

Network Working Group  
INTERNET-DRAFT  
Category: Standards Track  
Expires: January 11, 2013

A. Bierman  
Yumaworks  
D. Romascanu  
AVAYA  
J. Quittek  
NEC Europe Ltd.  
Mouli Chandramouli  
Cisco Systems, Inc.  
July 10, 2012

Entity MIB (Version 4)  
draft-chandramouli-eman-rfc4133bis-01.txt

## Abstract

This memo 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 used for managing multiple logical and physical entities managed by a single SNMP agent. This document specifies version of the Entity MIB, which obsoletes version 3 (RFC 4133).

## 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 January 3, 2013

#### Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

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

#### Conventions used in this document

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].

## Table of Contents

2. Overview . . . . .	4
2.1. Terms . . . . .	5
2.2. Relationship to Community Strings . . . . .	6
2.3. Relationship to SNMP Contexts . . . . .	6
2.4. Relationship to Proxy Mechanisms . . . . .	7
2.5. Relationship to a Chassis MIB . . . . .	7
2.6. Relationship to the Interfaces MIB . . . . .	7
2.7. Relationship to the Other MIBs . . . . .	8
2.8. Relationship to Naming Scopes . . . . .	8
2.9. Multiple Instances of the Entity MIB . . . . .	8
2.10. Re-Configuration of Entities . . . . .	9
2.11. Textual Convention Change . . . . .	9
2.12. MIB Structure . . . . .	9
2.12.1. entityPhysical Group . . . . .	10
2.12.2. entityLogical Group . . . . .	12
2.12.3. entityMapping Group . . . . .	12
2.12.4. entityGeneral Group . . . . .	13
2.12.5. entityNotifications Group . . . . .	13
2.13. Multiple Agents . . . . .	13
2.14. Changes Since RFC 2037 . . . . .	13
2.14.1. Textual Conventions . . . . .	13
2.14.2. New entPhysicalTable Objects . . . . .	13
2.14.3. New entLogicalTable Objects . . . . .	14
2.14.4. Bug Fixes . . . . .	14
2.15. Changes Since RFC 2737 . . . . .	14
2.15.1. Textual Conventions . . . . .	14
2.15.2. New Objects . . . . .	14
2.15.3. Bug Fixes . . . . .	15
2.16. Changes Since RFC 4133 . . . . .	15
2.16.1. MIB module addition . . . . .	15
2.16.2. Modification to some of the MIB objects . . . . .	15
3. MIB Definitions . . . . .	15
3.1. ENTITY MIB . . . . .	15
3.2. IANA-ENTITY-MIB . . . . .	45
4. Usage Examples . . . . .	49
4.1. Router/Bridge . . . . .	49
4.2. Repeaters . . . . .	55
5. Security Considerations . . . . .	62
6. IANA Considerations . . . . .	63
7. Acknowledgements . . . . .	64
8. Open Issues . . . . .	64
9. References . . . . .	64
9.1. Normative References . . . . .	64
9.2. Informative References . . . . .	65
Authors' Addresses . . . . .	66

## 1. The SNMP 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].

## 2. Overview

There is a need for a standardized way of representing a single agent, which supports multiple instances of one MIB. This is presently true for at least 3 standard MIBs, and is likely to become true for more and more MIBs as time passes. For example:

- multiple instances of a bridge supported within a single device that has a single agent;
- multiple repeaters supported by a single agent;
- multiple OSPF backbone areas, each operating as part of its own Autonomous System, and each identified by the same area-id (e.g., 0.0.0.0), supported inside a single router with one agent.

The single agent present in each of these cases implies a relationship binds these entities. Effectively, there is some "overall" physical entity which houses the sum of the things managed by that one agent, i.e., there are multiple "logical" entities within a single physical entity. Sometimes, the overall physical entity contains multiple (smaller) physical entities, and each logical entity is associated with a particular physical entity. Sometimes, the overall physical entity is a "compound" of multiple physical entities (e.g., a stack of stackable hubs).

What is needed is a way to determine exactly which logical entities are managed by the agent (with some version of SNMP) in order to communicate with the agent about a particular logical entity. When different logical entities are associated with different physical entities within the overall physical entity, it is also useful to be able to use this information to distinguish between logical entities.

In these situations, there is no need for varbinds for multiple logical entities to be referenced in the same SNMP message (although that might be useful in the future). Rather, it is sufficient, and in some situations preferable, to have the context/community in the message identify the logical entity to which the varbinds apply.

Version 2 of this MIB addresses new requirements, which have emerged since the publication of the first Entity MIB (RFC 2037 [RFC2037]). There is a need for a standardized way of providing non-volatile, administratively-assigned identifiers for physical components represented with the Entity MIB. There is also a need to align the Entity MIB with the SNMPv3 administrative framework (STD 62, RFC 3411 [RFC3411]). Implementation experience has shown that additional physical component attributes are also desirable.

Version 3 of this MIB addresses new requirements, which have emerged since the publication of the second Entity MIB (RFC 2737 [RFC2737]). There is a need to identify physical entities that are central processing units (CPUs) and a need to provide a textual convention that identifies an entPhysicalIndex value or zero, where the value zero has application-specific semantics. Two new objects have been added to the entPhysicalTable to identify the manufacturing date and provide additional URIs for a particular physical entity.

## 2.1. Terms

Some new terms are used throughout this document:

### - Naming Scope

A "naming scope" represents the set of information that may be potentially accessed through a single SNMP operation. All instances within the naming scope share the same unique identifier space. For SNMPv1, a naming scope is identified by the value of the associated 'entLogicalCommunity' instance. For SNMPv3, the term 'context' is used instead of 'naming scope'. The complete definition of an SNMP context can be found in section 3.3.1 of RFC 3411 [RFC3411].

### - Multi-Scoped Object

A MIB object, for which identical instance values identify different managed information in different naming scopes, is called a "multi-scoped" MIB object.

### - Single-Scoped Object

A MIB object, for which identical instance values identify the same managed information in different naming scopes, is called a "single-scoped" MIB object.

- Logical Entity

A managed system contains one or more logical entities, each represented by at most one instantiation of each of a particular set of MIB objects. A set of management functions is associated with each logical entity. Examples of logical entities include routers, bridges, print-servers, etc.

- Physical Entity

A "physical entity" or "physical component" represents an identifiable physical resource within a managed system. Zero or more logical entities may utilize a physical resource at any given time. Determining which physical components are represented by an agent in the EntPhysicalTable is an implementation-specific matter. Typically, physical resources (e.g., communications ports, backplanes, sensors, daughter-cards, power supplies, the overall chassis), which can be managed via functions associated with one or more logical entities, are included in the MIB.

- Containment Tree

Each physical component may be modeled as 'contained' within another physical component. A "containment-tree" is the conceptual sequence of entPhysicalIndex values that uniquely specifies the exact physical location of a physical component within the managed system. It is generated by 'following and recording' each 'entPhysicalContainedIn' instance 'up the tree towards the root', until a value of zero indicating no further containment is found.

## 2.2. Relationship to Community Strings

For community-based SNMP, differentiating logical entities is one (but not the only) purpose of the community string (RFC 1157 [RFC1157]). This is accommodated by representing each community string as a logical entity.

Note that different logical entities may share the same naming scope and, therefore, the same values of entLogicalCommunity. This is possible, providing they have no need for the same instance of a MIB object to represent different managed information.

## 2.3. Relationship to SNMP Contexts

Version 2 of the Entity MIB contains support for associating SNMPv3 contexts with logical entities. Two new MIB objects, defining an SnmpEngineID and ContextName pair, are used together to identify an SNMP context associated with a logical entity. This context can be used (in conjunction with the entLogicalTAddress and entLogicalTDomain MIB objects) to send SNMPv3 messages on behalf of a particular logical entity.

#### 2.4. Relationship to Proxy Mechanisms

The Entity MIB is designed to allow functional component discovery. The administrative relationships between different logical entities are not visible in any Entity MIB tables. A Network Management System (NMS) cannot determine whether MIB instances in different naming scopes are realized locally or remotely (e.g., via some proxy mechanism) by examining any particular Entity MIB objects.

The management of administrative framework functions is not an explicit goal of the Entity MIB WG at this time. This new area of functionality may be revisited after some operational experience with the Entity MIB is gained.

Note that for community-based versions of SNMP, a network administrator will likely be able to associate community strings with naming scopes that have proprietary mechanisms, as a matter of configuration. There are no mechanisms for managing naming scopes defined in this MIB.

#### 2.5. Relationship to a Chassis MIB

Some readers may recall that a previous IETF working group attempted to define a Chassis MIB. No consensus was reached by that working group, possibly because its scope was too broad. As such, it is not the purpose of this MIB to be a "Chassis MIB replacement", nor is it within the scope of this MIB to contain all the information which might be necessary to manage a "chassis". On the other hand, the entities represented by an implementation of this MIB might well be contained in a chassis.

#### 2.6. Relationship to the Interfaces MIB

The Entity MIB contains a mapping table identifying physical components that have 'external values' (e.g., ifIndex) associated with them within a given naming scope. This table can be used to identify the physical location of each interface in the ifTable (RFC 2863 [RFC2863]). Because ifIndex values in different contexts are not related to one another, the interface to physical component associations are relative to the same logical entity within the agent.

The Entity MIB also contains 'entPhysicalName' and 'entPhysicalAlias' objects, which approximate the semantics of the 'ifName' and 'ifAlias' objects (respectively) from the Interfaces MIB [RFC2863], for all types of physical components.

## 2.7. Relationship to the Other MIBs

The Entity MIB contains a mapping table identifying physical components that have identifiers from other standard MIBs associated with them. For example, this table can be used along with the physical mapping table to identify the physical location of each repeater port in the `rpTrPortTable`, or each interface in the `ifTable`.

## 2.8. Relationship to Naming Scopes

There is some question as to which MIB objects may be returned within a given naming scope. MIB objects which are not multi-scoped within a managed system are likely to ignore context information in implementation. In such a case, it is likely such objects will be returned in all naming scopes (e.g., not just the 'default' naming scope or the SNMPv3 default context).

For example, a community string used to access the management information for logical device 'bridge2' may allow access to all the non-bridge related objects in the 'default' naming scope, as well as a second instance of the Bridge MIB (RFC 1493 [RFC1493]).

The isolation of single-scoped MIB objects by the agent is an implementation-specific matter. An agent may wish to limit the objects returned in a particular naming scope to only the multi-scoped objects in that naming scope (e.g., system group and the Bridge MIB). In this case, all single-scoped management information would belong to a common naming scope (e.g., 'default'), which itself may contain some multi-scoped objects (e.g., system group).

## 2.9. Multiple Instances of the Entity MIB

It is possible that more than one agent may exist in a managed system. In such cases, multiple instances of the Entity MIB (representing the same managed objects) may be available to an NMS.

In order to reduce complexity for agent implementation, multiple instances of the Entity MIB are not required to be equivalent or even consistent. An NMS may be able to 'align' instances returned by different agents by examining the columns of each table, but vendor-specific identifiers and (especially) index values are likely to be different. Each agent may be managing different subsets of the entire chassis as well.

When all of a physically-modular device is represented by a single agent, the entry (for which `entPhysicalContainedIn` has the value zero) would likely have 'chassis' as the value of its `entPhysicalClass`. Alternatively, for an agent on a module where the



agent represents only the physical entities on that module (not those on other modules), the entry (for which `entPhysicalContainedIn` has the value zero) would likely have 'module' as the value of its `entPhysicalClass`.

