifies the virtual machine (guest) this
        storage device has been allocated to. The value zero
        indicates that the storage device is currently not
        allocated to any virtual machines."
    ::= { vmStorageEntry 1 }

```

vmStorageIndex OBJECT-TYPE  
SYNTAX VirtualMachineStorageIndex  
MAX-ACCESS not-accessible  
STATUS current  
DESCRIPTION  
    "A unique value identifying a virtual storage device  
    allocated to the virtual machine."  
 ::= { vmStorageEntry 2 }

vmStorageParent OBJECT-TYPE  
SYNTAX Integer32 (0..2147483647)  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
    "The value of hrStorageIndex which is the parent (i.e.,  
    physical) device of this virtual device on systems  
    implementing the HOST-RESOURCES-MIB. The value zero  
    denotes this virtual device is not any child represented  
    in the hrStorageTable."  
 ::= { vmStorageEntry 3 }

vmStorageSourceType OBJECT-TYPE  
SYNTAX VirtualMachineStorageSourceType  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
    "The source type of the virtual storage device."  
 ::= { vmStorageEntry 4 }

vmStorageSourceTypeString OBJECT-TYPE  
SYNTAX SnmpAdminString (SIZE (0..255))  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
    "A (detailed) textual string of the source type of the  
    virtual storage device. For example, this represents  
    the specific format name of the sparse file."  
 ::= { vmStorageEntry 5 }

vmStorageResourceID OBJECT-TYPE  
SYNTAX SnmpAdminString (SIZE (0..255))  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
    "A textual string that represents the resource  
    identifier of the virtual storage. For example, this  
    contains the path to the disk image file that  
    corresponds to the virtual storage."

```
 ::= { vmStorageEntry 6 }

vmStorageAccess OBJECT-TYPE
    SYNTAX      VirtualMachineStorageAccess
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The access permission of the virtual storage device."
    ::= { vmStorageEntry 7 }

vmStorageMediaType OBJECT-TYPE
    SYNTAX      VirtualMachineStorageMediaType
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The media type of the virtual storage device."
    ::= { vmStorageEntry 8 }

vmStorageMediaTypeString OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE (0..255))
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "A (detailed) textual string of the virtual storage
        media.  For example, this represents the specific driver
        name of the emulated media such as 'IDE' and 'SCSI'."
    ::= { vmStorageEntry 9 }

vmStorageSizeUnit OBJECT-TYPE
    SYNTAX      Integer32 (1..2147483647)
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The multiplication unit for vmStorageDefinedSize and
        vmStorageAllocatedSize.  For example, when this value is
        1048576, the storage size unit for vmStorageDefinedSize
        and vmStorageAllocatedSize is MiB."
    ::= { vmStorageEntry 10 }

vmStorageDefinedSize OBJECT-TYPE
    SYNTAX      Integer32 (-1|0..2147483647)
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The defined virtual storage size defined in the unit
        designated by vmStorageSizeUnit.  If this information is
        not available, this value shall be -1."
    ::= { vmStorageEntry 11 }
```

```
vmStorageAllocatedSize OBJECT-TYPE
    SYNTAX      Integer32 (-1|0..2147483647)
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The storage size allocated to the virtual storage from
        a physical storage in the unit designated by
        vmStorageSizeUnit.  When the virtual storage is block
        device or raw file, this value and vmStorageDefinedSize
        are supposed to equal.  This value MUST NOT be different
        from vmStorageDefinedSize when vmStorageSourceType is
        'block' or 'raw'.  If this information is not available,
        this value shall be -1."
    ::= { vmStorageEntry 12 }

vmStorageReadIOs OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of read I/O requests.

        Discontinuities in the value of this counter can occur
        at re-initialization of the hypervisor, and
        administrative state (vmAdminState) changes of the
        virtual machine."
    ::= { vmStorageEntry 13 }

vmStorageWriteIOs OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of write I/O requests.

        Discontinuities in the value of this counter can occur
        at re-initialization of the hypervisor, and
        administrative state (vmAdminState) changes of the
        virtual machine."
    ::= { vmStorageEntry 14 }

-- The virtual network interfaces on each virtual machine.
vmNetworkTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF VmNetworkEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "The conceptual table of virtual network interfaces
```



```

        attached to the virtual machine."
 ::= { vmObjects 8 }

vmNetworkEntry OBJECT-TYPE
    SYNTAX      VmNetworkEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "An entry for one virtual network interfaces attached to
        the virtual machine."
    INDEX { vmIndex, vmNetworkIndex }
    ::= { vmNetworkTable 1 }

VmNetworkEntry ::=
    SEQUENCE {
        vmNetworkIndex      VirtualMachineNetworkIndex,
        vmNetworkIfIndex    InterfaceIndexOrZero,
        vmNetworkParent     InterfaceIndexOrZero,
        vmNetworkModel      SnmpAdminString,
        vmNetworkPhysAddress PhysAddress
    }

vmNetworkIndex OBJECT-TYPE
    SYNTAX      VirtualMachineNetworkIndex
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "A unique value identifying a virtual network interface
        allocated to the virtual machine."
    ::= { vmNetworkEntry 1 }

vmNetworkIfIndex OBJECT-TYPE
    SYNTAX      InterfaceIndexOrZero
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "The value of ifIndex which corresponds to this virtual
        network interface.  If this device is not represented in
        the ifTable, then this value shall be zero."
    ::= { vmNetworkEntry 2 }

vmNetworkParent OBJECT-TYPE
    SYNTAX      InterfaceIndexOrZero
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "The value of ifIndex which corresponds to the parent
        (i.e., physical) device of this virtual device on.  The

```

```
        value zero denotes this virtual device is not any child
        represented in the ifTable."
 ::= { vmNetworkEntry 3 }

vmNetworkModel OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE (0..255))
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "A textual string containing the (emulated) model of
        virtual network interface. For example, this value is
        'virtio' when the emulation driver model is virtio."
 ::= { vmNetworkEntry 4 }

vmNetworkPhysAddress OBJECT-TYPE
    SYNTAX      PhysAddress
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The MAC address of the virtual network interface."
 ::= { vmNetworkEntry 5 }

-- Notification definitions:

vmPerVMNotificationsEnabled OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
        "Indicates if notification generator will send
        notifications per virtual machine."
 ::= { vmObjects 9 }

vmBulkNotificationsEnabled OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
        "Indicates if notification generator will send
        notifications per set of virtual machines."
 ::= { vmObjects 10 }

vmAffectedVMs OBJECT-TYPE
    SYNTAX      VirtualMachineList
    MAX-ACCESS   accessible-for-notify
    STATUS       current
    DESCRIPTION
```

```

        "A complete list of virtual machines whose state has
        changed. This object is the only object sent with bulk
        notifications."
 ::= { vmObjects 11 }

vmRunning NOTIFICATION-TYPE
OBJECTS      {
                vmName,
                vmUUID,
                vmOperState
            }
STATUS      current
DESCRIPTION
    "This notification is generated when the operational
    state of a virtual machine has been changed to
    running(4) from some other state. The other state is
    indicated by the included value of vmOperState."
 ::= { vmNotifications 1 }

vmShutdown NOTIFICATION-TYPE
OBJECTS      {
                vmName,
                vmUUID,
                vmOperState
            }
STATUS      current
DESCRIPTION
    "This notification is generated when the operational
    state of a virtual machine has been changed to
    shutdown(12) from some other state. The other state is
    indicated by the included value of vmOperState."
 ::= { vmNotifications 2 }

vmShuttingdown NOTIFICATION-TYPE
OBJECTS      {
                vmName,
                vmUUID,
                vmOperState
            }
STATUS      current
DESCRIPTION
    "This notification is generated when the operational
    state of a virtual machine has been changed to
    shuttingdown(11) from some other state. The other state
    is indicated by the included value of vmOperState."
 ::= { vmNotifications 3 }

vmPaused NOTIFICATION-TYPE
```

```
OBJECTS      {
                vmName,
                vmUUID,
                vmOperState
            }
STATUS      current
DESCRIPTION
    "This notification is generated when the operational
    state of a virtual machine has been changed to
    paused(9) from some other state.  The other state is
    indicated by the included value of vmOperState."
 ::= { vmNotifications 4 }

vmSuspending NOTIFICATION-TYPE
OBJECTS      {
                vmName,
                vmUUID,
                vmOperState
            }
STATUS      current
DESCRIPTION
    "This notification is generated when the operational
    state of a virtual machine has been changed to
    suspending(6) from some other state.  The other state is
    indicated by the included value of vmOperState."
 ::= { vmNotifications 5 }

vmSuspended NOTIFICATION-TYPE
OBJECTS      {
                vmName,
                vmUUID,
                vmOperState
            }
STATUS      current
DESCRIPTION
    "This notification is generated when the operational
    state of a virtual machine has been changed to
    suspended(7) from some other state.  The other state is
    indicated by the included value of vmOperState."
 ::= { vmNotifications 6 }

vmResuming NOTIFICATION-TYPE
OBJECTS      {
                vmName,
                vmUUID,
                vmOperState
            }
STATUS      current
```

```
DESCRIPTION
    "This notification is generated when the operational
    state of a virtual machine has been changed to
    resuming(8) from some other state. The other state is
    indicated by the included value of vmOperState."
 ::= { vmNotifications 7 }

vmMigrating NOTIFICATION-TYPE
OBJECTS
    {
        vmName,
        vmUUID,
        vmOperState
    }
STATUS
    current
DESCRIPTION
    "This notification is generated when the operational
    state of a virtual machine has been changed to
    migrating(10) from some other state. The other state is
    indicated by the included value of vmOperState."
 ::= { vmNotifications 8 }

vmCrashed NOTIFICATION-TYPE
OBJECTS
    {
        vmName,
        vmUUID,
        vmOperState
    }
STATUS
    current
DESCRIPTION
    "This notification is generated when a virtual machine
    has been crashed. The previos state of the virtual
    machine is indicated by the included value of
    vmOperState."
 ::= { vmNotifications 9 }

vmBlocked NOTIFICATION-TYPE
OBJECTS
    {
        vmName,
        vmUUID,
        vmOperState
    }
STATUS
    current
DESCRIPTION
    "This notification is generated when the operational
    state of a virtual machine has been changed to
    blocked(5). The previos state of the virtual machine is
    indicated by the included value of vmOperState."
 ::= { vmNotifications 10 }
```

```
vmDeleted NOTIFICATION-TYPE
  OBJECTS      {
                  vmName,
                  vmUUID,
                  vmOperState,
                  vmPersistent
                }
  STATUS        current
  DESCRIPTION   "This notification is generated when a virtual machine
                  has been deleted. The prior state of the virtual
                  machine is indicated by the included value of
                  vmOperState."
  ::= { vmNotifications 11 }

vmBulkRunning NOTIFICATION-TYPE
  OBJECTS      {
                  vmAffectedVMs
                }
  STATUS        current
  DESCRIPTION   "This notification is generated when the operational
                  state of one or more virtual machine has been changed to
                  running(4) from a all prior states except for
                  running(4). Management stations are encouraged to
                  subsequently poll the subset of virtual machines of
                  interest for vmOperState."
  ::= { vmNotifications 12 }

vmBulkShuttingdown NOTIFICATION-TYPE
  OBJECTS      {
                  vmAffectedVMs
                }
  STATUS        current
  DESCRIPTION   "This notification is generated when the operational
                  state of one or more virtual machine has been changed to
                  shuttingdown(11) from a state other than
                  shuttingdown(11). Management stations are encouraged to
                  subsequently poll the subset of virtual machines of
                  interest for vmOperState."
  ::= { vmNotifications 13 }

vmBulkShutdown NOTIFICATION-TYPE
  OBJECTS      {
                  vmAffectedVMs
                }
  STATUS        current
```

## DESCRIPTION

"This notification is generated when the operational state of one or more virtual machine has been changed to shutdown(12) from a state other than shutdown(12). Management stations are encouraged to subsequently poll the subset of virtual machines of interest for vmOperState."

::= { vmNotifications 14 }

## vmBulkPaused NOTIFICATION-TYPE

OBJECTS {  
vmAffectedVMs  
}

STATUS current

## DESCRIPTION

"This notification is generated when the operational state of one or more virtual machines have been changed to paused(9) from a state other than paused(9). Management stations are encouraged to subsequently poll the subset of virtual machines of interest for vmOperState."

::= { vmNotifications 15 }

## vmBulkSuspending NOTIFICATION-TYPE

OBJECTS {  
vmAffectedVMs  
}

STATUS current

## DESCRIPTION

"This notification is generated when the operational state of one or more virtual machines have been changed to suspending(6) from a state other than suspending(6). Management stations are encouraged to subsequently poll the subset of virtual machines of interest for vmOperState."

::= { vmNotifications 16 }

## vmBulkSuspended NOTIFICATION-TYPE

OBJECTS {  
vmAffectedVMs  
}

STATUS current

## DESCRIPTION

"This notification is generated when the operational state of one or more virtual machines have been changed to suspended(7) from a state other than suspended(7). Management stations are encouraged to subsequently poll

```

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

vmBulkResuming NOTIFICATION-TYPE
OBJECTS      {
                vmAffectedVMs
            }
STATUS      current
DESCRIPTION
    "This notification is generated when the operational
    state of one or more virtual machines have been changed
    to resuming(8) from a state other than resuming(8).
    Management stations are encouraged to subsequently poll
    the subset of virtual machines of interest for
    vmOperState."
 ::= { vmNotifications 18 }

vmBulkMigrating NOTIFICATION-TYPE
OBJECTS      {
                vmAffectedVMs
            }
STATUS      current
DESCRIPTION
    "This notification is generated when the operational
    state of one or more virtual machines have been changed
    to migrating(10) from a state other than migrating(10).
    Management stations are encouraged to subsequently poll
    the subset of virtual machines of interest for
    vmOperState."
 ::= { vmNotifications 19 }

vmBulkCrashed NOTIFICATION-TYPE
OBJECTS      {
                vmAffectedVMs
            }
STATUS      current
DESCRIPTION
    "This notification is generated when one or more virtual
    machines have been crashed. Management stations are
    encouraged to subsequently poll the subset of virtual
    machines of interest for vmOperState."
 ::= { vmNotifications 20 }

vmBulkBlocked NOTIFICATION-TYPE
OBJECTS      {
                vmAffectedVMs
            }
```



```

STATUS          current
DESCRIPTION
    "This notification is generated when the operational
    state of one or more virtual machines have been changed
    to blocked(5) from a state other than blocked(5).
    Management stations are encouraged to subsequently poll
    the subset of virtual machines of interest for
    vmOperState."
 ::= { vmNotifications 21 }

vmBulkDeleted NOTIFICATION-TYPE
OBJECTS          {
                  vmAffectedVMs
                }
STATUS          current
DESCRIPTION
    "This notification is generated when one or more virtual
    machines have been deleted. Management stations are
    encouraged to subsequently poll the subset of virtual
    machines of interest for vmOperState."
 ::= { vmNotifications 22 }

-- Compliance definitions:
vmGroups          OBJECT IDENTIFIER ::= { vmConformance 1 }
vmCompliances     OBJECT IDENTIFIER ::= { vmConformance 2 }

vmFullCompliances MODULE-COMPLIANCE
STATUS          current
DESCRIPTION
    "Compliance statement for implementations supporting
    read/write access, according to the object definitions."
MODULE          -- this module
MANDATORY-GROUPS {
    vmHypervisorGroup,
    vmVirtualMachineGroup,
    vmCpuGroup,
    vmCpuAffinityGroup,
    vmStorageGroup,
    vmNetworkGroup
}
GROUP vmPerVMNotificationOptionalGroup
DESCRIPTION
    "Support for per-VM notifications is optional. If not
    implemented then vmPerVMNotificationsEnabled must report
    false(2)."
```

```

GROUP vmBulkNotificationsVariablesGroup
DESCRIPTION
    "Necessary only if vmPerVMNotificationOptionalGroup is
```

```
        implemented."
GROUP    vmBulkNotificationOptionalGroup
DESCRIPTION
    "Support for bulk notifications is optional.  If not
    implemented then vmBulkNotificationsEnabled must report
    false(2)."
```

```
 ::= { vmCompliances 1 }
```

```
vmReadOnlyCompliances MODULE-COMPLIANCE
STATUS      current
DESCRIPTION
    "Compliance statement for implementations supporting
    only readonly access."
MODULE      -- this module
MANDATORY-GROUPS {
    vmHypervisorGroup,
    vmVirtualMachineGroup,
    vmCpuGroup,
    vmCpuAffinityGroup,
    vmStorageGroup,
    vmNetworkGroup
}

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

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

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

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

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

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

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

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

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

OBJECT vmBulkNotificationsEnabled
MIN-ACCESS    read-only
DESCRIPTION
    "Write access is not required."
 ::= { vmCompliances 2 }

vmHypervisorGroup OBJECT-GROUP
OBJECTS {
    vmHvSoftware,
    vmHvVersion,
    vmHvObjectID,
    vmHvUpTime,
    vmNumber,
    vmTableLastChange,
    vmPerVMNotificationsEnabled,
    vmBulkNotificationsEnabled
}
STATUS        current
DESCRIPTION
    "A collection of objects providing insight into the
    hypervisor itself."
 ::= { vmGroups 1 }

vmVirtualMachineGroup OBJECT-GROUP
OBJECTS {
    -- vmIndex
    vmName,
    vmUUID,
```

```
        vmOSType,
        vmAdminState,
        vmOperState,
        vmAutoStart,
        vmPersistent,
        vmCurCpuNumber,
        vmMinCpuNumber,
        vmMaxCpuNumber,
        vmMemUnit,
        vmCurMem,
        vmMinMem,
        vmMaxMem,
        vmUpTime,
        vmCpuTime
    }
    STATUS          current
    DESCRIPTION
        "A collection of objects providing insight into the
        virtual machines) controlled by a hypervisor."
    ::= { vmGroups 2 }

vmCpuGroup OBJECT-GROUP
    OBJECTS {
        -- vmCpuIndex,
        vmCpuCoreTime
    }
    STATUS          current
    DESCRIPTION
        "A collection of objects providing insight into the
        virtual machines) controlled by a hypervisor."
    ::= { vmGroups 3 }

vmCpuAffinityGroup OBJECT-GROUP
    OBJECTS {
        -- vmCpuPhysIndex,
        vmCpuAffinity
    }
    STATUS          current
    DESCRIPTION
        "A collection of objects providing insight into the
        virtual machines) controlled by a hypervisor."
    ::= { vmGroups 4 }

vmStorageGroup OBJECT-GROUP
    OBJECTS {
        -- vmStorageVmIndex,
        -- vmStorageIndex,
        vmStorageParent,
```

```
        vmStorageSourceType,
        vmStorageSourceTypeString,
        vmStorageResourceID,
        vmStorageAccess,
        vmStorageMediaType,
        vmStorageMediaTypeString,
        vmStorageSizeUnit,
        vmStorageDefinedSize,
        vmStorageAllocatedSize,
        vmStorageReadIOs,
        vmStorageWriteIOs
    }
    STATUS          current
    DESCRIPTION
        "A collection of objects providing insight into the
        virtual storage devices controlled by a hypervisor."
    ::= { vmGroups 5 }

vmNetworkGroup OBJECT-GROUP
    OBJECTS {
        -- vmNetworkIndex,
        vmNetworkIfIndex,
        vmNetworkParent,
        vmNetworkModel,
        vmNetworkPhysAddress
    }
    STATUS          current
    DESCRIPTION
        "A collection of objects providing insight into the
        virtual network interfaces controlled by a hypervisor."
    ::= { vmGroups 6 }

vmPerVMNotificationOptionalGroup NOTIFICATION-GROUP
    NOTIFICATIONS {
        vmRunning,
        vmShuttingdown,
        vmShutdown,
        vmPaused,
        vmSuspending,
        vmSuspended,
        vmResuming,
        vmMigrating,
        vmCrashed,
        vmBlocked,
        vmDeleted
    }
    STATUS          current
    DESCRIPTION
```

```
        "A collection of notifications for per-VM notification
        of changes to virtual machine state (vmOperState) as
        reported by a hypervisor."
 ::= { vmGroups 7 }

vmBulkNotificationsVariablesGroup OBJECT-GROUP
  OBJECTS {
    vmAffectedVMs
  }
  STATUS      current
  DESCRIPTION
    "The variables used in vmBulkNotificationOptionalGroup
    virtual network interfaces controlled by a hypervisor."
 ::= { vmGroups 8 }