An agent implementation of the `entLogicalTable` is not required to contain information about logical entities managed primarily by other agents. That is, the `entLogicalTAddress` and `entLogicalTDomain` objects in the `entLogicalTable` are provided to support an historical multiplexing mechanism, not to identify other SNMP agents.

Note that the Entity MIB is a single-scoped MIB, in the event an agent represents the MIB in different naming scopes.

#### 2.10. Re-Configuration of Entities

Most of the MIB objects defined in this MIB have, at most, a read-only MAX-ACCESS clause. This is a conscious decision by the working group to limit this MIB's scope. The second version of the Entity MIB allows a network administrator to configure some common attributes of physical components.

#### 2.11. Textual Convention Change

Version 1 of the Entity MIB contains three MIB objects defined with the (now obsolete) `DisplayString` textual convention. In version 2 of the Entity MIB, the syntax for these objects has been updated to use the (now preferred) `SnmpAdminString` textual convention.

The `entmib` working group (which was in charge with the document at that point) realized that this change is not strictly supported by SMIV2. In their judgment, the alternative of deprecating the old objects and defining new objects would have had a more adverse impact on backward compatibility and interoperability, given the particular semantics of these objects.

#### 2.12. MIB Structure

The Entity MIB contains five groups of MIB objects:

- `entityPhysical` group  
Describes the physical entities managed by a single agent.
- `entityLogical` group  
Describes the logical entities managed by a single agent.

- entityMapping group  
Describes the associations between the physical entities, logical entities, interfaces, and non-interface ports managed by a single agent.
- entityGeneral group  
Describes general system attributes shared by potentially all types of entities managed by a single agent.
- entityNotifications group  
Contains status indication notifications.

#### 2.12.1. entityPhysical Group

This group contains a single table to identify physical system components, called the entPhysicalTable.

The entPhysicalTable contains one row per physical entity, and must always contain at least one row for an "overall" physical entity, which should have an entPhysicalClass value of 'stack(11)', 'chassis(3)' or 'module(9)'.

Each row is indexed by an arbitrary, small integer, and contains a description and type of the physical entity. It also optionally contains the index number of another entPhysicalEntry, indicating a containment relationship between the two.

Version 2 of the Entity MIB provides additional MIB objects for each physical entity. Some common read-only attributes have been added, as well as three writable string objects.

- entPhysicalAlias  
This string can be used by an NMS as a non-volatile identifier for the physical component. Maintaining a non-volatile string for every physical component represented in the entPhysicalTable can be costly and unnecessary. An agent may algorithmically generate 'entPhysicalAlias' strings for particular entries (e.g., based on the entPhysicalClass value).
- entPhysicalAssetID  
This string is provided to store a user-specific asset identifier for removable physical components. In order to reduce the non-volatile storage needed by a particular agent, a network administrator should only assign asset identifiers to physical entities that are field-replaceable (i.e., not permanently contained within another physical entity).

- entPhysicalSerialNum  
This string is provided to store a vendor-specific serial number string for physical components. This writable object is used when an agent cannot identify the serial numbers of all installed physical entities, and a network administrator wishes to configure the non-volatile serial number strings manually (via an NMS application).

Version 3 of the Entity MIB provides two additional MIB objects for each physical entity:

- entPhysicalMfgDate  
This object contains the date of manufacturing of the managed entity. If the manufacturing date is unknown or not supported the object is not instantiated. The special value '0000000000000000'H may also be returned in this case.
- entPhysicalUris  
This object provides additional identification information about the physical entity.

This object contains one or more Uniform Resource Identifiers (URIs) and, therefore, the syntax of this object must conform to RFC 3986 [RFC3986] section 2. Uniform Resource Names (URNs), RFC 3406 [RFC3406], are resource identifiers with the specific requirements for enabling location independent identification of a resource, as well as longevity of reference. URNs are part of the larger URI family with the specific goal of providing persistent naming of resources. URI schemes and URN name spaces are registered by IANA (see <http://www.iana.org/assignments/uri-schemes> and <http://www.iana.org/assignments/urn-namespaces>).

For example, the entPhysicalUris object may be used to encode a URI containing a Common Language Equipment Identifier (CLEI) URN for the managed physical entity. The URN name space for CLEIs is defined in [RFC4152], and the CLEI format is defined in [T1.213][T1.213a]. For example, an entPhysicalUris instance may have the value of

URN:CLEI:D4CE18B7AA

[RFC3986] and [RFC4152] identify this as a URI in the CLEI URN name space. The specific CLEI code, D4CE18B7AA, is based on the example provided in [T1.213a].

Multiple URIs may be present and are separated by white space characters. Leading and trailing white space characters are ignored.

If no additional identification information is known about the physical entity or supported, the object is not instantiated.

#### 2.12.2. entityLogical Group

This group contains a single table to identify logical entities, called the entLogicalTable.

The entLogicalTable contains one row per logical entity. Each row is indexed by an arbitrary, small integer and contains a name, description, and type of the logical entity. It also contains information to allow access to the MIB information for the logical entity. This includes SNMP versions that use a community name (with some form of implied context representation) and SNMP versions that use the SNMP ARCH [RFC3411] method of context identification.

If an agent represents multiple logical entities with this MIB, then this group must be implemented for all logical entities known to the agent.

If an agent represents a single logical entity, or multiple logical entities within a single naming scope, then implementation of this group may be omitted by the agent.

#### 2.12.3. entityMapping Group

This group contains three tables to identify associations between different system components.

- entLPMappingTable

This table contains mappings between entLogicalIndex values (logical entities) and entPhysicalIndex values (the physical components supporting that entity). A logical entity can map to more than one physical component, and more than one logical entity can map to (share) the same physical component. If an agent represents a single logical entity, or multiple logical entities within a single naming scope, then implementation of this table may be omitted by the agent.

- entAliasMappingTable

This table contains mappings between entLogicalIndex, entPhysicalIndex pairs, and 'alias' object identifier values. This allows resources managed with other MIBs (e.g., repeater ports, bridge ports, physical and logical interfaces) to be identified in the physical entity hierarchy. Note that each alias identifier is only relevant in a particular naming scope. If an agent represents a single logical entity, or multiple logical entities within a

single naming scope, then implementation of this table may be omitted by the agent.

- entPhysicalContainsTable  
This table contains simple mappings between 'entPhysicalContainedIn' values for each container/'containee' relationship in the managed system. The indexing of this table allows an NMS to quickly discover the 'entPhysicalIndex' values for all children of a given physical entity.

#### 2.12.4. entityGeneral Group

This group contains general information relating to the other object groups.

At this time, the entGeneral group contains a single scalar object (entLastChangeTime), which represents the value of sysUptime when any part of the Entity MIB configuration last changed.

#### 2.12.5. entityNotifications Group

This group contains notification definitions relating to the overall status of the Entity MIB instantiation.

### 2.13. Multiple Agents

Even though a primary motivation for this MIB is to represent the multiple logical entities supported by a single agent, another motivation is to represent multiple logical entities supported by multiple agents (in the same "overall" physical entity). Indeed, it is implicit in the SNMP architecture that the number of agents is transparent to a network management station.

However, there is no agreement at this time as to the degree of cooperation that should be expected for agent implementations. Therefore, multiple agents within the same managed system are free to implement the Entity MIB independently. (For more information, refer to Section 2.9, "Multiple Instances of the Entity MIB".)

#### 2.14. Changes Since RFC 2037

##### 2.14.1. Textual Conventions

The PhysicalClass TC text has been clarified, and a new enumeration to support 'stackable' components has been added. The SnmpEngineIdOrNone TC has been added to support SNMPv3.

##### 2.14.2. New entPhysicalTable Objects

The entPhysicalHardwareRev, entPhysicalFirmwareRev, and entPhysicalSoftwareRev objects have been added for revision identification.

The entPhysicalSerialNum, entPhysicalMfgName, entPhysicalModelName, and entPhysicalIsFru objects have been added for better vendor identification for physical components. In the event the agent cannot identify this information, the entPhysicalSerialNum object can be set by a management station.

The entPhysicalAlias and entPhysicalAssetID objects have been added for better user component identification. These objects are intended to be set by a management station and preserved by the agent across restarts.

#### 2.14.3. New entLogicalTable Objects

The entLogicalContextEngineID and entLogicalContextName objects have been added to provide an SNMP context for SNMPv3 access on behalf of a logical entity.

#### 2.14.4. Bug Fixes

A bug was fixed in the entLogicalCommunity object. The subrange was incorrect (1..255) and is now (0..255). The description clause has also been clarified. This object is now deprecated.

The entLastChangeTime object description has been changed to generalize the events that cause an update to the last change timestamp.

The syntax was changed from DisplayString to SnmpAdminString for the entPhysicalDescr, entPhysicalName, and entLogicalDescr objects.

### 2.15. Changes Since RFC 2737

#### 2.15.1. Textual Conventions

The PhysicalIndexOrZero TC has been added to allow objects to reference an entPhysicalIndex value or zero. The PhysicalClass TC has been extended to support a new enumeration for central processing units.

#### 2.15.2. New Objects

The entPhysicalMfgDate object has been added to the entPhysicalTable to provide the date of manufacturing of the managed entity.

The entPhysicalUris object has been added to the entPhysicalTable to provide additional identification information about the physical entity, such as a Common Language Equipment Identifier (CLEI) URN.

#### 2.15.3. Bug Fixes

The syntax was changed from INTEGER to Integer32 for the entPhysicalParentRelPos, entLogicalIndex, and entAliasLogicalIndexOrZero objects, and from INTEGER to PhysicalIndexOrZero for the entPhysicalContainedIn object.

#### 2.16. Changes Since RFC 4133

2.16.1. MIB module addition Creation of a new MIB module IANA-ENTITY-MIB which makes the PhysicalIndex TC an IANA-maintained Textual Convention. Over time, there is the need to add new enumerated values for PhysicalClass. If the syntax of IANAPhysicalClass were defined in this MIB module then a new version of this MIB would have to be re-issued in order to define new values.

#### 2.16.2. Modification to some of the MIB objects

Creation of a new MODULE-COMPLIANCE module entity4LowCompliance for devices with constrained resources like batteries, which might require a limited number of objects to be supported (entPhysicalIndex, entPhysicalName, entIANAPhysicalClass, entPhysicalUris) with the entPhysicalUris object read-only and restricted to a fixed size to allow only for RFC 4122 [RFC4122] compliant values.

### 3. MIB Definitions

#### 3.1. ENTITY MIB

ENTITY-MIB DEFINITIONS ::= BEGIN

#### IMPORTS

MODULE-IDENTITY, OBJECT-TYPE, mib-2, NOTIFICATION-TYPE,  
Integer32  
FROM SNMPv2-SMI  
TDomain, TAddress, TEXTUAL-CONVENTION,  
AutonomousType, RowPointer, TimeStamp, TruthValue,  
DateAndTime  
FROM SNMPv2-TC  
MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP  
FROM SNMPv2-CONF  
SnmpAdminString  
FROM SNMP-FRAMEWORK-MIB

Uri

FROM URI-TC-MIB  
IANAPhysicalClass  
FROM IANA-ENTITY-MIB;

entityMIB MODULE-IDENTITY

LAST-UPDATED "201207100000Z"

ORGANIZATION "IETF Energy Management Working Group"

CONTACT-INFO

" WG E-mail: eman@ietf.org  
Mailing list subscription info:  
<http://www.ietf.org/mailman/listinfo/eman>

Andy Bierman  
Yumaworks  
andy@yumaworks.com

Dan Romascanu  
AVAYA  
Park Atidim, Bldg. #3  
Tel Aviv, 61581  
Israel  
Phone: +972-3-6458414  
dromasca@avaya.com

Juergen Quittek  
NEC Europe Ltd.  
Network Research Division  
Kurfuersten-Anlage 36  
Heidelberg 69115  
DE  
Phone: +49 6221 4342-115  
Email: quittek@neclab.eu

Mouli Chandramouli  
Cisco Systems, Inc.  
Sarjapur Outer Ring Road  
Bangalore 561013  
IN  
Phone: +91 80 4426 3947  
Email: moulchan@cisco.com"

#### DESCRIPTION

"The MIB module for representing multiple logical entities supported by a single SNMP agent.

Copyright (C) The Internet Society (2012). This version of this MIB module is part of RFC xxxx; see



the RFC itself for full legal notices."

REVISION "201207100000Z"

DESCRIPTION

"Entity MIB (Version 4).

This revision obsoletes RFC 4133.

Additions:

-

Changes:

- according to comments made on draft-chandramouli-00

This version published as RFC xxxx."

REVISION "201206100000Z"

DESCRIPTION

"Initial Version of Entity MIB (Version 4).

This revision obsoletes RFC 4133.

Additions:

-

Changes:

-

This version published as RFC xxxx."

REVISION "200508100000Z"

DESCRIPTION

"Initial Version of Entity MIB (Version 3).

This revision obsoletes RFC 2737.

Additions:

- cpu(12) enumeration added to IANAPhysicalClass TC

- DISPLAY-HINT clause to PhysicalIndex TC

- PhysicalIndexOrZero TC

- entPhysicalMfgDate object

- entPhysicalUris object

Changes:

- entPhysicalContainedIn SYNTAX changed from

INTEGER to PhysicalIndexOrZero

This version published as RFC 4133."

REVISION "199912070000Z"

DESCRIPTION

"Initial Version of Entity MIB (Version 2).

This revision obsoletes RFC 2037.

This version published as RFC 2737."

REVISION "199610310000Z"

DESCRIPTION

```
        "Initial version (version 1), published as
          RFC 2037."
 ::= { mib-2 47 }

entityMIBObjects OBJECT IDENTIFIER ::= { entityMIB 1 }

-- MIB contains four groups
entityPhysical OBJECT IDENTIFIER ::= { entityMIBObjects 1 }
entityLogical  OBJECT IDENTIFIER ::= { entityMIBObjects 2 }
entityMapping  OBJECT IDENTIFIER ::= { entityMIBObjects 3 }
entityGeneral  OBJECT IDENTIFIER ::= { entityMIBObjects 4 }


-- Textual Conventions
PhysicalIndex ::= TEXTUAL-CONVENTION
    DISPLAY-HINT    "d"
    STATUS           current
    DESCRIPTION
        "An arbitrary value that uniquely identifies the physical
        entity. The value should be a small, positive integer.
        Index values for different physical entities are not
        necessarily contiguous."
    SYNTAX Integer32 (1..2147483647)

PhysicalIndexOrZero ::= TEXTUAL-CONVENTION
    DISPLAY-HINT    "d"
    STATUS           current
    DESCRIPTION
        "This textual convention is an extension of the
        PhysicalIndex convention, which defines a greater than zero
        value used to identify a physical entity. This extension
        permits the additional value of zero. The semantics of the
        value zero are object-specific and must, therefore, be
        defined as part of the description of any object that uses
        this syntax. Examples of the usage of this extension are
        situations where none or all physical entities need to be
        referenced."
    SYNTAX Integer32 (0..2147483647)

SnmpEngineIdOrNone ::= TEXTUAL-CONVENTION
    STATUS           current
    DESCRIPTION
        "A specially formatted SnmpEngineID string for use with the
        Entity MIB."
```

If an instance of an object of SYNTAX SnmpEngineIdOrNone has a non-zero length, then the object encoding and semantics are defined by the SnmpEngineID textual convention (see STD 62, RFC 3411 [RFC3411]).

If an instance of an object of SYNTAX SnmpEngineIdOrNone contains a zero-length string, then no appropriate SnmpEngineID is associated with the logical entity (i.e., SNMPv3 is not supported)."

SYNTAX OCTET STRING (SIZE(0..32)) -- empty string or SnmpEngineID

-- The Physical Entity Table

entPhysicalTable OBJECT-TYPE

SYNTAX SEQUENCE OF EntPhysicalEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table contains one row per physical entity. There is always at least one row for an 'overall' physical entity."

::= { entityPhysical 1 }

entPhysicalEntry OBJECT-TYPE

SYNTAX EntPhysicalEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Information about a particular physical entity.

Each entry provides objects (entPhysicalDescr, entPhysicalVendorType, and entIANAPhysicalClass) to help an NMS identify and characterize the entry, and objects (entPhysicalContainedIn and entPhysicalParentRelPos) to help an NMS relate the particular entry to other entries in this table."

INDEX { entPhysicalIndex }

::= { entPhysicalTable 1 }

EntPhysicalEntry ::= SEQUENCE {

entPhysicalIndex	PhysicalIndex,
entPhysicalDescr	SnmpAdminString,
entPhysicalVendorType	AutonomousType,
entPhysicalContainedIn	PhysicalIndexOrZero,
entPhysicalClass	IANAPhysicalClass,
entPhysicalParentRelPos	Integer32,

```

    entPhysicalName          SnmpAdminString,
    entPhysicalHardwareRev   SnmpAdminString,
    entPhysicalFirmwareRev   SnmpAdminString,
    entPhysicalSoftwareRev   SnmpAdminString,
    entPhysicalSerialNum     SnmpAdminString,
    entPhysicalMfgName       SnmpAdminString,
    entPhysicalModelName     SnmpAdminString,
    entPhysicalAlias         SnmpAdminString,
    entPhysicalAssetID       SnmpAdminString,
    entPhysicalIsFRU         TruthValue,
    entPhysicalMfgDate       DateAndTime,
    entPhysicalUris          Uri
}

entPhysicalIndex OBJECT-TYPE
    SYNTAX      PhysicalIndex
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The index for this entry."
    ::= { entPhysicalEntry 1 }

entPhysicalDescr OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "A textual description of physical entity. This object
        should contain a string that identifies the manufacturer's
        name for the physical entity, and should be set to a
        distinct value for each version or model of the physical
        entity."
    ::= { entPhysicalEntry 2 }

entPhysicalVendorType OBJECT-TYPE
    SYNTAX      AutonomousType
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "An indication of the vendor-specific hardware type of the
        physical entity. Note that this is different from the
        definition of MIB-II's sysObjectID.

        An agent should set this object to an enterprise-specific

```

registration identifier value indicating the specific equipment type in detail. The associated instance of entIANAPhysicalClass is used to indicate the general type of hardware device.

If no vendor-specific registration identifier exists for this physical entity, or the value is unknown by this agent, then the value { 0 0 } is returned."

::= { entPhysicalEntry 3 }

entPhysicalContainedIn OBJECT-TYPE

SYNTAX PhysicalIndexOrZero

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of entPhysicalIndex for the physical entity which 'contains' this physical entity. A value of zero indicates this physical entity is not contained in any other physical entity. Note that the set of 'containment' relationships define a strict hierarchy; that is, recursion is not allowed.

In the event that a physical entity is contained by more than one physical entity (e.g., double-wide modules), this object should identify the containing entity with the lowest value of entPhysicalIndex."

::= { entPhysicalEntry 4 }

entPhysicalClass OBJECT-TYPE

SYNTAX IANAPhysicalClass

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"An indication of the general hardware type of the physical entity.

An agent should set this object to the standard enumeration value that most accurately indicates the general class of the physical entity, or the primary class if there is more than one entity.

If no appropriate standard registration identifier exists for this physical entity, then the value 'other(1)' is returned. If the value is unknown by this agent, then the value 'unknown(2)' is returned."

```
::= { entPhysicalEntry 5 }
```

entPhysicalParentRelPos OBJECT-TYPE

SYNTAX Integer32 (-1..2147483647)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"An indication of the relative position of this 'child' component among all its 'sibling' components. Sibling components are defined as entPhysicalEntries that share the same instance values of each of the entPhysicalContainedIn and entPhysicalClass objects.

An NMS can use this object to identify the relative ordering for all sibling components of a particular parent (identified by the entPhysicalContainedIn instance in each sibling entry).

If possible, this value should match any external labeling of the physical component. For example, for a container (e.g., card slot) labeled as 'slot #3', entPhysicalParentRelPos should have the value '3'. Note that the entPhysicalEntry for the module plugged in slot 3 should have an entPhysicalParentRelPos value of '1'.

If the physical position of this component does not match any external numbering or clearly visible ordering, then user documentation or other external reference material should be used to determine the parent-relative position. If this is not possible, then the agent should assign a consistent (but possibly arbitrary) ordering to a given set of 'sibling' components, perhaps based on internal representation of the components.

If the agent cannot determine the parent-relative position for some reason, or if the associated value of entPhysicalContainedIn is '0', then the value '-1' is returned. Otherwise, a non-negative integer is returned, indicating the parent-relative position of this physical entity.

Parent-relative ordering normally starts from '1' and continues to 'N', where 'N' represents the highest positioned child entity. However, if the physical entities (e.g., slots) are labeled from a starting position of zero,

then the first sibling should be associated with an entPhysicalParentRelPos value of '0'. Note that this ordering may be sparse or dense, depending on agent implementation.

The actual values returned are not globally meaningful, as each 'parent' component may use different numbering algorithms. The ordering is only meaningful among siblings of the same parent component.

The agent should retain parent-relative position values across reboots, either through algorithmic assignment or use of non-volatile storage."

::= { entPhysicalEntry 6 }

entPhysicalName OBJECT-TYPE

SYNTAX SnmpAdminString

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The textual name of the physical entity. The value of this object should be the name of the component as assigned by the local device and should be suitable for use in commands entered at the device's 'console'. This might be a text name (e.g., 'console') or a simple component number (e.g., port or module number, such as '1'), depending on the physical component naming syntax of the device.

If there is no local name, or if this object is otherwise not applicable, then this object contains a zero-length string.

Note that the value of entPhysicalName for two physical entities will be the same in the event that the console interface does not distinguish between them, e.g., slot-1 and the card in slot-1."

::= { entPhysicalEntry 7 }

entPhysicalHardwareRev OBJECT-TYPE

SYNTAX SnmpAdminString

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The vendor-specific hardware revision string for the physical entity. The preferred value is the hardware revision identifier actually printed on the component itself

(if present).

Note that if revision information is stored internally in a non-printable (e.g., binary) format, then the agent must convert such information to a printable format, in an implementation-specific manner.

If no specific hardware revision string is associated with the physical component, or if this information is unknown to the agent, then this object will contain a zero-length string."

::= { entPhysicalEntry 8 }

entPhysicalFirmwareRev OBJECT-TYPE

SYNTAX SnmpAdminString

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The vendor-specific firmware revision string for the physical entity.

Note that if revision information is stored internally in a non-printable (e.g., binary) format, then the agent must convert such information to a printable format, in an implementation-specific manner.

If no specific firmware programs are associated with the physical component, or if this information is unknown to the agent, then this object will contain a zero-length string."