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

END
```

#### 4. IANA Considerations

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

Descriptor	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`, `vmMinCpuNumber`, `vmMaxCpuNumber`, `vmMinMem`, `vmMaxMem`, and `vmCpuAffinity` in scenarios that DO NOT use SNMPv3 strong security (i.e. authentication and encryption). Extreme caution must be used to minimize the risk of cascading security vulnerabilities when SNMPv3 strong security is not used. When SNMPv3 strong security is not used, these objects should have access of read-only, not read-create.

SNMPv1 by itself is not a secure environment. Even if the network itself is secure (for example by using IPsec), even then, there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB.

It is recommended that the implementers consider the security features as provided by the SNMPv3 framework. Specifically, the use of the User-based Security Model [RFC3414] and the View-based Access Control Model [RFC3415] is recommended.

It is then a customer/user responsibility to ensure that the SNMP entity giving access to an instance of this MIB, is properly



configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

## 6. Acknowledgements

The authors like to thank Joe Marcus Clarke, Randy Presuhn, and David Black for providing helpful comments during the development of this specification.

Juergen Schoenwaelder was partly funded by Flamingo, a Network of Excellence project (ICT-318488) supported by the European Commission under its Seventh Framework Programme.

## 7. References

### 7.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.
- [RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.
- [RFC2580] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.
- [RFC2790] Waldbusser, S. and P. Grillo, "Host Resources MIB", RFC 2790, March 2000.
- [RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, June 2000.
- [RFC3413] Levi, D., Meyer, P., and B. Stewart, "Simple Network Management Protocol (SNMP) Applications", STD 62, RFC 3413, December 2002.
- [RFC3414] Blumenthal, U. and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", STD 62, RFC 3414, December 2002.
- [RFC3415] Wijnen, B., Presuhn, R., and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3415, December 2002.
- [RFC3418] Presuhn, R., "Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3418, December 2002.
- [RFC4122] Leach, P., Mealling, M., and R. Salz, "A Universally Unique IDentifier (UUID) URN Namespace", RFC 4122, July 2005.
- [RFC4133] Bierman, A. and K. McCloghrie, "Entity MIB (Version 3)", RFC 4133, August 2005.

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

## 7.2. Informative References

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

## Appendix A. State Transition Table

State	Action or (Event)	Next state	Notification
suspended	running	resuming	vmResuming   vmBulkResuming
suspending	(suspend operation completed)	suspended	vmSuspended   vmBulkSuspended
running	suspended	suspending	vmSuspending   vmBulkSuspending
	shutdown	shuttingdown	vmShuttingdown   vmBulkShuttingdown
	destroy	shutdown	vmShutdown   vmBulkShutdown
	(migration to other hypervisor initiated)	migrating	vmMigrating   vmBulkMigrating
resuming	(resume operation completed)	running	vmRunning   vmBulkRunning
paused	running	running	vmRunning   vmBulkRunning
shuttingdown	(shutdown operation completed)	shutdown	vmShutdown   vmBulkShutdown
shutdown	running	running	vmRunning   vmBulkRunning
	(if this state entry is created by a migration operation (*))	migrating	vmMigrating   vmBulkMigrating

	(deletion operation completed)	(no state)	vmDeleted   vmBulkDeleted
migrating	(migration from other hypervisor completed)	running	vmRunning   vmBulkRunning
	(migration to other hypervisor completed)	shutdown	vmShutdown   vmBulkShutdown
preparing	(preparation completed)	shutdown	vmShutdown   vmBulkShutdown
blocked	(blocking operation completed)	(previous state)	-
crashed	-	-	-
(any)	(blocking operation initiated)	blocked	vmBlocked   vmBulkBlocked
	(crashed)	crashed	vmCrashed   vmBulkCrashed
(no state)	(preparation initiated)	preparing	-
	(migrate from other hypervisor initiated)	shutdown (*)	vmShutdown   vmBulkShutdown

State transition table

## Authors' Addresses

Hirochika Asai  
The University of Tokyo  
7-3-1 Hongo  
Bunkyo-ku, Tokyo 113-8656  
JP

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

Michael MacFaden  
VMware Inc.  
  
Email: mrm@vmware.com

Juergen Schoenwaelder  
Jacobs University  
Campus Ring 1  
Bremen 28759  
Germany  
  
Email: j.schoenwaelder@jacobs-university.de

Yuji Sekiya  
The University of Tokyo  
2-11-16 Yayoi  
Bunkyo-ku, Tokyo 113-8658  
JP  
  
Email: sekiya@wide.ad.jp

Keiichi Shima  
IIJ Innovation Institute Inc.  
3-13 Kanda-Nishikicho  
Chiyoda-ku, Tokyo 101-0054  
JP  
  
Email: keiichi@iiijlab.net

Tina Tsou  
Huawei Technologies (USA)  
2330 Central Expressway  
Santa Clara CA 95050  
USA

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

Cathy Zhou  
Huawei Technologies  
Bantian, Longgang District  
Shenzhen 518129  
P.R. China

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

Hiroshi Esaki  
The University of Tokyo  
7-3-1 Hongo  
Bunkyo-ku, Tokyo 113-8656  
JP

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





Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: April 28, 2014

M. Ersue, Ed.  
Nokia Solutions and Networks  
D. Romascanu  
Avaya  
J. Schoenwaelder  
Jacobs University Bremen  
October 25, 2013

Management of Networks with Constrained Devices: Problem Statement and  
Requirements  
draft-ersue-opsawg-coman-probstate-reqs-00

Abstract

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

Status of this Memo

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

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

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

This Internet-Draft will expire on April 28, 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. Overview . . . . .	3
1.2. Terminology . . . . .	4
1.3. Networks Types and Characteristics in Focus . . . . .	5
1.4. Constrained Device Deployment Options . . . . .	9
1.5. Management Topology Options . . . . .	9
1.6. Managing the Constrainedness of a Device or Network . . . . .	10
2. Problem Statement . . . . .	14
3. Requirements on the Management of Networks with Constrained Devices . . . . .	16
3.1. Management Architecture/System . . . . .	16
3.2. Management protocols and data model . . . . .	21
3.3. Configuration management . . . . .	24
3.4. Monitoring functionality . . . . .	26
3.5. Self-management . . . . .	31
3.6. Security and Access Control . . . . .	32
3.7. Energy Management . . . . .	34
3.8. SW Distribution . . . . .	36
3.9. Traffic management . . . . .	36
3.10. Transport Layer . . . . .	38
3.11. Implementation Requirements . . . . .	39
4. IANA Considerations . . . . .	41
5. Security Considerations . . . . .	42
6. Contributors . . . . .	43
7. Acknowledgments . . . . .	44
8. References . . . . .	45
8.1. Normative References . . . . .	45
8.2. Informative References . . . . .	45
Appendix A. Related Development in other Bodies . . . . .	47
A.1. ETSI TC M2M . . . . .	47
A.2. OASIS . . . . .	48
A.3. OMA . . . . .	49
A.4. IPSO Alliance . . . . .	49
Appendix B. Related Research Projects . . . . .	51
Appendix C. Open issues . . . . .	52
Appendix D. Change Log . . . . .	53
D.1. draft-ersue-constrained-mgmt-03 - draft-ersue-opsawg-coman-probstate-reqs-00 . . . . .	53
D.2. draft-ersue-constrained-mgmt-02-03 . . . . .	53
D.3. draft-ersue-constrained-mgmt-01-02 . . . . .	54
D.4. draft-ersue-constrained-mgmt-00-01 . . . . .	55
Authors' Addresses . . . . .	56

## 1. Introduction

### 1.1. Overview

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

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

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

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

type of challenges compared to the management of a traditional IP network.

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

This document provides a problem statement and lists the requirements for the management of a network with constrained devices. Section 1.3 and Section 1.5 describe different topology options for the networking and management of constrained devices. Section 2 provides a problem statement on the issue of the management of networked constrained devices. Section 3 lists requirements on the management of applications and networks with constrained devices. Note that the requirements in Section 3 need to be seen as standalone requirements, where different implementer may decide to realize a different set of them.

The use cases in the context of networks with constrained devices can be found in the companion document [COM-US].

## 1.2. Terminology

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

The following terms are additionally used throughout this documentation:

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

C0: Class 0 constrained device as defined in Section 3. of [I-D.ietf-lwig-terminology].

C1: Class 1 constrained device as defined in Section 3. of [I-D.ietf-lwig-terminology].

C2: Class 2 constrained device as defined in Section 3. of [I-D.ietf-lwig-terminology].

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

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

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

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

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

For a detailed discussion on the constrained networks as well as classes of constrained devices and their capabilities please see [I-D.ietf-lwig-terminology].

### 1.3. Networks Types and Characteristics in Focus

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

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

1. Wireline non-constrained networks, e.g. an Ethernet-LAN with non-constrained and constrained devices involved.
2. A combination of wireline and wireless networks, which may or may not be mesh-based but have a multi-hop connectivity between constrained devices, utilizing dynamic routing in both the wireless and wireline portions of the network. Such networks usually support highly distributed applications with many nodes (e.g. environmental monitoring) and tend to deal with large-scale

multipoint-to-point systems with massive data flows. Wireless Mesh Networks (WMN), as a specific variant, use off-the-shelf radio technology such as Wi-Fi, WiMax, and cellular 3G/4G. WMNs are reliable based on the redundancy they offer and have often a more planned deployment to provide dynamic and cost effective connectivity over a certain geographic area.

3. A combination of wireline and wireless networks with point-to-point or point-to-multipoint communication generally with single-hop connectivity to constrained devices, utilizing static routing over the wireless network. Such networks support short-range, point-to-point, low-data-rate, source-to-sink type of applications such as RFID systems, light switches, fire and smoke detectors, and home appliances. This type of networks also support confined short-range spaces such as a home, a factory, a building, or the human body. IEEE 802.15.1 (Bluetooth) and IEEE 802.15.4 are well-known examples of applicable standards for such networks.
4. Mobile Adhoc networks (MANET) are self-configuring \_infrastructureless\_ networks of mobile devices connected by wireless technologies. MANETs are based on point-to-point communications of devices moving independently in any direction and changing the links to other devices frequently. MANET devices do act as a router to forward traffic unrelated to their own use.

Wireline non-constrained networks are mainly used for specific applications like Building Automation or Infrastructure Monitoring. However, wireline and wireless networks with multi-hop or point-to-multipoint connectivity are especially in the interest of the analysis on the management of constrained devices in this document.

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

#### Network Topology:

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

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

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

Dynamicity of the network topology:

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

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

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

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

Traffic flows:

The traffic flow in a network determines from which sources data traffic is sent to which destinations in the network. Several different traffic flows are defined in [I-D.ietf-roll-terminology],



including "point-to-point" (P2P), "multipoint-to-point" (MP2P), and "point-to-multipoint" (P2MP) flows as:

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

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

#### Bandwidth:

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

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

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

#### Loss rate:

The loss rate (or bit error rate) is the number of bit errors divided by the total number of bits transmitted. For wired networks, loss rates are typically extremely low, e.g. around  $10^{-12}$  or  $10^{-13}$  for the latest 10Gbit Ethernet. For wireless networks, such as 802.15.4,

the bit error rate can be as high as  $10^{-1}$  to  $10^{-0}$  in case of interferences. Even when using a reliable transport protocol, management operations can fail if the loss rate is too high, unless they are specifically designed to cope with these situations.

Note: The discussion on the management requirements of MANETs is currently not in the focus of this document. [COM-US] provides a use case to make it clear how a MANET-based application differs from others.

#### 1.4. Constrained Device Deployment Options

We differentiate following Deployment options for the constrained devices:

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

#### 1.5. Management Topology Options

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

- o A network of constrained devices managed by one central manager. A logically centralized management might be implemented in a hierarchical fashion for scalability and robustness reasons. The manager and the management application logic might have a gateway/

proxy in between or might be on different nodes in different networks, e.g., management application running on a cloud server.

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

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

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

A constrained device:

- o might only support an unreliable radio with lossy links, i.e. the client and server of a management protocol need to gracefully ignore incomplete commands or repeat commands as necessary.
- o might only be able to go online from time-to-time, where it is reachable, i.e. a command might be necessary to repeat after a longer timeout or the timeout value with which one endpoint waits on a response needs to be sufficiently high.
- o might only be able to support a limited operating time (e.g. based on the available battery), i.e. the devices need to economize their energy usage with suitable mechanisms and the managing entity needs to monitor and control the energy status of the constrained devices it manages.
- o might only be able to support one simple communication protocol, i.e. the management protocol needs to be possible to downscale from constrained (C2) to very constrained (C0) devices with

modular implementation and a very basic version with just a few simple commands.

- o might only be able to support limited or no user and/or transport security, i.e. the management system needs to support a less-costly and simple but sufficiently secure authentication mechanism.
- o might not be able to support compression and decompression of exchanged data based on limited CPU power, i.e. an intermediary entity which is capable of data compression should be able to communicate with both, devices, which support data compression (e.g. C2) and devices, which do not support data compression (e.g. C1 and C0).
- o might only be able to support very simple encryption, i.e. it would be efficient if the devices use cryptographic algorithms that are supported in hardware.
- o might only be able to communicate with one single managing entity and cannot support the parallel access of many managing entities.
- o might depend on a self-configuration feature, i.e. the managing entity might not know all devices in a network and the device needs to be able to initiate connection setup for the device configuration.
- o might depend on self- or neighbor-monitoring feature, i.e. the managing entity might not be able to monitor all devices in a network continuously.
- o might only be able to communicate with its neighbors, i.e. the device should be able to get its configuration from a neighbor.
- o might only be able to support parsing of data models with limited size, i.e. the device data models need to be compact containing the most necessary data and if possible parsable as a stream.
- o might only be able to support a limited or no failure detection, i.e. the managing entity needs to handle the situation, where a failure does not get detected or gets detected late gracefully e.g. with asking repeatedly.
- o might only be able to support the reporting of just one or a limited set failure types.
- o might only be able to support a limited set of notifications, possible only an "I-am-alive" message.

- o might only be able to support a soft-reset from failure recovery.
- o might possibly generate a huge amount of redundant reporting data, i.e. the intermediary management entity (see [I-D.ietf-core-coap]) should be able to filter and aggregate redundant data.

A constrained network:

- o might only support an unreliable radio with lossy links, i.e. the client and server of a management protocol need to repeat commands as necessary or gracefully ignore incomplete commands.
- o might be necessary to manage based on multicast communication, i.e. the managing entity needs to be prepared to configure many devices at once based on the same data model.
- o might have a very large topology supporting 10.000 or more nodes for some applications and as such node naming is a specific issue for constrained networks.
- o must be able to self-organize, i.e. given the large number of nodes and their potential placement in hostile locations and frequently changing topology, manual configuration is typically not feasible. As such the network must be able to reconfigure itself so that it can continue to operate properly and support reliable connectivity.
- o needs a management solution, which is energy-efficient, using as little wireless bandwidth as possible since communication is highly energy demanding.
- o needs to support localization schemes to determine the location of devices since the devices might be moving and location information is important for some applications.
- o needs a management solution, which is scalable as the network may consist of thousands of nodes and may need to be extended continuously.
- o needs to provide fault tolerance. Faults in network operation including hardware and software errors or failures detected by the transport protocol should be handled smoothly enabling. In such a case it should be possible to run the protocol possibly at a reduced level but avoiding to fail completely. E.g. self-monitoring mechanisms or graceful degradation of features can be used to provide fault tolerance.

- o might require new management capabilities: for example, network coverage information and a constrained device power-distribution-map.
- o might require a new management function for data management, since the type and amount of data collected in constrained networks is different from those of the traditional networks.
- o might also need energy-efficient key management algorithms for security.

## 2. Problem Statement

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

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

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

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

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

In an ideal world, we would have only one network management protocol for monitoring, configuration, and exchanging management data,

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

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

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



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

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

Note that the requirements in this section need to be seen as standalone requirements. A device might be able to provide only a particular profile of requirements (i.e. selected set of requirements) and might not be capable to provide all requirements in this document. On the other hand a device vendor might select a subset of the requirements to implement. As of today this document does not recommend the realization of a profile of requirements.

Following template is used for the definition of the requirements.

Req-ID: An ID uniquely identified by a three-digit number

Title: The title of the requirement.

Description: The rational and description of the requirement.

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

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

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

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

#### 3.1. Management Architecture/System

Req-ID: 4.1.001

Title: Support multiple device classes within a single network.

Description: Larger networks usually are made up of devices belonging to different device classes (e.g., constrained mesh endpoints and less constrained routers) that work together. Hence, the management architecture must be applicable to networks that have a mix of different device classes. See Section 3. of [I-D.ietf-lwig-terminology] for the definition of Constrained Device Classes.

Source: All use cases.

Requirement Type: Non-Functional Requirement

Device type: Managing and intermediary entities.

Priority: High

---

Req-ID: 4.1.002

Title: Management scalability.

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

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

Requirement Type: Non-Functional Requirement

Device type: C0, C1, and C2

Priority: High

---

Req-ID: 4.1.003

Title: Hierarchical management

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

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

Requirement Type: Non-Functional Requirement

Device type: Managing and intermediary entities.

Priority: Medium

---

Req-ID: 4.1.004

Title: Minimize state maintained on constrained devices.

Description: The amount of state that needs to be maintained on constrained devices should be minimized. This is important in order to save memory (especially relevant for C0 and C1 devices) and in order to allow devices to restart for example to apply configuration changes or to recover from extended periods of inactivity. One way to achieve this is to adopt a RESTful architecture that minimizes the amount of state maintained by managed constrained devices and that makes resources of a device addressable via URIs.

Source: Basic requirement which concerns all use cases.

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: High

---

Req-ID: 4.1.005

Title: Automatic re-synchronization with eventual consistency.

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

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

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: High

---

Req-ID: 4.1.006

Title: Support for lossy links and unreachable devices.

Description: Some constrained devices will only be able to support lossy and unreliable links characterized by a limited data rate, a high latency, and a high transmission error rate. Furthermore constrained devices often duty cycle their radio or the whole device in order to save energy. In both cases the management system must not assume that constrained devices are always reachable. The management protocol(s) must act gracefully if a constrained device is not reachable and provide a high degree of resilience. Intermediaries may be used that provide information for devices currently inactive or that take responsibility to re-synchronize devices when they become reachable again after an extended offline period.

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

Requirement Type: Non-Functional Requirement

Device type: C0, C1, and C2

Priority: High

---

Req-ID: 4.1.007

Title: Network-wide configuration

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

Source: In general all use cases, which want to configure the network and its devices based on a network view in a top-down manner.

Requirement Type: Non-Functional Requirement

Device type: C0, C1, and C2

Priority: Medium

---

Req-ID: 4.1.008

Title: Distributed Management

Description: Provide a means of simple distributed management, where a constrained network can be managed or monitored by more than one manager. Since the connectivity to a server cannot be guaranteed at all times, a distributed approach may provide a higher reliability, at the cost of increased complexity. This

requirement implies the handling of data consistency in case of concurrent read and write access to the device datastore. It might also happen that no management (configuration) server is accessible and the only reachable node is a peer device. In this case the device should be able to obtain its configuration from peer devices.

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

Requirement Type: Non-Functional Requirement

Device type: C1 and C2

Priority: Medium

### 3.2. Management protocols and data model

Req-ID: 4.2.001

Title: Modular implementation of management protocols

Description: Management protocols should allow modular implementations, i.e., it should be possible to implement only a basic set of protocol primitives on highly constrained devices while devices with additional resources may provide more support for additional protocol primitives. It should be possible to discover the management protocol primitives by a device.

Source: Basic requirement interesting for all use cases.

Requirement Type: Non-Functional Requirement

Device type: C0, C1, and C2

Priority: High

---

Req-ID: 4.2.002

Title: Compact encoding of management data

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

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

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: High

---

Req-ID: 4.2.003

Title: Compression of management data or complete messages

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

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

Requirement Type: Functional Requirement

Device type: C1 and C2

Priority: Medium

---

Req-ID: 4.2.004

Title: Mapping of management protocol interactions.

Description: It is desirable to have a loss-less automated mapping between the management protocol used to manage constrained devices and the management protocols used to manage regular devices. In the ideal case, the same core management protocol can be used with certain restrictions taking into account the resource limitations of constrained devices. However, for very resource constrained devices, this goal might not be achievable. Hence this requirement is marked optional for device class C2.

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

Requirement Type: Functional Requirement

Device type: C2

Priority: Medium

---

Req-ID: 4.2.005

Title: Consistency of data models with the underlying information model.

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

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

Requirement Type: Non-Functional Requirement

Device type: C0, C1, and C2

Priority: High

---

Req-ID: 4.2.006

Title: Loss-less mapping of management data models.

Description: It is desirable to have a loss-less automated mapping between the management data models used to manage regular devices and the management data models used for managing constrained devices. In the ideal case, the same core data models can be used with certain restrictions taking into account the resource



limitations of constrained devices. However, for very resource constrained devices, this goal might not be achievable. Hence this requirement is marked optional for device class C2.

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

Requirement Type: Functional Requirement

Device type: C2

Priority: Medium

---

Req-ID: 4.2.007

Title: Protocol extensibility

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

Source: Basic requirement useful for all use cases.

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: High

### 3.3. Configuration management

Req-ID: 4.3.001

Title: Self-configuration capability

Description: Automatic configuration and re-configuration of devices without manual intervention. Compared to the traditional management of devices where the management application is the central entity configuring the devices, in the auto-configuration scenario the device is the active part and initiates the configuration process. Self-configuration can be initiated during the initial configuration or for subsequent configurations, where the configuration data needs to be refreshed. Self-configuration should be also supported during the initialization phase or in the

event of failures, where prior knowledge of the network topology is not available or the topology of the network is uncertain.

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

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

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

---

Req-ID: 4.3.002

Title: Capability Discovery

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

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

Requirement Type: Functional Requirement

Device type: C1 and C2

Priority: Medium

---

Req-ID: 4.3.003

Title: Asynchronous Transaction Support

Description: Provide configuration management with asynchronous transaction support. Configuration operations must support a transactional model, with asynchronous indications that the transaction was completed.

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

Requirement Type: Functional Requirement

Device type: C1 and C2

Priority: Medium

---

Req-ID: 4.3.004

Title: Network reconfiguration

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

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

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: Medium

### 3.4. Monitoring functionality

Req-ID: 4.4.001

Title: Device status monitoring

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

Source: All use cases

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: High, Medium for neighbor-monitoring.

---

Req-ID: 4.4.002

Title: Energy status monitoring

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

Source: Use case Energy Management

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

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

---

Req-ID: 4.4.003

Title: Monitoring of current and estimated device availability

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

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

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: Medium

---

Req-ID: 4.4.004

Title: Network status monitoring

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

Source: All use cases.

Requirement Type: Functional Requirement

Device type: C1 and C2

Priority: Low, based on the realization complexity.

---

Req-ID: 4.4.005

Title: Self-monitoring

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

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

Requirement Type: Functional Requirement

Device type: C1 and C2

Priority: High for C2, Medium for C1

---

Req-ID: 4.4.006

Title: Performance Monitoring

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

Source: Use cases Building automation, and Transport applications

Requirement Type: Functional Requirement

Device type: C1 and C2

Priority: Low

---

Req-ID: 4.4.007

Title: Fault detection monitoring

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

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

Requirement Type: Functional Requirement

Device type: C0, C1 and C2

Priority: Medium

---

Req-ID: 4.4.008

Title: Passive and Reactive Monitoring

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

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

Requirement Type: Functional Requirement

Device type: C2

Priority: Medium

---

Req-ID: 4.4.009

Title: Recovery

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

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

Requirement Type: Functional Requirement

Device type: C2

Priority: Medium

---

Req-ID: 4.4.010

Title: Network topology discovery

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

Source: Use cases Community Network Applications and Mobile Applications

Requirement Type: Functional Requirement

Device type: C1 and C2

Priority: Low, based on the realization complexity.

---

Req-ID: 4.4.011

Title: Notifications

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

Source: All use cases.

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: Medium for C2, Low for C1

---

Req-ID: 4.4.012

Title: Logging

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

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

Requirement Type: Functional Requirement

Device type: C2

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

### 3.5. Self-management

Req-ID: 4.5.001



Title: Self-management - Self-healing

Description: Enable event-driven and/or periodic self-management functionality in a device. The device should be able to react in case of a failure e.g. by initiating a fully or partly reset and initiate a self-configuration or management data update as necessary. A device might be further able to check for failures cyclically or schedule-controlled to trigger self-management as necessary. It is a matter of device design and subject for discussion how much self-management a C1 device can support. A minimal failure detection and self-management logic is assumed to be generally useful for the self-healing of a device.

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

Requirement Type: Functional Requirement

Device type: C1 and C2

Priority: High for C2, Medium for C1

### 3.6. Security and Access Control

Req-ID: 4.6.001

Title: Authentication of management system and devices.

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

Source: Basic security requirement for all use cases.

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

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

---

Req-ID: 4.6.002

Title: Support suitable security bootstrapping mechanisms

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

Source: Basic security requirement for all use cases.

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: High

---

Req-ID: 4.6.003

Title: Access control on management system and devices

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

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

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: High

---

Req-ID: 4.6.004

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

Description: Cryptographic algorithms have a major impact in terms of both code size and overall execution time. It is therefore necessary to select mandatory to implement cryptographic algorithms (like some elliptic curve algorithm) that are reasonable to implement with the available code space and that have a small impact at runtime. Furthermore some wireless technologies (e.g., IEEE 802.15.4) require the support of certain cryptographic algorithms. It might be useful to choose algorithms that are likely to be supported in wireless chipsets for certain wireless technologies.

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

Requirement Type: Non-Functional Requirement

Device type: C0, C1, and C2

Priority: High, Medium for hardware-supported algorithms.

### 3.7. Energy Management

Req-ID: 4.7.001

Title: Management of Energy Resources

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

Source: Use case Energy Management

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

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

---

Req-ID: 4.7.002

Title: Support of energy-optimized communication protocols

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

Source: Use cases Energy Management and Mobile Applications.

Requirement Type: Non-Functional Requirement

Device type: C2

Priority: Medium

---

Req-ID: 4.7.003

Title: Support for layer 2 energy-aware protocols

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

Source: Use case Energy Management

Requirement Type: Non-Functional Requirement

Device type: C0, C1, and C2

Priority: Medium

---

Req-ID: 4.7.004

Title: Dying gasp

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

Source: Use case Energy Management

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: Medium

### 3.8. SW Distribution

Req-ID: 4.8.001

Title: Group-based provisioning

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

Source: All use cases

Requirement Type: Non-Functional Requirement

Device type: C0, C1, and C2

Priority: Medium

### 3.9. Traffic management

Req-ID: 4.9.001

Title: Congestion avoidance

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

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

Requirement Type: Non-Functional Requirement

Device type: C1 and C2

Priority: Medium

---

Req-ID: 4.9.002

Title: Redirect traffic

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

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

Requirement Type: Non-Functional Requirement

Device type: Intermediary entity in the network.

Priority: Medium

---

Req-ID: 4.9.003

Title: Traffic delay schemes.

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

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

Requirement Type: Non-Functional Requirement

Device type: Intermediary entity in the network.

Priority: Medium

### 3.10. Transport Layer

Req-ID: 4.10.001

Title: Scalable transport layer

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

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

Requirement Type: Non-Functional Requirement

Device type: C0, C1 and C2

Priority: Medium

---

Req-ID: 4.10.002

Title: Reliable unicast transport of messages

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

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

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: High

---

Req-ID: 4.10.003

Title: Best-effort multicast

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

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

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: Medium

Req-ID: 4.10.004

Title: Secure message transport.

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

Source: All use cases.

Requirement Type: Non-Functional Requirements

Device type: C1 and C2

Priority: High

### 3.11. Implementation Requirements

Req-ID: 4.11.001

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



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

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

Requirement Type: Non-Functional Requirement

Device type: C0, C1, and C2

Priority: High

Req-ID: 4.11.002

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

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

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

Requirement Type: Non-Functional Requirement

Device type: C0, C1, and C2

Priority: High

#### 4. IANA Considerations

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

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

## 5. Security Considerations

This document discusses the problem statement and requirements on the network of constrained devices. If specific requirements for security will be identified, they will be described in future versions of this document.

## 6. Contributors

Ulrich Herberg (Fujitsu Laboratories of America) contributed to the Section 1.3 on Networks Types and Characteristics in Focus.

## 7. Acknowledgments

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

Dominique Barthel, Carsten Bormann, Zhen Cao, Benoit Claise, Bert Greevenbosch, Ulrich Herberg, James Nguyen, Anuj Sehgal, Zach Shelby, and Peter van der Stok.

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

## 8. References

### 8.1. Normative References

### 8.2. Informative References

- [RFC6632] Ersue, M. and B. Claise, "An Overview of the IETF Network Management Standards", RFC 6632, June 2012.
- [I-D.ietf-lwig-terminology]  
Bormann, C., Ersue, M., and A. Keranen, "Terminology for Constrained Node Networks", draft-ietf-lwig-terminology-05 (work in progress), July 2013.
- [I-D.ietf-core-coap]  
Shelby, Z., Hartke, K., and C. Bormann, "Constrained Application Protocol (CoAP)", draft-ietf-core-coap-18 (work in progress), June 2013.
- [I-D.ietf-roll-terminology]  
Vasseur, J., "Terms used in Routing for Low power And Lossy Networks", draft-ietf-roll-terminology-13 (work in progress), October 2013.
- [M2MDEVCLASS]  
Open Mobile Alliance, "OMA M2M Device Classification v1.0", October 2012, <[http://technical.openmobilealliance.org/Technical/release\\_program/m2m\\_Device\\_class\\_v1\\_0.aspx](http://technical.openmobilealliance.org/Technical/release_program/m2m_Device_class_v1_0.aspx)>.
- [EU-IOT-A]  
EU Commission Seventh Framework Programme, "EU FP7 Project Internet-of-Things Architecture", <<http://www.iot-a.eu/>>.
- [EU-SENSEI]  
EU Commission Seventh Framework Programme, "EU Project SENSEI", <<http://www.sensei-project.eu/>>.
- [EU-FI-WARE]  
EU Commission Future Internet Public Private Partnership (FI-PPP), "EU Project Future Internet-Core Platform", <<http://www.iot-butler.eu/>>.
- [EU-IOT-BUTLER]  
EU Commission Seventh Framework Programme, "EU FP7 Project Butler Smartlife", <<http://www.iot-butler.eu/>>.
- [COM-US] Ersue, M., "Constrained Management: Use Cases",

draft-ersue-coman-use-cases (work in progress),  
October 2013.

## Appendix A. Related Development in other Bodies

Note that over time the summary on the related work in other bodies might become outdated.

### A.1. ETSI TC M2M

ETSI Technical Committee Machine-to-Machine (ETSI TC M2M) aims to provide an end-to-end view of M2M standardization, which enables the integration of multiple vertical M2M applications. The main goal is to overcome the current M2M market fragmentation and to reuse existing mechanisms from telecom standards such as from OMA or 3GPP.

ETSI Release 1 is functionally frozen. The main focus is on use cases for Smart Metering (Technical Report (TR) 102 691) but it also includes eHealth use cases (TR 102 732) and some others. The Service requirements (Technical Standard (TS) 102 689) derived from the use cases, and the functional architecture specification (TS 102 690), will together define the M2M platform. The architecture consists of Service Capabilities (SC), which are basic functional building blocks for building the M2M platform.

Smart Metering is seen as the important showcase for M2M. It is believed that the Service Enablers that were defined based on the work done for Smart Metering and eHealth segments will also allow the building of other services like vending machines, alarm systems etc.

The functional architecture includes following management-related definitions:

- o Network Management Functions: consists of all functions required to manage the Access, Transport and Core networks: these include Provisioning, Supervision, Fault Management, etc.
- o M2M Management Functions: consists of functions required to manage generic functionalities of M2M Applications and M2M Service Capabilities in the Network and Applications Domain. The management of the M2M Devices and Gateways may use specific M2M Service Capabilities.

The Release 2 work of ETSI TC M2M has started beginning of 2012. Following is a list of networking- and management-related topics under work:

- o Interworking with 3GPP networks. This is a new work item, and no discussion has been held on technical details. The intent is to define which ETSI TC M2M functions are applicable when 3GPP NW is used as transport. It is possible that this work would also cover



details on how to use 3GPP interfaces, e.g. those defined in the SIMTC work, but also for charging and policy control.

- o Creating a Semantic Model or Data Abstraction layer for vertical industries and interworking. This would provide some high level information description that would be usable for interworking with local networks (e.g. ZigBee), and also for verticals, and it would allow the ETSI Service Enablement layer to also understand the data, instead of being just a bit storage and bit pipe. All technical details are still under discussion, but it has been agreed that a function for this exists in the architecture at least for interworking.

#### A.2. OASIS

Developments in OASIS related to management of constrained networks are following:

- o The Energy Interoperation TC works to define interaction between Smart Grids and their end nodes, including Smart Buildings, Enterprises, Industry, Homes, and Vehicles. The TC develops data and communication models that enable the interoperable and standard exchange of signals for dynamic pricing, reliability, and emergencies. The TC's agenda also extends to the communication of market participation data (such as bids), load predictability, and generation information. The first version of the Energy Interoperation specification is in final review.
- o OASIS Open Data Protocol (OData) aims to simplify the querying and sharing of data across disparate applications and multiple stakeholders for re-use in the enterprise, Cloud, and mobile devices. As a REST-based protocol, OData builds on HTTP, AtomPub, and JSON using URIs to address and access data feed resources. It enables information to be accessed from a variety of sources including (but not limited to) relational databases, file systems, content management systems, and traditional Web sites.
- o Open Building Information Exchange (oBIX) aims to enable the mechanical and electrical control systems in buildings to communicate with enterprise applications, and to provide a platform for developing new classes of applications that integrate control systems with other enterprise functions. Enterprise functions include processes such as Human Resources, Finance, Customer Relationship Management (CRM), and Manufacturing.

### A.3. OMA

OMA is currently working on Lightweight M2M Enabler, OMA Device Management (OMA DM) Next Generation, and a white paper on M2M Device Classification.

The Lightweight M2M Enabler covers both M2M device management and service management for constrained devices. In the case of less constrained devices, OMA DM Next Generation Enabler may be more appropriate. OMA DM is structured around Management Objects (MO), each specified for a specific purpose. There is also ongoing work with various other MOs such as the Gateway Management Object (GwMO). A draft for the "Lightweight M2M Requirements" is available.

OMA Lightweight M2M and OMA DM Next Generation are important to M2M device management, provisioning and service managements in both the protocol and management objects. OMA Lightweight M2M work seems to have grown from its original scope of being targeted for very simple devices only, i.e. such that could not handle all those protocols that ETSI M2M requires.

The white paper on the M2M Device Classification [M2MDEVCLASS] provides an M2M device classification framework based on the horizontal attributes (e.g., wide or local area communication interface, IP stack, I/O capabilities) of interest to communication service providers and M2M service providers, independent of vertical markets, such as smart grid, connected cars, e-health, etc. The white paper can be used as a tool to analyze the applicability of existing requirements and specifications developed by OMA and other cooperative standards development organizations.

### A.4. IPSO Alliance

IPSO Alliance developed a profile for Device Functions supporting devices such as sensors with a limited user interface, where the configuration of even basic parameters is impossible to do manually. This is a challenge especially for consumer devices that are managed by non-professional users. The configuration of a web service application running on a constrained device goes beyond the autoconfiguration of the IP stack and local information (e.g. proxy address). Constrained devices need additionally service provider and user account related configuration, such as an address/locator and the username for a web server.

IPSO discusses the use cases and requirements for user friendly configuration of such information on a constrained device, and specifies how IPSO profile Device Function Set can be used in the process. It furthermore defines a standard format for the basic

application configuration information.

## Appendix B. Related Research Projects

- o The EU project IoT-A (Internet-of-Things Architecture) develops an architectural reference model together with the definition of an initial set of key building blocks. These enable the integration of IoT into the service layer of the Future Internet, and realize a novel resolution infrastructure, as well as a network infrastructure that allows the seamless communication flow between IoT devices and services. The development includes a conceptual model of a smart object as well as a basic Internet of Things reference model defining the interaction and communication between IoT devices and relevant entities. The requirements document includes also network and information management requirements (see [EU-IOT-A]).
- o The EU project SENSEI specified the document on 'End to End Networking and Management' for Wireless Sensor and Actuator Networks. This report presents several research results carried out in SENSEI's tasks related to End-to-End Networking and Management. Particular analyses have been addressed related to naming and addressing of resources, management of resources, resource plug and play, resource level mobility and traffic modelling. The detailed analysis on each of these topics is intended to identify possible gaps between their specific mechanisms and the functional requirements in the SENSEI reference architecture (see [EU-SENSEI]).
- o The EU project FI-WARE is developing the Things Management GE (generic enabler), which uses a data model derived from the OMA DM NGSI data model. Using the abstraction level of things which include non-technical things like rooms, places and people, Things Management GE aims to discover and look up IoT resources that can provide information about things or actuate on these things. The system aims to manage the dynamic associations between IoT resources and things in order to allow internal components as well as external applications to interact with the system using the thing abstraction as the core concept (see [EU-FI-WARE]).
- o EU project BUTLER Smart Life discusses different IoT management aspects and collects requirements for smart life use cases (e.g. smart home or smart city) mainly from service management pov. (see [EU-IOT-BUTLER]).

Appendix C. Open issues

- o Section 4 on the management requirements, as the core section in the document, needs further consolidation.

## Appendix D. Change Log

- D.1. draft-ersue-constrained-mgmt-03 -  
draft-ersue-opsawg-coman-probstate-reqs-00
- o Reduced the terminology section for terminology addressed in the LWIG terminology draft. Referenced the LWIG terminology draft.
  - o Checked and aligned all terminology against the LWIG terminology draft.
  - o Moved section 1.4. Constrained Device Deployment Options and section 3. Use Cases to the companion document [COM-US].
  - o Renamed Section 1.3. Class of Networks in Focus to "Network Types in Focus" and removed abbreviations C0, C1 and C2 for network classes as they have not been used.
  - o Changed requirement priority classes to be High, Medium and Low.
  - o Changed requirement types to be Functional and Non-Functional and added text to explain the requirement types.
  - o Reformulation of some text parts for more clarity.
- D.2. draft-ersue-constrained-mgmt-02-03
- o Extended the terminology section and removed some of the terminology addressed in the new LWIG terminology draft. Referenced the LWIG terminology draft.
  - o Moved Section 1.3. on Constrained Device Classes to the new LWIG terminology draft.
  - o Class of networks considering the different type of radio and communication technologies in use and dimensions extended.
  - o Extended the Problem Statement in Section 2. following the requirements listed in Section 4.
  - o Following requirements, which belong together and can be realized with similar or same kind of solutions, have been merged.
    - \* Distributed Management and Peer Configuration,
    - \* Device status monitoring and Neighbor-monitoring,

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

#### D.3. draft-ersue-constrained-mgmt-01-02

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

- o Added Section 3 on the detailed requirements on constrained management matched to management tasks like fault, monitoring, configuration management, Security and Access Control, Energy Management, etc.
- o Solved nits and added references.
- o Added Appendix A on the related development in other bodies.
- o Added Appendix B on the work in related research projects.

D.4. draft-ersue-constrained-mgmt-00-01

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



Authors' Addresses

Mehmet Ersue (editor)  
Nokia Solutions and Networks

Email: mehmet.ersue@nsn.com

Dan Romascanu  
Avaya

Email: dromasca@avaya.com

Juergen Schoenwaelder  
Jacobs University Bremen

Email: j.schoenwaelder@jacobs-university.de



Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: April 28, 2014

M. Ersue, Ed.  
Nokia Solutions and Networks  
D. Romascanu  
Avaya  
J. Schoenwaelder  
Jacobs University Bremen  
October 25, 2013

Management of Networks with Constrained Devices: Use Cases  
draft-ersue-opsawg-coman-use-cases-00

Abstract

This document discusses the use cases concerning the management of networks, where constrained devices are involved. A problem statement, deployment options and the requirements on the networks with constrained devices can be found in the companion document [COM-REQ].

Status of this Memo

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

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

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

This Internet-Draft will expire on April 28, 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. Overview . . . . .	3
1.2. Terminology . . . . .	4
2. Use Cases . . . . .	5
2.1. Environmental Monitoring . . . . .	5
2.2. Medical Applications . . . . .	5
2.3. Industrial Applications . . . . .	6
2.4. Home Automation . . . . .	7
2.5. Building Automation . . . . .	8
2.6. Energy Management . . . . .	9
2.7. Transport Applications . . . . .	11
2.8. Infrastructure Monitoring . . . . .	12
2.9. Community Network Applications . . . . .	13
2.10. Mobile Applications . . . . .	15
2.11. Automated Metering Infrastructure (AMI) . . . . .	16
2.12. MANET Concept of Operations (CONOPS) in Military . . . . .	18
3. IANA Considerations . . . . .	24
4. Security Considerations . . . . .	25
5. Contributors . . . . .	26
6. Acknowledgments . . . . .	27
7. References . . . . .	28
7.1. Normative References . . . . .	28
7.2. Informative References . . . . .	28
Appendix A. Open issues . . . . .	29
Appendix B. Change Log . . . . .	30
B.1. draft-ersue-constrained-mgmt-03 - draft-ersue-opsawg-coman-use-cases-00 . . . . .	30
B.2. draft-ersue-constrained-mgmt-02-03 . . . . .	30
B.3. draft-ersue-constrained-mgmt-01-02 . . . . .	31
B.4. draft-ersue-constrained-mgmt-00-01 . . . . .	32
Authors' Addresses . . . . .	33

## 1. Introduction

### 1.1. Overview

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

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

This document aims to understand the use cases for the management of a network, where constrained devices are involved. The document lists and discusses diverse use cases for the management from the network as well as from the application point of view. The application scenarios discussed aim to show where networks of constrained devices are expected to be deployed. For each application scenario, we first briefly describe the characteristics followed by a discussion on how network management can be provided, who is likely going to be responsible for it, and on which time-scale management operations are likely to be carried out.

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

## 1.2. Terminology

This documents builds on the terminology defined in [I-D.ietf-lwig-terminology] and [COM-REQ]. [I-D.ietf-lwig-terminology] is a base document for the terminology concerning constrained devices and constrained networks.

## 2. Use Cases

### 2.1. Environmental Monitoring

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

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

Management responsibility typically rests with the organization running the environmental monitoring application. Since these monitoring applications must be designed to tolerate a number of failures, the time scale for detecting and recording failures is for some of these applications likely measured in hours and repairs might easily take days. However, for certain environmental monitoring applications, much tighter time scales may exist and might be enforced by regulations (e.g., monitoring of nuclear radiation).

### 2.2. Medical Applications

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

Constrained devices that are part of medical applications are managed

either by the users of those devices or by an organization providing medical (monitoring) services for physicians. In the first case, management must be automatic and or easy to install and setup by average people. In the second case, it can be expected that devices be controlled by specially trained people. In both cases, however, it is crucial to protect the privacy of the people to which medical devices are attached. Even though the data collected by a heart beat monitor might be protected, the pure fact that someone carries such a device may need protection. As such, certain medical appliances may not want to participate in discovery and self-configuration protocols in order to remain invisible.

Many medical devices are likely to be used (and relied upon) to provide data to physicians in critical situations since the biggest market is likely elderly and handicapped people. As such, fault detection of the communication network or the constrained devices becomes a crucial function that must be carried out with high reliability and, depending on the medical appliance and its application, within seconds.

### 2.3. Industrial Applications

Industrial Applications and smart manufacturing refer not only to production equipment, but also to a factory that carries out centralized control of energy, HVAC (heating, ventilation, and air conditioning), lighting, access control, etc. via a network. For the management of a factory it is becoming essential to implement smart capabilities. From an engineering standpoint, industrial applications are intelligent systems enabling rapid manufacturing of new products, dynamic response to product demand, and real-time optimization of manufacturing production and supply chain networks. Potential industrial applications e.g. for smart factories and smart manufacturing are:

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



Sensor networks are an essential technology used for smart manufacturing. Measurements, automated controls, plant optimization, health and safety management, and other functions are provided by a large number of networked sectors. Data interoperability and seamless exchange of product, process, and project data are enabled through interoperable data systems used by collaborating divisions or business systems. Intelligent automation and learning systems are vital to smart manufacturing but must be effectively integrated with the decision environment. Wireless sensor networks (WSN) have been developed for machinery Condition-based Maintenance (CBM) as they offer significant cost savings and enable new functionalities. Inaccessible locations, rotating machinery, hazardous areas, and mobile assets can be reached with wireless sensors. WSNs can provide today wireless link reliability, real-time capabilities, and quality-of-service and enable industrial and related wireless sense and control applications.

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

Management responsibility is typically owned by the organization running the industrial application. Since the monitoring applications must handle a potentially large number of failures, the time scale for detecting and recording failures is for some of these applications likely measured in minutes. However, for certain industrial applications, much tighter time scales may exist, e.g. in real-time, which might be enforced by the manufacturing process or the use of critical material.

#### 2.4. Home Automation

Home automation includes the control of lighting, heating, ventilation, air conditioning, appliances, and entertainment devices to improve convenience, comfort, energy efficiency, and security. It can be seen as a residential extension of building automation.

Home automation networks need a certain amount of configuration (associating switches or sensors to actors) that is either provided by electricians deploying home automation solutions or done by residents by using the application user interface to configure (parts

of) the home automation solution. Similarly, failures may be reported via suitable interfaces to residents or they might be recorded and made available to electricians in charge of the maintenance of the home automation infrastructure.

The management responsibility lies either with the residents or it may be outsourced to electricians providing management of home automation solutions as a service. The time scale for failure detection and resolution is in many cases likely counted in hours to days.

## 2.5. Building Automation

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

Building automation requires the deployment of a large number (10-100.000) of sensors that monitor the status of devices, and parameters inside the building and controllers with different specialized functionality for areas within the building or the totality of the building. Inter-node distances between neighboring nodes vary between 1 to 20 meters. Contrary to home automation, in building management the devices are expected to be managed assets and known to a set of commissioning tools and a data storage, such that every connected device has a known origin. The management includes verifying the presence of the expected devices and detecting the presence of unwanted devices.

Examples of functions performed by such controllers are regulating the quality, humidity, and temperature of the air inside the building and lighting. Other systems may report the status of the machinery inside the building like elevators, or inside the rooms like projectors in meeting rooms. Security cameras and sensors may be deployed and operated on separate dedicated infrastructures connected to the common backbone. The deployment area of a BAS is typically inside one building (or part of it) or several buildings geographically grouped in a campus. A building network can be composed of subnets, where a subnet covers a floor, an area on the floor, or a given functionality (e.g. security cameras).

Some of the sensors in Building Automation Systems (for example fire alarms or security systems) register, record and transfer critical alarm information and therefore must be resilient to events like loss of power or security attacks. This leads to the need that some components and subsystems operate in constrained conditions and are separately certified. Also in some environments, the malfunctioning of a control system (like temperature control) needs to be reported in the shortest possible time. Complex control systems can misbehave, and their critical status reporting and safety algorithms need to be basic and robust and perform even in critical conditions.

Building Automation solutions are deployed in some cases in newly designed buildings, in other cases it might be over existing infrastructures. In the first case, there is a broader range of possible solutions, which can be planned for the infrastructure of the building. In the second case the solution needs to be deployed over an existing structure taking into account factors like existing wiring, distance limitations, the propagation of radio signals over walls and floors. As a result, some of the existing WLAN solutions (e.g. IEEE 802.11 or IEEE 802.15) may be deployed. In mission-critical or security sensitive environments and in cases where link failures happen often, topologies that allow for reconfiguration of the network and connection continuity may be required. Some of the sensors deployed in building automation may be very simple constrained devices for which class 0 or class 1 may be assumed.

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

## 2.6. Energy Management

EMAN working group developed [I-D.ietf-eman-framework], which defines a framework for providing Energy Management for devices within or connected to communication networks. This document observes that one of the challenges of energy management is that a power distribution

network is responsible for the supply of energy to various devices and components, while a separate communication network is typically used to monitor and control the power distribution network. Devices that have energy management capability are defined as Energy Devices and identified components within a device (Energy Device Components) can be monitored for parameters like Power, Energy, Demand and Power Quality. If a device contains batteries, they can be also monitored and managed.

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

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

Energy Management is especially relevant to Smart Grid. A Smart Grid is an electrical grid that uses data networks to gather and act on energy and power-related information, in an automated fashion with the goal to improve the efficiency, reliability, economics, and sustainability of the production and distribution of electricity. As such Smart Grid provides sustainable and reliable generation, transmission, distribution, storage and consumption of electrical energy based on advanced energy and ICT solutions and as such enables e.g. following specific application areas: Smart transmission systems, Demand Response/Load Management, Substation Automation, Advanced Distribution Management, Advanced Metering Infrastructure (AMI), Smart Metering, Smart Home and Building Automation, E-mobility, etc.

Smart Metering is a good example of a M2M application and can be realized as one of the vertical applications in an M2M environment. Different types of possibly wireless small meters produce all together a huge amount of data, which is collected by a central entity and processed by an application server. The M2M infrastructure can be provided by a mobile network operator as the

meters in urban areas will have most likely a cellular or WiMAX radio.

Smart Grid is built on a distributed and heterogeneous network and can use a combination of diverse networking technologies, such as wireless Access Technologies (WiMAX, Cellular, etc.), wireline and Internet Technologies (e.g., IP/MPLS, Ethernet, SDH/PDH over Fiber optic, etc.) as well as low-power radio technologies enabling the networking of smart meters, home appliances, and constrained devices (e.g. BT-LE, ZigBee, Z-Wave, Wi-Fi, etc.). The operational effectiveness of the smart grid is highly dependent on a robust, two-way, secure, and reliable communications network with suitable availability.

The management of a distributed system like smart grid requires an end-to-end management of and information exchange through different type of networks. However, as of today there is no integrated smart grid management approach and no common smart grid information model available. Specific smart grid applications or network islands use their own management mechanisms. For example, the management of smart meters depends very much on the AMI environment they have been integrated to and the networking technologies they are using. In general, smart meters do only need seldom reconfiguration and they send a small amount of redundant data to a central entity. For a discussion on the management needs of an AMI network see Section 2.11. The management needs for Smart Home and Building Automation are discussed in Section 2.4 and Section 2.5.

## 2.7. Transport Applications

Transport Application is a generic term for the integrated application of communications, control, and information processing in a transportation system. Transport telematics or vehicle telematics are used as a term for the group of technologies that support transportation systems. Transport applications running on such a transportation system cover all modes of the transport and consider all elements of the transportation system, i.e. the vehicle, the infrastructure, and the driver or user, interacting together dynamically. The overall aim is to improve decision making, often in real time, by transport network controllers and other users, thereby improving the operation of the entire transport system. As such, transport applications can be seen as one of the important M2M service scenarios with the involvement of manifold small devices.

The definition encompasses a broad array of techniques and approaches that may be achieved through stand-alone technological applications or as enhancements to other transportation communication schemes. Examples for transport applications are inter and intra vehicular

communication, smart traffic control, smart parking, electronic toll collection systems, logistic and fleet management, vehicle control, and safety and road assistance.

As a distributed system, transport applications require an end-to-end management of different types of networks. It is likely that constrained devices in a network (e.g. a moving in-car network) have to be controlled by an application running on an application server in the network of a service provider. Such a highly distributed network including mobile devices on vehicles is assumed to include a wireless access network using diverse long distance wireless technologies such as WiMAX, 3G/LTE or satellite communication, e.g. based on an embedded hardware module. As a result, the management of constrained devices in the transport system might be necessary to plan top-down and might need to use data models obliged from and defined on the application layer. The assumed device classes in use are mainly C2 devices. In cases, where an in-vehicle network is involved, C1 devices with limited capabilities and a short-distance constrained radio network, e.g. IEEE 802.15.4 might be used additionally.

Management responsibility typically rests within the organization running the transport application. The constrained devices in a moving transport network might be initially configured in a factory and a reconfiguration might be needed only rarely. New devices might be integrated in an ad-hoc manner based on self-management and -configuration capabilities. Monitoring and data exchange might be necessary to do via a gateway entity connected to the back-end transport infrastructure. The devices and entities in the transport infrastructure need to be monitored more frequently and can be able to communicate with a higher data rate. The connectivity of such entities does not necessarily need to be wireless. The time scale for detecting and recording failures in a moving transport network is likely measured in hours and repairs might easily take days. It is likely that a self-healing feature would be used locally.

## 2.8. Infrastructure Monitoring

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

The infrastructure to monitor might be in a factory or spread over a wider area but difficult to access. As such, the network in use

might be based on a combination of fixed and wireless technologies, which use robust networking equipment and support reliable communication. It is likely that constrained devices in such a network are mainly C2 devices and have to be controlled centrally by an application running on a server. In case such a distributed network is widely spread, the wireless devices might use diverse long-distance wireless technologies such as WiMAX, or 3G/LTE, e.g. based on embedded hardware modules. In cases, where an in-building network is involved, the network can be based on Ethernet or wireless technologies suitable for in-building usage.

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

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

## 2.9. Community Network Applications

Community networks are comprised of constrained routers in a multi-hop mesh topology, communicating over a lossy, and often wireless channel. While the routers are mostly non-mobile, the topology may be very dynamic because of fluctuations in link quality of the (wireless) channel caused by, e.g., obstacles, or other nearby radio transmissions. Depending on the routers that are used in the community network, the resources of the routers (memory, CPU) may be more or less constrained - available resources may range from only a few kilobytes of RAM to several megabytes or more, and CPUs may be small and embedded, or more powerful general-purpose processors. Examples of such community networks are the FunkFeuer network (Vienna, Austria), FreiFunk (Berlin, Germany), Seattle Wireless (Seattle, USA), and AWMN (Athens, Greece). These community networks are public and non-regulated, allowing their users to connect to each other and - through an uplink to an ISP - to the Internet. No fee, other than the initial purchase of a wireless router, is charged for

these services. Applications of these community networks can be diverse, e.g., location based services, free Internet access, file sharing between users, distributed chat services, social networking etc, video sharing etc.

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

While the routers are non-mobile, fluctuations in link quality require an ad hoc routing protocol that allows for quick convergence to reflect the effective topology of the network (such as NHDP [RFC6130] and OLSRV2 [I-D.ietf-manet-olsrv2] developed in the MANET WG). Usually, no human interaction is required for these protocols, as all variable parameters required by the routing protocol are either negotiated in the control traffic exchange, or are only of local importance to each router (i.e. do not influence interoperability). However, external management and monitoring of an ad hoc routing protocol may be desirable to optimize parameters of the routing protocol. Such an optimization may lead to a more stable perceived topology and to a lower control traffic overhead, and therefore to a higher delivery success ratio of data packets, a lower end-to-end delay, and less unnecessary bandwidth and energy usage.

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

- o One single Network Management Station (NMS), e.g. a border gateway providing connectivity to the Internet, requires managing or monitoring routers in the community network, in order to investigate problems (monitoring) or to improve performance by changing parameters (managing). As the topology of the network is dynamic, constant connectivity of each router towards the management station cannot be guaranteed. Current network management protocols, such as SNMP and Netconf, may be used (e.g., using interfaces such as the NHDP-MIB [RFC6779]). However, when routers in the community network are constrained, existing protocols may require too many resources in terms of memory and CPU; and more importantly, the bandwidth requirements may exceed



the available channel capacity in wireless mesh networks. Moreover, management and monitoring may be unfeasible if the connection between the NMS and the routers is frequently interrupted.

- o A distributed network monitoring, in which more than one management station monitors or manages other routers. Because connectivity to a server cannot be guaranteed at all times, a distributed approach may provide a higher reliability, at the cost of increased complexity. Currently, no IETF standard exists for distributed monitoring and management.
- o Monitoring and management of a whole network or a group of routers. Monitoring the performance of a community network may require more information than what can be acquired from a single router using a network management protocol. Statistics, such as topology changes over time, data throughput along certain routing paths, congestion etc., are of interest for a group of routers (or the routing domain) as a whole. As of 2012, no IETF standard allows for monitoring or managing whole networks, instead of single routers.

#### 2.10. Mobile Applications

M2M services are increasingly provided by mobile service providers as numerous devices, home appliances, utility meters, cars, video surveillance cameras, and health monitors, are connected with mobile broadband technologies. This diverse range of machines brings new network and service requirements and challenges. Different applications e.g. in a home appliance or in-car network use Bluetooth, Wi-Fi or Zigbee and connect to a cellular module acting as a gateway between the constrained environment and the mobile cellular network.

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

- o a smart phone with 3G/4G and WLAN radio might use BT-LE to connect to the devices in a home area network,
- o a femtocell might be combined with home gateway functionality acting as a low-power cellular base station connecting smart devices to the application server of a mobile service provider.
- o an embedded cellular module with LTE radio connecting the devices in the car network with the server running the telematics service,

- o an M2M gateway connected to the mobile operator network supporting diverse IoT connectivity technologies including ZigBee and CoAP over 6LoWPAN over IEEE 802.15.4.

Common to all scenarios above is that they are embedded in a service and connected to a network provided by a mobile service provider. Usually there is a hierarchical deployment and management topology in place where different parts of the network are managed by different management entities and the count of devices to manage is high (e.g. many thousands). In general, the network is comprised by manifold type and size of devices matching to different device classes. As such, the managing entity needs to be prepared to manage devices with diverse capabilities using different communication or management protocols. In case the devices are directly connected to a gateway they most likely are managed by a management entity integrated with the gateway, which itself is part of the Network Management System (NMS) run by the mobile operator. Smart phones or embedded modules connected to a gateway might be themselves in charge to manage the devices on their level. The initial and subsequent configuration of such a device is mainly based on self-configuration and is triggered by the device itself.

The challenges in the management of devices in a mobile application are manifold. Firstly, the issues caused through the device mobility need to be taken into consideration. While the cellular devices are moving around or roaming between different regional networks, they should report their status to the corresponding management entities with regard to their proximity and management hierarchy. Secondly, a variety of device troubleshooting information needs to be reported to the management system in order to provide accurate service to the customer. Third but not least, the NMS and the used management protocol need to be tailored to keep the cellular devices lightweight and as energy efficient as possible.

The data models used in these scenario are mostly derived from the models of the operator NMS and might be used to monitor the status of the devices and to exchange the data sent by or read from the devices. The gateway might be in charge of filtering and aggregating the data received from the device as the information sent by the device might be mostly redundant.

#### 2.11. Automated Metering Infrastructure (AMI)

An AMI network enables an electric utility to retrieve frequent electric usage data from each electric meter installed at a customer's home or business. With an AMI network, a utility can also receive immediate notification of power outages when they occur, directly from the electric meters that are experiencing those

outages. In addition, if the AMI network is designed to be open and extensible, it could serve as the backbone for communicating with other distribution automation devices besides meters, which could include transformers and reclosers.

In this use case, each meter in the AMI network contains a constrained device. These devices are typically C2 devices. Each meter connects to a constrained mesh network with a low-bandwidth radio. These radios can be 50, 150, or 200 kbps at raw link speed, but actual network throughput may be significantly lower due to forward error correction, multihop delays, MAC delays, lossy links, and protocol overhead.

The constrained devices are used to connect the metering logic with the network, so that usage data and outage notifications can be sent back to the utility's headend systems over the network. These headend systems are located in a data center managed by the utility, and may include meter data collection systems, meter data management systems, and outage management systems.

The meters are connected to a mesh network, and each meter can act as both a source of traffic and as a router for other meters' traffic. In a typical AMI application, smaller amounts of traffic (read requests, configuration) flow "downstream" from the headend to the mesh, and larger amounts of traffic flow "upstream" from the mesh to the headend. However, during a firmware update operation, larger amounts of traffic might flow downstream while smaller amounts flow upstream. Other applications that make use of the AMI network may have their own distinct traffic flows.

The mesh network is anchored by a collection of higher-end devices, which contain a mesh radio that connects to the constrained network as well as a backhaul link that connects to a less-constrained network. The backhaul link could be cellular, WiMAX, or Ethernet, depending on the backhaul networking technology that the utility has chosen. These higher-end devices (termed "routers" in this use case) are typically installed on utility poles throughout the service territory. Router devices are typically less constrained than meters, and often contain the full routing table for all the endpoints routing through them.

In this use case, the utility typically installs on the order of 1000 meters per router. The collection of meters comprised in a local network that are routing through a specific router is called in this use case a Local Meter Network (LMN). When powered on, each meter is designed to discover the nearby LMNs, select the optimal LMN to join, and select the optimal meters in that LMN to route through when sending data to the headend. After joining the LMN, the meter is

designed to continuously monitor and optimize its connection to the LMN, and it may change routes and LMNs as needed.

Each LMN may be configured e.g. to share an encryption key, providing confidentiality for all data traffic within the LMN. This key may be obtained by a meter only after an end-to-end authentication process based on certificates, ensuring that only authorized and authenticated meters are allowed to join the LMN, and by extension, the mesh network as a whole.

After joining the LMN, each endpoint obtains a routable and possibly private IPv6 address that enables end-to-end communication between the headend systems and each meter. In this use case, the meters are always-on. However, due to lossy links and network optimization, not every meter will be immediately accessible, though eventually every meter will be able to exchange data with the headend.

In a large AMI deployment, there may be 10 million meters supported by 10,000 routers, spread across a very large geographic area. Within a single LMN, the meters may range between 1 and approx. 20 hops from the router. During the deployment process, these meters are installed and turned on in large batches, and those meters must be authenticated, given addresses, and provisioned with any configuration information necessary for their operation. During deployment and after deployment is finished, the network must be monitored continuously and failures must be handled. Configuration parameters may need to be changed on large numbers of devices, but most of the devices will be running the same configuration. Moreover, eventually, the firmware in those meters will need to be upgraded, and this must also be done in large batches because most of the devices will be running the same firmware image.

Because there may be thousands of routers, this operational model (batch deployment, automatic provisioning, continuous monitoring, batch reconfiguration, batch firmware update) should also apply to the routers as well as the constrained devices. The scale is different (thousands instead of millions) but still large enough to make individual management impractical for routers as well.

## 2.12. MANET Concept of Operations (CONOPS) in Military

The use case on the Concept of Operations (CONOPS) focuses on the configuration and monitoring of networks that are currently being used in military and as such, it offers insights and challenges of network management that military agencies are facing.

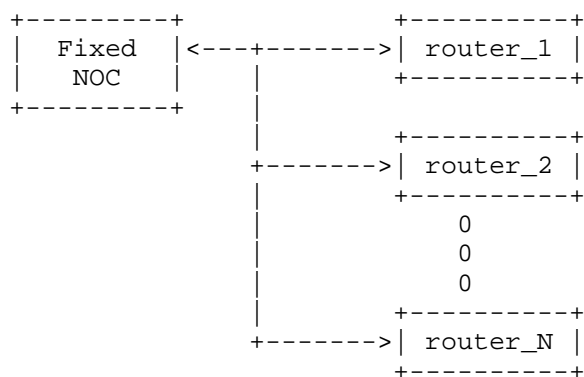
As technology advances, military networks nowadays become large and consist of varieties of different types of equipments that run

different protocols and tools that obviously increase complexity of the tactical networks. Moreover, lacks of open common interfaces and Application Programming Interface (API) are often a challenge to network management. Configurations are, most likely, manually performed. Some devices do not support IP networks. Integration and evaluation process are no longer trivial for a large set of protocols and tools. In addition, majority of protocols and tools developed by vendors that are being used are proprietary which makes integration more difficult. The main reason that leads to this problem is that there is no clearly defined standard for the MANET Concept of Operations (CONOPS). In the following, a set of scenarios of network operations are described, which might lead to the development of network management protocols and a framework that can potentially be used in military networks.

Note: The term "node" is used at IETF for either a host or router. The term "unit" or "mobile unit" in military (e.g. Humvees, tanks) is a unit that contains multiple routers, hosts, and/or other non-IP-based communication devices.

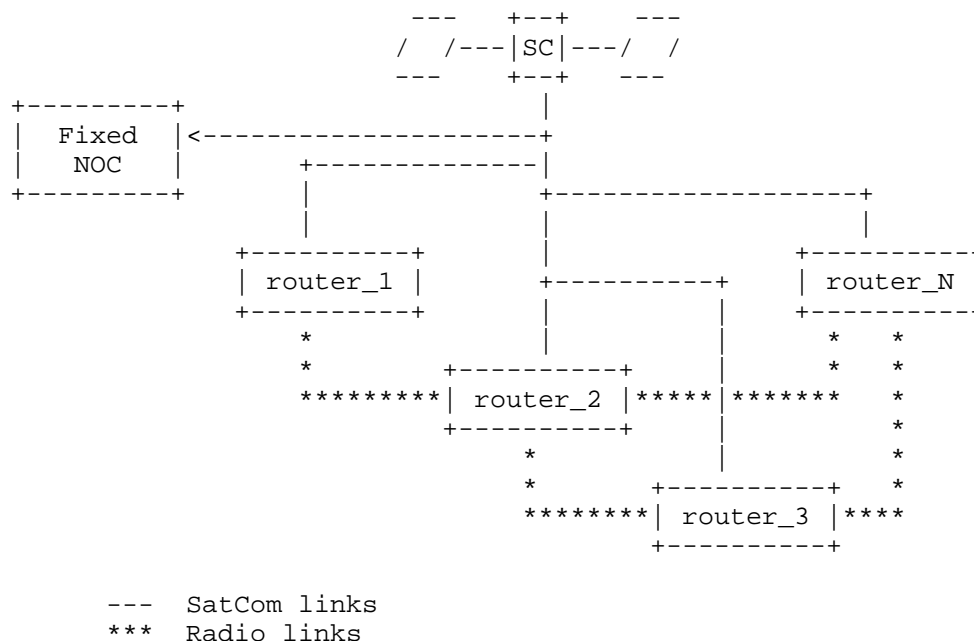
#### Scenario: Parking Lot Staging Area:

The Parking Lot Staging Area is the most common network operation that is currently widely used in military prior to deployment. MANET routers, which can be identical such as the platoon leader's or rifleman's radio, are shipped to a remote location along with a Fixed Network Operations Center (NOC), where they are all connected over traditional wired or wireless networks. The Fixed NOC then performs mass-configuration and evaluation of configuration processes. The same concept can be applied to mobile units. Once all units are successfully configured, they are ready to be deployed.



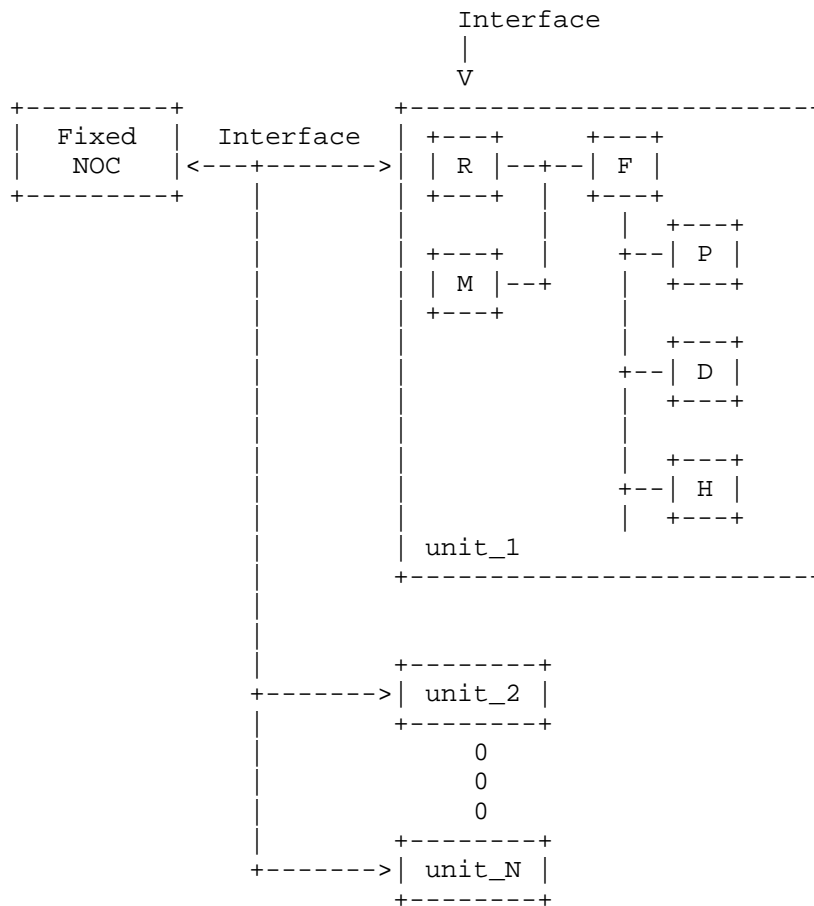
Scenario: Monitoring with SatCom Reachback:

The Monitoring with SatCom Reachback, which is considered another possible common scenario to military's network operations, is similar to the Parking Lot Staging Area. Instead, the Fixed NOC and MANET routers are connected through a Satellite Communications (SatCom) network. The Monitoring with SatCom Reachback is a scenario where MANET routers are augmented with SatCom Reachback capabilities while On-The-Move (OTM). Vehicles carrying MANET routers support multiple types of wireless interfaces, including High Capacity Short Range Radio interfaces as well as Low Capacity OTM SatCom interfaces. The radio interfaces are the preferred interfaces for carrying data traffic due to their high capacity, but the range is limiting with respect to connectivity to a Fixed NOC. Hence, OTM SatCom interfaces offer a more persistent but lower capacity reachback capability. The existence of a SatCom persistent Reachback capability offers the NOC the ability to monitor and manage the MANET routers over the air. Similarly to the Parking Lot Staging Area, the same concept can be applied to mobile units.



#### Scenario: Hierarchical Management:

Another reasonable scenario common to military operations in a MANET environment is the Hierarchical Management scenario. Vehicles carry a rather complex set of networking devices, including routers running MANET control protocols. In this hierarchical architecture, the MANET mobile unit has a rather complex internal architecture where a local manager within the unit is responsible for local management. The local management includes management of the MANET router and control protocols, the firewall, servers, proxies, hosts and applications. In addition, a standard management interface is required in this architecture. Moreover, in addition to requiring standard management interfaces into the components comprising the MANET nodal architecture, the local manager is responsible for local monitoring and the generation of periodic reports back to the Fixed NOC.



Key: R-Router  
 F-Firewall  
 P-PEP (Performance Enhancing Proxy)  
 D-Servers, e.g., DNS  
 H-hosts  
 M-Local Manager

Figure 3: Hierarchical Management

Scenario: Management over Lossy/Intermittent Links:

In the future of military operations, the standard management will be done over lossy and intermittent links and ideally the Fixed NOC will become mobile. In this architecture, the nature and current quality



of each link are distinct. However, there are a number of issues that would arise and need to be addressed:

1. Common and specific configurations are undefined:
  - A. When mass-configuring devices, common set of configurations are undefined at this time.
  - B. Similarly, when performing a specific device, set of specific configurations is unknown.
2. Once the total number of units becomes quite large, scalability would be an issue and need to be addressed.
3. The state of the devices are different and may be in various states of operations, e.g., ON/OFF, etc.
4. Pushing large data files over reliable transport, e.g., TCP, would be problematic. Would a new mechanism of transmitting large configurations over the air in low bandwidth be implemented? Which protocol would be used at transport layer?
5. How to validate network configuration (and local configuration) is complex, even when to cutover is an interesting question.
6. Security as a general issue needs to be addressed as it could be problematic in military operations.

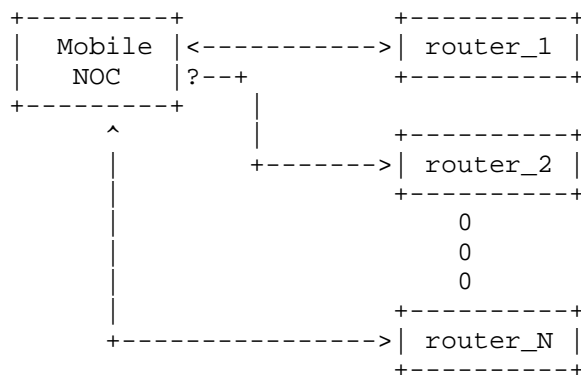


Figure 4: Management over Lossy/intermittent Links

### 3. IANA Considerations

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

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

#### 4. Security Considerations

This document discusses the use cases for a network of constrained devices and does not introduce any security issues by itself.

## 5. Contributors

Following persons made significant contributions to and reviewed this document:

- o Ulrich Herberg (Fujitsu Laboratories of America) contributed the Section 2.9 on Community Network Applications.
- o Peter van der Stok contributed to Section 2.5 on Building Automation.
- o Zhen Cao contributed to Section 2.10 on Mobile Applications.
- o Gilman Tolle contributed the Section 2.11 on Automated Metering Infrastructure.
- o James Nguyen and Ulrich Herberg contributed the Section 2.12 on MANET Concept of Operations (CONOPS) in Military.

## 6. Acknowledgments

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

Dominique Barthel, Carsten Bormann, Zhen Cao, Benoit Claise, Bert Greevenbosch, Ulrich Herberg, James Nguyen, Anuj Sehgal, Zach Shelby, and Peter van der Stok.

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

## 7. References

### 7.1. Normative References

### 7.2. Informative References

- [RFC6130] Clausen, T., Dearlove, C., and J. Dean, "Mobile Ad Hoc Network (MANET) Neighborhood Discovery Protocol (NHDP)", RFC 6130, April 2011.
- [RFC6779] Herberg, U., Cole, R., and I. Chakeres, "Definition of Managed Objects for the Neighborhood Discovery Protocol", RFC 6779, October 2012.
- [RFC6988] Quittek, J., Chandramouli, M., Winter, R., Dietz, T., and B. Claise, "Requirements for Energy Management", RFC 6988, September 2013.
- [I-D.ietf-lwig-terminology] Bormann, C., Ersue, M., and A. Keranen, "Terminology for Constrained Node Networks", draft-ietf-lwig-terminology-05 (work in progress), July 2013.
- [I-D.ietf-eman-framework] Parello, J., Claise, B., Schoening, B., and J. Quittek, "Energy Management Framework", draft-ietf-eman-framework-11 (work in progress), October 2013.
- [I-D.ietf-manet-olsrv2] Clausen, T., Dearlove, C., Jacquet, P., and U. Herberg, "The Optimized Link State Routing Protocol version 2", draft-ietf-manet-olsrv2-19 (work in progress), March 2013.
- [COM-REQ] Ersue, M., "Constrained Management: Problem statement and Requirements", draft-ersue-coman-prostate-reqs (work in progress), October 2013.

Appendix A. Open issues

- o It has been noted that the use cases the Industrial Application, Home Automation and Building Automation have an intersect.

## Appendix B. Change Log

B.1. draft-ersue-constrained-mgmt-03 -  
draft-ersue-opsawg-coman-use-cases-00

- o Reduced the terminology section for terminology addressed in the LWIG and Coman Requirements drafts. Referenced the other drafts.
- o Checked and aligned all terminology against the LWIG terminology draft.
- o Spent some effort to resolve the intersection between the Industrial Application, Home Automation and Building Automation use cases.
- o Moved section section 3. Use Cases from the companion document [COM-REQ] to this draft.
- o Reformulation of some text parts for more clarity.

## B.2. draft-ersue-constrained-mgmt-02-03

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



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

#### B.3. draft-ersue-constrained-mgmt-01-02

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

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

B.4. draft-ersue-constrained-mgmt-00-01

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

Authors' Addresses

Mehmet Ersue (editor)  
Nokia Solutions and Networks

Email: mehmet.ersue@nsn.com

Dan Romascanu  
Avaya

Email: dromasca@avaya.com

Juergen Schoenwaelder  
Jacobs University Bremen

Email: j.schoenwaelder@jacobs-university.de



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

Y. Chen  
China Mobile  
D. Liu

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

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

Abstract

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

Status of This Memo

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

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

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

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

Copyright Notice

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

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

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

## Table of Contents

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

## 1. Introduction

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

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

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

## 2. Terminology

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

This document uses the following abbreviations:

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

## 3. CAPWAP 802.11n Support

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

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

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

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

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

### 3.1. CAPWAP Extension for 802.11n Support

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

#### 3.1.1. 802.11n Radio Capability Information

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

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

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

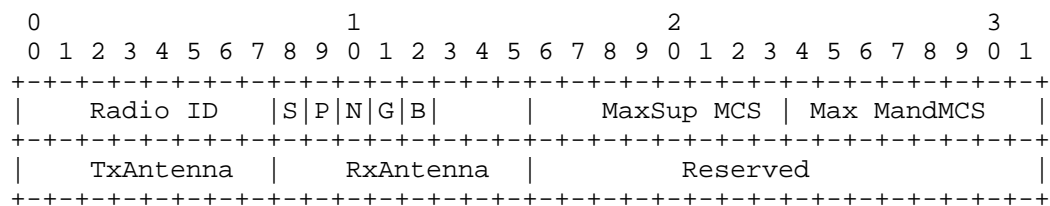


Figure 1: 802.11n Radio Configuration Message Element

Type: TBD1 for 802.11n Radio Configuration Message Element.

Length: 16.



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

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

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

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

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

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

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

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

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

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

The detail definition of TxAntenna/RxAntenna is as follows:

```

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

```

Figure 2: Definition of TxAntenna/RxAntenna

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

### 3.1.3. 802.11n Station Information

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

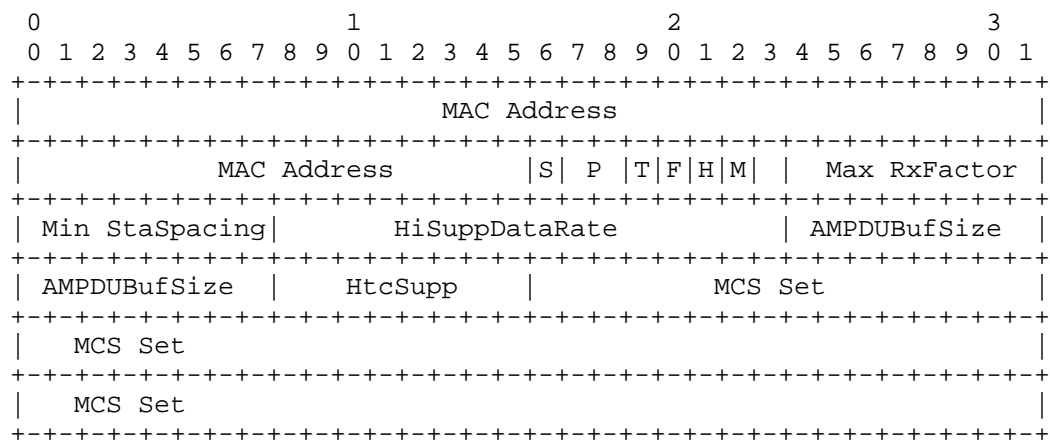


Figure 3: 802.11n Station Information

MAC Address: The station's MAC Address.

Type: TBD2 for 802.11 Station Information.

Length: 24.

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

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

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

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

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

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

Max RxFactor: The maximal receiving A-MPDU factor.

Min StaSpacing: Minimum MPDU Start Spacing.

HiSuppDataRate: Maximal transmission speed (Mbps).

AMPDUBufSize: A-MPDU buffer size (Byte).

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

MCS Set: The MCS bitmap that the station supports.

#### 4. Power and Channel Autoconfiguration

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

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

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

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

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

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

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

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

#### 4.3. Channel/Power Auto Adjustment

The Channel Scan Procedure is illustrated by the figure 4.

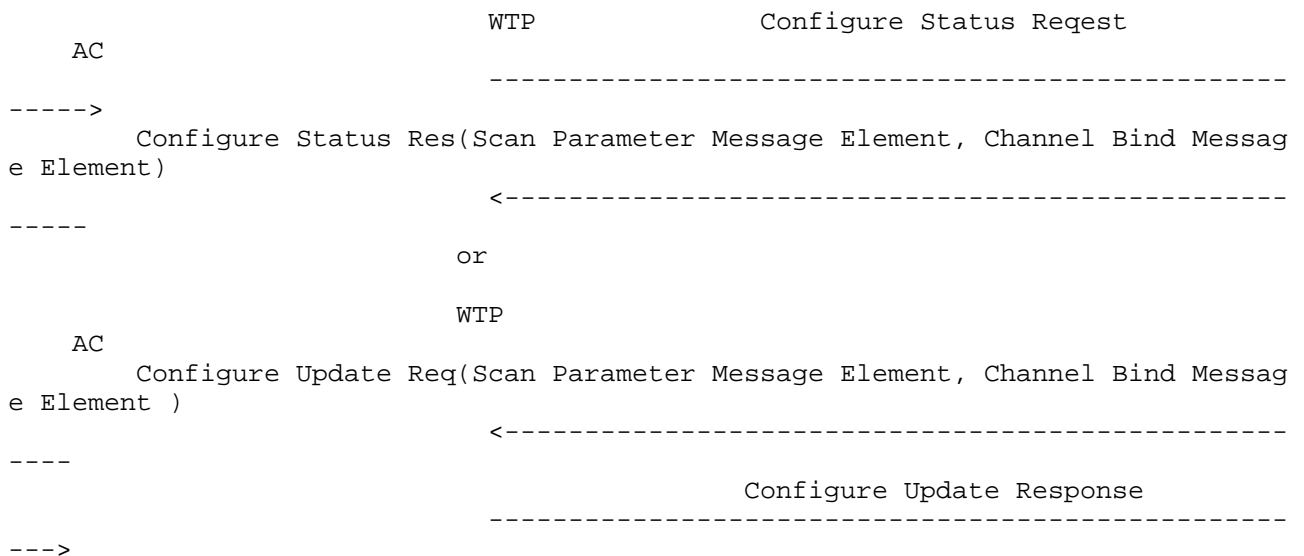


Figure 4: Channel Scan Procedure

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

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

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

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

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

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

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

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

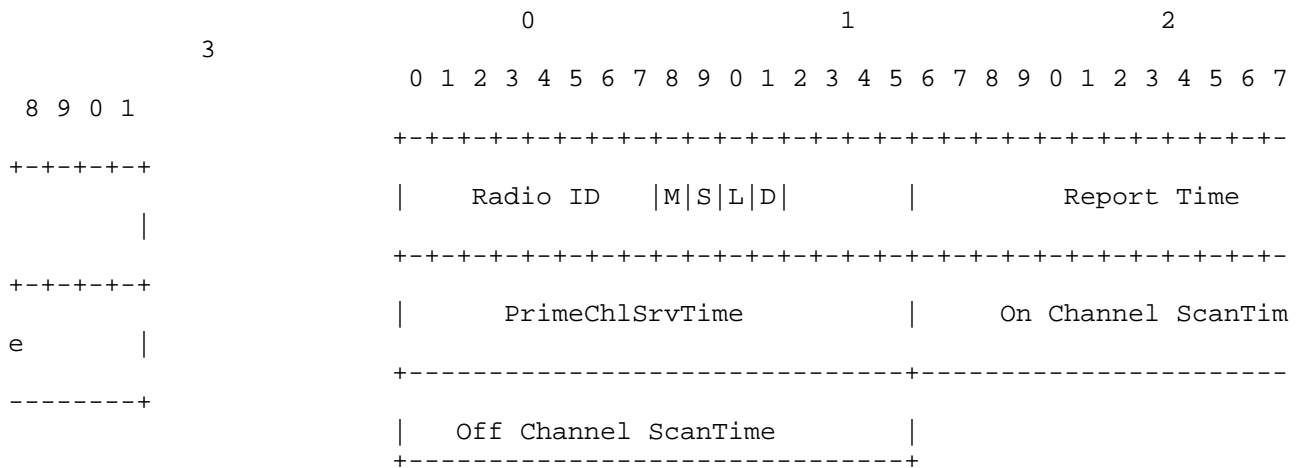


Figure 5: IEEE 802.11 Scan Parameters Message Element

Type: TBD3 for IEEE 802.11 Scan Parameters Message Element.

Length: 10.

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

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

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

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

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

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

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

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

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

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

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

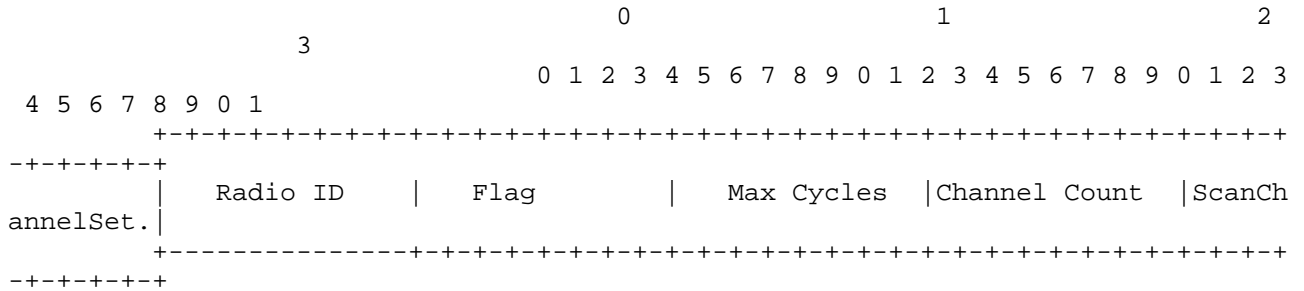


Figure 6: IEEE 802.11 Scan Channel Bind Message Element

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

Length: variable.

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

Flag: reserved.

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

Channel Count: The number of channels will be scanned.

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

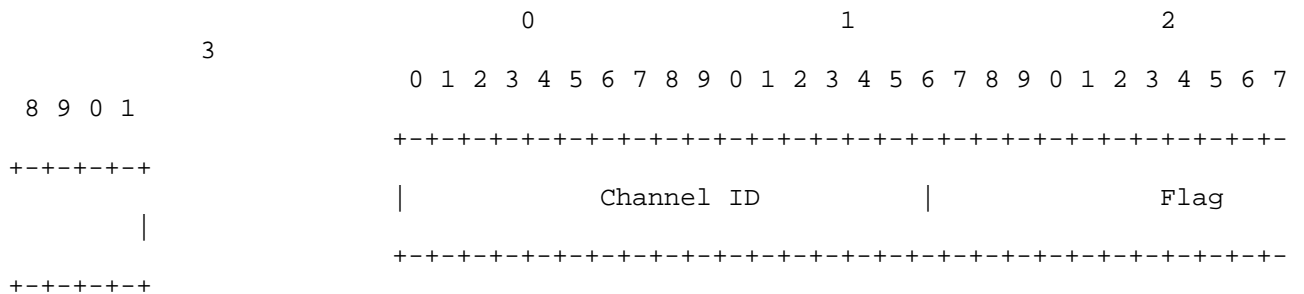


Figure 7: Channel Information Format

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

Flag: Bitmap, reserved for future use.

#### 4.3.3. IEEE 802.11 Channel Scan Report

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

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

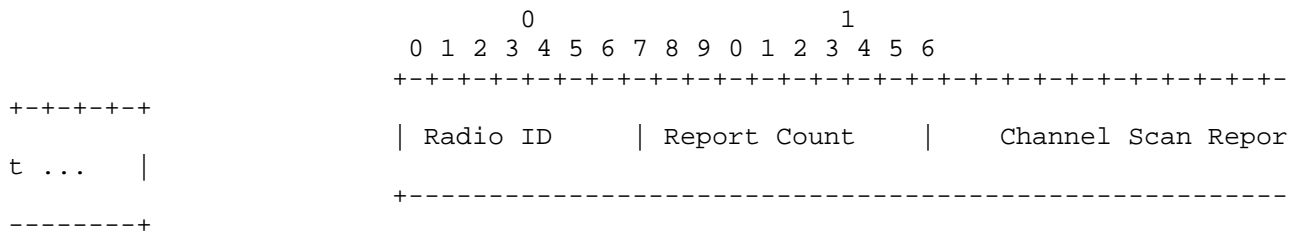


Figure 8: IEEE 802.11 Channel Scan Report Message Element

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

Length: >=29.

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

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

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



			0										1										2																			
3			0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0									
1			+-----+																																							
+--+			Channel Number										Radar Statistics										Mean																			
			+-----+																																							
+--+			Time										Mean RSSI										Screen Packet Count																			
			+-----+																																							
+--+			NeighborCount										Mean Noise										Interference										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

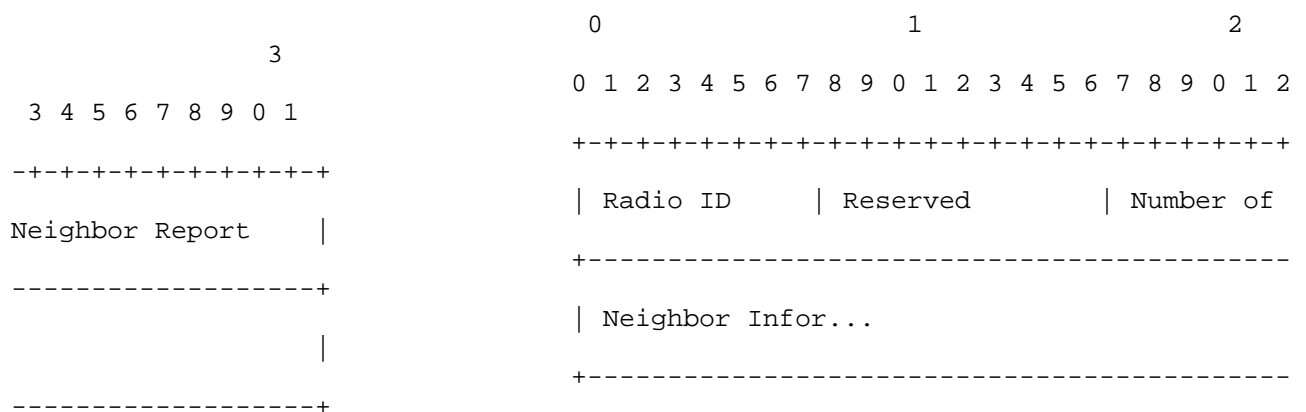


Figure 10: WTP Neighbor Report TLV

The definition of Neighbor info is as follows:

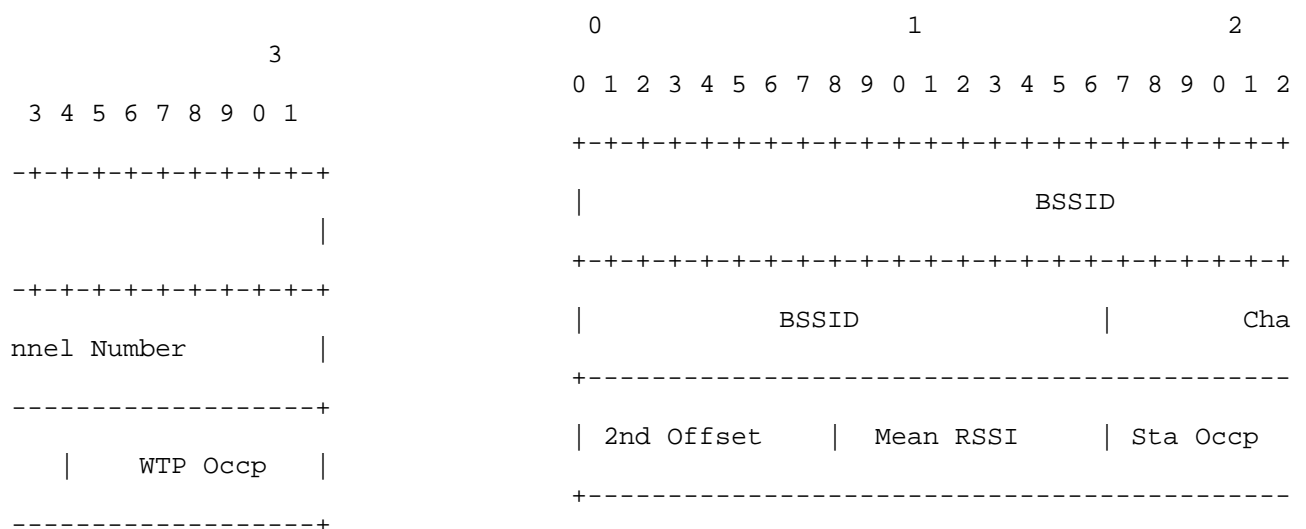


Figure 11: Neighbor info

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 Bei qing Road, Haidian District  
Beijing 100095  
China

Email: lei.zhu@huawei.com

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

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

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

Abstract

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

Status of This Memo

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

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

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



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

#### Copyright Notice

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

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

#### Table of Contents

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

#### 1. Introduction

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

experience interoperability issues with WTPs and ACs from different vendors.

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

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

Figure 1: Functions in Local MAC and Split MAC

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

## 2. IEEE MAC Profile Descriptions

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

### 2.1. Split MAC with WTP encryption

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

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

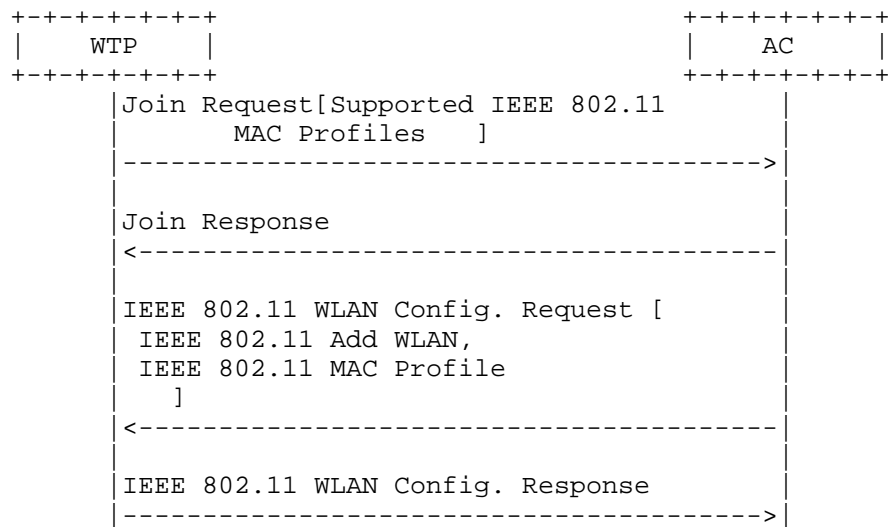


Figure 4: Message Exchange For Negotiating MAC Profile

### 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.

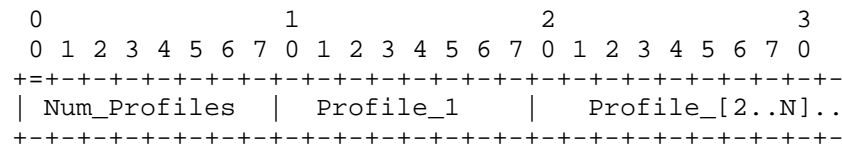


Figure 5: IEEE 802.11 Supported MAC Profiles

- 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).

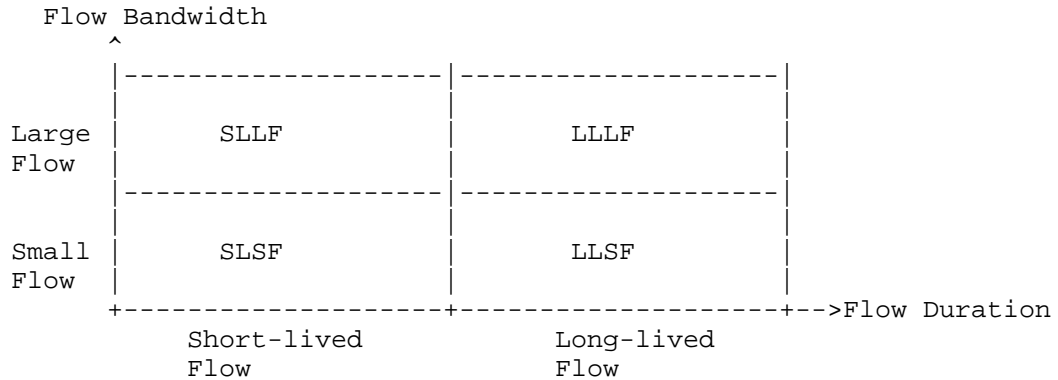


Figure 1: Flow Categorization

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".

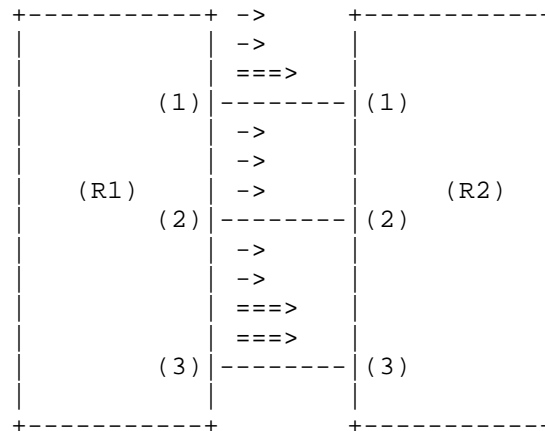
### 3. Hash-based Load Distribution in LAG/ECMP

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.)

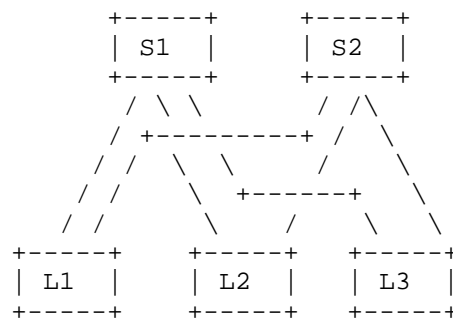


Figure 3: Two-level Clos Network

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

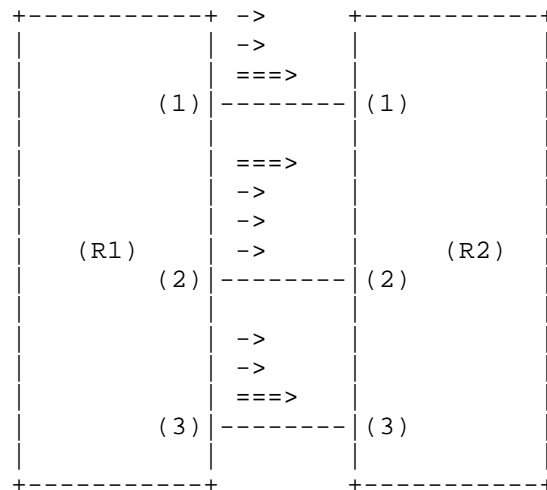
Specific algorithms for placement of large flows are out of scope of this document. One possibility is to formulate the problem for large flow placement as the well-known bin-packing problem and make use of the various heuristics that are available for that problem [bin-pack].



#### 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.

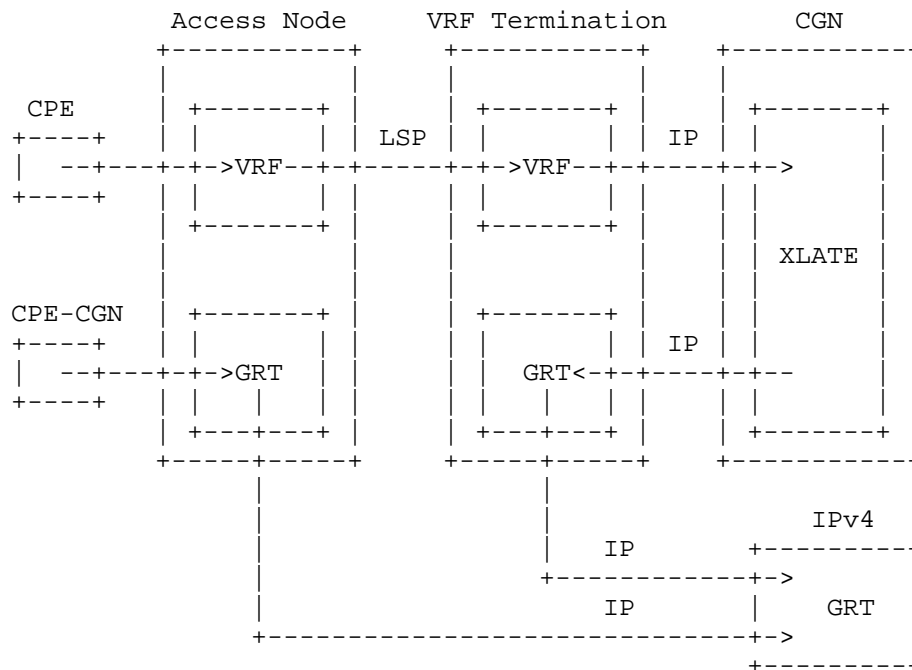


Figure 1: Basic BGP/MPLS IP VPN CGN Model

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.



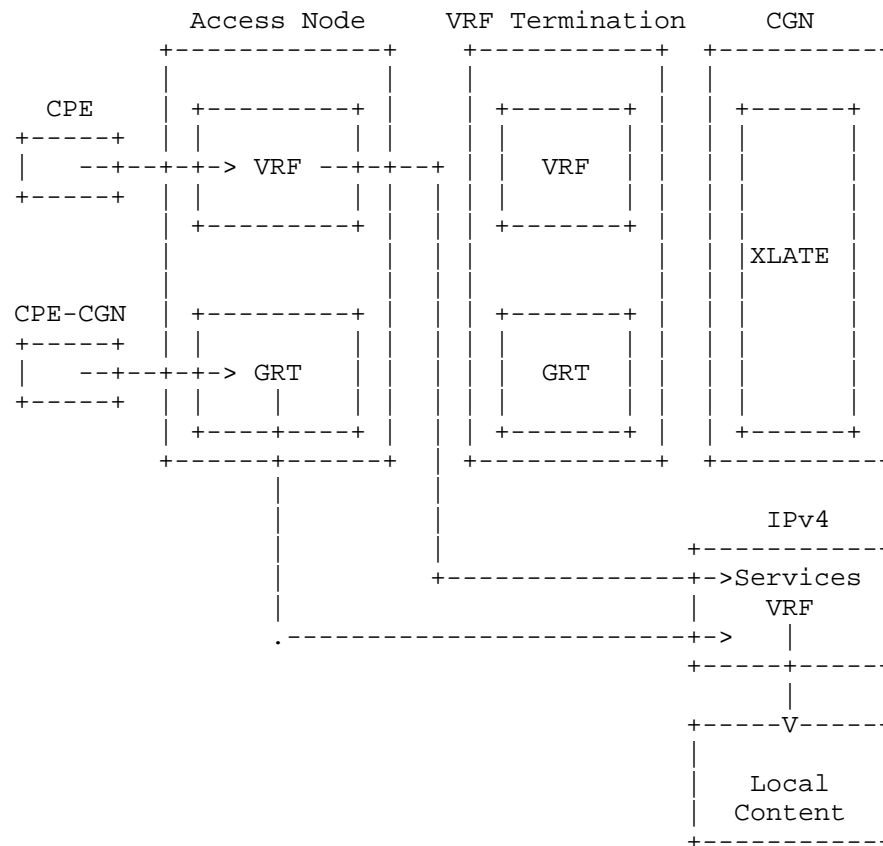
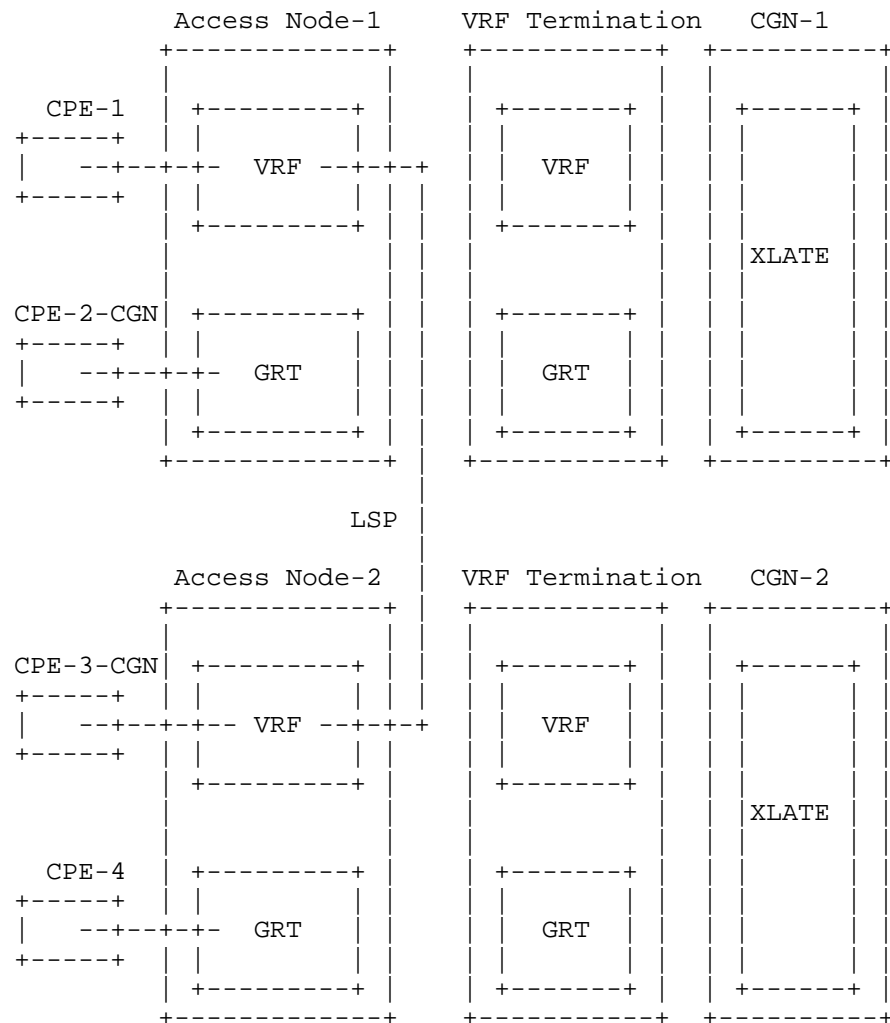


Figure 2: Internal Services and CGN By-Pass

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.

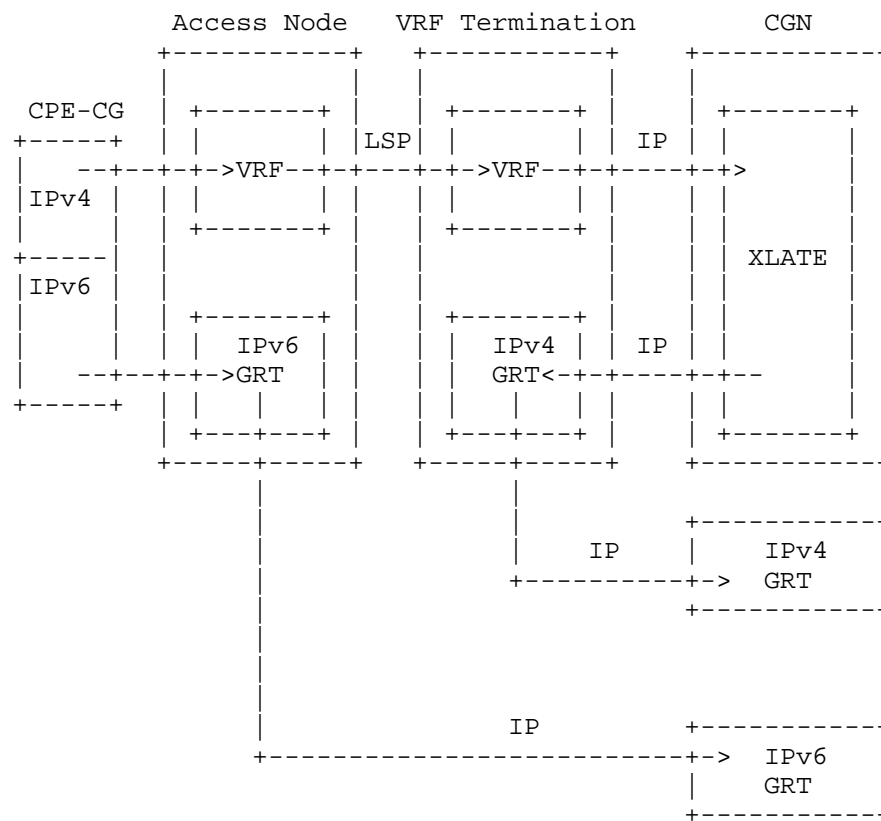


Figure 4: CGN with IPv6 Dual Stack Operation

#### 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>

Operations and Management Area Working Group  
Internet Draft  
Intended status: Informational  
Expires: September 2014

T. Mizrahi  
Marvell  
N. Sprecher  
Nokia Solutions and Networks  
E. Bellagamba  
Ericsson  
Y. Weingarten

March 28, 2014

An Overview of  
Operations, Administration, and Maintenance (OAM) Tools  
draft-ietf-opsawg-oam-overview-16.txt

## Abstract

Operations, Administration, and Maintenance (OAM) is a general term that refers to a toolset for fault detection and isolation, and for performance measurement. Over the years various OAM tools have been defined for various layers in the protocol stack.

This document summarizes some of the OAM tools defined in the IETF in the context of IP unicast, MPLS, MPLS Transport Profile (MPLS-TP), pseudowires, and TRILL. This document focuses on tools for detecting and isolating failures in networks and for performance monitoring. Control and management aspects of OAM are outside the scope of this document. Network repair functions such as Fast Reroute (FRR) and protection switching, which are often triggered by OAM protocols, are also out of the scope of this document.

The target audience of this document includes network equipment vendors, network operators and standards development organizations, and can be used as an index to some of the main OAM tools defined in the IETF. This document provides a brief description of each of the OAM tools in the IETF. At the end of the document a list of the OAM toolsets and a list of the OAM functions are presented as a summary.

## Status of this Memo

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

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 September 28, 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 .....	4
1.1. Background .....	4
1.2. Target Audience.....	5
1.3. OAM-related Work in the IETF .....	6
1.4. Focusing on the Data Plane .....	7
2. Terminology .....	7
2.1. Abbreviations .....	7
2.2. Terminology used in OAM Standards .....	9
2.2.1. General Terms .....	9
2.2.2. Operations, Administration and Maintenance .....	9
2.2.3. Functions, Tools and Protocols .....	10
2.2.4. Data Plane, Control Plane and Management Plane ....	11
2.2.5. The Players .....	12
2.2.6. Proactive and On-demand Activation .....	12
2.2.7. Connectivity Verification and Continuity Checks ...	13
2.2.8. Connection Oriented vs. Connectionless Communication	14
2.2.9. Point-to-point vs. Point-to-multipoint Services ...	14

2.2.10. Failures .....	15
3. OAM Functions .....	16
4. OAM Tools in the IETF - a Detailed Description .....	16
4.1. IP Ping .....	17
4.2. IP Traceroute .....	17
4.3. Bidirectional Forwarding Detection (BFD) .....	18
4.3.1. Overview .....	18
4.3.2. Terminology .....	19
4.3.3. BFD Control .....	19
4.3.4. BFD Echo .....	19
4.4. MPLS OAM .....	20
4.4.1. LSP Ping .....	20
4.4.2. BFD for MPLS .....	21
4.4.3. OAM for Virtual Private Networks (VPN) over MPLS ..	21
4.5. MPLS-TP OAM .....	21
4.5.1. Overview .....	21
4.5.2. Terminology .....	22
4.5.3. Generic Associated Channel .....	24
4.5.4. MPLS-TP OAM Toolset .....	24
4.5.4.1. Continuity Check and Connectivity Verification	25
4.5.4.2. Route Tracing .....	25
4.5.4.3. Lock Instruct .....	25
4.5.4.4. Lock Reporting .....	25
4.5.4.5. Alarm Reporting .....	26
4.5.4.6. Remote Defect Indication .....	26
4.5.4.7. Client Failure Indication .....	26
4.5.4.8. Performance Monitoring .....	26
4.5.4.8.1. Packet Loss Measurement (LM) .....	26
4.5.4.8.2. Packet Delay Measurement (DM) .....	27
4.6. Pseudowire OAM .....	27
4.6.1. Pseudowire OAM using Virtual Circuit Connectivity	
Verification (VCCV) .....	27
4.6.2. Pseudowire OAM using G-ACh .....	29
4.6.3. Attachment Circuit - Pseudowire Mapping .....	29
4.7. OWAMP and TWAMP.....	29
4.7.1. Overview .....	29
4.7.2. Control and Test Protocols .....	30
4.7.3. OWAMP .....	31
4.7.4. TWAMP .....	31
4.8. TRILL .....	32
5. Summary .....	32
5.1. Summary of OAM Tools .....	32
5.2. Summary of OAM Functions .....	35
5.3. Guidance to Network Equipment Vendors .....	36
6. Security Considerations .....	36
7. IANA Considerations .....	37
8. Acknowledgments .....	37

9. References .....	37
9.1. Normative References .....	37
9.2. Informative References .....	37
Appendix A. List of OAM Documents .....	43
A.1. List of IETF OAM Documents .....	43
A.2. List of Selected Non-IETF OAM Documents .....	48

## 1. Introduction

OAM is a general term that refers to a toolset for detecting, isolating and reporting failures and for monitoring the network performance.

There are several different interpretations to the "OAM" acronym. This document refers to Operations, Administration and Maintenance, as recommended in Section 3 of [OAM-Def].

This document summarizes some of the OAM tools defined in the IETF in the context of IP unicast, MPLS, MPLS Transport Profile (MPLS-TP), pseudowires, and TRILL.

This document focuses on tools for detecting and isolating failures and for performance monitoring. Hence, this document focuses on the tools used for monitoring and measuring the data plane; control and management aspects of OAM are outside the scope of this document. Network repair functions such as Fast Reroute (FRR) and protection switching, which are often triggered by OAM protocols, are also out of the scope of this document.

### 1.1. Background

OAM was originally used in traditional communication technologies such as E1 and T1, evolving into PDH and then later in SONET/SDH. ATM was probably the first technology to include inherent OAM support from day one, while in other technologies OAM was typically defined in an ad hoc manner after the technology was already defined and deployed. Packet-based networks were traditionally considered unreliable and best-effort. As packet-based networks evolved, they have become the common transport for both data and telephony, replacing traditional transport protocols. Consequently, packet-based networks were expected to provide a similar "carrier grade" experience, and specifically to support more advanced OAM functions, beyond ICMP and router hellos, that were traditionally used for fault detection.

As typical networks have a multi-layer architecture, the set of OAM protocols similarly take a multi-layer structure; each layer has its

own OAM protocols. Moreover, OAM can be used at different levels of hierarchy in the network to form a multi-layer OAM solution, as shown in the example in Figure 1.

Figure 1 illustrates a network in which IP traffic between two customer edges is transported over an MPLS provider network. MPLS OAM is used at the provider-level for monitoring the connection between the two provider edges, while IP OAM is used at the customer-level for monitoring the end-to-end connection between the two customer edges.

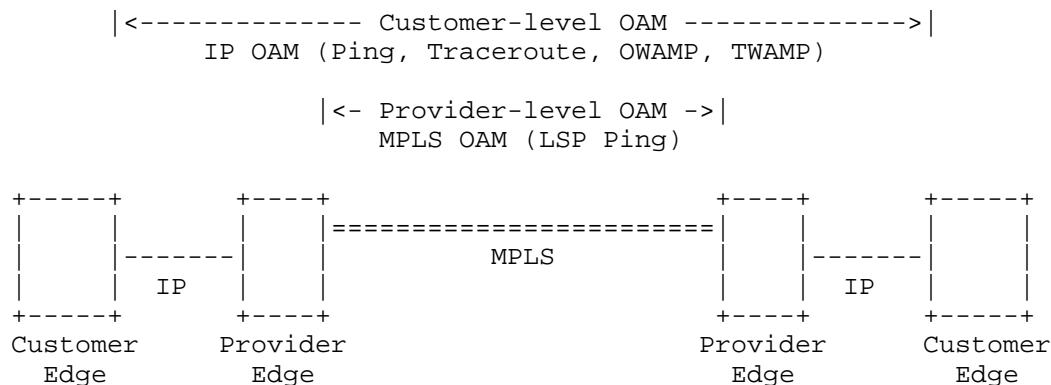


Figure 1 Example: Multi-layer OAM

## 1.2. Target Audience

The target audience of this document includes:

- o Standards development organizations - both IETF working groups and non-IETF organizations can benefit from this document when designing new OAM protocols, or when looking to reuse existing OAM tools for new technologies.
- o Network equipment vendors and network operators - can use this document as an index to some of the common IETF OAM tools.

It should be noted that some background in OAM is necessary in order to understand and benefit from this document. Specifically, the reader is assumed to be familiar with the term OAM [OAM-Def], the motivation for using OAM, and the distinction between OAM and network management [OAM-Mng].

### 1.3. OAM-related Work in the IETF

This memo provides an overview of the different sets of OAM tools defined by the IETF. The set of OAM tools described in this memo are applicable to IP unicast, MPLS, pseudowires, MPLS Transport Profile (MPLS-TP), and TRILL. While OAM tools that are applicable to other technologies exist, they are beyond the scope of this memo.

This document focuses on IETF documents that have been published as RFCs, while other ongoing OAM-related work is outside the scope.

The IETF has defined OAM protocols and tools in several different contexts. We roughly categorize these efforts into a few sets of OAM-related RFCs, listed in Table 1. Each set defines a logically-coupled set of RFCs, although the sets are in some cases intertwined by common tools and protocols.

The discussion in this document is ordered according to these sets (the acronyms and abbreviations are listed in Section 2.1.).

Toolset	Transport Technology
IP Ping	IPv4/IPv6
IP Traceroute	IPv4/IPv6
BFD	generic
MPLS OAM	MPLS
MPLS-TP OAM	MPLS-TP
Pseudowire OAM	Pseudowires
OWAMP and TWAMP	IPv4/IPv6
TRILL OAM	TRILL

Table 1 OAM Toolset Packages in the IETF Documents

This document focuses on OAM tools that have been developed in the IETF. A short summary of some of the significant OAM standards that have been developed in other standard organizations is presented in Appendix A.2.

#### 1.4. Focusing on the Data Plane

OAM tools may, and quite often do, work in conjunction with a control plane and/or management plane. OAM provides instrumentation tools for measuring and monitoring the data plane. OAM tools often use control plane functions, e.g., to initialize OAM sessions and to exchange various parameters. The OAM tools communicate with the management plane to raise alarms, and often OAM tools may be activated by the management (as well as by the control plane), e.g., to locate and localize problems.

The considerations of the control plane maintenance tools and the functionality of the management plane are out of scope for this document, which concentrates on presenting the data plane tools that are used for OAM. Network repair functions such as Fast Reroute (FRR) and protection switching, which are often triggered by OAM protocols, are also out of the scope of this document.

Since OAM protocols are used for monitoring the data plane, it is imperative for OAM tools to be capable of testing the actual data plane with as much accuracy as possible. Thus, it is important to enforce fate-sharing between OAM traffic that monitors the data plane and the data plane traffic it monitors.

## 2. Terminology

### 2.1. Abbreviations

ACH	Associated Channel Header
AIS	Alarm Indication Signal
ATM	Asynchronous Transfer Mode
BFD	Bidirectional Forwarding Detection
CC	Continuity Check
CV	Connectivity Verification
DM	Delay Measurement



ECMP	Equal Cost Multiple Paths
FEC	Forwarding Equivalence Class
FRR	Fast Reroute
G-ACh	Generic Associated Channel
GAL	Generic Associated Label
ICMP	Internet Control Message Protocol
L2TP	Layer Two Tunneling Protocol
L2VPN	Layer Two Virtual Private Network
L3VPN	Layer Three Virtual Private Network
LCCE	L2TP Control Connection Endpoint
LDP	Label Distribution Protocol
LER	Label Edge Router
LM	Loss Measurement
LSP	Label Switched Path
LSR	Label Switched Router
ME	Maintenance Entity
MEG	Maintenance Entity Group
MEP	MEG End Point
MIP	MEG Intermediate Point
MP	Maintenance Point
MPLS	Multiprotocol Label Switching
MPLS-TP	MPLS Transport Profile
MTU	Maximum Transmission Unit
OAM	Operations, Administration, and Maintenance

OWAMP	One-way Active Measurement Protocol
PDH	Plesiochronous Digital Hierarchy
PE	Provider Edge
PSN	Public Switched Network
PW	Pseudowire
PWE3	Pseudowire Emulation Edge-to-Edge
RBridge	Routing Bridge
RDI	Remote Defect Indication
SDH	Synchronous Digital Hierarchy
SONET	Synchronous Optical Networking
TRILL	Transparent Interconnection of Lots of Links
TTL	Time To Live
TWAMP	Two-way Active Measurement Protocol
VCCV	Virtual Circuit Connectivity Verification
VPN	Virtual Private Network

## 2.2. Terminology used in OAM Standards

### 2.2.1. General Terms

A wide variety of terms is used in various OAM standards. This section presents a comparison of the terms used in various OAM standards, without fully quoting the definition of each term.

An interesting overview of the term OAM and its derivatives is presented in [OAM-Def]. A thesaurus of terminology for MPLS-TP terms is presented in [TP-Term], and provides a good summary of some of the OAM related terminology.

### 2.2.2. Operations, Administration and Maintenance

The following definition of OAM is quoted from [OAM-Def]:

The components of the "OAM" acronym (and provisioning) are defined as follows:

- o Operations - Operation activities are undertaken to keep the network (and the services that the network provides) up and running. It includes monitoring the network and finding problems. Ideally these problems should be found before users are affected.
- o Administration - Administration activities involve keeping track of resources in the network and how they are used. It includes all the bookkeeping that is necessary to track networking resources and the network under control.
- o Maintenance - Maintenance activities are focused on facilitating repairs and upgrades -- for example, when equipment must be replaced, when a router needs a patch for an operating system image, or when a new switch is added to a network. Maintenance also involves corrective and preventive measures to make the managed network run more effectively, e.g., adjusting device configuration and parameters.

### 2.2.3. Functions, Tools and Protocols

#### OAM Function

An OAM function is an instrumentation measurement type or diagnostic.

OAM functions are the atomic building blocks of OAM, where each function defines an OAM capability.

Typical examples of OAM functions are presented in Section 3.

#### OAM Protocol

A protocol used for implementing one or more OAM functions.

The OWAMP-Test [OWAMP] is an example of an OAM protocol.

#### OAM Tool

An OAM tool is a specific means of applying one or more OAM functions.

In some cases an OAM protocol *is* an OAM tool, e.g., OWAMP-Test. In other cases an OAM tool uses a set of protocols that are not strictly OAM-related; for example, Traceroute (Section 4.2.) can be

implemented using UDP and ICMP messages, without using an OAM protocol per se.

#### 2.2.4. Data Plane, Control Plane and Management Plane

##### Data Plane

The data plane is the set of functions used to transfer data in the stratum or layer under consideration [ITU-Terms].

The Data Plane is also known as the Forwarding Plane or the User Plane.

##### Control Plane

The control plane is the set of protocols and mechanisms that enable routers to efficiently learn how to forward packets towards their final destination (based on [Comp]).

##### Management Plane

The term Management Plane, as described in [Mng], is used to describe the exchange of management messages through management protocols (often transported by IP and by IP transport protocols) between management applications and the managed entities such as network nodes.

#### Data Plane vs. Control Plane vs. Management Plane

The distinction between the planes is at times a bit vague. For example, the definition of "Control Plane" above may imply that OAM tools such as ping, BFD and others are in fact in the control plane.

This document focuses on tools used for monitoring the data plane. While these tools could arguably be considered to be in the control plane, these tools monitor the data plane, and hence it is imperative to have fate-sharing between OAM traffic that monitors the data plane and the data plane traffic it monitors.

Another potentially vague distinction is between the management plane and control plane. The management plane should be seen as separate from, but possibly overlapping with, the control plane (based on [Mng]).

### 2.2.5. The Players

An OAM tool is used between two (or more) peers. Various terms are used in IETF documents to refer to the players that take part in OAM. Table 2 summarizes the terms used in each of the toolsets discussed in this document.

Toolset	Terms
Ping / Traceroute ([ICMPv4], [ICMPv6], [TCPIP-Tools])	-Host -Node -Interface -Gateway
BFD [BFD]	System
MPLS OAM [MPLS-OAM-FW]	LSR
MPLS-TP OAM [TP-OAM-FW]	-End Point - MEP -Intermediate Point - MIP
Pseudowire OAM [VCCV]	-PE -LCCE
OWAMP and TWAMP ([OWAMP], [TWAMP])	-Host -End system
TRILL OAM [TRILL-OAM]	-RBridge

Table 2 Maintenance Point Terminology

### 2.2.6. Proactive and On-demand Activation

The different OAM tools may be used in one of two basic types of activation:

Proactive

Proactive activation - indicates that the tool is activated on a continual basis, where messages are sent periodically, and errors are detected when a certain number of expected messages are not received.

On-demand

On-demand activation - indicates that the tool is activated "manually" to detect a specific anomaly.

#### 2.2.7. Connectivity Verification and Continuity Checks

Two distinct classes of failure management functions are used in OAM protocols, connectivity verification and continuity checks. The distinction between these terms is defined in [MPLS-TP-OAM], and is used similarly in this document.

Continuity Check

Continuity checks are used to verify that a destination is reachable, and are typically sent proactively, though they can be invoked on-demand as well.

Connectivity Verification

A connectivity verification function allows Alice to check whether she is connected to Bob or not. It is noted that while the CV function is performed in the data plane, the "expected path" is predetermined either in the control plane or in the management plane. A connectivity verification (CV) protocol typically uses a CV message, followed by a CV reply that is sent back to the originator. A CV function can be applied proactively or on-demand.

Connectivity verification tools often perform path verification as well, allowing Alice to verify that messages from Bob are received through the correct path, thereby verifying not only that the two MPs are connected, but also that they are connected through the expected path, allowing detection of unexpected topology changes.

Connectivity verification functions can also be used for checking the MTU of the path between the two peers.

Connectivity verification and continuity checks are considered complementary mechanisms, and are often used in conjunction with each other.

#### 2.2.8. Connection Oriented vs. Connectionless Communication

##### Connection Oriented

In Connection Oriented technologies an end-to-end connection is established (by a control protocol or provisioned by a management system) prior to the transmission of data.

Typically a connection identifier is used to identify the connection. In connection oriented technologies it is often the case (although not always) that all packets belonging to a specific connection use the same route through the network.

##### Connectionless

In Connectionless technologies data is typically sent between end points without prior arrangement. Packets are routed independently based on their destination address, and hence different packets may be routed in a different way across the network.

##### Discussion

The OAM tools described in this document include tools that support connection oriented technologies, as well as tools for connectionless technologies.

In connection oriented technologies OAM is used to monitor a \*specific\* connection; OAM packets are forwarded through the same route as the data traffic and receive the same treatment. In connectionless technologies, OAM is used between a source and destination pair without defining a specific connection. Moreover, in some cases the route of OAM packets may differ from the one of the data traffic. For example, the connectionless IP Ping (Section 4.1.) tests the reachability from a source to a given destination, while the connection oriented LSP Ping (Section 4.4.) is used for monitoring a specific LSP (connection), and provides the capability to monitor all the available paths used by an LSP.

It should be noted that in some cases connectionless protocols are monitored by connection oriented OAM protocols. For example, while IP is a connectionless protocol, it can be monitored by BFD (Section 4.3.), which is connection oriented.

#### 2.2.9. Point-to-point vs. Point-to-multipoint Services

##### Point-to-point (P2P)

A P2P service delivers data from a single source to a single destination.

#### Point-to-multipoint (P2MP)

A P2MP service delivers data from a single source to a one or more destinations (based on [Signal]).

An MP2MP service is a service that delivers data from more than one source to one or more receivers (based on [Signal]).

Note: the two definitions for P2MP and MP2MP are quoted from [Signal]. Although [Signal] describes a specific case of P2MP and MP2MP which is MPLS-specific, these two definitions also apply to non-MPLS cases.

#### Discussion

The OAM tools described in this document include tools for P2P services, as well as tools for P2MP services.

The distinction between P2P services and P2MP services affects the corresponding OAM tools. A P2P service is typically simpler to monitor, as it consists of a single pair of end points. P2MP and MP2MP services present several challenges. For example, in a P2MP service, the OAM mechanism not only verifies that each of the destinations is reachable from the source, but also verifies that the P2MP distribution tree is intact and loop-free.

#### 2.2.10. Failures

The terms Failure, Fault, and Defect are used interchangeably in the standards, referring to a malfunction that can be detected by a connectivity or a continuity check. In some standards, such as 802.1ag [IEEE802.1Q], there is no distinction between these terms, while in other standards each of these terms refers to a different type of malfunction.

The terminology used in IETF MPLS-TP OAM is based on the ITU-T terminology, which distinguishes between these three terms in [ITU-T-G.806];

#### Fault

The term Fault refers to an inability to perform a required action, e.g., an unsuccessful attempt to deliver a packet.



## Defect

The term Defect refers to an interruption in the normal operation, such as a consecutive period of time where no packets are delivered successfully.

## Failure

The term Failure refers to the termination of the required function. While a Defect typically refers to a limited period of time, a failure refers to a long period of time.

## 3. OAM Functions

This subsection provides a brief summary of the common OAM functions used in OAM-related standards. These functions are used as building blocks in the OAM standards described in this document.

- o Connectivity Verification (CV), Path Verification and Continuity Checks (CC):  
As defined in Section 2.2.7.
- o Path Discovery / Fault Localization:  
This function can be used to trace the route to a destination, i.e., to identify the nodes along the route to the destination. When more than one route is available to a specific destination, this function traces one of the available routes. When a failure occurs, this function attempts to detect the location of the failure.  
Note that the term route tracing (or Traceroute) that is used in the context of IP and MPLS, is sometimes referred to as path tracing in the context of other protocols, such as TRILL.
- o Performance Monitoring:  
Typically refers to:
  - o Loss Measurement (LM) - monitors the packet loss rate.
  - o Delay Measurement (DM) - monitors the delay and delay variation (jitter).

## 4. OAM Tools in the IETF - a Detailed Description

This section presents a detailed description of the sets of OAM-related tools in each of the toolsets in Table 1.

#### 4.1. IP Ping

Ping is a common network diagnosis application for IP networks that uses ICMP. According to [NetTerms], 'Ping' is an abbreviation for Packet internet groper, although the term has been so commonly used that it stands on its own. As defined in [NetTerms], it is a program used to test reachability of destinations by sending them an ICMP echo request and waiting for a reply.

The ICMP Echo request/reply exchange in Ping is used as a continuity check function for the Internet Protocol. The originator transmits an ICMP Echo request packet, and the receiver replies with an Echo reply. ICMP ping is defined in two variants, [ICMPv4] is used for IPv4, and [ICMPv6] is used for IPv6.

Ping can be invoked either to a unicast destination or to a multicast destination. In the latter case, all members of the multicast group send an Echo reply back to the originator.