::= { entPhysicalEntry 9 }

entPhysicalSoftwareRev OBJECT-TYPE

SYNTAX SnmpAdminString

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The vendor-specific software revision string for the physical entity.

Note that if revision information is stored internally in a

non-printable (e.g., binary) format, then the agent must convert such information to a printable format, in an implementation-specific manner.

If no specific software programs are associated with the



physical component, or if this information is unknown to the agent, then this object will contain a zero-length string."  
 ::= { entPhysicalEntry 10 }

entPhysicalSerialNum OBJECT-TYPE

SYNTAX SnmpAdminString (SIZE (0..32))

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The vendor-specific serial number string for the physical entity. The preferred value is the serial number string actually printed on the component itself (if present).

On the first instantiation of an physical entity, the value of entPhysicalSerialNum associated with that entity is set to the correct vendor-assigned serial number, if this information is available to the agent. If a serial number is unknown or non-existent, the entPhysicalSerialNum will be set to a zero-length string instead.

Note that implementations that can correctly identify the serial numbers of all installed physical entities do not need to provide write access to the entPhysicalSerialNum object. Agents which cannot provide non-volatile storage for the entPhysicalSerialNum strings are not required to implement write access for this object.

Not every physical component will have a serial number, or even need one. Physical entities for which the associated value of the entPhysicalIsFRU object is equal to 'false(2)' (e.g., the repeater ports within a repeater module), do not need their own unique serial number. An agent does not have to provide write access for such entities, and may return a zero-length string.

If write access is implemented for an instance of entPhysicalSerialNum, and a value is written into the instance, the agent must retain the supplied value in the entPhysicalSerialNum instance (associated with the same physical entity) for as long as that entity remains instantiated. This includes instantiations across all re-initializations/reboots of the network management system, including those resulting in a change of the physical

entity's entPhysicalIndex value."  
 ::= { entPhysicalEntry 11 }

**entPhysicalMfgName** OBJECT-TYPE

SYNTAX SnmpAdminString

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The name of the manufacturer of this physical component.  
The preferred value is the manufacturer name string actually  
printed on the component itself (if present)."

Note that comparisons between instances of the  
entPhysicalModelName, entPhysicalFirmwareRev,  
entPhysicalSoftwareRev, and the entPhysicalSerialNum  
objects, are only meaningful amongst entPhysicalEntries with  
the same value of entPhysicalMfgName.

If the manufacturer name string associated with the physical  
component is unknown to the agent, then this object will  
contain a zero-length string."

::= { entPhysicalEntry 12 }

**entPhysicalModelName** OBJECT-TYPE

SYNTAX SnmpAdminString

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The vendor-specific model name identifier string associated  
with this physical component. The preferred value is the  
customer-visible part number, which may be printed on the  
component itself."

If the model name string associated with the physical  
component is unknown to the agent, then this object will  
contain a zero-length string."

::= { entPhysicalEntry 13 }

**entPhysicalAlias** OBJECT-TYPE

SYNTAX SnmpAdminString (SIZE (0..32))

MAX-ACCESS read-write

STATUS current

## DESCRIPTION

"This object is an 'alias' name for the physical entity, as  
specified by a network manager, and provides a non-volatile  
'handle' for the physical entity."

On the first instantiation of a physical entity, the value

of entPhysicalAlias associated with that entity is set to the zero-length string. However, the agent may set the value to a locally unique default value, instead of a zero-length string.

If write access is implemented for an instance of entPhysicalAlias, and a value is written into the instance, the agent must retain the supplied value in the entPhysicalAlias instance (associated with the same physical entity) for as long as that entity remains instantiated. This includes instantiations across all re-initializations/reboots of the network management system, including those resulting in a change of the physical entity's entPhysicalIndex value."

```
::= { entPhysicalEntry 14 }
```

entPhysicalAssetID OBJECT-TYPE

SYNTAX SnmpAdminString (SIZE (0..32))

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This object is a user-assigned asset tracking identifier (as specified by a network manager) for the physical entity, and provides non-volatile storage of this information.

On the first instantiation of a physical entity, the value of entPhysicalAssetID associated with that entity is set to the zero-length string.

Not every physical component will have an asset tracking identifier, or even need one. Physical entities for which the associated value of the entPhysicalIsFRU object is equal to 'false(2)' (e.g., the repeater ports within a repeater module), do not need their own unique asset tracking identifier. An agent does not have to provide write access for such entities, and may instead return a zero-length string.

If write access is implemented for an instance of entPhysicalAssetID, and a value is written into the instance, the agent must retain the supplied value in the entPhysicalAssetID instance (associated with the same physical entity) for as long as that entity remains instantiated. This includes instantiations across all re-initializations/reboots of the network management system, including those resulting in a change of the physical entity's entPhysicalIndex value.

If no asset tracking information is associated with the physical component, then this object will contain a zero-length string."

```
::= { entPhysicalEntry 15 }
```

entPhysicalIsFRU OBJECT-TYPE

```
SYNTAX      TruthValue
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
```

"This object indicates whether or not this physical entity is considered a 'field replaceable unit' by the vendor. If this object contains the value 'true(1)' then this entPhysicalEntry identifies a field replaceable unit. For all entPhysicalEntries that represent components permanently contained within a field replaceable unit, the value 'false(2)' should be returned for this object."

```
::= { entPhysicalEntry 16 }
```

entPhysicalMfgDate OBJECT-TYPE

```
SYNTAX      DateAndTime
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
```

"This object contains the date of manufacturing of the managed entity. If the manufacturing date is unknown or not supported, the object is not instantiated. The special value '0000000000000000'H may also be returned in this case."

```
::= { entPhysicalEntry 17 }
```

entPhysicalUri OBJECT-TYPE

```
SYNTAX      Uri
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
```

"This object contains additional identification information about the physical entity. The object contains URIs and, therefore, the syntax of this object must conform to RFC 3986, section 2.

Multiple URIs may be present and are separated by white space characters. Leading and trailing white space characters are ignored.

If no additional identification information is known about the physical entity or supported, the object is not instantiated. A zero length octet string may also be

returned in this case."

REFERENCE

"RFC 3986, Uniform Resource Identifiers (URI): Generic Syntax, section 2, August 1998."

::= { entPhysicalEntry 18 }

-- The Logical Entity Table

entLogicalTable OBJECT-TYPE

SYNTAX SEQUENCE OF EntLogicalEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table contains one row per logical entity. For agents that implement more than one naming scope, at least one entry must exist. Agents which instantiate all MIB objects within a single naming scope are not required to implement this table."

::= { entityLogical 1 }

entLogicalEntry OBJECT-TYPE

SYNTAX EntLogicalEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Information about a particular logical entity. Entities may be managed by this agent or other SNMP agents (possibly) in the same chassis."

INDEX { entLogicalIndex }

::= { entLogicalTable 1 }

EntLogicalEntry ::= SEQUENCE {

entLogicalIndex	Integer32,
entLogicalDescr	SnmpAdminString,
entLogicalType	AutonomousType,
entLogicalCommunity	OCTET STRING,
entLogicalTAddress	TAddress,
entLogicalTDomain	TDomain,
entLogicalContextEngineID	SnmpEngineIdOrNone,
entLogicalContextName	SnmpAdminString

}

entLogicalIndex OBJECT-TYPE

SYNTAX Integer32 (1..2147483647)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The value of this object uniquely identifies the logical entity. The value should be a small positive integer; index values for different logical entities are not necessarily contiguous."

::= { entLogicalEntry 1 }

entLogicalDescr OBJECT-TYPE  
SYNTAX SnmpAdminString  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
"A textual description of the logical entity. This object should contain a string that identifies the manufacturer's name for the logical entity, and should be set to a distinct value for each version of the logical entity."  
 ::= { entLogicalEntry 2 }

entLogicalType OBJECT-TYPE  
SYNTAX AutonomousType  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
"An indication of the type of logical entity. This will typically be the OBJECT IDENTIFIER name of the node in the SMI's naming hierarchy which represents the major MIB module, or the majority of the MIB modules, supported by the logical entity. For example:  
a logical entity of a regular host/router -> mib-2  
a logical entity of a 802.1d bridge -> dot1dBridge  
a logical entity of a 802.3 repeater -> snmpDot3RptrMgmt  
If an appropriate node in the SMI's naming hierarchy cannot be identified, the value 'mib-2' should be used."  
 ::= { entLogicalEntry 3 }

entLogicalCommunity OBJECT-TYPE  
SYNTAX OCTET STRING (SIZE (0..255))  
MAX-ACCESS read-only  
STATUS deprecated  
DESCRIPTION  
"An SNMPv1 or SNMPv2C community-string, which can be used to access detailed management information for this logical entity. The agent should allow read access with this community string (to an appropriate subset of all managed objects) and may also return a community string based on the privileges of the request used to read this object. Note that an agent may return a community string with read-only privileges, even if this object is accessed with a read-write community string. However, the agent must take

care not to return a community string that allows more privileges than the community string used to access this object.

A compliant SNMP agent may wish to conserve naming scopes by representing multiple logical entities in a single 'default' naming scope. This is possible when the logical entities, represented by the same value of entLogicalCommunity, have no object instances in common. For example, 'bridge1' and 'repeater1' may be part of the main naming scope, but at least one additional community string is needed to represent 'bridge2' and 'repeater2'.

Logical entities 'bridge1' and 'repeater1' would be represented by sysOREntries associated with the 'default' naming scope.

For agents not accessible via SNMPv1 or SNMPv2C, the value of this object is the empty string. This object may also contain an empty string if a community string has not yet been assigned by the agent, or if no community string with suitable access rights can be returned for a particular SNMP request.

Note that this object is deprecated. Agents which implement SNMPv3 access should use the entLogicalContextEngineID and entLogicalContextName objects to identify the context associated with each logical entity. SNMPv3 agents may return a zero-length string for this object, or may continue to return a community string (e.g., tri-lingual agent support)."

::= { entLogicalEntry 4 }

entLogicalTAddress OBJECT-TYPE

SYNTAX TAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The transport service address by which the logical entity receives network management traffic, formatted according to the corresponding value of entLogicalTDomain.

For snmpUDPDDomain, a TAddress is 6 octets long: the initial 4 octets contain the IP-address in network-byte order and the last 2 contain the UDP port in network-byte order. Consult 'Transport Mappings for the Simple Network Management Protocol' (STD 62, RFC 3417 [RFC3417]) for further information on snmpUDPDDomain."

```
::= { entLogicalEntry 5 }
```

entLogicalTDomain OBJECT-TYPE

SYNTAX TDomain

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Indicates the kind of transport service by which the logical entity receives network management traffic. Possible values for this object are presently found in the Transport Mappings for Simple Network Management Protocol' (STD 62, RFC 3417 [RFC3417])."

```
::= { entLogicalEntry 6 }
```

entLogicalContextEngineID OBJECT-TYPE

SYNTAX SnmpEngineIdOrNone

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The authoritative contextEngineID that can be used to send an SNMP message concerning information held by this logical entity, to the address specified by the associated 'entLogicalTAddress/entLogicalTDomain' pair.

This object, together with the associated entLogicalContextName object, defines the context associated with a particular logical entity, and allows access to SNMP engines identified by a contextEngineId and contextName pair.