Ping implementations typically use ICMP messages. UDP Ping is a variant that uses UDP messages instead of ICMP echo messages.

Ping is a single-ended continuity check, i.e., it allows the \*initiator\* of the Echo request to test the reachability. If it is desirable for both ends to test the reachability, both ends have to invoke Ping independently.

Note that since ICMP filtering is deployed in some routers and firewalls, the usefulness of Ping is sometimes limited in the wider internet. This limitation is equally relevant to Traceroute.

#### 4.2. IP Traceroute

Traceroute ([TCPIP-Tools], [NetTools]) is an application that allows users to discover a path between an IP source and an IP destination.

The most common way to implement Traceroute [TCPIP-Tools] is described as follows. Traceroute sends a sequence of UDP packets to UDP port 33434 at the destination. By default, Traceroute begins by sending three packets (the number of packets is configurable in most Traceroute implementations), each with an IP Time-To-Live (or Hop Limit in IPv6) value of one to the destination. These packets expire as soon as they reach the first router in the path. Consequently, that router sends three ICMP Time Exceeded Messages back to the Traceroute application. Traceroute now sends another three UDP packets, each with the TTL value of 2. These messages cause the second router to return ICMP messages. This process continues, with

ever increasing values for the TTL field, until the packets actually reach the destination. Because no application listens to port 33434 at the destination, the destination returns ICMP Destination Unreachable Messages indicating an unreachable port. This event indicates to the Traceroute application that it is finished. The Traceroute program displays the round-trip delay associated with each of the attempts.

While Traceroute is a tool that finds *a* path from A to B, it should be noted that traffic from A to B is often forwarded through Equal Cost Multiple Paths (ECMP). Paris Traceroute [PARIS] is an extension to Traceroute that attempts to discover all the available paths from A to B by scanning different values of header fields (such as UDP ports) in the probe packets.

It is noted that Traceroute is an application, and not a protocol. As such, it has various different implementations. One of the most common ones uses UDP probe packets, as described above. Other implementations exist that use other types of probe messages, such as ICMP or TCP.

Note that IP routing may be asymmetric. While Traceroute discovers a path between a source and destination, it does not reveal the reverse path.

A few ICMP extensions ([ICMP-MP], [ICMP-Int]) have been defined in the context of Traceroute. These documents define several extensions, including extensions to the ICMP Destination Unreachable message, that can be used by Traceroute applications.

Traceroute allows path discovery to *unicast* destination addresses. A similar tool [mtrace] was defined for multicast destination addresses, allowing to trace the route that a multicast IP packet takes from a source to a particular receiver.

#### 4.3. Bidirectional Forwarding Detection (BFD)

##### 4.3.1. Overview

While multiple OAM tools have been defined for various protocols in the protocol stack, Bidirectional Forwarding Detection [BFD], defined by the IETF BFD working group, is a generic OAM tool that can be deployed over various encapsulating protocols, and in various medium types. The IETF has defined variants of the protocol for IP ([BFD-IP], [BFD-Multi]), for MPLS LSPs [BFD-LSP], and for pseudowires [BFD-VCCV]. The usage of BFD in MPLS-TP is defined in [TP-CC-CV].

BFD includes two main OAM functions, using two types of BFD packets: BFD Control packets, and BFD Echo packets.

#### 4.3.2. Terminology

BFD operates between *\*systems\**. The BFD protocol is run between two or more systems after establishing a *\*session\**.

#### 4.3.3. BFD Control

BFD supports a bidirectional continuity check, using BFD control packets, that are exchanged within a BFD session. BFD sessions operate in one of two modes:

- o Asynchronous mode (i.e., proactive): in this mode BFD control packets are sent periodically. When the receiver detects that no BFD control packets have been received during a predetermined period of time, a failure is reported.
- o Demand mode: in this mode, BFD control packets are sent on-demand. Upon need, a system initiates a series of BFD control packets to check the continuity of the session. BFD control packets are sent independently in each direction.

Each of the end-points (referred to as systems) of the monitored path maintains its own session identification, called a Discriminator, both of which are included in the BFD Control Packets that are exchanged between the end-points. At the time of session establishment, the Discriminators are exchanged between the two-end points. In addition, the transmission (and reception) rate is negotiated between the two end-points, based on information included in the control packets. These transmission rates may be renegotiated during the session.

During normal operation of the session, i.e., when no failures have been detected, the BFD session is in the Up state. If no BFD Control packets are received during a period of time called the Detection Time, the session is declared to be Down. The detection time is a function of the pre-configured or negotiated transmission rate, and a parameter called Detect Mult. Detect Mult determines the number of missing BFD Control packets that cause the session to be declared as Down. This parameter is included in the BFD Control packet.

#### 4.3.4. BFD Echo

A BFD echo packet is sent to a peer system, and is looped back to the originator. The echo function can be used proactively, or on-demand.

The BFD echo function has been defined in BFD for IPv4 and IPv6 ([BFD-IP]), but is not used in BFD for MPLS LSPs, PWs, or in BFD for MPLS-TP.

#### 4.4. MPLS OAM

The IETF MPLS working group has defined OAM for MPLS LSPs. The requirements and framework of this effort are defined in [MPLS-OAM-FW] and [MPLS-OAM], respectively. The corresponding OAM tool defined, in this context, is LSP Ping [LSP-Ping]. OAM for P2MP services is defined in [MPLS-P2MP].

BFD for MPLS [BFD-LSP] is an alternative means for detecting data-plane failures, as described below.

##### 4.4.1. LSP Ping

LSP Ping is modeled after the Ping/Traceroute paradigm and thus it may be used in one of two modes:

- o "Ping" mode: In this mode LSP Ping is used for end-to-end connectivity verification between two LERs.
- o "Traceroute" mode: This mode is used for hop-by-hop fault isolation.

LSP Ping is based on ICMP Ping operation (of data-plane connectivity verification) with additional functionality to verify data-plane vs. control-plane consistency for a Forwarding Equivalence Class (FEC) and also identify Maximum Transmission Unit (MTU) problems.

The Traceroute functionality may be used to isolate and localize MPLS faults, using the Time-to-live (TTL) indicator to incrementally identify the sub-path of the LSP that is successfully traversed before the faulty link or node.

The challenge in MPLS networks is that the traffic of a given LSP may be load balanced across Equal Cost Multiple paths (ECMP). LSP Ping monitors all the available paths of an LSP by monitoring its different Forwarding Equivalence Classes (FEC). Note that MPLS-TP does not use ECMP, and thus does not require OAM over multiple paths.

Another challenge is that an MPLS LSP does not necessarily have a return path; traffic that is sent back from the egress LSR to the ingress LSR is not necessarily sent over an MPLS LSP, but can be sent through a different route, such as an IP route. Thus, responding to an LSP Ping message is not necessarily as trivial as in IP Ping,

where the responder just swaps the source and destination IP addresses. Note that this challenge is not applicable to MPLS-TP, where a return path is always available.

It should be noted that LSP Ping supports unique identification of the LSP within an addressing domain. The identification is checked using the full FEC identification. LSP Ping is extensible to include additional information needed to support new functionality, by use of Type-Length-Value (TLV) constructs. The usage of TLVs is typically handled by the control plane, as it is not easy to implement in hardware.

LSP Ping supports both asynchronous, as well as, on-demand activation.

#### 4.4.2. BFD for MPLS

BFD [BFD-LSP] can be used to detect MPLS LSP data plane failures.

A BFD session is established for each MPLS LSP that is being monitored. BFD Control packets must be sent along the same path as the monitored LSP. If the LSP is associated with multiple FECs, a BFD session is established for each FEC.

While LSP Ping can be used for detecting MPLS data plane failures and for verifying the MPLS LSP data plane against the control plane, BFD can only be used for the former. BFD can be used in conjunction with LSP Ping, as is the case in MPLS-TP (see Section 4.5.4.).

#### 4.4.3. OAM for Virtual Private Networks (VPN) over MPLS

The IETF has defined two classes of VPNs, Layer 2 VPNs (L2VPN) and Layer 3 VPNs (L3VPN). [L2VPN-OAM] provides the requirements and framework for OAM in the context of Layer 2 Virtual Private Networks (L2VPN), and specifically it also defines the OAM layering of L2VPNs over MPLS. [L3VPN-OAM] provides a framework for the operation and management of Layer 3 Virtual Private Networks (L3VPNs).

### 4.5. MPLS-TP OAM

#### 4.5.1. Overview

The MPLS working group has defined the OAM toolset that fulfills the requirements for MPLS-TP OAM. The full set of requirements for MPLS-TP OAM are defined in [MPLS-TP-OAM], and include both general requirements for the behavior of the OAM tools and a set of operations that should be supported by the OAM toolset. The set of

mechanisms required are further elaborated in [TP-OAM-FW], which describes the general architecture of the OAM system as well as giving overviews of the functionality of the OAM toolset.

Some of the basic requirements for the OAM toolset for MPLS-TP are:

- o MPLS-TP OAM must be able to support both an IP based and non-IP based environment. If the network is IP based, i.e., IP routing and forwarding are available, then the MPLS-TP OAM toolset should rely on the IP routing and forwarding capabilities. On the other hand, in environments where IP functionality is not available, the OAM tools must still be able to operate without dependence on IP forwarding and routing.
- o OAM packets and the user traffic are required to be congruent (i.e., OAM packets are transmitted in-band) and there is a need to differentiate OAM packets from ordinary user packets in the data plane. Inherent in this requirement is the principle that MPLS-TP OAM be independent of any existing control-plane, although it should not preclude use of the control-plane functionality. OAM packets are identified by the Generic Associated Label (GAL), which is a reserved MPLS label value (13).

#### 4.5.2. Terminology

##### Maintenance Entity (ME)

The MPLS-TP OAM tools are designed to monitor and manage a Maintenance Entity (ME). An ME, as defined in [TP-OAM-FW], defines a relationship between two points of a transport path to which maintenance and monitoring operations apply.

The term Maintenance Entity (ME) is used in ITU-T Recommendations (e.g., [ITU-T-Y1731]), as well as in the MPLS-TP terminology ([TP-OAM-FW]).

##### Maintenance Entity Group (MEG)

The collection of one or more MEs that belongs to the same transport path and that are maintained and monitored as a group are known as a Maintenance Entity Group (based on [TP-OAM-FW]).

##### Maintenance Point (MP)

A Maintenance Point (MP) is a functional entity that is defined at a node in the network, and can initiate and/or react to OAM messages. This document focuses on the data-plane functionality of MPs, while

MPs interact with the control plane and with the management plane as well.

The term MP is used in IEEE 802.1ag, and was similarly adopted in MPLS-TP ([TP-OAM-FW]).

#### Maintenance End Point (MEP)

A Maintenance End Point (MEP) is one of the end points of an ME, and can initiate OAM messages and respond to them (based on [TP-OAM-FW]).

#### Maintenance Intermediate Point (MIP)

In between MEPs, there are zero or more intermediate points, called Maintenance Entity Group Intermediate Points (based on [TP-OAM-FW]).

A Maintenance Intermediate Point (MIP) is an intermediate point that does not generally initiate OAM frames (one exception to this is the use of AIS notifications), but is able to respond to OAM frames that are destined to it. A MIP in MPLS-TP identifies OAM packets destined to it by the expiration of the TTL field in the OAM packet. The term Maintenance Point is a general term for MEPs and MIPs.

#### Up and Down MEPs

The IEEE 802.1ag [IEEE802.1Q] defines a distinction between Up MEPs and Down MEPs. A MEP monitors traffic either in the direction facing the network, or in the direction facing the bridge. A Down MEP is a MEP that receives OAM packets from, and transmits them to the direction of the network. An Up MEP receives OAM packets from, and transmits them to the direction of the bridging entity. MPLS-TP ([TP-OAM-FW]) uses a similar distinction on the placement of the MEP - either at the ingress, egress, or forwarding function of the node (Down / Up MEPs). This placement is important for localization of a failure.

Note that the terms Up and Down MEPs are entirely unrelated to the conventional up/down terminology, where down means faulty, and up is nonfaulty.

The distinction between Up and Down MEPs was defined in [TP-OAM-FW], but has not been used in other MPLS-TP RFCs, as of the writing of this document.



#### 4.5.3. Generic Associated Channel

In order to address the requirement for in-band transmission of MPLS-TP OAM traffic, MPLS-TP uses a Generic Associated Channel (G-ACh), defined in [G-ACh] for LSP-based OAM traffic. This mechanism is based on the same concepts as the PWE3 ACH [PW-ACH] and VCCV [VCCV] mechanisms. However, to address the needs of LSPs as differentiated from PW, the following concepts were defined for [G-ACh]:

- o An Associated Channel Header (ACH), that uses a format similar to the PW Control Word [PW-ACH], is a 4-byte header that is prepended to OAM packets.
- o A Generic Associated Label (GAL). The GAL is a reserved MPLS label value (13) that indicates that the packet is an ACH packet and the payload follows immediately after the label stack.

It should be noted that while the G-ACh was defined as part of the MPLS-TP definition effort, the G-ACh is a generic tool that can be used in MPLS in general, and not only in MPLS-TP.

#### 4.5.4. MPLS-TP OAM Toolset

To address the functionality that is required of the OAM toolset, the MPLS WG conducted an analysis of the existing IETF and ITU-T OAM tools and their ability to fulfill the required functionality. The conclusions of this analysis are documented in [OAM-Analys]. MPLS-TP uses a mixture of OAM tools that are based on previous standards, and adapted to the requirements of [MPLS-TP-OAM]. Some of the main building blocks of this solution are based on:

- o Bidirectional Forwarding Detection ([BFD], [BFD-LSP]) for proactive continuity check and connectivity verification.
- o LSP Ping as defined in [LSP-Ping] for on-demand connectivity verification.
- o New protocol packets, using G-ACh, to address different functionality.
- o Performance measurement protocols that are based on the functionality that is described in [ITU-T-Y1731].

The following sub-sections describe the OAM tools defined for MPLS-TP as described in [TP-OAM-FW].

#### 4.5.4.1. Continuity Check and Connectivity Verification

Continuity Check and Connectivity Verification are presented in Section 2.2.7. of this document. As presented there, these tools may be used either proactively or on-demand. When using these tools proactively, they are generally used in tandem.

For MPLS-TP there are two distinct tools, the proactive tool is defined in [TP-CC-CV] while the on-demand tool is defined in [OnDemand-CV]. In on-demand mode, this function should support monitoring between the MEPs and, in addition, between a MEP and MIP. [TP-OAM-FW] highlights, when performing Connectivity Verification, the need for the CC-V messages to include unique identification of the MEG that is being monitored and the MEP that originated the message.

The proactive tool [TP-CC-CV] is based on extensions to BFD (see Section 4.3.) with the additional limitation that the transmission and receiving rates are based on configuration by the operator. The on-demand tool [OnDemand-CV] is an adaptation of LSP Ping (see Section 4.4.) for the required behavior of MPLS-TP.

#### 4.5.4.2. Route Tracing

[MPLS-TP-OAM] defines that there is a need for functionality that would allow a path end-point to identify the intermediate and end-points of the path. This function would be used in on-demand mode. Normally, this path will be used for bidirectional PW, LSP, and sections, however, unidirectional paths may be supported only if a return path exists. The tool for this is based on the LSP Ping (see Section 4.4.) functionality and is described in [OnDemand-CV].

#### 4.5.4.3. Lock Instruct

The Lock Instruct function [Lock-Loop] is used to notify a transport path end-point of an administrative need to disable the transport path. This functionality will generally be used in conjunction with some intrusive OAM function, e.g., Performance measurement, Diagnostic testing, to minimize the side-effect on user data traffic.

#### 4.5.4.4. Lock Reporting

Lock Reporting is a function used by an end-point of a path to report to its far-end end-point that a lock condition has been affected on the path.

#### 4.5.4.5. Alarm Reporting

Alarm Reporting [TP-Fault] provides the means to suppress alarms following detection of defect conditions at the server sub-layer. Alarm reporting is used by an intermediate point of a path, that becomes aware of a fault on the path, to report to the end-points of the path. [TP-OAM-FW] states that this may occur as a result of a defect condition discovered at a server sub-layer. This generates an Alarm Indication Signal (AIS) that continues until the fault is cleared. The consequent action of this function is detailed in [TP-OAM-FW].

#### 4.5.4.6. Remote Defect Indication

Remote Defect Indication (RDI) is used proactively by a path end-point to report to its peer end-point that a defect is detected on a bidirectional connection between them. [MPLS-TP-OAM] points out that this function may be applied to a unidirectional LSP only if a return path exists. [TP-OAM-FW] points out that this function is associated with the proactive CC-V function.

#### 4.5.4.7. Client Failure Indication

Client Failure Indication (CFI) is defined in [MPLS-TP-OAM] to allow the propagation information from one edge of the network to the other. The information concerns a defect to a client, in the case that the client does not support alarm notification.

#### 4.5.4.8. Performance Monitoring

The definition of MPLS performance monitoring was motivated by the MPLS-TP requirements [MPLS-TP-OAM], but was defined generically for MPLS in [MPLS-LM-DM]. An additional document [TP-LM-DM] defines a performance monitoring profile for MPLS-TP.

##### 4.5.4.8.1. Packet Loss Measurement (LM)

Packet Loss Measurement is a function used to verify the quality of the service. Packet loss, as defined in [IPPM-1LM] and [MPLS-TP-OAM], indicates the ratio of the number of user packets lost to the total number of user packets sent during a defined time interval.

There are two possible ways of determining this measurement:

- o Using OAM packets, it is possible to compute the statistics based on a series of OAM packets. This, however, has the disadvantage of being artificial, and may not be representative since part of the packet loss may be dependent upon packet sizes and upon the implementation of the MEPs that take part in the protocol.
- o Sending delimiting messages for the start and end of a measurement period during which the source and sink of the path count the packets transmitted and received. After the end delimiter, the ratio would be calculated by the path OAM entity.

#### 4.5.4.8.2. Packet Delay Measurement (DM)

Packet Delay Measurement is a function that is used to measure one-way or two-way delay of a packet transmission between a pair of the end-points of a path (PW, LSP, or Section). Where:

- o One-way packet delay, as defined in [IPPM-1DM], is the time elapsed from the start of transmission of the first bit of the packet by a source node until the reception of the last bit of that packet by the destination node. Note that one-way delay measurement requires the clocks of the two end-points to be synchronized.
- o Two-way packet delay, as defined in [IPPM-2DM], is the time elapsed from the start of transmission of the first bit of the packet by a source node until the reception of the last bit of the loop-backed packet by the same source node, when the loopback is performed at the packet's destination node. Note that due to possible path asymmetry, the one-way packet delay from one end-point to another is not necessarily equal to half of the two-way packet delay.  
As opposed to one-way delay measurement, two-way delay measurement does not require the two end-points to be synchronized.

For each of these two metrics, the DM function allows the MEP to measure the delay, as well as the delay variation. Delay measurement is performed by exchanging timestamped OAM packets between the participating MEPs.

#### 4.6. Pseudowire OAM

##### 4.6.1. Pseudowire OAM using Virtual Circuit Connectivity Verification (VCCV)

VCCV, as defined in [VCCV], provides a means for end-to-end fault detection and diagnostics tools to be used for PWs (regardless of the

underlying tunneling technology). The VCCV switching function provides a control channel associated with each PW. [VCCV] defines three Control Channel (CC) types, i.e., three possible methods for transmitting and identifying OAM messages:

- o CC Type 1: In-band VCCV, as described in [VCCV], is also referred to as "PWE3 Control Word with 0001b as first nibble". It uses the PW Associated Channel Header [PW-ACH].
- o CC Type 2: Out-of-band VCCV [VCCV], is also referred to as "MPLS Router Alert Label". In this case the control channel is created by using the MPLS router alert label [MPLS-ENCAPS] immediately above the PW label.
- o CC Type 3: TTL expiry VCCV [VCCV], is also referred to as "MPLS PW Label with TTL == 1", i.e., the control channel is identified when the value of the TTL field in the PW label is set to 1.

VCCV currently supports the following OAM tools: ICMP Ping, LSP Ping, and BFD. ICMP and LSP Ping are IP encapsulated before being sent over the PW ACH. BFD for VCCV [BFD-VCCV] supports two modes of encapsulation - either IP/UDP encapsulated (with IP/UDP header) or PW-ACH encapsulated (with no IP/UDP header) and provides support to signal the AC status. The use of the VCCV control channel provides the context, based on the MPLS-PW label, required to bind and bootstrap the BFD session to a particular pseudo wire (FEC), eliminating the need to exchange Discriminator values.

VCCV consists of two components: (1) signaled component to communicate VCCV capabilities as part of VC label, and (2) switching component to cause the PW payload to be treated as a control packet.

VCCV is not directly dependent upon the presence of a control plane. The VCCV capability advertisement may be performed as part of the PW signaling when LDP is used. In case of manual configuration of the PW, it is the responsibility of the operator to set consistent options at both ends. The manual option was created specifically to handle MPLS-TP use cases where no control plane was a requirement. However, new use cases such as pure mobile backhaul find this functionality useful too.

The PWE3 working group has conducted an implementation survey of VCCV [VCCV-SURVEY], which analyzes which VCCV mechanisms are used in practice.

#### 4.6.2. Pseudowire OAM using G-ACh

As mentioned above, VCCV enables OAM for PWs by using a control channel for OAM packets. When PWs are used in MPLS-TP networks, rather than the control channels defined in VCCV, the G-ACh can be used as an alternative control channel. The usage of the G-ACh for PWs is defined in [PW-G-ACh].

#### 4.6.3. Attachment Circuit - Pseudowire Mapping

The PWE3 working group has defined a mapping and notification of defect states between a pseudowire (PW) and the Attachment Circuits (ACs) of the end-to-end emulated service. This mapping is of key importance to the end-to-end functionality. Specifically, the mapping is provided by [PW-MAP], by [L2TP-EC] for L2TPv3 pseudowires, and Section 5.3 of [ATM-L2] for ATM.

[L2VPN-OAM] provides the requirements and framework for OAM in the context of Layer 2 Virtual Private Networks (L2VPN), and specifically it also defines the OAM layering of L2VPNs over pseudowires.

The mapping defined in [Eth-Int] allows an end-to-end emulated Ethernet service over pseudowires.

### 4.7. OWAMP and TWAMP

#### 4.7.1. Overview

The IPPM working group in the IETF defines common criteria and metrics for measuring performance of IP traffic ([IPPM-FW]). Some of the key RFCs published by this working group have defined metrics for measuring connectivity [IPPM-Con], delay ([IPPM-1DM], [IPPM-2DM]), and packet loss [IPPM-1LM]. It should be noted that the work of the IETF in the context of performance metrics is not limited to IP networks; [PM-CONS] presents general guidelines for considering new performance metrics.

The IPPM working group has defined not only metrics for performance measurement, but also protocols that define how the measurement is carried out. The One-way Active Measurement Protocol [OWAMP] and the Two-Way Active Measurement Protocol [TWAMP] define a method and protocol for measuring performance metrics in IP networks.

OWAMP [OWAMP] enables measurement of one-way characteristics of IP networks, such as one-way packet loss and one-way delay. For its proper operation OWAMP requires accurate time of day setting at its end points.

TWAMP [TWAMP] is a similar protocol that enables measurement of both one-way and two-way (round trip) characteristics.

OWAMP and TWAMP are both comprised of two separate protocols:

- o OWAMP-Control/TWAMP-Control: used to initiate, start, and stop test sessions and to fetch their results. Continuity Check and Connectivity Verification are tested and confirmed by establishing the OWAMP/TWAMP Control Protocol TCP connection.
- o OWAMP-Test/TWAMP-Test: used to exchange test packets between two measurement nodes. Enables the loss and delay measurement functions, as well as detection of other anomalies, such as packet duplication and packet reordering.

It should be noted that while [OWAMP] and [TWAMP] define tools for performance measurement, they do not define the accuracy of these tools. The accuracy depends on scale, implementation and network configurations.

Alternative protocols for performance monitoring are defined, for example, in MPLS-TP OAM ([MPLS-LM-DM], [TP-LM-DM]), and in Ethernet OAM [ITU-T-Y1731].

#### 4.7.2. Control and Test Protocols

OWAMP and TWAMP control protocols run over TCP, while the test protocols run over UDP. The purpose of the control protocols is to initiate, start, and stop test sessions, and for OWAMP to fetch results. The test protocols introduce test packets (which contain sequence numbers and timestamps) along the IP path under test according to a schedule, and record statistics of packet arrival. Multiple sessions may be simultaneously defined, each with a session identifier, and defining the number of packets to be sent, the amount of padding to be added (and thus the packet size), the start time, and the send schedule (which can be either a constant time between test packets or exponentially distributed pseudo-random). Statistics recorded conform to the relevant IPPM RFCs.

From a security perspective, OWAMP and TWAMP test packets are hard to detect because they are simply UDP streams between negotiated port numbers, with potentially nothing static in the packets. OWAMP and TWAMP also include optional authentication and encryption for both control and test packets.

#### 4.7.3. OWAMP

OWAMP defines the following logical roles: Session-Sender, Session-Receiver, Server, Control-Client, and Fetch-Client. The Session-Sender originates test traffic that is received by the Session-Receiver. The Server configures and manages the session, as well as returning the results. The Control-Client initiates requests for test sessions, triggers their start, and may trigger their termination. The Fetch-Client requests the results of a completed session. Multiple roles may be combined in a single host - for example, one host may play the roles of Control-Client, Fetch-Client, and Session-Sender, and a second playing the roles of Server and Session-Receiver.

In a typical OWAMP session the Control-Client establishes a TCP connection to port 861 of the Server, which responds with a server greeting message indicating supported security/integrity modes. The Control-Client responds with the chosen communications mode and the Server accepts the mode. The Control-Client then requests and fully describes a test session to which the Server responds with its acceptance and supporting information. More than one test session may be requested with additional messages. The Control-Client then starts a test session and the Server acknowledges, and instructs the Session-Sender to start the test. The Session-Sender then sends test packets with pseudorandom padding to the Session-Receiver until the session is complete or until the Control-client stops the session. Once finished, the Session-Sender reports to the Server which recovers data from the Session-Receiver. The Fetch-Client can then send a fetch request to the Server, which responds with an acknowledgement and immediately thereafter the result data.

#### 4.7.4. TWAMP

TWAMP defines the following logical roles: session-sender, session-reflector, server, and control-client. These are similar to the OWAMP roles, except that the Session-Reflector does not collect any packet information, and there is no need for a Fetch-Client.

In a typical TWAMP session the Control-Client establishes a TCP connection to port 862 of the Server, and mode is negotiated as in OWAMP. The Control-Client then requests sessions and starts them. The Session-Sender sends test packets with pseudorandom padding to the Session-Reflector which returns them with insertion of timestamps.



#### 4.8. TRILL

The requirements of OAM in TRILL are defined in [TRILL-OAM]. The challenge in TRILL OAM, much like in MPLS networks, is that traffic between RBridges RB1 and RB2 may be forwarded through more than one path. Thus, an OAM protocol between RBridges RB1 and RB2 must be able to monitor all the available paths between the two RBridge.

During the writing of this document the detailed definition of the TRILL OAM tools are still work in progress. This subsection presents the main requirements of TRILL OAM.

The main requirements defined in [TRILL-OAM] are:

- o Continuity Checking (CC) - the TRILL OAM protocol must support a function for CC between any two RBridges RB1 and RB2.
- o Connectivity Verification (CV) - connectivity between two RBridges RB1 and RB2 can be verified on a per-flow basis.
- o Path Tracing - allows an RBridge to trace all the available paths to a peer RBridge.
- o Performance monitoring - allows an RBridge to monitor the packet loss and packet delay to a peer RBridge.

#### 5. Summary

This section summarizes the OAM tools and functions presented in this document. This summary is an index to some of the main OAM tools defined in the IETF. This compact index that can be useful to all readers from network operators to standards development organizations. The summary includes a short subsection that presents some guidance to network equipment vendors.

##### 5.1. Summary of OAM Tools

This subsection provides a short summary of each of the OAM toolsets described in this document.

A detailed list of the RFCs related to each toolset is given in Appendix A.1.

+-----+-----+-----+		+-----+-----+	
Toolset	Description	Transport	Technology

IP Ping	Ping ([IntHost], [NetTerms]) is a simple application for testing reachability that uses ICMP Echo messages ([ICMPv4], [ICMPv6]).	IPv4/IPv6
IP Traceroute	Traceroute ([TCPIP-Tools], [NetTools]) is an application that allows users to trace the path between an IP source and an IP destination, i.e., to identify the nodes along the path. If more than one path exists between the source and destination Traceroute traces *a* path. The most common implementation of Traceroute uses UDP probe messages, although there are other implementations that use different probes, such as ICMP or TCP. Paris Traceroute [PARIS] is an extension that attempts to discover all the available paths from A to B by scanning different values of header fields.	IPv4/IPv6
BFD	Bidirectional Forwarding Detection (BFD) is defined in [BFD] as a framework for a lightweight generic OAM tool. The intention is to define a base tool that can be used with various encapsulation types, network environments, and in various medium types.	generic
MPLS OAM	MPLS LSP Ping, as defined in [MPLS-OAM], [MPLS-OAM-FW] and [LSP-Ping], is an OAM tool for point-to-point and point-to-multipoint MPLS LSPs. It includes two main functions: Ping and Traceroute. BFD [BFD-LSP] is an alternative means for detecting MPLS LSP data plane failures.	MPLS
MPLS-TP OAM	MPLS-TP OAM is defined in a set of RFCs.	MPLS-TP

	The OAM requirements for MPLS Transport Profile (MPLS-TP) are defined in [MPLS-TP-OAM]. Each of the tools in the OAM toolset is defined in its own RFC, as specified in Section A.1.	
Pseudowire OAM	The PWE3 OAM architecture defines control channels that support the use of existing IETF OAM tools to be used for a pseudowire (PW). The control channels that are defined in [VCCV] and [PW-G-ACh] may be used in conjunction with ICMP Ping, LSP Ping, and BFD to perform CC and CV functionality. In addition the channels support use of any of the MPLS-TP based OAM tools for completing their respective OAM functionality for a PW.	Pseudowire
OWAMP and TWAMP	The One Way Active Measurement Protocol [OWAMP] and the Two Way Active Measurement Protocols [TWAMP] are two protocols defined in the IP Performance Metrics (IPPM) working group in the IETF. These protocols allow various performance metrics to be measured, such as packet loss, delay and delay variation, duplication and reordering.	IPv4/IPv6
TRILL OAM	The requirements of OAM in TRILL are defined in [TRILL-OAM]. These requirements include continuity checking, connectivity verification, path tracing and performance monitoring. During the writing of this document the detailed definition of the TRILL OAM tools is work in progress.	TRILL

Table 3 Summary of OAM-related IETF Tools

## 5.2. Summary of OAM Functions

Table 4 summarizes the OAM functions that are supported in each of the toolsets that were analyzed in this section. The columns of this table are the typical OAM functions described in Section 1.3.

Toolset	Continuity Check	Connectivity Verification	Path Discovery	Performance Monitoring	Other Functions
IP Ping	Echo				
IP Traceroute			Traceroute		
BFD	BFD Control / Echo	BFD Control			RDI using BFD Control
MPLS OAM (LSP Ping)		"Ping" mode	"Traceroute" mode		
MPLS-TP OAM	CC	CV/pro-active or on-demand	Route Tracing	-LM -DM	-Diagnostic Test -Lock -Alarm Reporting -Client Failure Indication -RDI
Pseudowire OAM	BFD	-BFD -ICMP Ping -LSP-Ping	LSP-Ping		
OWAMP and	- control			-Delay	

	TWAMP	protocol			measur ement -Packet loss measur ement	
	TRILL OAM	CC	CV	Path tracing	-Delay measur ement -Packet loss measur ement	

Table 4 Summary of the OAM Functionality in IETF OAM Tools

### 5.3. Guidance to Network Equipment Vendors

As mentioned in Section 1.4. , it is imperative for OAM tools to be capable of testing the actual data plane in as much accuracy as possible. While this guideline may appear obvious, it is worthwhile to emphasize the key importance of enforcing fate-sharing between OAM traffic that monitors the data plane and the data plane traffic it monitors.

## 6. Security Considerations

OAM is tightly coupled with the stability of the network. A successful attack on an OAM protocol can create a false illusion of non-existent failures, or prevent the detection of actual ones. In both cases the attack may result in denial of service.

Some of the OAM tools presented in this document include security mechanisms that provide integrity protection, thereby preventing attackers from forging or tampering with OAM packets. For example, [BFD] includes an optional authentication mechanism for BFD Control packets, using either SHA1, MD5, or a simple password. [OWAMP] and [TWAMP] have 3 modes of security: unauthenticated, authenticated, and encrypted. The authentication uses SHA1 as the HMAC algorithm, and the encrypted mode uses AES encryption.

Confidentiality is typically not considered a requirement for OAM protocols. However, the use of encryption (e.g., [OWAMP] and

[TWAMP]) can make it difficult for attackers to identify OAM packets, thus making it more difficult to attack the OAM protocol.

OAM can also be used as a means for network reconnaissance; information about addresses, port numbers and about the network topology and performance can be gathered either by passively eavesdropping to OAM packets, or by actively sending OAM packets and gathering information from the respective responses. This information can then be used maliciously to attack the network. Note that some of this information, e.g., addresses and port numbers, can be gathered even when encryption is used ([OWAMP], [TWAMP]).

For further details about the security considerations of each OAM protocol, the reader is encouraged to review the Security Considerations section of each document referenced by this memo.

## 7. IANA Considerations

There are no new IANA considerations implied by this document.

## 8. Acknowledgments

The authors gratefully acknowledge Sasha Vainshtein, Carlos Pignataro, David Harrington, Dan Romascanu, Ron Bonica, Benoit Claise, Stewart Bryant, Tom Nadeau, Elwyn Davies, Al Morton, Sam Aldrin, Thomas Narten, and other members of the OPSA WG for their helpful comments on the mailing list.

This document was prepared using 2-Word-v2.0.template.dot.

## 9. References

### 9.1. Normative References

[OAM-Def]        Andersson, L., Van Helvoort, H., Bonica, R., Romascanu, D., Mansfield, S., "Guidelines for the use of the OAM acronym in the IETF ", RFC 6291, June 2011.

### 9.2. Informative References

[ATM-L2]        Singh, S., Townsley, M., and C. Pignataro, "Asynchronous Transfer Mode (ATM) over Layer 2 Tunneling Protocol Version 3 (L2TPv3)", RFC 4454, May 2006.

[BFD]           Katz, D., Ward, D., "Bidirectional Forwarding Detection (BFD)", RFC 5880, June 2010.

- [BFD-Gen] Katz, D., Ward, D., "Generic Application of Bidirectional Forwarding Detection (BFD)", RFC 5882, June 2010.
- [BFD-IP] Katz, D., Ward, D., "Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop)", RFC 5881, June 2010.
- [BFD-LSP] Aggarwal, R., Kompella, K., Nadeau, T., and Swallow, G., "Bidirectional Forwarding Detection (BFD) for MPLS Label Switched Paths (LSPs)", RFC 5884, June 2010.
- [BFD-Multi] Katz, D., Ward, D., "Bidirectional Forwarding Detection (BFD) for Multihop Paths", RFC 5883, June 2010.
- [BFD-VCCV] Nadeau, T., Pignataro, C., "Bidirectional Forwarding Detection (BFD) for the Pseudowire Virtual Circuit Connectivity Verification (VCCV)", RFC 5885, June 2010.
- [Comp] Bonaventure, O., "Computer Networking: Principles, Protocols and Practice", 2008.
- [Dup] Uijterwaal, H., "A One-Way Packet Duplication Metric", RFC 5560, May 2009.
- [Eth-Int] Mohan, D., Bitar, N., Sajassi, A., Delord, S., Niger, P., Qiu, R., "MPLS and Ethernet Operations, Administration, and Maintenance (OAM) Interworking", RFC 7023, October 2013.
- [G-ACh] Bocci, M., Vigoureux, M., Bryant, S., "MPLS Generic Associated Channel", RFC 5586, June 2009.
- [ICMP-Ext] Bonica, R., Gan, D., Tappan, D., Pignataro, C., "ICMP Extensions for Multiprotocol Label Switching", RFC 4950, August 2007.
- [ICMP-Int] Atlas, A., Bonica, R., Pignataro, C., Shen, N., Rivers, JR., "Extending ICMP for Interface and Next-Hop Identification", RFC 5837, April 2010.
- [ICMP-MP] Bonica, R., Gan, D., Tappan, D., Pignataro, C., "Extended ICMP to Support Multi-Part Messages", RFC 4884, April 2007.

- [ICMPv4] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, September 1981.
- [ICMPv6] Conta, A., Deering, S., and M. Gupta, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 4443, March 2006.
- [IEEE802.1Q] IEEE 802.1Q, "IEEE Standard for Local and metropolitan area networks - Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks", October 2012.
- [IEEE802.3ah] IEEE 802.3, "IEEE Standard for Information technology - Local and metropolitan area networks - Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications", clause 57, December 2008.
- [IntHost] Braden, R., "Requirements for Internet Hosts -- Communication Layers", RFC 1122, October 1989.
- [IPPM-1DM] Almes, G., Kalidindi, S., Zekauskas, M., "A One-way Delay Metric for IPPM", RFC 2679, September 1999.
- [IPPM-1LM] Almes, G., Kalidindi, S., Zekauskas, M., "A One-way Packet Loss Metric for IPPM", RFC 2680, September 1999.
- [IPPM-2DM] Almes, G., Kalidindi, S., Zekauskas, M., "A Round-trip Delay Metric for IPPM", RFC 2681, September 1999.
- [IPPM-Con] Mahdavi, J., Paxson, V., "IPPM Metrics for Measuring Connectivity", RFC 2678, September 1999.
- [IPPM-FW] Paxson, V., Almes, G., Mahdavi, J., and Mathis, M., "Framework for IP Performance Metrics", RFC 2330, May 1998.
- [ITU-G8113.1] ITU-T Recommendation G.8113.1/Y.1372.1, "Operations, Administration and Maintenance mechanism for MPLS-TP in Packet Transport Network (PTN)", November 2012.
- [ITU-G8113.2] ITU-T Recommendation G.8113.2/Y.1372.2, "Operations, administration and maintenance mechanisms for MPLS-TP networks using the tools defined for MPLS", November 2012.



- [ITU-T-CT] Betts, M., "Allocation of a Generic Associated Channel Type for ITU-T MPLS Transport Profile Operation, Maintenance, and Administration (MPLS-TP OAM)", RFC 6671, November 2012.
- [ITU-T-G.806] ITU-T Recommendation G.806, "Characteristics of transport equipment - Description methodology and generic functionality", January 2009.
- [ITU-T-Y1711] ITU-T Recommendation Y.1711, "Operation & Maintenance mechanism for MPLS networks", February 2004.
- [ITU-T-Y1731] ITU-T Recommendation G.8013/Y.1731, "OAM Functions and Mechanisms for Ethernet-based Networks", July 2011.
- [ITU-Terms] ITU-R/ITU-T Terms and Definitions, online, 2013.
- [L2TP-EC] McGill, N. and C. Pignataro, "Layer 2 Tunneling Protocol Version 3 (L2TPv3) Extended Circuit Status Values", RFC 5641, August 2009.
- [L2VPN-OAM] Sajassi, A., Mohan, D., "Layer 2 Virtual Private Network (L2VPN) Operations, Administration, and Maintenance (OAM) Requirements and Framework", RFC 6136, March 2011.
- [L3VPN-OAM] El Mghazli, Y., Nadeau, T., Boucadair, M., Chan, K., Gonguet, A., "Framework for Layer 3 Virtual Private Networks (L3VPN) Operations and Management", RFC 4176, October 2005.
- [Lock-Loop] Boutros, S., Sivabalan, S., Aggarwal, R., Vigoureux, M., Dai, X., "MPLS Transport Profile Lock Instruct and Loopback Functions", RFC 6435, November 2011.
- [LSP-Ping] Kompella, K., Swallow, G., "Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures", RFC 4379, February 2006.
- [Mng] Farrel, A., "Inclusion of Manageability Sections in Path Computation Element (PCE) Working Group Drafts", RFC 6123, February 2011.
- [MPLS-ENCAPS] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T. and A. Conta, "MPLS Label Stack Encoding", RFC 3032, January 2001.

- [MPLS-LM-DM] Frost, D., Bryant, S., "Packet Loss and Delay Measurement for MPLS Networks", RFC 6374, September 2011.
- [MPLS-OAM] Nadeau, T., Morrow, M., Swallow, G., Allan, D., Matsushima, S., "Operations and Management (OAM) Requirements for Multi-Protocol Label Switched (MPLS) Networks", RFC 4377, February 2006.
- [MPLS-OAM-FW] Allan, D., Nadeau, T., "A Framework for Multi-Protocol Label Switching (MPLS) Operations and Management (OAM)", RFC 4378, February 2006.
- [MPLS-P2MP] Yasukawa, S., Farrel, A., King, D., Nadeau, T., "Operations and Management (OAM) Requirements for Point-to-Multipoint MPLS Networks", RFC 4687, September 2006.
- [MPLS-TP-OAM] Vigoureux, M., Ward, D., Betts, M., "Requirements for OAM in MPLS Transport Networks", RFC 5860, May 2010.
- [mtrace] Fenner, W., Casner, S., "A "traceroute" facility for IP Multicast", draft-ietf-idmr-traceroute-ipm-07 (expired), July 2000.
- [NetTerms] Jacobsen, O., Lynch, D., "A Glossary of Networking Terms", RFC 1208, March 1991.
- [NetTools] Enger, R., Reynolds, J., "FYI on a Network Management Tool Catalog: Tools for Monitoring and Debugging TCP/IP Internets and Interconnected Devices", RFC 1470, June 1993.
- [OAM-Analys] Sprecher, N., Fang, L., "An Overview of the OAM Tool Set for MPLS based Transport Networks", RFC 6669, July 2012.
- [OAM-Label] Ohta, H., "Assignment of the 'OAM Alert Label' for Multiprotocol Label Switching Architecture (MPLS) Operation and Maintenance (OAM) Functions", RFC 3429, November 2002.
- [OAM-Mng] Ersue, M., Claise, B., "An Overview of the IETF Network Management Standards", RFC 6632, June 2012.

- [OnDemand-CV] Gray, E., Bahadur, N., Boutros, S., Aggarwal, R. "MPLS On-Demand Connectivity Verification and Route Tracing", RFC 6426, November 2011.
- [OWAMP] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and Zekauskas, M., "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, September 2006.
- [PARIS] Brice Augustin, Timur Friedman and Renata Teixeira, "Measuring Load-balanced Paths in the Internet", IMC, 2007.
- [PM-CONS] Clark, A. and B. Claise, "Guidelines for Considering New Performance Metric Development", BCP 170, RFC 6390, October 2011.
- [PW-ACH] Bryant, S., Swallow, G., Martini, L., McPherson, D., "Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN", RFC 4385, February 2006.
- [PW-G-ACh] Li, H., Martini, L., He, J., Huang, F., "Using the Generic Associated Channel Label for Pseudowire in the MPLS Transport Profile (MPLS-TP)", RFC 6423, November 2011.
- [PW-MAP] Aissaoui, M., Busschbach, P., Martini, L., Morrow, M., Nadeau, T., and Y(J). Stein, "Pseudowire (PW) Operations, Administration, and Maintenance (OAM) Message Mapping", RFC 6310, July 2011.
- [Reorder] Morton, A., Ciavattone, L., Ramachandran, G., Shalunov, S., and J. Perser, "Packet Reordering Metrics", RFC 4737, November 2006.
- [Signal] Yasukawa, S., "Signaling Requirements for Point-to-Multipoint Traffic-Engineered MPLS Label Switched Paths (LSPs)", RFC 4461, April 2006.
- [TCPIP-Tools] Kessler, G., Shepard, S., "A Primer On Internet and TCP/IP Tools and Utilities", RFC 2151, June 1997.
- [TP-CC-CV] Allan, D., Swallow, G., Drake, J., "Proactive Connectivity Verification, Continuity Check and Remote Defect indication for MPLS Transport Profile", RFC 6428, November 2011.

- [TP-Fault] Swallow, G., Fulignoli, A., Vigoureux, M., Boutros, S., "MPLS Fault Management Operations, Administration, and Maintenance (OAM)", RFC 6427, November 2011.
- [TP-LM-DM] Frost, D., Bryant, S., "A Packet Loss and Delay Measurement Profile for MPLS-Based Transport Networks", RFC 6375, September 2011.
- [TP-OAM-FW] Busi, I., Allan, D., "Operations, Administration and Maintenance Framework for MPLS-based Transport Networks ", RFC 6371, September 2011.
- [TP-Term] Van Helvoort, H., Andersson, L., Sprecher, N., "A Thesaurus for the Terminology used in MPLS Transport Profile (MPLS-TP) Internet-Drafts and RFCs in the Context of the ITU-T's Transport Network Recommendations", RFC 7087, December 2013.
- [TRILL-OAM] Senevirathne, T., Bond, D., Aldrin, S., Li, Y., Watve, R., "Requirements for Operations, Administration, and Maintenance (OAM) in Transparent Interconnection of Lots of Links (TRILL)", RFC 6905, March 2013.
- [TWAMP] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and Babiarz, J., "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, October 2008.
- [VCCV] Nadeau, T., Pignataro, C., "Pseudowire Virtual Circuit Connectivity Verification (VCCV): A Control Channel for Pseudowires", RFC 5085, December 2007.
- [VCCV-SURVEY] Del Regno, N., Malis, A., "The Pseudowire (PW) and Virtual Circuit Connectivity Verification (VCCV) Implementation Survey Results", RFC 7079, November 2013.

## Appendix A.

### List of OAM Documents

#### A.1. List of IETF OAM Documents

Table 5 summarizes the OAM related RFCs published by the IETF.

It is important to note that the table lists various RFCs that are different by nature. For example, some of these documents define OAM tools or OAM protocols (or both), while others define protocols that

are not strictly OAM-related, but are used by OAM tools. The table also includes RFCs that define the requirements or the framework of OAM in a specific context (e.g., MPLS-TP).

The RFCs in the table are categorized in a few sets as defined in Section 1.3.

Toolset	Title	RFC
IP Ping	Requirements for Internet Hosts -- Communication Layers [IntHost]	RFC 1122
	A Glossary of Networking Terms [NetTerms]	RFC 1208
	Internet Control Message Protocol [ICMPv4]	RFC 792
	Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification [ICMPv6]	RFC 4443
IP Traceroute	A Primer On Internet and TCP/IP Tools and Utilities [TCPIP-Tools]	RFC 2151
	FYI on a Network Management Tool Catalog: Tools for Monitoring and Debugging TCP/IP Internets and Interconnected Devices [NetTools]	RFC 1470
	Internet Control Message Protocol [ICMPv4]	RFC 792
	Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification [ICMPv6]	RFC 4443
	Extended ICMP to Support Multi-Part Messages [ICMP-MP]	RFC 4884

	Extending ICMP for Interface and Next-Hop Identification [ICMP-Int]	RFC 5837
BFD	Bidirectional Forwarding Detection [BFD]	RFC 5880
	Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop) [BFD-IP]	RFC 5881
	Generic Application of Bidirectional Forwarding Detection [BFD-Gen]	RFC 5882
	Bidirectional Forwarding Detection (BFD) for Multihop Paths [BFD-Multi]	RFC 5883
	Bidirectional Forwarding Detection for MPLS Label Switched Paths (LSPs) [BFD-LSP]	RFC 5884
	Bidirectional Forwarding Detection for the Pseudowire Virtual Circuit Connectivity Verification (VCCV) [BFD-VCCV]	RFC 5885
MPLS OAM	Operations and Management (OAM) Requirements for Multi-Protocol Label Switched (MPLS) Networks [MPLS-OAM]	RFC 4377
	A Framework for Multi-Protocol Label Switching (MPLS) Operations and Management (OAM) [MPLS-OAM-FW]	RFC 4378
	Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures [LSP-Ping]	RFC 4379
	Operations and Management (OAM) Requirements for Point-to-Multipoint MPLS Networks [MPLS-P2MP]	RFC 4687

MPLS-TP OAM	ICMP Extensions for Multiprotocol Label Switching [ICMP-Ext]	RFC 4950
	Bidirectional Forwarding Detection for MPLS Label Switched Paths (LSPs) [BFD-LSP]	RFC 5884
	Requirements for OAM in MPLS-TP [MPLS-TP-OAM]	RFC 5860
	MPLS Generic Associated Channel [G-ACh]	RFC 5586
	MPLS-TP OAM Framework [TP-OAM-FW]	RFC 6371
	Proactive Connectivity Verification, Continuity Check, and Remote Defect Indication for the MPLS Transport Profile [TP-CC-CV]	RFC 6428
	MPLS On-Demand Connectivity Verification and Route Tracing [OnDemand-CV]	RFC 6426
	MPLS Fault Management Operations, Administration, and Maintenance (OAM) [TP-Fault]	RFC 6427
	MPLS Transport Profile Lock Instruct and Loopback Functions [Lock-Loop]	RFC 6435
	Packet Loss and Delay Measurement for MPLS Networks [MPLS-LM-DM]	RFC 6374
	A Packet Loss and Delay Measurement Profile for MPLS-Based Transport Networks [TP-LM-DM]	RFC 6375
Pseudowire	Pseudowire Virtual Circuit	RFC 5085

OAM	Connectivity Verification (VCCV): A Control Channel for Pseudowires [VCCV]	
	Bidirectional Forwarding Detection for the Pseudowire Virtual Circuit Connectivity Verification (VCCV) [BFD-VCCV]	RFC 5885
	Using the Generic Associated Channel Label for Pseudowire in the MPLS Transport Profile (MPLS-TP) [PW-G-ACh]	RFC 6423
	Pseudowire (PW) Operations, Administration, and Maintenance (OAM) Message Mapping [PW-MAP]	RFC 6310
	MPLS and Ethernet Operations, Administration, and Maintenance (OAM) Interworking [Eth-Int]	RFC 7023
OWAMP and TWAMP	A One-way Active Measurement Protocol [OWAMP]	RFC 4656
	A Two-Way Active Measurement Protocol [TWAMP]	RFC 5357
	Framework for IP Performance Metrics [IPPM-FW]	RFC 2330
	IPPM Metrics for Measuring Connectivity [IPPM-Con]	RFC 2678
	A One-way Delay Metric for IPPM [IPPM-1DM]	RFC 2679
	A One-way Packet Loss Metric for IPPM [IPPM-1LM]	RFC 2680
	A Round-trip Delay Metric for IPPM	RFC 2681



	[IPPM-2DM]	
	Packet Reordering Metrics [Reorder]	RFC 4737
	A One-Way Packet Duplication Metric [Dup]	RFC 5560
TRILL OAM	Requirements for Operations, Administration, and Maintenance (OAM) in Transparent Interconnection of Lots of Links (TRILL)	RFC 6905

Table 5 Summary of IETF OAM Related RFCs

## A.2. List of Selected Non-IETF OAM Documents

In addition to the OAM tools defined by the IETF, the IEEE and ITU-T have also defined various OAM tools that focus on Ethernet, and various other transport network environments. These various tools, defined by the three standard organizations, are often tightly coupled, and have had a mutual effect on each other. The ITU-T and IETF have both defined OAM tools for MPLS LSPs, [ITU-T-Y1711] and [LSP-Ping]. The following OAM standards by the IEEE and ITU-T are to some extent linked to IETF OAM tools listed above and are mentioned here only as reference material:

- o OAM tools for Layer 2 have been defined by the ITU-T in [ITU-T-Y1731], and by the IEEE in 802.1ag [IEEE802.1Q] . The IEEE 802.3 standard defines OAM for one-hop Ethernet links [IEEE802.3ah].
- o The ITU-T has defined OAM for MPLS LSPs in [ITU-T-Y1711], and MPLS-TP OAM in [ITU-G8113.1] and [ITU-G8113.2].

It should be noted that these non-IETF documents deal in many cases with OAM functions below the IP layer (Layer 2, Layer 2.5) and in some cases operators use a multi-layered OAM approach, which is a function of the way their networks are designed.

Table 6 summarizes some of the main OAM standards published by non-IETF standard organizations. This document focuses on IETF OAM standards, but these non-IETF standards are referenced in this document where relevant.

	Title	Standard/Draft
ITU-T MPLS OAM	Operation & Maintenance mechanism for MPLS networks [ITU-T-Y1711]	ITU-T Y.1711
	<p>Assignment of the 'OAM Alert Label' for Multiprotocol Label Switching Architecture (MPLS) Operation and Maintenance (OAM) Functions [OAM-Label]</p> <p>Note: although this is an IETF document, it is listed as one of the non-IETF OAM standards, since it was defined as a complementary part of ITU-T Y.1711.</p>	RFC 3429
ITU-T MPLS-TP OAM	<p>Operations, administration and Maintenance mechanisms for MPLS-TP networks using the tools defined for MPLS [ITU-G8113.2]</p> <p>Note: this document describes the OAM toolset defined by the IETF for MPLS-TP, whereas ITU-T G.8113.1 describes the OAM toolset defined by the ITU-T.</p>	ITU-T G.8113.2
	Operations, Administration and Maintenance mechanism for MPLS-TP in Packet Transport Network (PTN)	ITU-T G.8113.1
	<p>Allocation of a Generic Associated Channel Type for ITU-T MPLS Transport Profile Operation, Maintenance, and Administration (MPLS-TP OAM) [ITU-T-CT]</p> <p>Note: although this is an IETF document, it is listed as one of the</p>	RFC 6671

	non-IETF OAM standards, since it was defined as a complementary part of ITU-T G.8113.1.	
ITU-T Ethernet OAM	OAM Functions and Mechanisms for Ethernet-based Networks [ITU-T-Y1731]	ITU-T Y.1731
IEEE CFM	Connectivity Fault Management [IEEE802.1Q]  Note: CFM was originally published as IEEE 802.1ag, but is now incorporated in the 802.1Q standard.	IEEE 802.1ag
IEEE DDCFM	Management of Data Driven and Data Dependent Connectivity Faults [IEEE802.1Q]  Note: DDCFM was originally published as IEEE 802.1Qaw, but is now incorporated in the 802.1Q standard.	IEEE 802.1ag
IEEE 802.3 link level OAM	Media Access Control Parameters, Physical Layers, and Management Parameters for Subscriber Access Networks [IEEE802.3ah]  Note: link level OAM was originally defined in IEEE 802.3ah, and is now incorporated in the 802.3 standard.	IEEE 802.3ah

Table 6 Non-IETF OAM Standards Mentioned in this Document

Authors' Addresses

Tal Mizrahi  
Marvell  
6 Hamada St.  
Yokneam, 20692  
Israel

Email: [talmi@marvell.com](mailto:talmi@marvell.com)

Nurit Sprecher  
Nokia Solutions and Networks  
3 Hanagar St. Neve Ne'eman B  
Hod Hasharon, 45241  
Israel

Email: [nurit.sprecher@nsn.com](mailto:nurit.sprecher@nsn.com)

Elisa Bellagamba  
Ericsson  
6 Farogatan St.  
Stockholm, 164 40  
Sweden

Phone: +46 761440785  
Email: [elisa.bellagamba@ericsson.com](mailto:elisa.bellagamba@ericsson.com)

Yaacov Weingarten  
34 Hagefen St.  
Karnei Shomron, 4485500  
Israel

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



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

E. Beili  
Actelis Networks  
December 10, 2013

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

Abstract

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

Status of This Memo

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

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

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

This Internet-Draft will expire on June 13, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

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

described in the Simplified BSD License.

#### Table of Contents

1. Introduction . . . . .	3
2. The Internet-Standard Management Framework . . . . .	3
3. Mapping between EFM-CU-MIB and IEEE8023-EFM-CU-MIB . . . . .	3
4. Updating the MIB Modules . . . . .	4
5. Security Considerations . . . . .	4
6. IANA Considerations . . . . .	5
7. Acknowledgments . . . . .	5
8. References . . . . .	5
8.1. Normative References . . . . .	5
8.2. Informative References . . . . .	6

## 1. Introduction

RFC 5066 [RFC5066] defines two MIB modules:

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

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

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

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

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

## 2. The Internet-Standard Management Framework

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

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

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

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



} sub-tree.

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

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

#### 4. Updating the MIB Modules

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

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

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

#### 5. Security Considerations

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

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

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

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

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

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

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

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

## 6. IANA Considerations

No action is required from IANA.

## 7. Acknowledgments

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

Dan Romascanu

David Harrington

Michael MacFaden

Tom Petch

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

## 8. References

### 8.1. Normative References

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

- [RFC3414] Blumenthal, U. and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", STD 62, RFC 3414, December 2002.
- [RFC3826] Blumenthal, U., Maino, F., and K. McCloghrie, "The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model", RFC 3826, June 2004.
- [RFC5066] Beili, E., "Ethernet in the First Mile Copper (EFMCu) Interfaces MIB", RFC 5066, November 2007.

## 8.2. Informative References

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

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

Author's Address

Edward Beili  
Actelis Networks  
Bazel 25  
Petach-Tikva  
Israel

Phone: +972-73-237-6852  
EMail: edward.beili@actelis.com



Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: April 13, 2014

D. Satyanarayana  
V. Prakash  
Cisco Systems  
October 10, 2013

Local Auth MIB  
draft-sdanda-localauth-mib-01

## Abstract

This draft defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it describes managed objects for managing Locally authenticated users.

## 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 BCP 14, RFC 2119.

## Status of This Memo

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

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

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

This Internet-Draft will expire on April 13, 2014.

## Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

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

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

## Table of Contents

1. The Internet-Standard Management Framework . . . . .	2
2. Introduction . . . . .	2
3. Terminology . . . . .	3
4. Brief Description of MIB Objects . . . . .	3
4.1. Local Auth User Table (localAuthUserTable) . . . . .	3
5. Local Auth User MIB Module Definitions . . . . .	3
6. Security Considerations . . . . .	11
7. IANA Considerations . . . . .	12
8. References . . . . .	13
8.1. Normative References . . . . .	13
8.2. Informative References . . . . .	13
Appendix A. Acknowledgments . . . . .	13

## 1. 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.

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, STD 58, RFC 2579 and STD 58, RFC 2580.

## 2. Introduction

Authentication, Authorization and Accounting enables the user to control the access of the system resources. Dedicated AAA servers cannot be used for small enterprise network deployments that provide network access to hundreds of users. For such scenarios, the user information or profiles can be stored locally at the network element.

This MIB can be used by the central controller to manage Local authentication information on the central controller. One of the use-cases would be to monitor user access on multiple vendor devices like - user login/logout notifications - user account lifetime expiry notifications - User account creation/deletion notifications

This draft defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it describes managed objects to monitor Local authenticated users.

Comments should be made directly to the opsawg@ietf.org mailing alias.

### 3. Terminology

This document adopts the definitions, acronyms and mechanisms described in [RFC2903]. Unless otherwise stated, the mechanisms described therein will not be re-described here.

### 4. Brief Description of MIB Objects

This section describes objects pertaining to Local Authenticated users with specific information related to the MIB module specified in this document.

The Local Authenticated MIB has one module named LocalAuthMIB which is focussed on describing users authenticated locally by Network Access Server.

#### 4.1. Local Auth User Table (localAuthUserTable)

The localAuthUserTable lists the currently configured local users. For each user object, it provides information and statistics about the local users.

### 5. Local Auth User MIB Module Definitions