If no value has been configured by the agent, a zero-length string is returned, or the agent may choose not to instantiate this object at all."

```
::= { entLogicalEntry 7 }
```

entLogicalContextName OBJECT-TYPE

SYNTAX SnmpAdminString

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The contextName that can be used to send an SNMP message concerning information held by this logical entity, to the address specified by the associated 'entLogicalTAddress/entLogicalTDomain' pair.

This object, together with the associated entLogicalContextEngineID object, defines the context associated with a particular logical entity, and allows



access to SNMP engines identified by a contextEngineId and contextName pair.

If no value has been configured by the agent, a zero-length string is returned, or the agent may choose not to instantiate this object at all."

::= { entLogicalEntry 8 }

entLPMappingTable OBJECT-TYPE

SYNTAX SEQUENCE OF EntLPMappingEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table contains zero or more rows of logical entity to physical equipment associations. For each logical entity known by this agent, there are zero or more mappings to the physical resources, which are used to realize that logical entity.

An agent should limit the number and nature of entries in this table such that only meaningful and non-redundant information is returned. For example, in a system that contains a single power supply, mappings between logical entities and the power supply are not useful and should not be included.

Also, only the most appropriate physical component, which is closest to the root of a particular containment tree, should be identified in an entLPMapping entry.

For example, suppose a bridge is realized on a particular module, and all ports on that module are ports on this bridge. A mapping between the bridge and the module would be useful, but additional mappings between the bridge and each of the ports on that module would be redundant (because the entPhysicalContainedIn hierarchy can provide the same information). On the other hand, if more than one bridge were utilizing ports on this module, then mappings between each bridge and the ports it used would be appropriate.

Also, in the case of a single backplane repeater, a mapping for the backplane to the single repeater entity is not necessary."

::= { entityMapping 1 }

entLPMappingEntry OBJECT-TYPE

SYNTAX EntLPMappingEntry

MAX-ACCESS not-accessible

```
STATUS      current
DESCRIPTION
    "Information about a particular logical entity to physical
    equipment association. Note that the nature of the
    association is not specifically identified in this entry.
    It is expected that sufficient information exists in the
    MIBs used to manage a particular logical entity to infer how
    physical component information is utilized."
INDEX       { entLogicalIndex, entLPPhysicalIndex }
 ::= { entLPMappingTable 1 }

EntLPMappingEntry ::= SEQUENCE {
    entLPPhysicalIndex      PhysicalIndex
}

entLPPhysicalIndex OBJECT-TYPE
    SYNTAX      PhysicalIndex
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The value of this object identifies the index value of a
        particular entPhysicalEntry associated with the indicated
        entLogicalEntity."
    ::= { entLPMappingEntry 1 }

-- logical entity/component to alias table
entAliasMappingTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF EntAliasMappingEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "This table contains zero or more rows, representing
        mappings of logical entity and physical component to
        external MIB identifiers. Each physical port in the system
        may be associated with a mapping to an external identifier,
        which itself is associated with a particular logical
        entity's naming scope. A 'wildcard' mechanism is provided
        to indicate that an identifier is associated with more than
        one logical entity."
    ::= { entityMapping 2 }

entAliasMappingEntry      OBJECT-TYPE
    SYNTAX      EntAliasMappingEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "Information about a particular physical equipment, logical
```

entity to external identifier binding. Each logical entity/physical component pair may be associated with one alias mapping. The logical entity index may also be used as a 'wildcard' (refer to the entAliasLogicalIndexOrZero object DESCRIPTION clause for details.)

Note that only entPhysicalIndex values that represent physical ports (i.e., associated entPhysicalClass value is 'port(10)') are permitted to exist in this table."

```
INDEX { entPhysicalIndex, entAliasLogicalIndexOrZero }
::= { entAliasMappingTable 1 }
```

```
EntAliasMappingEntry ::= SEQUENCE {
    entAliasLogicalIndexOrZero      Integer32,
    entAliasMappingIdentifier       RowPointer
}
```

```
entAliasLogicalIndexOrZero OBJECT-TYPE
    SYNTAX      Integer32 (0..2147483647)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
```

"The value of this object identifies the logical entity that defines the naming scope for the associated instance of the 'entAliasMappingIdentifier' object.

If this object has a non-zero value, then it identifies the logical entity named by the same value of entLogicalIndex.

If this object has a value of zero, then the mapping between the physical component and the alias identifier for this entAliasMapping entry is associated with all unspecified logical entities. That is, a value of zero (the default mapping) identifies any logical entity that does not have an explicit entry in this table for a particular entPhysicalIndex/entAliasMappingIdentifier pair.

For example, to indicate that a particular interface (e.g., physical component 33) is identified by the same value of ifIndex for all logical entities, the following instance might exist:

```
entAliasMappingIdentifier.33.0 = ifIndex.5
```

In the event an entPhysicalEntry is associated differently for some logical entities, additional entAliasMapping entries may exist, e.g.:

```
entAliasMappingIdentifier.33.0 = ifIndex.6
entAliasMappingIdentifier.33.4 = ifIndex.1
entAliasMappingIdentifier.33.5 = ifIndex.1
entAliasMappingIdentifier.33.10 = ifIndex.12
```

Note that entries with non-zero entAliasLogicalIndexOrZero index values have precedence over zero-indexed entries. In this example, all logical entities except 4, 5, and 10, associate physical entity 33 with ifIndex.6."

```
::= { entAliasMappingEntry 1 }
```

entAliasMappingIdentifier OBJECT-TYPE

SYNTAX RowPointer

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object identifies a particular conceptual row associated with the indicated entPhysicalIndex and entLogicalIndex pair.

Because only physical ports are modeled in this table, only entries that represent interfaces or ports are allowed. If an ifEntry exists on behalf of a particular physical port, then this object should identify the associated 'ifEntry'. For repeater ports, the appropriate row in the 'rpPtrPortGroupTable' should be identified instead.

For example, suppose a physical port was represented by entPhysicalEntry.3, entLogicalEntry.15 existed for a repeater, and entLogicalEntry.22 existed for a bridge. Then there might be two related instances of

entAliasMappingIdentifier:

```
entAliasMappingIdentifier.3.15 == rpPtrPortGroupIndex.5.2
```

```
entAliasMappingIdentifier.3.22 == ifIndex.17
```

It is possible that other mappings (besides interfaces and repeater ports) may be defined in the future, as required.

Bridge ports are identified by examining the Bridge MIB and appropriate ifEntries associated with each 'dot1dBasePort', and are thus not represented in this table."

```
::= { entAliasMappingEntry 2 }
```

-- physical mapping table

entPhysicalContainsTable OBJECT-TYPE

SYNTAX SEQUENCE OF EntPhysicalContainsEntry

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

"A table that exposes the container/'containeer' relationships between physical entities. This table provides all the information found by constructing the virtual containment tree for a given entPhysicalTable, but in a more direct format.

In the event a physical entity is contained by more than one other physical entity (e.g., double-wide modules), this table should include these additional mappings, which cannot be represented in the entPhysicalTable virtual containment tree."

::= { entityMapping 3 }

## entPhysicalContainsEntry OBJECT-TYPE

SYNTAX EntPhysicalContainsEntry

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

"A single container/'containeer' relationship."

INDEX { entPhysicalIndex, entPhysicalChildIndex }

::= { entPhysicalContainsTable 1 }

EntPhysicalContainsEntry ::= SEQUENCE {  
    entPhysicalChildIndex PhysicalIndex  
}

## entPhysicalChildIndex OBJECT-TYPE

SYNTAX PhysicalIndex

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The value of entPhysicalIndex for the contained physical entity."

::= { entPhysicalContainsEntry 1 }

-- last change time stamp for the whole MIB

## entLastChangeTime OBJECT-TYPE

SYNTAX TimeStamp

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The value of sysUpTime at the time a conceptual row is created, modified, or deleted in any of these tables:

- entPhysicalTable
- entLogicalTable
- entLPMappingTable
- entAliasMappingTable

```

        - entPhysicalContainsTable
        "
 ::= { entityGeneral 1 }

-- Entity MIB Trap Definitions
entityMIBTraps      OBJECT IDENTIFIER ::= { entityMIB 2 }
entityMIBTrapPrefix OBJECT IDENTIFIER ::= { entityMIBTraps 0 }

entConfigChange NOTIFICATION-TYPE
    STATUS          current
    DESCRIPTION
        "An entConfigChange notification is generated when the value
        of entLastChangeTime changes.  It can be utilized by an NMS
        to trigger logical/physical entity table maintenance polls.

        An agent should not generate more than one entConfigChange
        'notification-event' in a given time interval (five seconds
        is the suggested default).  A 'notification-event' is the
        transmission of a single trap or inform PDU to a list of
        notification destinations.

        If additional configuration changes occur within the
        throttling period, then notification-events for these
        changes should be suppressed by the agent until the current
        throttling period expires.  At the end of a throttling
        period, one notification-event should be generated if any
        configuration changes occurred since the start of the
        throttling period.  In such a case, another throttling
        period is started right away.

        An NMS should periodically check the value of
        entLastChangeTime to detect any missed entConfigChange
        notification-events, e.g., due to throttling or transmission
        loss."
 ::= { entityMIBTrapPrefix 1 }

-- conformance information
entityConformance OBJECT IDENTIFIER ::= { entityMIB 3 }

entityCompliances OBJECT IDENTIFIER ::= { entityConformance 1 }
entityGroups      OBJECT IDENTIFIER ::= { entityConformance 2 }

-- compliance statements
entityCompliance MODULE-COMPLIANCE
    STATUS deprecated
```

```
DESCRIPTION
    "The compliance statement for SNMP entities that implement
    version 1 of the Entity MIB."
MODULE -- this module
    MANDATORY-GROUPS {
        entityPhysicalGroup,
        entityLogicalGroup,
        entityMappingGroup,
        entityGeneralGroup,
        entityNotificationsGroup
    }
 ::= { entityCompliances 1 }

entity2Compliance MODULE-COMPLIANCE
STATUS deprecated
DESCRIPTION
    "The compliance statement for SNMP entities that implement
    version 2 of the Entity MIB."
MODULE -- this module
    MANDATORY-GROUPS {
        entityPhysicalGroup,
        entityPhysical2Group,
        entityGeneralGroup,
        entityNotificationsGroup
    }
    GROUP entityLogical2Group
DESCRIPTION
    "Implementation of this group is not mandatory for agents
    that model all MIB object instances within a single naming
    scope."

    GROUP entityMappingGroup
DESCRIPTION
    "Implementation of the entPhysicalContainsTable is mandatory
    for all agents. Implementation of the entLPMappingTable and
    entAliasMappingTables are not mandatory for agents that
    model all MIB object instances within a single naming scope.

    Note that the entAliasMappingTable may be useful for all
    agents; however, implementation of the entityLogicalGroup or
    entityLogical2Group is required to support this table."

OBJECT entPhysicalSerialNum
MIN-ACCESS not-accessible
DESCRIPTION
    "Read and write access is not required for agents that
    cannot identify serial number information for physical
    entities, and/or cannot provide non-volatile storage for
```

NMS-assigned serial numbers.

Write access is not required for agents that can identify serial number information for physical entities, but cannot provide non-volatile storage for NMS-assigned serial numbers.

Write access is not required for physical entities for which the associated value of the entPhysicalIsFRU object is equal to 'false(2)'."

OBJECT entPhysicalAlias

MIN-ACCESS read-only

DESCRIPTION

"Write access is required only if the associated entPhysicalClass value is equal to 'chassis(3)'."

OBJECT entPhysicalAssetID

MIN-ACCESS not-accessible

DESCRIPTION

"Read and write access is not required for agents that cannot provide non-volatile storage for NMS-assigned asset identifiers.

Write access is not required for physical entities for which the associated value of the entPhysicalIsFRU object is equal to 'false(2)'."

OBJECT entPhysicalClass

SYNTAX INTEGER {

other(1),  
unknown(2),  
chassis(3),  
backplane(4),  
container(5),  
powerSupply(6),  
fan(7),  
sensor(8),  
module(9),  
port(10),  
stack(11)

}

DESCRIPTION

"Implementation of the 'cpu(12)' enumeration is not required."

::= { entityCompliances 2 }



```
entity3Compliance MODULE-COMPLIANCE
  STATUS current
  DESCRIPTION
    "The compliance statement for SNMP entities that implement
    version 3 and 4 (full compliance) of the Entity MIB."
  MODULE -- this module
    MANDATORY-GROUPS {
      entityPhysicalGroup,
      entityPhysical2Group,
      entityGeneralGroup,
      entityNotificationsGroup
    }
  GROUP entityLogical2Group
  DESCRIPTION
    "Implementation of this group is not mandatory for agents
    that model all MIB object instances within a single naming
    scope."

  GROUP entityMappingGroup
  DESCRIPTION
    "Implementation of the entPhysicalContainsTable is mandatory
    for all agents. Implementation of the entLPMappingTable and
    entAliasMappingTables are not mandatory for agents that
    model all MIB object instances within a single naming scope.

    Note that the entAliasMappingTable may be useful for all
    agents; however, implementation of the entityLogicalGroup or
    entityLogical2Group is required to support this table."

  OBJECT entPhysicalSerialNum
  MIN-ACCESS not-accessible
  DESCRIPTION
    "Read and write access is not required for agents that
    cannot identify serial number information for physical
    entities, and/or cannot provide non-volatile storage for
    NMS-assigned serial numbers.

    Write access is not required for agents that can identify
    serial number information for physical entities, but cannot
    provide non-volatile storage for NMS-assigned serial
    numbers.

    Write access is not required for physical entities for
    which the associated value of the entPhysicalIsFRU object
    is equal to 'false(2)'."

  OBJECT entPhysicalAlias
```

```
MIN-ACCESS    read-only
DESCRIPTION
    "Write access is required only if the associated
    entPhysicalClass value is equal to 'chassis(3)'."

OBJECT entPhysicalAssetID
MIN-ACCESS    not-accessible
DESCRIPTION
    "Read and write access is not required for agents that
    cannot provide non-volatile storage for NMS-assigned asset
    identifiers.

    Write access is not required for physical entities for which
    the associated value of entPhysicalIsFRU is equal to
    'false(2)'."
::= { entityCompliances 3 }
```

```
entity4CRCompliance MODULE-COMPLIANCE
STATUS    current
DESCRIPTION
    "The compliance statement for SNMP entities that implement
    version 4 of the Entity MIB on devices with constrained
    resources."
MODULE -- this module
MANDATORY-GROUPS {
    entityPhysicalCRGroup
}

OBJECT entPhysicalUri
SYNTAX Uri    (SIZE (0 | 45))
MIN-ACCESS    read-only
DESCRIPTION
    "Write access is not required for agents that
    implement version 4 of the Entity MIB on devices with
    constrained resources.

    Size is constrained to 45 to allow only for RFC 4122
    compliant values and 0 for cases when the object is
    not instantiated."

::= { entityCompliances 4 }
```

```
-- MIB groupings
entityPhysicalGroup      OBJECT-GROUP
    OBJECTS {
        entPhysicalDescr,
        entPhysicalVendorType,
        entPhysicalContainedIn,
        entPhysicalClass,
        entPhysicalParentRelPos,
        entPhysicalName
    }
    STATUS current
    DESCRIPTION
        "The collection of objects used to represent physical
        system components, for which a single agent provides
        management information."
    ::= { entityGroups 1 }

entityLogicalGroup       OBJECT-GROUP
    OBJECTS {
        entLogicalDescr,
        entLogicalType,
        entLogicalCommunity,
        entLogicalTAddress,
        entLogicalTDomain
    }
    STATUS deprecated
    DESCRIPTION
        "The collection of objects used to represent the list of
        logical entities, for which a single agent provides
        management information."

    ::= { entityGroups 2 }

entityMappingGroup       OBJECT-GROUP
    OBJECTS {
        entLPPPhysicalIndex,
        entAliasMappingIdentifier,
        entPhysicalChildIndex
    }
    STATUS current
    DESCRIPTION
        "The collection of objects used to represent the
        associations between multiple logical entities, physical
        components, interfaces, and port identifiers, for which a
        single agent provides management information."
    ::= { entityGroups 3 }
```

```
entityGeneralGroup      OBJECT-GROUP
  OBJECTS {
    entLastChangeTime
  }
  STATUS current
  DESCRIPTION
    "The collection of objects used to represent general entity
    information, for which a single agent provides management
    information."
  ::= { entityGroups 4 }

entityNotificationsGroup NOTIFICATION-GROUP
  NOTIFICATIONS { entConfigChange }
  STATUS current
  DESCRIPTION
    "The collection of notifications used to indicate Entity MIB
    data consistency and general status information."
  ::= { entityGroups 5 }

entityPhysical2Group     OBJECT-GROUP
  OBJECTS {
    entPhysicalHardwareRev,
    entPhysicalFirmwareRev,
    entPhysicalSoftwareRev,
    entPhysicalSerialNum,
    entPhysicalMfgName,
    entPhysicalModelName,
    entPhysicalAlias,
    entPhysicalAssetID,
    entPhysicalIsFRU
  }
  STATUS current

  DESCRIPTION
    "The collection of objects used to represent physical
    system components, for which a single agent provides
    management information. This group augments the objects
    contained in the entityPhysicalGroup."
  ::= { entityGroups 6 }

entityLogical2Group      OBJECT-GROUP
  OBJECTS {
    entLogicalDescr,
    entLogicalType,
    entLogicalTAddress,
    entLogicalTDomain,
```

```
        entLogicalContextEngineID,
        entLogicalContextName
    }
    STATUS current
    DESCRIPTION
        "The collection of objects used to represent the
        list of logical entities, for which a single SNMP entity
        provides management information."
    ::= { entityGroups 7 }

entityPhysicalCRGroup    OBJECT-GROUP
    OBJECTS {
        entPhysicalName,
        entPhysicalMfgDate,
        entPhysicalUris
    }
    STATUS current
    DESCRIPTION
        "The collection of objects used to represent physical
        system components for constrained resourced devices,
        for which a single agent provides
        management information."
    ::= { entityGroups 8 }
```

END

### 3.2. IANA-ENTITY-MIB

IANA-ENTITY-MIB DEFINITIONS ::= BEGIN

IMPORTS

```
    MODULE-IDENTITY, mib-2
    FROM SNMPv2-SMI
    TEXTUAL-CONVENTION
    FROM SNMPv2-TC
    ;
```

ianaEntityMIB MODULE-IDENTITY

```
    LAST-UPDATED "201206100000Z" -- June 10, 2011
    ORGANIZATION "IANA"
    CONTACT-INFO "        Internet Assigned Numbers Authority
```

```
        Postal: ICANN
                4676 Admiralty Way, Suite 330
                Marina del Rey, CA 90292
```

Tel: +1-310-823-9358  
EMail: iana@iana.org"

## DESCRIPTION

"This MIB module

Copyright (C) The IETF Trust (2012).  
The initial version of this MIB module was published in  
RFC yyyy; for full legal notices see the RFC itself.  
Supplementary information may be available at:  
<http://www.ietf.org/copyrights/ianamib.html>"

REVISION "201206100000Z" -- June 10, 2012

DESCRIPTION "Initial version of this MIB as published in  
RFC yyyy."

::= { mib-2 xxx }

-- RFC Editor, please replace xxx with the IANA allocation for  
-- this MIB module and yyyy with the number of the approved RFC

-- Textual Conventions

IANAPhysicalClass ::= TEXTUAL-CONVENTION

STATUS current

## DESCRIPTION

"An enumerated value which provides an indication of the  
general hardware type of a particular physical entity.  
There are no restrictions as to the number of  
entPhysicalEntries of each entPhysicalClass, which must  
be instantiated by an agent.

The enumeration 'other' is applicable if the physical  
entity class is known, but does not match any of the  
supported values.

The enumeration 'unknown' is applicable if the physical  
entity class is unknown to the agent.

The enumeration 'chassis' is applicable if the physical  
entity class is an overall container for networking  
equipment. Any class of physical entity, except a stack,  
may be contained within a chassis; and a chassis may only  
be contained within a stack.

The enumeration 'backplane' is applicable if the physical entity class is some sort of device for aggregating and forwarding networking traffic, such as a shared backplane in a modular ethernet switch. Note that an agent may model a backplane as a single physical entity, which is actually implemented as multiple discrete physical components (within a chassis or stack).

The enumeration 'container' is applicable if the physical entity class is capable of containing one or more removable physical entities, possibly of different types. For example, each (empty or full) slot in a chassis will be modeled as a container. Note that all removable physical entities should be modeled within a container entity, such as field-replaceable modules, fans, or power supplies. Note that all known containers should be modeled by the agent, including empty containers.

The enumeration 'powerSupply' is applicable if the physical entity class is a power-supplying component.

The enumeration 'fan' is applicable if the physical entity class is a fan or other heat-reduction component.

The enumeration 'sensor' is applicable if the physical entity class is some sort of sensor, such as a temperature sensor within a router chassis.

The enumeration 'module' is applicable if the physical entity class is some sort of self-contained sub-system. If the enumeration 'module' is removable, then it should be modeled within a container entity, otherwise it should be modeled directly within another physical entity (e.g., a chassis or another module).

The enumeration 'port' is applicable if the physical entity class is some sort of networking port, capable of receiving and/or transmitting networking traffic.

The enumeration 'stack' is applicable if the physical entity class is some sort of super-container (possibly virtual), intended to group together multiple chassis entities. A stack may be realized by a 'virtual' cable, a real interconnect cable, attached to multiple chassis, or may in fact be comprised of multiple interconnect cables. A stack should not be modeled within any other physical entities, but a stack may be contained within another stack. Only chassis entities should be contained within a stack.

The enumeration 'cpu' is applicable if the physical entity class is some sort of central processing unit.

The enumeration ?energyObject? is applicable if the physical entity is some sort of a energy object i.e. a piece of equipment that is part of or attached to a communications network that is monitored, controlled, or aids in the management of another device for Energy Management.

The enumeration ?battery? is applicable of the physical entity class is some sort of an energy battery device. "

```
SYNTAX      INTEGER  {
    other(1),
    unknown(2),
    chassis(3),
    backplane(4),
    container(5),      -- e.g., chassis slot or daughter-card holder
    powerSupply(6),
    fan(7),
    sensor(8),
    module(9),         -- e.g., plug-in card or daughter-card
    port(10),
    stack(11),         -- e.g., stack of multiple chassis entities
    cpu(12),
    energyObject(13),
    battery (14)
}
```

END



#### 4. Usage Examples

The following sections iterate the instance values for two example networking devices. These examples are kept simple to make them more understandable. Auxiliary components such as fans, sensors, empty slots, and sub-modules are not shown, but might be modeled in real implementations.