```
LOCAL-AUTH-MIB DEFINITIONS ::= BEGIN
```

```
IMPORTS
```

```
    MODULE-IDENTITY,  
    OBJECT-TYPE,  
    NOTIFICATION-TYPE,  
    Counter32,  
    Unsigned32,  
    mib-2  
        FROM SNMPv2-SMI  
    MODULE-COMPLIANCE,  
    NOTIFICATION-GROUP,  
    OBJECT-GROUP  
        FROM SNMPv2-CONF  
    TruthValue,  
    DateAndTime
```



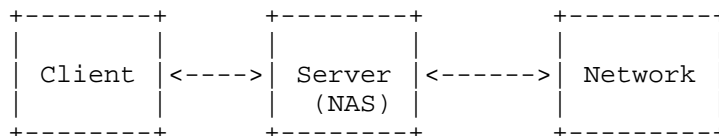
```
FROM SNMPv2-TC
SnmpAdminString
FROM SNMP-FRAMEWORK-MIB;
```

```
localAuthMIB MODULE-IDENTITY
    LAST-UPDATED      "201305090000Z"
    ORGANIZATION      "Operations and Management Area
                        Working Group"
    CONTACT-INFO
        "Satyanarayana Danda,
         Cisco Systems, Inc
         Email: sdanda@cisco.com

         Prakash Vijayaragavan
         Cisco Systems, Inc
         Email: pravijay@cisco.com"
```

#### DESCRIPTION

"This MIB module defines objects describing users authenticated locally by a Network Access Server (NAS).



A client is a telnet or SSH user needing access to the NAS box directly. Network user like PPP or dot1x will request NAS box for authentication to access the network.

NAS box authenticates user present in the local user database.

#### GLOSSARY

##### Network Access Server (NAS)

A single point of access to a remote resource and is exclusively used with Authentication, Authorization and Accounting.

##### Point-to-Point Protocol (PPP)

A data link protocol commonly used in establishing a direct connection between two networking nodes.

##### Secure Shell (SSH)

It is a cryptographic network protocol for secure data communication.

dot1x  
dot1x also known as IEEE 802.1X is an IEEE Standard for Port-based Network Access Control."

REVISION "201305100000Z"

DESCRIPTION

"Initial version of MIB"

::= { mib-2 999 }

-- Default Notification Type

localAuthMIBNotifs OBJECT IDENTIFIER

::= { localAuthMIB 0 }

-- Local authenticated user MIB object definition

localAuthMIBObjects OBJECT IDENTIFIER

::= { localAuthMIB 1 }

localAuthMIBConform OBJECT IDENTIFIER

::= { localAuthMIB 2 }

-- Notification Configuration

localAuthNotifEnable OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This object specifies whether the system generates localAuthUserAdded, localAuthUserDeleted, localAuthUserLoggedIn and localAuthUserLoggedOut notifications."

DEFVAL { false }

::= { localAuthMIBObjects 1 }

localAuthUserTable OBJECT-TYPE

SYNTAX SEQUENCE OF LocalAuthUserEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table lists the currently configured local users."

```
::= { localAuthMIBObjects 2 }

localAuthUserEntry OBJECT-TYPE
    SYNTAX          LocalAuthUserEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "An entry describes a local user identified by its index.

        An entry is created or modified when a user is defined in
        system through configuration. An entry is removed when
        a user is undefined with configuration commands via CLI
        or by automatic expiry of users when lifetime of the user is
        expired."
    INDEX            { localAuthUserIndex }
    ::= { localAuthUserTable 1 }

LocalAuthUserEntry ::= SEQUENCE {
    localAuthUserIndex      Unsigned32,
    localAuthUserName       SnmpAdminString,
    localAuthUserType       INTEGER,
    localAuthUserCreationTime DateAndTime,
    localAuthUserLifetime   Unsigned32,
    localAuthUserLoginSuccessCount Counter32,
    localAuthUserLoginFailureCount Counter32,
    localAuthUserLastLoginTime DateAndTime,
    localAuthUserOTPEnabled TruthValue,
    localAuthUserPrivilegeLevel Unsigned32,
    localAuthUserLoginStatus TruthValue,
    localAuthUserPasswordLifetime Unsigned32
}

localAuthUserIndex OBJECT-TYPE
    SYNTAX          Unsigned32 (1..4294967295)
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This object indicates an integer-value that uniquely
        identifies a local user."
    ::= { localAuthUserEntry 1 }

localAuthUserName OBJECT-TYPE
    SYNTAX          SnmpAdminString
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "A textual string containing the name of the locally
        authenticated user."
```

```
::= { localAuthUserEntry 2 }

localAuthUserType OBJECT-TYPE
    SYNTAX          INTEGER {
                        defaultUser(1),
                        lobbyUser(2),
                        managementUser(3),
                        networkUser(4),
                        guestUser(5)
                    }
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "This object indicates the type of local user:

        defaultUser      - Default user account type.
        lobbyUser        - Management user with lobby admin
                           privileges, can create and manage
                           guest user account type.
        managementUser   - Management user account type.
        networkUser      - User requires accessing the network.
        guestUser        - Type of networkUser with lifetime configured
                           such that they can stay alive for a given
                           time period and will expire thereafter."
    ::= { localAuthUserEntry 3 }

localAuthUserCreationTime OBJECT-TYPE
    SYNTAX          DateAndTime
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "This object indicates the time the local user was created."
    ::= { localAuthUserEntry 4 }

localAuthUserLifetime OBJECT-TYPE
    SYNTAX          Unsigned32
    UNITS           "seconds"
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "This object indicates the expiry duration of the local user;
        that is, the duration the local user is valid from the
        creation time."
    ::= { localAuthUserEntry 5 }

localAuthUserLoginSuccessCount OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
```

```
STATUS          current
DESCRIPTION
    "This object indicates the number of times, the user
    logged-in successfully."
 ::= { localAuthUserEntry 6 }

localAuthUserLoginFailureCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "This object indicates the number of times, the user failed
        to authenticate successfully."
    ::= { localAuthUserEntry 7 }

localAuthUserLastLoginTime OBJECT-TYPE
    SYNTAX      DateAndTime
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "This object indicates the last time the local user was
        logged in successfully."
    ::= { localAuthUserEntry 8 }

localAuthUserOTPEnabled OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "This object specifies whether One Time Password is
        enabled for the user."
    ::= { localAuthUserEntry 9 }

localAuthUserPrivilegeLevel OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "This object indicates the privilege level of the
        local user."
    ::= { localAuthUserEntry 10 }

localAuthUserLoginStatus OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "This object indicates the current login status of
```

```
        the local user."
 ::= { localAuthUserEntry 11 }

localAuthUserPasswordLifetime OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "seconds"
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "This object indicates the expiry duration of the
        password of the local user."
 ::= { localAuthUserEntry 12 }

localAuthMIBCompliances OBJECT IDENTIFIER
 ::= { localAuthMIBConform 1 }

localAuthUserAdded NOTIFICATION-TYPE
    OBJECTS      {
                    localAuthUserName,
                    localAuthUserType,
                    localAuthUserLifetime
                }
    STATUS      current
    DESCRIPTION
        "This notification indicates when the system has added a
        user."
 ::= { localAuthMIBNotifs 1 }

localAuthUserDeleted NOTIFICATION-TYPE
    OBJECTS      {
                    localAuthUserName,
                    localAuthUserType
                }
    STATUS      current
    DESCRIPTION
        "This notification indicates when the system has deleted a
        user."
 ::= { localAuthMIBNotifs 2 }

localAuthUserLoggedIn NOTIFICATION-TYPE
    OBJECTS      {
                    localAuthUserName,
                    localAuthUserType
                }
    STATUS      current
    DESCRIPTION
```

```
        "This notification indicates when the user has logged
        into the system."
 ::= { localAuthMIBNotifs 3 }

localAuthUserLoggedIn NOTIFICATION-TYPE
    OBJECTS          {
                        localAuthUserName,
                        localAuthUserType
                      }
    STATUS            current
    DESCRIPTION
        "This notification indicates when the user has logged
        out of the system"
 ::= { localAuthMIBNotifs 4 }

localAuthUserPasswordExpired NOTIFICATION-TYPE
    OBJECTS          {
                        localAuthUserName,
                        localAuthUserType
                      }
    STATUS            current
    DESCRIPTION
        "This notification indicates when the user password
        is expired."
 ::= { localAuthMIBNotifs 5 }

localAuthMIBGroups OBJECT IDENTIFIER
 ::= { localAuthMIBConform 2 }

localAuthMIBCompliance MODULE-COMPLIANCE
    STATUS            current
    DESCRIPTION
        "This is a default module-compliance
        containing default object groups."
    MODULE            -- this module
    MANDATORY-GROUPS {
                        localAuthMIBMainObjectGroup,
                        localAuthMIBNotificationGroup
                      }
 ::= { localAuthMIBCompliances 1 }

-- Units of Conformance

localAuthMIBMainObjectGroup OBJECT-GROUP
    OBJECTS          {
```

```

        localAuthNotifEnable,
        localAuthUserType,
        localAuthUserCreationTime,
        localAuthUserLifetime,
        localAuthUserName,
        localAuthUserLoginSuccessCount,
        localAuthUserLoginFailureCount,
        localAuthUserLastLoginTime,
        localAuthUserOTPEnabled,
        localAuthUserPrivilegeLevel,
        localAuthUserLoginStatus,
        localAuthUserPasswordLifetime
    }
    STATUS          current
    DESCRIPTION
        "The is a local Authenticated User MIB Main Object group."
    ::= { localAuthMIBGroups 1 }

localAuthMIBNotificationGroup NOTIFICATION-GROUP
    NOTIFICATIONS    {
        localAuthUserAdded,
        localAuthUserDeleted,
        localAuthUserLoggedIn,
        localAuthUserLoggedOut,
        localAuthUserPasswordExpired
    }
    STATUS          current
    DESCRIPTION
        "The is a local Authenticated User MIB
        Notification group."
    ::= { localAuthMIBGroups 2 }

END

```

## 6. Security Considerations

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

Management object localAuthNotifEnable can be modified by the network operators which will effect in large number of notification being generated by the NAS.



localAuthUserName object exposed via this MIB may not be considered as a risk for an attacker. Username as an identity in the network transport would mostly be a clear test. If this object is not exposed via MIB, intruder can get this information via packet capture or by any other means. With knowing username, risk can be mitigated by enforcing strong password encryption schemes.

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

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

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

## 7. 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
-----	-----
localAuthUserMIB	{ mib-2 XXX }

[Editor's Note (to be removed prior to publication): the IANA is requested to assign a value for "XXX" under the 'mib-2' subtree and to record the assignment in the SMI Numbers registry. When the assignment has been made, the RFC Editor is asked to replace "XXX" (here and in the MIB module) with the assigned value and to remove this note.]

## 8. References

### 8.1. Normative References

[RFC2903] de Laat, C., Gross, G., Gommans, L., Vollbrecht, J., and D. Spence, "Generic AAA Architecture", RFC 2903, August 2000.

### 8.2. Informative References

[RFC4001] Daniele, M., Haberman, B., Routhier, S., and J. Schoenwaelder, "Textual Conventions for Internet Network Addresses", RFC 4001, February 2005.

[RFC3413] Levi, D., Meyer, P., and B. Stewart, "Simple Network Management Protocol (SNMP) Applications", STD 62, RFC 3413, December 2002.

## Appendix A. Acknowledgments

Authors would like to thank Mouli Chandramouli, Peddareddappa Gonichettipalli, Arun Kudur, Naresh Sunkara and Biju Raju for their comments and suggestions.

## Authors' Addresses

Satyanarayana Danda  
Cisco Systems

EMail: sdanda@cisco.com

Prakash Vijayaragavan  
Cisco Systems

EMail: pravijay@cisco.com

OPSAWG  
Internet-Draft  
Intended status: Informational  
Expires: April 21, 2014

H. Song  
Huawei  
Z. Cao  
China Mobile  
October 18, 2013

The Problems of Virtual Network Function Configuration  
draft-song-opsawg-virtual-network-function-config-01

Abstract

This document describes the problem space of remote service installation and configuration in the provider's network through a centralized management system. This is a typical scenario for virtual function installation and dynamic configuration in network function virtualization (NFV) context. It is also a typical scenario in cloud computing environment where end users do not have to install applications in their end hosts, but can install their own featured powerful software application in the cloud. This specification also identifies the scope that needs standardization based on the problems.

Status of This Memo

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

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

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

This Internet-Draft will expire on April 21, 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. Background . . . . .	2
2. Terminology . . . . .	5
3. Problems of Service Configuration . . . . .	5
4. Scope for Standardization . . . . .	7
5. Security Considerations . . . . .	8
6. Acknowledgments . . . . .	8
7. References . . . . .	8
7.1. Normative References . . . . .	8
7.2. Informative References . . . . .	8
Authors' Addresses . . . . .	9

## 1. Background

This document describes the problems in the context of remote virtual network function installation and service configuration through a central management system, called a controller. Four main roles are involved, including the user, the software provider, the controller and the infrastructure resources.

Users always have different requirements when they need a software. So a software vendor often provides a "full-set" of functions to satisfy a majority of users in the market. But for each individual user it might only need a few sub-set functions according to his own requirements. For example, a firewall could provide many functions, including anti-DDoS attack, anti-phishing attack, IP filtering function, MAC filtering function, network address translator function, and etc.. But a home user who is going to install a virtual firewall from the network operator, (which may be installed in a virtual machine inside the provider's network and the operator can make the traffic from/to this user go through that virtual machine,) may contempt his own environment and determines that he only needs anti-phishing attack function and MAC filtering function for his traffic. Other functions may be not useful for this user, for example, DDoS attacks may not happen to this user in this context. Typically, these functions exist in the software as different components. There are several possibilities for the user to acquire the relative components that he needs:

(1) A software vendor distinguishes users as several classes, and provides related versions of software to the users accordingly, for example, a "home edition" version, an "enterprise edition" version and etc. In this case, the specific version may satisfy most users in that class, but for each individual user, it may also contain many function components that the user does not need.

(2) When a user requests a software, the user negotiates with a customer service person from the software vendor about his requirements, and the software vendor makes a specified version of software to the user, in this version, it enables the components that the user need, and disables those unneeded. In this case, it costs more human energy, and is not efficient. The user has to wait days or even longer for his specific software version after the negotiation.

(3) The user get a license and software packet, and with the license, it allows the user to choose inside a range of components for installation. The user enables components that he wants in that range. In this case, it gives the user more flexibility to operate the software components, but from another perspective, it also authorizes the user with more components than the user wants.

These methods either too complex, or authorize the user with more components than what he wants. A real-time, exact matching and flexible way for the user to choose his software components is desirable.

In the context of network function virtualization (NFV), more and more network functions become to be available in a virtualized function way. It adopts the common IT infrastructure instead of physical hardware box to implement these network functions. The benefits of this method is to reduce cost through improved infrastructure reusability and lower entry of the industry, which allows more software vendors. Various virtual functions exist in the network. They are deployed into virtual machines through the NFV controller. These virtual functions can be replaced with new virtual functions when needed, with only re-configuring it with a new software through the controller. In this case, NFV controller is just like a broker for many software applications.

The user may also have his own requirements or want to put his constraints on the network properties of the software that is to be installed in the provider's network. For example, the bandwidth requirements for this software. This is different from the virtual machine level or host level bandwidth limitation. It is an application specific bandwidth limitation. A number of software applications can be installed on a same virtual machine, and they

share the bandwidth of this virtual machine. But they are different applications and have different requirements on the bandwidth. In this case, the user specifies his requirements on the bandwidth of this software, and the NFV controller will enforce that constraints to the application level through some kind of configuration.

Besides the network properties, during the remote installation, the user would also need to notify the controller about the virtual network function's storage space requirements, memory requirements, CPU requirements, operating system requirements, location constraints for the software installation. These constraints can be mapped to a virtual machine if a virtual network function is mapped to a single virtual machine, or a subset resources of a virtual machine if multiple virtual network functions share a virtual machine. It is the controller's responsibility to select the most appropriate hardware resources for the virtual network function based on some mapping algorithm, but it is totally an implementation issue. These constraints are also relative to the software vendor, and the software vendor should describe the basic hardware and software requirements of the installation environment to the user, and the user should combine it with capacity requirements from himself and make a final decision on these parameters.

The previous two paragraphs describe the explicit way between the user and the controller for resource requirements. But there could also be an implicit way. In the implicit way, the user only describes what software components he needs. But the controller will choose appropriate resources for the user. When the user needs more capacity due to the service expansion, for example, in a scenario where an enterprise has a virtual firewall in the provider's network, the controller is responsible for the redirection of the traffic from or to this enterprise go through the virtual firewall. The configuration could be that the controller sends flow rules to the network equipment, so as to forward the related traffic to the virtual firewall. And the virtual firewall notifies the controller about its traffic load status. If the load is above some threshold, the controller will automatically create new virtual firewall instances, and allocate additional CPU/memory/storage/bandwidth resources to handle that traffic. The controller will configure new flow rules to make a portion of the traffic go through the new instances. If the traffic volume is shrinking, the controller will automatically reduce the number of virtual instances for the user. The resource requirements for each virtual instance can be from the recommendation of the software provider. This kind of automatic scale-out and scale-in mechanism can make a better utilization of the network, computing and storage resources. And users do not need to have a deep understanding of the resource consumption model, but only pay as much as the resources he used.

In the context of this document, the operator owns the controller. OSS/BSS for the NFV service is part of the controller. The network administrator from the operator can also act as a user to install and configure VNFs in the provider's (operator's) network for the customers or for himself. Note that the controller could be third party SP software components for NFV, and in this case, there should be the interface for the operator to configure the controller beforehand. But we assume the controller has already been configured for the NFV services in this document. And the configuration of the controller itself is out of scope.

## 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]. And the following terms used in this document have their definitions from the NFV end to end architecture [NFVE2E].

NFV: network function virtualization. NFV technology uses the commodity servers to replace the dedicated hardware boxes for the network functions, for example, home gateway, enterprise access router, carrier grade NAT and etc. So as to improve the reusability, allow more vendors into the market, and reduce time to market. NFV architecture includes a NFV controller (orchestrator) to manage the virtual network functions and the infrastructure resources. (Note: will use terms defined by ETSI NFV ISG in the next version.)

NF: A functional building block within an operator's network infrastructure, which has well-defined external interfaces and a well-defined functional behaviour. Note that the totality of all network functions constitutes the entire network and services infrastructure of an operator/service provider. In practical terms, a Network Function is today often a network node or physical appliance.

vNF: virtual network function, an implementation of an executable software program that constitutes the whole or a part of an NF that can be deployed on a virtualisation infrastructure.

VM: virtual machines, a program and configuration of part of a host computer server. Note that the Virtual Machine inherits the properties of its host computer server e.g. location, network interfaces.

## 3. Problems of Service Configuration

There are several problems in the context of remote software installation, which makes it different from the traditional ways.

First, it is a remote operation. For example, in the NFV framework, a software is installed according to the user (home user, enterprise user or the operator network administrator) requirements through NFV controller. It is not installed locally in the user's equipment, but remotely in the provider's network. NFV's control center needs to coordinate the necessary infrastructure resources for the installation. So the user does not have direct control over the software installation position or the hardware and software resources. But the controller has the direct control. In a result, the user needs to interact with the controller to accomplish the configuration of the components and his preferred locations for a software installation. This process includes the service parameters selection and event notification, for example, the operating system selection, software components selection, installation location selection, notification of the number of the instances of the same virtual network functions, installation status notification and etc.

Service chaining also needs a descriptor from the user. A user can directly establish a service chain. For the service graph, there can be two layers, one is the stable virtual/physical link layer, and the other is conditional/stable service forwarding layer.

Second, the NFV controller is just like a broker for various software applications. There are different methods for the software installation. A proprietary method is that every software vendor has a plug-in in the NFV controller platform, and each end user uses the proprietary messages to interact with that software vendor's plug-in for the software installation. Another way is using standard messages to allow users to select their preferred software components for all different kinds of software installation. The drawback for every software vendor has its own proprietary messages for the software installation component configuration, is that it will make both the controller and the user environment more complex. A uniform and standard component configuration is more appropriate for this context.

Third, if the software vendor does not provide a clear description of these software components, then users do not know how to choose among those components. So the controller also needs a standard format to communicate with the software vendors, so as to acquire the detailed descriptions of the software components.

Fourth, dynamic configuration is another problem. A user may want to change its service configuration when the software is running. In the traditional context, a user logs into the server, and changes the



service template in the server, then save it. It may become effect immediately or after reboot. But in the context of NFV, a user's virtual function may be installed in many virtual machines. It gets too complex if we let the user maintains the installed virtual machines information and logs into each virtual machine to reconfigure the service template one by one. A centralized service template configuration modification is much more easier. The controller may be or not be aware of the meaning of these dynamic configurations, But it needs to know that this is a configuration file and the range of VMs that it applies to.

There are also resource requirements for the remote software installation, which are complement to the software components selection. There is lack of a standard for a user to tell the controller how much bandwidth, storage, CPU, memory are allocated to a specific software in the provider's network (perhaps there is existing standard for the virtual machine or host level resources), or just tell the controller allocate the resources dynamically for him. Note that VNF level and VM level resource allocation are different. Because one VM might be holding multiple VNFs. The resource allocation interfaces provided by some platforms such like Openstack-Neutron are combined with a VM, and are not easy to make a change. But here when the resource is combined with a VNF. It is much easier to change it.

A recommended resource requirements notification for a service instance is also needed between a software vendor and the controller.

#### 4. Scope for Standardization

The key point is the information model. Network Function Virtualization needs standard information model so as to improve the interoperation. How to represent the user's functional and resource requirements, and how to map and apply these requirements to the underlying infrastructure is the key point to success. This specification on the stage only focuses on the virtual network function level at the beginning, but virtual overlay network of network functions should be extended in the near future. The narrowed scope for the current stage is:

(1) A protocol between the user and controller for software installation components choices, dynamic service configuration through the controller, and the resource requirements for the installation.

(2) A protocol between software vendor and the controller for the detailed description of the software components and the recommended resource requirements for service instance.

Existing protocols can be extended for the description of NFV service configurations and related resource requirements. YANG has deployment in the existing network. So if YANG is extended for the VNF configuration and modeling, it can be seamlessly integrated with other related network management systems.

JavaScript Object Notation is a text-based open standard, so it is human readable. There is also effort in IETF to use JSON to describe the application layer information, as well as the network layer information. For example, in ALTO protocol [I-D.ietf-alto-protocol], it uses JSON to describe the routing cost between peers of network hosts.

## 5. Security Considerations

This document does not introduce any new security threats. But for any solution to solve these problems, authentication is required between a user and the controller to verify whether the user is authorized to install that software. And the messages among the user, the controller and software vendor must be encrypted to prevent from interception attack.

## 6. Acknowledgments

The authors would like to thank Zhen Cao for his valuable 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.

### 7.2. Informative References

[I-D.ietf-alto-protocol]  
Alimi, R., Penno, R., and Y. Yang, "ALTO Protocol", draft-ietf-alto-protocol-20 (work in progress), October 2013.

[NFVE2E] , "Network Functions Virtualisation: End to End Architecture, <http://docbox.etsi.org/ISG/NFV/70-DRAFT/0010/NFV-0010v016.zip>", .

Authors' Addresses

Haibin Song  
Huawei

Email: [haibin.song@huawei.com](mailto:haibin.song@huawei.com)

Cao Zhen  
China Mobile

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

Internet Working Group

W. Xu  
Y. Jiang  
C. Zhou  
Huawei

Internet Draft

Intended status: Informational

Expires: February 2014

September 01, 2013

Problem Statement of Network Functions Virtualization Model  
draft-xjz-nfv-model-problem-statement-00.txt

Status of this Memo

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

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 February 30, 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.

## Abstract

This document discusses the problem space of the Network Functions Virtualisation (NFV) models in the NFV environment. Possible use cases and the language for the models will also be identified.

## Table of Contents

1.	Introduction .....	2
2.	Conventions used in this document .....	3
3.	Terminology .....	4
4.	Problems and Use Cases .....	5
5.	Aspects of the Problems .....	6
5.1.	NFV Modeling Objectives .....	6
5.2.	Modeling Language considerations .....	7
6.	Related IETF work.....	7
7.	Security Considerations .....	9
8.	IANA Considerations .....	9
9.	References .....	9
9.1.	Normative References .....	9
9.2.	Informative References .....	9
10.	Acknowledgments .....	9

## 1. Introduction

Network Functions Virtualisation (NFV), as currently being in progress in ETSI NFV, leverages standard IT virtualisation technology to consolidate many network equipment types onto industry standard high volume servers, switches and storage. The traditional physical network device could be implemented using the Virtualized Network Function (VNF) with the introduction of the NFV technology, which will bring some benefit, e.g., reduced costs, increased speed of Time to Market, one platform for different applications, users and tenants, and inspired service innovation.

The NFV Orchestrator (NFVO) is one of the key entities in the end-to-end NFV reference architecture, which manages and coordinates VNFs and the respective NFVI provides a northbound interface with some candidate protocols to offer the NFV management and orchestration functions to other entities. The E2E NFV architecture abstracts the network function information model to the external network. However, those protocols do not include a modeling language or accompanying

rules that can be used to model the management information that is to be configured using protocol.

Another key entity in ETSI NFV is the VNF. We need an abstracted information model for VNFD (Virtual Network Function Descriptor) to specify the configuration data model for the virtual network function. The VNFD describes the resource requirements of the VNF, the VNF Forwarding Graphs describe how service providers wish to have multiple VNFs connected together to form customized services for end users, and the NFV service describes how a network service can be realized using NFs. The VNF configuration data model which is included by the VNFD may need a standard data modeling language to specify it. This allows the OSS/orchestrator to dynamically extract and parse the data model from the VNFD at run-time and to implement some rudimentary flow-through provisioning of any service. VNF Forwarding Graphs and NFV service also need a standard data modeling language for their description.

The NETCONF Data Modeling Language (netmod) working group in the Internet Engineering Task Force (IETF) has defined the data modeling language YANG and has developed a set of YANG data models and other activities for configuration and management of network elements. Since the VNF Forwarding Graphs are organized independent of the existing network topology and protocols, a new set of data models (e.g., YANG) may be needed to provide implementation guidance for the operators to configure and manage the virtual network functions.

Therefore, NFV model (NFVMOD) is proposed to be studied in IETF, with the goal to provide management entities with standardized information they can use to deploy, instantiate, configure and manage network services based on NFV infrastructure. The information is organized with VNFD, the VNF Forwarding Graph and the NFV Service. Network topologies constructed with Network Service Chaining (NSC mechanism) may also be in the scope of NFVMOD if its work starts. The models to be developed in NFVMOD may also help the understanding of NFV and the development of NSC protocol. Finally, the modeling language used for NFVMOD may be YANG.

## 2. Conventions used in this document

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

### 3. Terminology

CMS: Cloud Management System.

Network Function (NF): A functional building block within an operator's network infrastructure, which has well-defined external interfaces and a well-defined functional behaviour.

NF set: A collection of NFs with unspecified connectivity between them.

Network Functions Virtualization Orchestrator (NFVO): a function that deploys, operates, manages, and coordinates VNFs and the respective NFVI. The Orchestrator has control and visibility of all VNF running inside the NFVI.

Network Functions Virtualization Infrastructure (NFVI): the totality of all hardware and software components that constitute the environment in which VNFs are deployed, managed and executed. The NFVI includes resources for computation, networking and storage.

Physical Network Function (PNF): An implementation of a NF via a tightly coupled software and hardware system.

Virtual Machine (VM): A program and configuration of part of a host computer server. Note that the Virtual Machine inherits the properties of its host computer server e.g. location, network interfaces.

Virtualised Network Function (VNF): An implementation of an executable software program that constitutes the whole or a part of an NF and can be deployed on a virtualisation infrastructure.

VNFD: Virtualized Network Function Description, used to describe all properties associated in the NFV infrastructure.

VNF Forwarding Graph: A graph specified by a Network Service Provider of bi-directional logical links connecting NF nodes where at least one node is a VNF through which network traffic is directed.

NFV Service: A network service utilizing NFs, where at least some NFs are VNFs. A VNF Forwarding Graph is an example of such a service.

#### 4. Problems and Use Cases

Much of the following materials and all definitions are brought from the ongoing ETSI NFV work, which may undergo frequent changes in the future.

From a top down viewpoint of NFV (that is, from an end to end service to its serving infrastructure), three layers of logical abstraction are defined in NFV architecture: NFV Service, VNF Forwarding Graph and VNF.

##### -NFV Service

NFV Service is a network service utilizing NFs, where at least some NFs are VNFs. It can be regarded as a service presentation layer.

An NFV service requires a modeling language which can describe how an end to end service is to be assembled including which VNFs and PNFs are required and how they are connected, and the relevant service parameters (e.g. relationship with other VNFs in terms of latency, co-location, etc). It can optionally contain a reliability class description that specifies the reliability, availability, and scale metrics for each VNF.

##### -VNF Forwarding Graph

A VNF forwarding graph requires a modeling language which can describe how a forwarding graph is formed, including which VNFs are required and how forwarding from one VNF to another VNF is to be realized. It can also describe the link options available in the infrastructure, such as E-LAN, E-LINE, and E-TREE services. These enable the connection of VNFs to other VNFs or the existing networks that can be selected by the NFVO and be used to configure the Infrastructure network.

##### -VNF

A VNF also requires a detailed data model, written in a standardized data modeling language, describing its configurable parameters, operational state variables, operations and notifications, thus enabling the resources required for an instance to be deployed by the NFVO and CMS onto an appropriate server. Five separate logical interfaces are envisioned (see GS NFV-SWA001) to enable effective orchestration of a VNF.



These modeling languages must have a well-defined grammar in textual form so that automation tools can process and translate the models described in them.

These modeling languages must have well-defined upgrade rules, so that clients (OSS and/or Orchestrators) can work with different versions of a VNF data model. Following these upgrade rules ensures that upgrade scenarios are handled properly in the network.

ETSI NFV has proposed some information models on the NFV architecture, e.g., NFVMAN(13)20\_003r4. But it is not clear which modeling language will be used and what is the data model for configuration and management when using these languages.

## 5. Aspects of the Problems

Some considerations on the NFV modeling issue are provided in this section.

### 5.1. NFV Modeling Objectives

The main objective of VNF modeling may include:

- Basic VNF attributes: VNF name, function description, sharing or non-sharing attribute.
- Deployment attributes: environment requirements of VNF deployment such as the number of VMs, virtual CPU, memory and disk requirements, image of each VM, and QoS requirements such as bandwidth and delay of VNF.
- Operational attributes: which defines the operational and management behavior, such as start, stop, pause, migration and etc.
- Interface attributes: external interface, such as interface type, configuration parameters of these interfaces.

The main objective of VNF Forwarding Graph modeling may include:

- VNFs in a Forwarding Graph.
- VNF connectivity, such as virtual links between VNFs.
- Attributes of service flows in a VNF forwarding graph.

The main objective of NFV service modeling may include:

- VNFs and PNFs in an NFV service.
- VNF connectivity between PNFs and VNFs.
- Service attributes of an NFV service, such as entry and exit point, QoS of an NFV service, and etc.

## 5.2. Modeling Language considerations

YANG is a modular language representing data structures in an XML tree format. A YANG module defines a hierarchy of data that can be used for NETCONF-based operations, including configuration, state data, Remote Procedure Calls (RPCs), and notifications. This allows a complete description of all data sent between a NETCONF client and server.

The IETF NETMOD working group has defined the data modeling language YANG and a set of core YANG data models, which can be used by network operators for the configuration and management of their physical network elements.

YANG can also be used for NFV data modeling, the advantages include:

- Its simplicity and flexibility in modeling semantics.
- Consistence with the modeling of its physical network elements, thus facilitate an end to end service constructed with both virtual and physical network functions, and across both virtual and physical networks.

## 6. Related IETF work

The following subsections discuss related IETF work and are provided for reference. They may be deleted when the document is published.

- NVO3

NVO3 WG is mainly developing the tunnel technology for DC inter-connection, and solving the problem of VM migration.

The NVO3 mechanism can be used to accommodate the needs of end to end services in NFV, especially when the virtualization platforms (e.g., server pools) involve multiple DCs.

The virtual link between Virtual Network Functions (VNFs) can also use NVO3 technology as a tunneling mechanism (if intra DC protocols such as VXLAN and NVGRE are adopted into charter and specified there).

Thus, NVO3 at most solves the problem of supporting virtual links. When VMs serving a VNF (or VNFs) migrate from one DC to another DC, NVO3 can also have its use.

#### - NSC

Service chaining is a broad term used to describe a common model for delivering multiple services in a specific order. Service chaining decouples service delivery from the underlying network topology and creates a dynamic services plane that addresses the requirements of cloud and virtual application delivery. Packets and/or flows that require service chaining are classified and redirected to the appropriate, available services. Additionally, context can be shared between the network and the services.

During the 87th IETF meeting, NSC BoF had triggered quite a lot of interests in the attendees. Though the charter of NSC is still unclear thus far, it is believed to cover the routing of a service chain, i.e., providing an end to end path to serve a service chain.

As shown in discussions, NSC can provide routing and forwarding mechanisms for service chains. Its main focus is on data plane, and some control plane issues may also be involved.

#### -NETMOD

The NETCONF Working Group has completed a base protocol to be used for configuration management. However, the NETCONF protocol does not include a modeling language or accompanying rules that can be used to model the management information that is to be configured using NETCONF. The NETMOD working group has defined the data modeling language YANG but no IETF models exist yet. The purpose of the NETMOD working group is to support the ongoing deployment of YANG by developing a set of core YANG data models and other activities that will allow network operators to use YANG for configuration and management of network elements.

The NETMOD Working Group currently works on the following items: Core system data model, Core interface data model, Core routing data model

and Data model for configuring SNMP engines. Virtual network functions and virtual network is not in the scope of NETMOD.

## 7. Security Considerations

TBD.

## 8. IANA Considerations

No IANA action is needed for this document.

## 9. References

### 9.1. Normative References

### 9.2. Informative References

ETSI GS NFV-SWA001

ETSI NFVMAN(13)20

## 10. Acknowledgments

TBD

Authors' Addresses

Weiping Xu  
Huawei Technologies Co., Ltd.  
Bantian, Longgang district  
Shenzhen 518129, China  
Email: xuweiping@huawei.com

Yuanlong Jiang  
Huawei Technologies Co., Ltd.  
Bantian, Longgang district  
Shenzhen 518129, China  
Email: jiangyuanlong@huawei.com

Cathy Zhou  
Huawei Technologies Co., Ltd.  
Bantian, Longgang district  
Shenzhen 518129, China  
Email: cathy.zhou@huawei.com

Expires February 30, 2014

[Page 10]



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   |

```

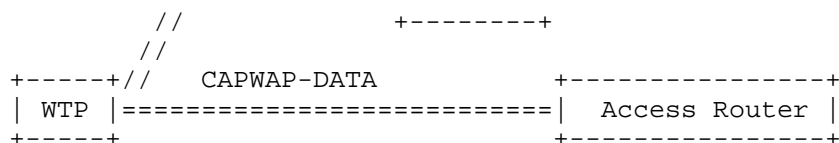


Figure 1: Split between CAPWAP-CTL and CAPWAP-DATA Plane

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.