##### 4.1. Router/Bridge

The first example is a router containing two slots. Each slot contains a 3 port router/bridge module. Each port is represented in the ifTable. There are two logical instances of OSPF running and two logical bridges:

```
Physical entities -- entPhysicalTable:
1 Field-replaceable physical chassis:
  entPhysicalDescr.1 ==      'Acme Chassis Model 100'
  entPhysicalVendorType.1 == acmeProducts.chassisTypes.1
  entPhysicalContainedIn.1 == 0
  entPhysicalClass.1 ==      chassis(3)
  entPhysicalParentRelPos.1 == 0
  entPhysicalName.1 ==        '100-A'
  entPhysicalHardwareRev.1 == 'A(1.00.02)'
  entPhysicalSoftwareRev.1 == ''
  entPhysicalFirmwareRev.1 == ''
  entPhysicalSerialNum.1 ==   'C100076544'
  entPhysicalMfgName.1 ==     'Acme'
  entPhysicalModelName.1 ==   '100'
  entPhysicalAlias.1 ==       'cl-SJ17-3-006:rack1:rtr-U3'
  entPhysicalAssetID.1 ==     '0007372293'
  entPhysicalIsFRU.1 ==       true(1)
  entPhysicalMfgDate.1 ==     '2002-5-26,13:30:30.0,-4:0'
  entPhysicalUris.1 ==        'URN:CLEI:CNME120ARA'
2 slots within the chassis:
  entPhysicalDescr.2 ==      'Acme Chassis Slot Type AA'
  entPhysicalVendorType.2 == acmeProducts.slotTypes.1
  entPhysicalContainedIn.2 == 1
  entPhysicalClass.2 ==      container(5)
  entPhysicalParentRelPos.2 == 1
  entPhysicalName.2 ==        'S1'
  entPhysicalHardwareRev.2 == 'B(1.00.01)'
  entPhysicalSoftwareRev.2 == ''
  entPhysicalFirmwareRev.2 == ''
  entPhysicalSerialNum.2 ==   ''
  entPhysicalMfgName.2 ==     'Acme'
  entPhysicalModelName.2 ==   'AA'
  entPhysicalAlias.2 ==       ''
```

```

entPhysicalAssetID.2 == ''
entPhysicalIsFRU.2 == false(2)
entPhysicalMfgDate.2 == '2002-7-26,12:22:12.0,-4:0'
entPhysicalUris.2 == 'URN:CLEI:CNME123ARA'

entPhysicalDescr.3 == 'Acme Chassis Slot Type AA'
entPhysicalVendorType.3 == acmeProducts.slotTypes.1
entPhysicalContainedIn.3 == 1
entPhysicalClass.3 == container(5)
entPhysicalParentRelPos.3 == 2
entPhysicalName.3 == 'S2'
entPhysicalHardwareRev.3 == '1.00.07'
entPhysicalSoftwareRev.3 == ''
entPhysicalFirmwareRev.3 == ''
entPhysicalSerialNum.3 == ''
entPhysicalMfgName.3 == 'Acme'
entPhysicalModelName.3 == 'AA'
entPhysicalAlias.3 == ''
entPhysicalAssetID.3 == ''
entPhysicalIsFRU.3 == false(2)
entPhysicalMfgDate.3 == '2002-7-26,12:12:12.0,-4:0'
entPhysicalUris.3 == 'URN:CLEI:CNME123ARA'

```

## 2 Field-replaceable modules:

Slot 1 contains a module with 3 ports:

```

entPhysicalDescr.4 == 'Acme Router-100'
entPhysicalVendorType.4 == acmeProducts.moduleTypes.14
entPhysicalContainedIn.4 == 2
entPhysicalClass.4 == module(9)
entPhysicalParentRelPos.4 == 1
entPhysicalName.4 == 'M1'
entPhysicalHardwareRev.4 == '1.00.07'
entPhysicalSoftwareRev.4 == '1.4.1'
entPhysicalFirmwareRev.4 == 'A(1.1)'
entPhysicalSerialNum.4 == 'C100087363'
entPhysicalMfgName.4 == 'Acme'
entPhysicalModelName.4 == 'R100-FE'
entPhysicalAlias.4 == 'rtr-U3:m1:SJ17-3-eng'
entPhysicalAssetID.4 == '0007372462'
entPhysicalIsFRU.4 == true(1)
entPhysicalMfgDate.4 == '2003-7-18,13:30:30.0,-4:0'
entPhysicalUris.4 == 'URN:CLEI:CNRU123CAA'

entPhysicalDescr.5 == 'Acme Ethernet-100 Port'
entPhysicalVendorType.5 == acmeProducts.portTypes.2
entPhysicalContainedIn.5 == 4
entPhysicalClass.5 == port(10)
entPhysicalParentRelPos.5 == 1

```

```

entPhysicalName.5 ==          'P1'
entPhysicalHardwareRev.5 ==   'G(1.02)'
entPhysicalSoftwareRev.5 ==   ''
entPhysicalFirmwareRev.5 ==   '1.1'
entPhysicalSerialNum.5 ==     ''
entPhysicalMfgName.5 ==       'Acme'
entPhysicalModelName.5 ==     'FE-100'
entPhysicalAlias.5 ==         ''
entPhysicalAssetID.5 ==       ''
entPhysicalIsFRU.5 ==         false(2)
entPhysicalMfgDate.5 ==       '2003-7-18,14:20:22.0,-4:0'
entPhysicalUris.5 ==          'URN:CLEI:CNMES23ARA'

entPhysicalDescr.6 ==         'Acme Ethernet-100 Port'
entPhysicalVendorType.6 ==    acmeProducts.portTypes.2
entPhysicalContainedIn.6 ==    4
entPhysicalClass.6 ==         port(10)
entPhysicalParentRelPos.6 ==   2
entPhysicalName.6 ==          'P2'
entPhysicalHardwareRev.6 ==   'G(1.02)'
entPhysicalSoftwareRev.6 ==   ''
entPhysicalFirmwareRev.6 ==   '1.1'
entPhysicalSerialNum.6 ==     ''
entPhysicalMfgName.6 ==       'Acme'
entPhysicalModelName.6 ==     'FE-100'
entPhysicalAlias.6 ==         ''
entPhysicalAssetID.6 ==       ''
entPhysicalIsFRU.6 ==         false(2)
entPhysicalMfgDate.6 ==       '2003-7-19,10:15:15.0,-4:0'
entPhysicalUris.6 ==          'URN:CLEI:CNMES23ARA'

entPhysicalDescr.7 ==         'Acme Router-100 FDDI-Port'
entPhysicalVendorType.7 ==    acmeProducts.portTypes.3
entPhysicalContainedIn.7 ==    4
entPhysicalClass.7 ==         port(10)
entPhysicalParentRelPos.7 ==   3
entPhysicalName.7 ==          'P3'
entPhysicalHardwareRev.7 ==   'B(1.03)'
entPhysicalSoftwareRev.7 ==   '2.5.1'
entPhysicalFirmwareRev.7 ==   '2.5F'
entPhysicalSerialNum.7 ==     ''
entPhysicalMfgName.7 ==       'Acme'
entPhysicalModelName.7 ==     'FDDI-100'
entPhysicalAlias.7 ==         ''
entPhysicalAssetID.7 ==       ''
entPhysicalIsFRU.7 ==         false(2)

```

Slot 2 contains another 3-port module:

```

entPhysicalDescr.8 ==      'Acme Router-100 Comm Module'
entPhysicalVendorType.8 == acmeProducts.moduleTypes.15
entPhysicalContainedIn.8 == 3
entPhysicalClass.8 ==      module(9)
entPhysicalParentRelPos.8 == 1
entPhysicalName.8 ==        'M2'
entPhysicalHardwareRev.8 == '2.01.00'
entPhysicalSoftwareRev.8 == '3.0.7'
entPhysicalFirmwareRev.8 == 'A(1.2)'
entPhysicalSerialNum.8 ==   'C100098732'
entPhysicalMfgName.8 ==     'Acme'
entPhysicalModelName.8 ==   'C100'
entPhysicalAlias.8 ==       'rtr-U3:m2:SJ17-2-eng'
entPhysicalAssetID.8 ==     '0007373982'
entPhysicalIsFRU.8 ==       true(1)
entPhysicalMfgDate.8 ==     '2002-5-26,13:30:15.0,-4:0'
entPhysicalUris.8 ==        'URN:CLEI:CNRT321MAA'

entPhysicalDescr.9 ==      'Acme Fddi-100 Port'
entPhysicalVendorType.9 == acmeProducts.portTypes.5
entPhysicalContainedIn.9 == 8
entPhysicalClass.9 ==      port(10)
entPhysicalParentRelPos.9 == 1
entPhysicalName.9 ==        'FDDI Primary'
entPhysicalHardwareRev.9 == 'CC(1.07)'
entPhysicalSoftwareRev.9 == '2.0.34'
entPhysicalFirmwareRev.9 == '1.1'
entPhysicalSerialNum.9 ==   ''
entPhysicalMfgName.9 ==     'Acme'
entPhysicalModelName.9 ==   'FDDI-100'
entPhysicalAlias.9 ==       ''
entPhysicalAssetID.9 ==     ''
entPhysicalIsFRU.9 ==       false(2)

entPhysicalDescr.10 ==     'Acme Ethernet-100 Port'
entPhysicalVendorType.10 == acmeProducts.portTypes.2
entPhysicalContainedIn.10 == 8
entPhysicalClass.10 ==     port(10)
entPhysicalParentRelPos.10 == 2
entPhysicalName.10 ==      'Ethernet A'
entPhysicalHardwareRev.10 == 'G(1.04)'
entPhysicalSoftwareRev.10 == ''
entPhysicalFirmwareRev.10 == '1.3'
entPhysicalSerialNum.10 ==  ''
entPhysicalMfgName.10 ==    'Acme'
entPhysicalModelName.10 ==  'FE-100'
entPhysicalAlias.10 ==     ''

```

```

entPhysicalAssetID.10 ==      ''
entPhysicalIsFRU.10 ==      false(2)
entPhysicalMfgDate.10 ==     '2002-7-26,13:30:15.0,-4:0'
entPhysicalUris.10 ==        'URN:CLEI:CNMES23ARA'

entPhysicalDescr.11 ==       'Acme Ethernet-100 Port'
entPhysicalVendorType.11 ==  acmeProducts.portTypes.2
entPhysicalContainedIn.11 ==  8
entPhysicalClass.11 ==       port(10)
entPhysicalParentRelPos.11 == 3
entPhysicalName.11 ==        'Ethernet B'
entPhysicalHardwareRev.11 == 'G(1.04)'
entPhysicalSoftwareRev.11 == ''
entPhysicalFirmwareRev.11 == '1.3'
entPhysicalSerialNum.11 ==   ''
entPhysicalMfgName.11 ==     'Acme'
entPhysicalModelName.11 ==   'FE-100'
entPhysicalAlias.11 ==       ''
entPhysicalAssetID.11 ==     ''
entPhysicalIsFRU.11 ==       false(2)
entPhysicalMfgDate.11 ==     '2002-8-16,15:35:15.0,-4:0'
entPhysicalUris.11 ==        'URN:CLEI:CNMES23ARA'