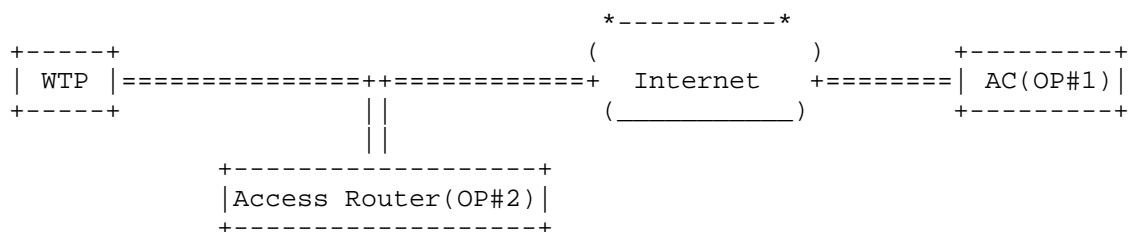


Figure 2: Access Service and Authentication Service Provided by different Operators

### 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.

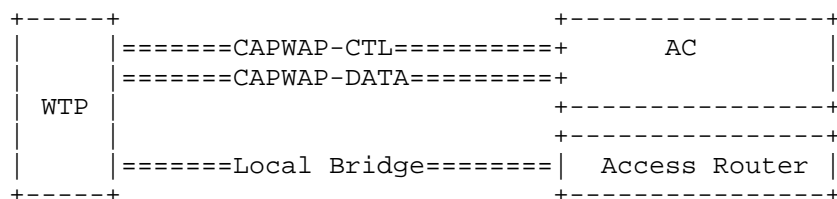


Figure 3: Local Bridging Model

### 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.

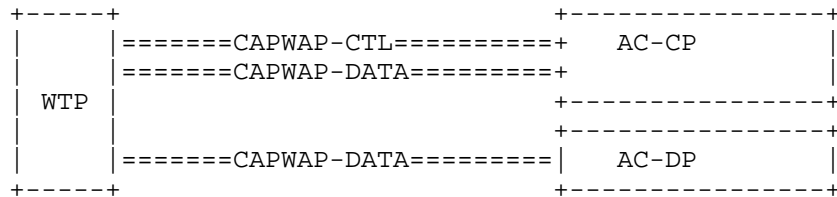


Figure 4: Multiple DATA tunnels Model

## 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

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: September 6, 2014

R. Zhang  
China Telecom  
Z. Cao  
H. Deng  
China Mobile  
R. Pazhyannur  
S. Gundavelli  
Cisco  
L. Xue  
Huawei  
March 5, 2014

Alternate Tunnel Encapsulation for Data Frames in CAPWAP  
draft-zhang-opsawg-capwap-cds-03

Abstract

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

Status of This Memo

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

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

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



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

This Internet-Draft will expire on September 6, 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 . . . . .	2
1.1. Conventions used in this document . . . . .	5
1.2. Terminology . . . . .	5
2. Alternate Tunnel Encapsulation . . . . .	6
2.1. Description . . . . .	6
2.2. Supported Alternate Tunnel Encapsulations . . . . .	8
2.3. Alternate Tunnel Encapsulations Type . . . . .	8
3. IANA Considerations . . . . .	9
4. Security Considerations . . . . .	9
5. Contributors . . . . .	10
6. References . . . . .	10
6.1. Normative References . . . . .	10
6.2. Informative References . . . . .	10
Authors' Addresses . . . . .	10

#### 1. Introduction

Service Providers are deploying very large Wi-Fi deployments (ranging from hundreds of thousands of APs to millions of APs). These networks are designed to carry traffic generated from mobile users. The volume in mobile user traffic is already very large (in the order of petabytes per day) and expected to continue growing rapidly. As a result, operators are looking for solutions that can scale to meet the increasing demand. One way to meet the scalability requirement is to split the control/management plane from the data plane. This separation enables the data plane be scaled independently of the

control/management plane. This document provides a description of a CAPWAP specification change that enables the separation of data plane from control plane.

CAPWAP ([RFC5415], [RFC5416]) defines a tunnel mode that specifies the frame tunneling type to be used for 802.11 data frames from stations associated with the WLAN. The following types are supported:

- o Local Bridging: All user traffic is to be locally bridged.
- o 802.3 Tunnel: All user traffic is to be tunneled to the AC in 802.3 format.
- o 802.11 Tunnel: All user traffic is to be tunneled to the AC in 802.11 format.

There are two shortcomings with currently specified tunneled modes: 1) it does not allow the WTP to tunnel data frames to an endpoint different from the AC and 2) it does not allow the WTP to tunnel data frames using any encapsulation other than CAPWAP (as specified in Section 4.4.2 of [RFC5415]). Next, we describe what is driving the above mentioned two requirements.

Some operators deploying large number of Access Points prefer to centralize the management and control of Access Points while distributing the handling of data traffic to increase scaling. This motivates an architecture as shown in Figure 1 that has the AC in a centralized location and one or more tunnel gateways (or Access Routers) that terminate the data tunnels from the various WTPs. This split architecture has two benefits over an architecture where data traffic is aggregated at the AC: 1) reduces the scale requirement on data traffic handling capability of the AC and 2) leads to more efficient/optimal routing of data traffic.

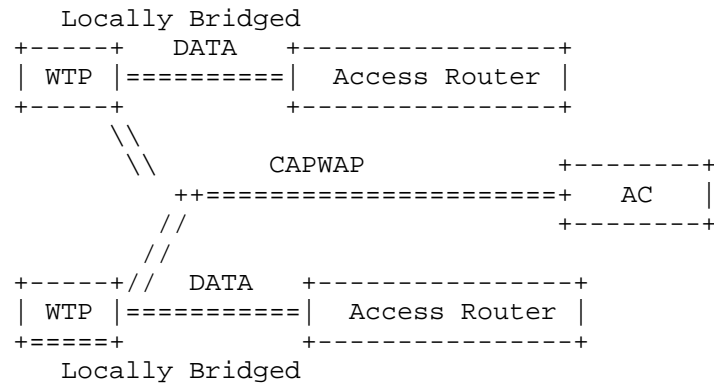


Figure 1: Centralized Control with Distributed Data

The above system (shown in Figure 1) could be achieved by setting the tunnel mode to Local bridging. In such a case the AC would handle control of WTPs as well as handle the management traffic to/from the stations. There is CAPWAP Control and Data Channel between the WTP and the AC. The CAPWAP Data channel carries the IEEE 802.11 management traffic (like IEEE 802.11 Action Frames). The station's data frames are locally bridged, i.e., not carried over the CAPWAP data channel. The station's data frames are handled by the Access Router. However, in many deployments the operator managing the WTPs/AC may be different from the operator providing the internet connectivity to the WTPs. Further, the WTP operator may want (or be required by legal/regulatory requirements) to tunnel the traffic back to an Access Router in its network as shown in Figure 2. The tunneling requirement may be driven by the need to apply policy at the Access Router or a legal requirement to support lawful intercept of user traffic. What this means is that local bridging does not meet their requirements. Their requirements are met either by having the WTP tunnel the station's traffic to the AC or the WTP support an alternate tunnel, i.e., a tunnel to an alternate entity different from the AC. This is the motivation for Alternate Tunnel encapsulation support where the data tunnels from the WTP are terminated at an AR (and more specifically at an end point different from the AC).

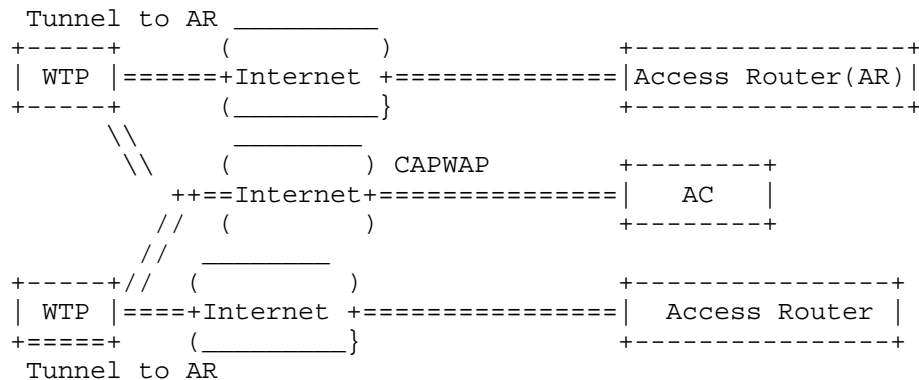


Figure 2: Centralized Control with Distributed Data

In the case where the WTP is tunneling data frames to an AR (and not the AC), the choice of tunnel encapsulation need not be restricted only to CAPWAP (as described in Section 4.4.2 of [RFC5415]). In fact, the WTP may additionally support other widely used encapsulation types such as L2TP, L2TPv3, IP-in-IP, IP/GRE, etc. The WTP may advertise the different alternate tunnel encapsulation types supported and the AC can select one of the supported encapsulation types. As shown in the figure there is still a CAPWAP control and data channel between the WTP and AC wherein the CAPWAP data channel carries the stations' management traffic. Thus the WTP will maintain three tunnels: CAPWAP Control, CAPWAP Data, and another (alternate) tunnel to the AR. The main reason to maintain a CAPWAP data channel is to minimize the changes on the WTP and AC required to transport stations' management frames (like EAP, IEEE 802.11 Action Frames). These management frames are transported over the CAPWAP data channel as they are done for case when the WTP's tunnel mode is configured as the local bridging. In this specification we describe how the WTP can be configured with this alternate tunnel.

### 1.1. Conventions used in this document

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

### 1.2. Terminology

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

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

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

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

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

## 2. Alternate Tunnel Encapsulation

### 2.1. Description

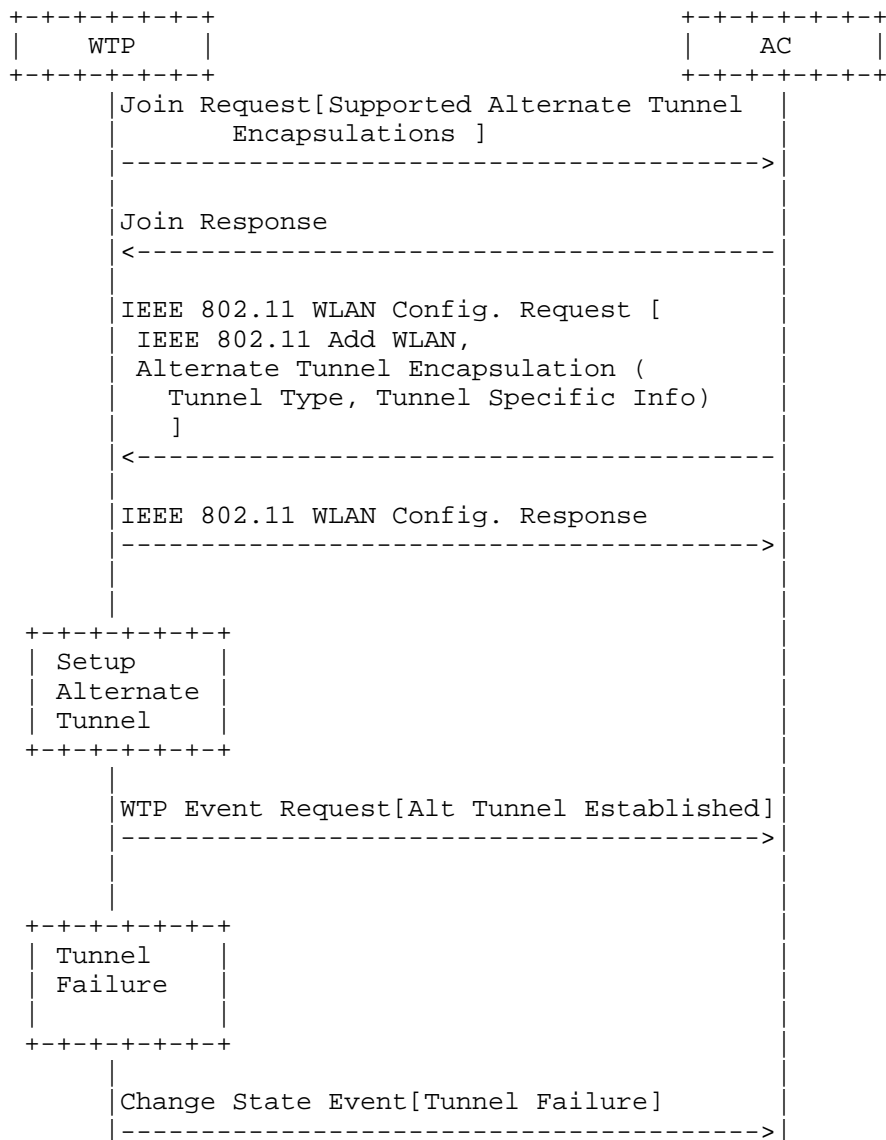


Figure 3: Setup of Alternate Tunnel

The above example describes how the alternate tunnel encapsulation may be established. When the WTP joins the AC, it should indicate its alternate tunnel encapsulation capability. The AC would determine whether an alternate tunnel configuration is required. If required, it would select an appropriate alternate tunnel encapsulation. The AC provides the alternate tunnel encapsulation

message element that provides both the tunnel-type and tunnel specific information. The tunnel specific information may contain configuration information to help the WTP setup the tunnel. For example, the IP address of the access router that will terminate the WTP tunnel. Once the WTP sets up the tunnel, the WTP may inform the AC about the tunnel setup. Correspondingly, if the WTP discovers that the tunneled link to the AR has failed, then it may inform the AC.

## 2.2. Supported Alternate Tunnel Encapsulations

This message element enables a WTP to communicate its capability to support alternate tunnel encapsulations to the AC. The WTP may communicate its capability during the discovery or join process.

```

      0           1           2           3
      0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0
      +-----+-----+-----+-----+-----+-----+
      | Num_Tunnels | Tunnel_1 | Tunnel_[2..N]..
      +-----+-----+-----+-----+-----+-----+

```

Figure 4: Supported Alternate Tunnel Encapsulations

- o Type: TBD for Supported Tunnel Encapsulations
- o Num\_Tunnels >=1: This refers to number of profiles present in this message element. There must be at least one profile.
- o Tunnel: Each Tunnel is identified by value defined in the Tunnel Type field in Section 2.3

## 2.3. Alternate Tunnel Encapsulations Type

The IEEE 802.11 Alternate Tunnel Encapsulation message element allows the AC to select the alternate tunnel encapsulation. This message element may be provided along with the IEEE 802.11 Add WLAN message element. When the message element is present the following fields of the IEEE 802.11 Add WLAN element shall be set as follows: MAC mode is set to 0 (Local MAC) and Tunnel Mode is set to 0 (Local Bridging).

```

      0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
      +-----+-----+-----+-----+-----+-----+-----+
      | Tunnel Type | Tunnel Specific
      |             | Information
      +-----+-----+-----+-----+-----+-----+-----+

```

Figure 5: Alternate Tunnel Encapsulations Type

- o Type: TBD for Alternate Tunnel Encapsulation Type

- o Tunnel Type: The profile is identified by a value given below
  - \* 0: CAPWAP data channel as described in [RFC5415][RFC5416]
  - \* 1: L2TP
  - \* 2: L2TPv3
  - \* 3: IP-in-IP
  - \* 4: IP/GRE
- o Tunnel Specific Information: This field contains tunnel specific information that is used to configure the WTP with parameters needed for alternate tunnel setup.

```

  0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| Length          | Data                                           |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

```

Figure 6: Tunnel Specific Information

- \* Length:
- \* Data: The data field would contain tunnel specific information to assist the WTP in setting up the alternate tunnel. For example if the tunnel type is CAPWAP then the data field would contain the following (non-exhaustive) list of parameters

- + Access Router IPv4 address
- + Access Router IPv6 address
- + Tunnel DTLS Policy
- + IEEE 802.11 Tagging Policy

This specification only defines a generic container for such message elements. We anticipate that these message elements (for the different protocols) will be defined in separate documents, potentially one for each tunneling protocols. See [I-D.xue-opsawg-capwap-separation-capability] for example of such a specification.

### 3. IANA Considerations

To be specified in later versions

### 4. Security Considerations

To be specified in later versions.



## 5. Contributors

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

## 6. References

### 6.1. Normative References

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

### 6.2. Informative References

- [I-D.xue-opsawg-capwap-separation-capability]  
Xue, L., Du, Z., Liu, D., Zhang, R., and J. Kaippallimalil, "Capability Announcement and AR Discovery in CAPWAP Control and Data Channel Separation", draft-xue-opsawg-capwap-separation-capability-01 (work in progress), October 2013.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

## 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  
No.32 Xuanwumen West Street  
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

Li Xue  
Huawei  
No.156 Beiqing Rd. Z-park, Shi-Chuang-Ke-Ji-Shi-Fan-Yuan, HaiDian District  
Beijing  
China

Email: xueli@huawei.com

OPSAWG  
Internet-Draft  
Intended status: Informational  
Expires: January 4, 2015

H. Zhou  
H. Song  
Huawei  
Q. Fu  
China Mobile  
July 4, 2014

Virtual Network Function Configuration Architecture  
draft-zhou-opsawg-vnf-config-arch-01

Abstract

This document describes the architecture of remote service installation and configuration in the provider's network through a centralized management system. This is a typical scenario for virtual network function installation and dynamic configuration in network function virtualization (NFV) context. It is also a typical scenario in cloud computing environment where end users do not have to install applications on their end hosts, but can install their own featured powerful software application in the cloud. This specification describes an architecture of virtual network function installation and configuration. It is also applicable to other use cases with similar requirements.

Status of This Memo

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

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

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

This Internet-Draft will expire on March 27, 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. Terminology	4
3. Architecture	5
3.1. Design Principals	5
3.2. User-controller	6
3.3. Software Vendor-Controller	7
3.4. Controller-VNF	7
3.5. Controller-Infrastructure	8
4. Security Considerations	8
5. IANA Considerations	8
6. References	8
6.1. Normative References	9
6.2. Informative References	9
Authors' Addresses	9

## 1. Introduction

This document describes the architecture to solve the problems described in [I-D.song-opsawg-virtual-network-function-config], In virtual network function configuration. The network function virtualization (NFV) controller is a broker for various software vendors. It manages how to install a virtual network function in the provider's network according to user's requirements. There needs standard protocols between user and NFV controller to describe the components requirements and resource requirements for a (or a batch of) virtual network function installation and configuration. There also needs a standard protocol between software vendor and NFV controller to describe the software basic information, running environment resource requirements, components information description. The component of a virtual network function may be another virtual network function. And the NFV controller may combine some virtual network functions together to form another virtual network function. If layer N (N can be 1, 2, 3...) virtual network function is consisted of several layer N-1 virtual network functions, the NFV controller must communicate with the user on how to construct the upper layer VNF with lower layer VNFS, and what are the

forwarding sequences and what are the conditions to trigger that forwarding sequence. The resource requirements from the user can be either explicit or implicit. If the resource requirement is "on-demand", then the NFV controller has to communicate with the virtual network function layer to create new service instances when the service load is increasing or delete extra service instances when the service load is decreasing. The resource requirements for each service instance is based on the recommendation from the software vendor.

This specification gives a basic management architecture for virtual network function (VNF) configuration. It lists the main functional modules that plays important roles in VNF configuration, and specifies what each functional module does, and what are the contents communicated among these functional modules, as depicted in Figure 1. The main concepts from this document are from ETSI NFV ISG [NFVE2E]. But there is slightly difference from ETSI. For example, we have a central NFV controller in this architecture that contains OSS/BSS, VNF management and Infrastructure management, unlike ETSI NFV ISG, where it separates the OSS/BSS and a management and orchestration system. In real world implementation, these functional units usually reside in the same physical equipment(s) called network management system.

OSS/BSS (operations support system / business support system) functional module is responsible for communication with users. It provides software (VNF) provider's information to the users, and get components requirements and resource requirements from the users. It also communicates with VNF software vendors to get the software information and the recommended resource information, and etc.. VNF management module is responsible for VNF instance creation/deletion, VNF status management, VNF resource management, and VNF connection management when one VNF is represented as a conditional/static connection of multiple VNFs. Infrastructure management module is responsible for physical layer resource (CPU, memory, storage, bandwidth etc.) allocation for the virtual machine that a VNF is resided on.

This document will describe the basic principle for the architecture first, then give the description of the interfaces and what contents are exchanged among those interfaces. At the end, it will discuss the security threats for the VNF configuration.

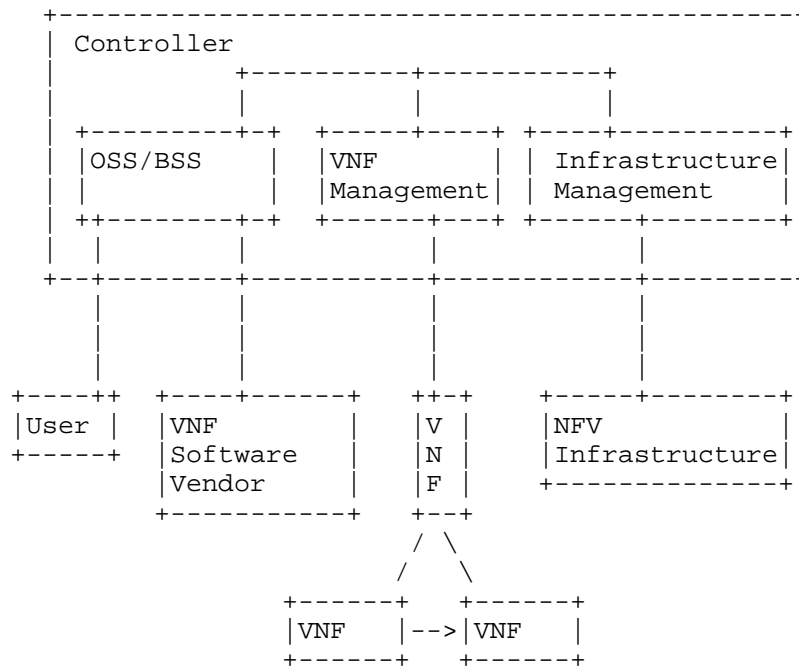


Figure 1 Virtual Network Function Configuration Framework

## 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]. And the following terms used in this document have their definitions from the NFV end to end architecture [NFVE2E].

**User:** a person, home or an enterprise customer who installs their own virtual network function (such as virtual enterprise firewall, virtual home gateway) in the provider's network. A user can also be the provider's network system administrator who want to install a general virtual network function (such like the carrier grade network address translator (NAT) ) that can server various customers inside the provider's network

**NFV:** network function virtualization. NFV technology uses the commodity servers to replace the dedicated hardware boxes for the network functions, for example, home gateway, enterprise access router, carrier grade NAT and etc. So as to improve the reusability, allow more vendors into the market, and reduce time to market. NFV

architecture includes a NFV controller (orchestrator) to manage the virtual network functions and the infrastructure resources.

NF: A functional building block within an operator's network infrastructure, which has well-defined external interfaces and a well-defined functional behavior. Note that the totality of all network functions constitutes the entire network and services infrastructure of an operator/service provider. In practical terms, a Network Function today is often a network node or physical appliance.

vNF: virtual network function, an implementation of an executable software program that constitutes the whole or a part of an NF that can be deployed on a virtualization infrastructure.

VM: virtual machines, a program and configuration of part of a host computer server. Note that the Virtual Machine inherits the properties of its host computer server e.g. location, network interfaces.

### 3. Architecture

#### 3.1. Design Principals

There are five key modules in the architecture. They are controller, software vendor, user, VNF, and infrastructure. And the controller has three key functional modules, OSS/BSS, VNF management, and infrastructure management.

(1) Controller is the brain of all operations. Controller creates virtual network functions in the provider's network. Although the users may communicate with the virtual network functions in the service layer directly, but it has not to directly communicate with the VNF in the management level. A user can have multiple VNFs in the provider's network, and can configure them or a subset of them through a simple communication with the controller. The controller is also a broker for various software vendors. It provides standard interfaces to convey information model with software vendors and users. It also monitors the VNF resource consumption status when the resource consumption model is "on-demand".

(2) Controller MUST NOT be aware of the service logic of each VNF. When a user configures multiple VNFs by sending a configuration template to the controller. The controller does not understand the contents in the configuration template. It only has to know how to apply the configuration to the appropriate VNFs.

(3) The key point for the protocol development SHOULD focus on the information model to describe the VNF itself, the resource requirements, the service/forwarding graph, and the status report.

### 3.2. User-controller

User communicates with the OSS/BSS module of the controller, to query, buy, or configure VNFs. User gets the VNF list from the query interface. Each VNF in the list contains: VNF type, function description, description of components that can be installed optionally, installation resource requirements, possible performance standard (which may include capable number of users per instance, throughput, concurrent connections, and etc.), and pricing information. User can select the appropriate VNF to install from the list. The information that the user can provide to the OSS/BSS before the installation can be the VNF name, quantity, the preferred components of the VNF, the preferred installation locations in high level (user do not have to know the detailed location such like on which virtual machine), and the possible performance requirement at the beginning (which will be considered when allocating resources, so that even the same VNF provided by the same software vendor can have different initial resource allocation when it is installed to serve different customers that have different amount of users/traffic). User should also tell the controller whether it needs specific resources or it only needs resources on-demand. In the meantime, user should also provide the data forwarding graph to the controller, so that the controller can figure out how the new VNF is connected to the existing network, and how the data flow should forward after the installation of this new VNF. Such forwarding graph should be written in a fixed format in spite of the difference of users. User can get the relative report from the controller, the report contains information of which VNFs have been installed, the status of each VNF, the number of VNF instances, logging information, and accounting information. User can also configure its VNFs from the controller, the details of the configuration are relative to the specific VNF vendors. User needs to specify the ID of which VNFs the configuration file is used to configure. When update package is released by the vendor, possible procedure may also need for the OSS/BSS to inquire the user whether he would like to update the VNF of his own. The protocol between the user and OSS/BSS module of controller could be a new protocol or an extension of existing protocols such like HTTP, NetConf, YANG or XML.



### 3.3. Software Vendor-Controller

Software Vendor communicates with the OSS/BSS module of the controller, to publish a VNF, update a VNF, off-the-shelf a VNF. To the details, if a software vendor sends a request to publish a VNF, the identity of the software vendor must be verified before any further action. The software vendor needs to provide the software description information and the software package itself. The description information should include the following: VNF name, classification (classification types can be provided by the controller to the software vendor in advance), function description, components description, installation environment description (operation system, CPU, memory, storage and etc.), and the capacity description under the recommended installation environment (for example, the capable number of users per instance, throughput, concurrent connections, and etc.), as well as the pricing information if needed. Please note that the software package can be submitted to the OSS/BSS together with the description information, or be provided from some out-of-band methods. For example, the software vendor can provide a URL where the OSS/BSS can get this VNF package, or the OSS/BSS provides a URL where the software vendor can upload its package. The software vendor should provide the VNF name and the update package to the controller when updating a certain VNF. Modification of functions, components, and the other basic information should also be provided to the controller. The protocol between software vendor and the OSS/BSS module of controller could be a new protocol or an extension of existing protocols such like HTTP, NetConf, YANG and XML.

### 3.4. Controller-VNF

Another important role for the controller is the VNF management. VNF management module collaborates with the OSS/BSS and infrastructure management module for the VNF creation, deletion, update, monitoring, scale-out/scale-in, and software configuration. OSS/BSS receives the VNF request from the user, and then invokes its interface with the VNF management module to create VNFs. VNF management module gets VMs with requested resources from the infrastructure module, or directly gets the physical host resources. And then it mounts operation system on the VM or the host, then installs the customized VNF(s), sets up the forwarding rules between VNFs according to the forwarding graph provided by the user. VNF management is also in charge of VNF update when update request is made by the vendor. When update request is agreed by the user, the OSS/BSS invoke the update procedure of the VNF management to install the update version VNF(s) on the corresponding VM. Each VNF will report its status to the VNF management module, such like the CPU, memory, storage usage, current traffic volume information, the link usage on the forwarding graph.

User can configure his VNFs through the OSS/BSS interface, but the configurations are finally carried out by the VNF management module. VNF management module can automatically scale-out or scale-in the number of VNF instances according to the load information on the current VNFs. For example, when a user's resource model is "on-demand", then when the VNF management module finds a relative VNF is beyond the load threshold, then it creates another same VNF to offload the relative VNF. And if the VNF function is process network traffic for the user, then VNF management module needs to coordinate with the NFV infrastructure to configure the relative switches (including soft switches), routers (including soft routers), so as to make traffic be divided to different VNFs. And user SHOULD be agnostic of this traffic division operation.

The protocol between the VNF management module can be a new protocol or the extension of existing protocols, such like HTTP, NetConf, and YANG.

### 3.5. Controller-Infrastructure

The third module in the controller is the infrastructure management module. It manages the infrastructure resources that the VNFs are using. It can create, delete, query about VMs. It configures the underlying network, including IP address, VLAN, ACL rules and flow tables. It can utilize OpenStack, CloudStack for some network management system similar to that. Infrastructure management can use the existing open interfaces. Infrastructure management also configure the network for the forwarding graph among the VNFs.

## 4. Security Considerations

Because VNFs are running in the provider's network, so the privacy concern should be the most important aspect that a user would think about. The operator of NFV must guarantee that no third party can access the user's VNFs or any information of the VNFs without the user's permission.

VNFs are generally considered to be run on cloud computing environment, so the security threats to a cloud computing system are also applicable here.

## 5. IANA Considerations

There is no IANA considerations in this document.

## 6. References

## 6.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [I-D.song-opsawg-virtual-network-function-config]  
Song, H., "The Problems of Virtual Network Function Configuration", draft-song-opsawg-virtual-network-function-config-00 (work in progress), September 2013.

## 6.2. Informative References

- [NFVE2E] , "Network Functions Virtualisation: End to End Architecture, <http://docbox.etsi.org/ISG/NFV/70-DRAFT/0010/NFV-0010v016.zip>", .

## Authors' Addresses

Hong Zhou  
Huawei

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

Haibin Song  
Huawei

Email: [haibin.song@huawei.com](mailto:haibin.song@huawei.com)

Fu Qiao  
China Mobile

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