```

Logical entities -- entLogicalTable; no SNMPv3 support

2 OSPF instances:

```

entLogicalDescr.1 ==         'Acme OSPF v1.1'
entLogicalType.1 ==          ospf
entLogicalCommunity.1 ==     'public-ospf1'
entLogicalTAddress.1 ==      192.0.2.1:161
entLogicalTDomain.1 ==       snmpUDPDomain
entLogicalContextEngineID.1 == ''
entLogicalContextName.1 ==   ''

entLogicalDescr.2 ==         'Acme OSPF v1.1'
entLogicalType.2 ==          ospf
entLogicalCommunity.2 ==     'public-ospf2'
entLogicalTAddress.2 ==      192.0.2.1:161
entLogicalTDomain.2 ==       snmpUDPDomain
entLogicalContextEngineID.2 == ''
entLogicalContextName.2 ==   ''

```

2 logical bridges:

```

entLogicalDescr.3 ==         'Acme Bridge v2.1.1'
entLogicalType.3 ==          dot1dBridge
entLogicalCommunity.3 ==     'public-bridge1'
entLogicalTAddress.3 ==      192.0.2.1:161
entLogicalTDomain.3 ==       snmpUDPDomain
entLogicalContextEngineID.3 == ''

```

```

entLogicalContextName.3 ==      ''

entLogicalDescr.4 ==            'Acme Bridge v2.1.1'
entLogicalType.4 ==             dot1dBridge
entLogicalCommunity.4 ==        'public-bridge2'
entLogicalTAddress.4 ==         192.0.2.1:161
entLogicalTDomain.4 ==          snmpUDPDomain
entLogicalContextEngineID.4 ==  ''
entLogicalContextName.4 ==      ''

```

#### Logical to Physical Mappings:

```

1st OSPF instance: uses module 1-port 1
entLPPPhysicalIndex.1.5 ==      5

```

```

2nd OSPF instance: uses module 2-port 1
entLPPPhysicalIndex.2.9 ==      9

```

1st bridge group: uses module 1, all ports

[ed. -- Note that these mappings are included in the table because another logical entity (1st OSPF) utilizes one of the ports. If this were not the case, then a single mapping to the module (e.g., entLPPPhysicalIndex.3.4) would be present instead.]

```

entLPPPhysicalIndex.3.5 ==      5
entLPPPhysicalIndex.3.6 ==      6
entLPPPhysicalIndex.3.7 ==      7

```

2nd bridge group: uses module 2, all ports

```

entLPPPhysicalIndex.4.9 ==      9
entLPPPhysicalIndex.4.10 ==     10
entLPPPhysicalIndex.4.11 ==     11

```

#### Physical to Logical to MIB Alias Mappings -- entAliasMappingTable:

Example 1: ifIndex values are global to all logical entities

```

entAliasMappingIdentifier.5.0 == ifIndex.1
entAliasMappingIdentifier.6.0 == ifIndex.2
entAliasMappingIdentifier.7.0 == ifIndex.3
entAliasMappingIdentifier.9.0 == ifIndex.4
entAliasMappingIdentifier.10.0 == ifIndex.5
entAliasMappingIdentifier.11.0 == ifIndex.6

```

Example 2: ifIndex values are not shared by all logical entities;  
(Bridge-1 uses ifIndex values 101 - 103 and Bridge-2 uses  
ifIndex values 204-206.)

```

entAliasMappingIdentifier.5.0 == ifIndex.1
entAliasMappingIdentifier.5.3 == ifIndex.101
entAliasMappingIdentifier.6.0 == ifIndex.2

```

```

entAliasMappingIdentifier.6.3 == ifIndex.102
entAliasMappingIdentifier.7.0 == ifIndex.3
entAliasMappingIdentifier.7.3 == ifIndex.103
entAliasMappingIdentifier.9.0 == ifIndex.4
entAliasMappingIdentifier.9.4 == ifIndex.204
entAliasMappingIdentifier.10.0 == ifIndex.5
entAliasMappingIdentifier.10.4 == ifIndex.205
entAliasMappingIdentifier.11.0 == ifIndex.6
entAliasMappingIdentifier.11.4 == ifIndex.206

```

#### Physical Containment Tree -- entPhysicalContainsTable

chassis has two containers:

```

entPhysicalChildIndex.1.2 ==      2
entPhysicalChildIndex.1.3 ==      3

```

container 1 has a module:

```

entPhysicalChildIndex.2.4 ==      4

```

container 2 has a module:

```

entPhysicalChildIndex.3.8 ==      8

```

module 1 has 3 ports:

```

entPhysicalChildIndex.4.5 ==      5
entPhysicalChildIndex.4.6 ==      6
entPhysicalChildIndex.4.7 ==      7

```

module 2 has 3 ports:

```

entPhysicalChildIndex.8.9 ==      9
entPhysicalChildIndex.8.10 ==     10
entPhysicalChildIndex.8.11 ==     11

```

#### 4.2. Repeaters

The second example is a 3-slot Hub with 2 backplane ethernet segments. Slot three is empty, and the remaining slots contain ethernet repeater modules.

Note that this example assumes an older Repeater MIB implementation, (RFC 1516 [RFC1516]) rather than the new Repeater MIB (RFC 2108 [RFC2108]). The new version contains an object called 'rpPtrPortRpPtrId', which should be used to identify repeater port groupings, rather than using community strings or contexts.

Physical entities -- entPhysicalTable:

1 Field-replaceable physical chassis:

```

entPhysicalDescr.1 ==      'Acme Chassis Model 110'
entPhysicalVendorType.1 ==  acmeProducts.chassisTypes.2
entPhysicalContainedIn.1 == 0

```

```
entPhysicalClass.1 ==      chassis(3)
entPhysicalParentRelPos.1 ==0
entPhysicalName.1 ==      '110-B'
entPhysicalHardwareRev.1 == 'A(1.02.00)'
entPhysicalSoftwareRev.1 == ''
entPhysicalFirmwareRev.1 == ''
entPhysicalSerialNum.1 ==  'C100079294'
entPhysicalMfgName.1 ==    'Acme'
entPhysicalModelName.1 ==  '110'
entPhysicalAlias.1 ==      'bldg09:floor1:rp1r18:0067eea0229f'
entPhysicalAssetID.1 ==    '0007386327'
entPhysicalIsFRU.1 ==      true(1)
```

## 2 Chassis Ethernet Backplanes:

```
entPhysicalDescr.2 ==      'Acme Ethernet Backplane Type A'
entPhysicalVendorType.2 ==  acmeProducts.backplaneTypes.1
entPhysicalContainedIn.2 == 1
entPhysicalClass.2 ==      backplane(4)
entPhysicalParentRelPos.2 == 1
entPhysicalName.2 ==        'B1'
entPhysicalHardwareRev.2 == 'A(2.04.01)'
entPhysicalSoftwareRev.2 == ''
entPhysicalFirmwareRev.2 == ''
entPhysicalSerialNum.2 ==   ''
entPhysicalMfgName.2 ==     'Acme'
entPhysicalModelName.2 ==   'BK-A'
entPhysicalAlias.2 ==       ''
entPhysicalAssetID.2 ==     ''
entPhysicalIsFRU.2 ==       false(2)
```

```
entPhysicalDescr.3 ==      'Acme Ethernet Backplane Type A'
entPhysicalVendorType.3 ==  acmeProducts.backplaneTypes.1
entPhysicalContainedIn.3 == 1
entPhysicalClass.3 ==      backplane(4)
entPhysicalParentRelPos.3 == 2
entPhysicalName.3 ==        'B2'
entPhysicalHardwareRev.3 == 'A(2.04.01)'
entPhysicalSoftwareRev.3 == ''
entPhysicalFirmwareRev.3 == ''
entPhysicalSerialNum.3 ==   ''
entPhysicalMfgName.3 ==     'Acme'
entPhysicalModelName.3 ==   'BK-A'
entPhysicalAlias.3 ==       ''
entPhysicalAssetID.3 ==     ''
entPhysicalIsFRU.3 ==       false(2)
```



```
3 slots within the chassis:
  entPhysicalDescr.4 ==      'Acme Hub Slot Type RB'
  entPhysicalVendorType.4 == acmeProducts.slotTypes.5
  entPhysicalContainedIn.4 == 1
  entPhysicalClass.4 ==      container(5)
  entPhysicalParentRelPos.4 == 1
  entPhysicalName.4 ==        'Slot 1'
  entPhysicalHardwareRev.4 == 'B(1.00.03)'
  entPhysicalSoftwareRev.4 == ''
  entPhysicalFirmwareRev.4 == ''
  entPhysicalSerialNum.4 ==   ''
  entPhysicalMfgName.4 ==     'Acme'
  entPhysicalModelName.4 ==   'RB'
  entPhysicalAlias.4 ==       ''
  entPhysicalAssetID.4 ==     ''
  entPhysicalIsFRU.4 ==       false(2)

  entPhysicalDescr.5 ==      'Acme Hub Slot Type RB'
  entPhysicalVendorType.5 == acmeProducts.slotTypes.5
  entPhysicalContainedIn.5 == 1
  entPhysicalClass.5 ==      container(5)
  entPhysicalParentRelPos.5 == 2
  entPhysicalName.5 ==        'Slot 2'
  entPhysicalHardwareRev.5 == 'B(1.00.03)'
  entPhysicalSoftwareRev.5 == ''
  entPhysicalFirmwareRev.5 == ''
  entPhysicalSerialNum.5 ==   ''
  entPhysicalMfgName.5 ==     'Acme'
  entPhysicalModelName.5 ==   'RB'
  entPhysicalAlias.5 ==       ''
  entPhysicalAssetID.5 ==     ''
  entPhysicalIsFRU.5 ==       false(2)

  entPhysicalDescr.6 ==      'Acme Hub Slot Type RB'
  entPhysicalVendorType.6 == acmeProducts.slotTypes.5
  entPhysicalContainedIn.6 == 1
  entPhysicalClass.6 ==      container(5)
  entPhysicalParentRelPos.6 == 3
  entPhysicalName.6 ==        'Slot 3'
  entPhysicalHardwareRev.6 == 'B(1.00.03)'
  entPhysicalSoftwareRev.6 == ''
  entPhysicalFirmwareRev.6 == ''
  entPhysicalSerialNum.6 ==   ''
  entPhysicalMfgName.6 ==     'Acme'
  entPhysicalModelName.6 ==   'RB'
  entPhysicalAlias.6 ==       ''
  entPhysicalAssetID.6 ==     ''
  entPhysicalIsFRU.6 ==       false(2)
```

Slot 1 contains a plug-in module with 4 10-BaseT ports:

```
entPhysicalDescr.7 ==      'Acme 10Base-T Module 114'
entPhysicalVendorType.7 == acmeProducts.moduleTypes.32
entPhysicalContainedIn.7 == 4
entPhysicalClass.7 ==      module(9)
entPhysicalParentRelPos.7 == 1
entPhysicalName.7 ==       'M1'
entPhysicalHardwareRev.7 == 'A(1.02.01)'
entPhysicalSoftwareRev.7 == '1.7.2'
entPhysicalFirmwareRev.7 == 'A(1.5)'
entPhysicalSerialNum.7 ==   'C100096244'
entPhysicalMfgName.7 ==     'Acme'
entPhysicalModelName.7 ==   '114'
entPhysicalAlias.7 ==       'bldg09:floor1:eng'
entPhysicalAssetID.7 ==     '0007962951'
entPhysicalIsFRU.7 ==       true(1)

entPhysicalDescr.8 ==      'Acme 10Base-T Port RB'
entPhysicalVendorType.8 == acmeProducts.portTypes.10
entPhysicalContainedIn.8 == 7
entPhysicalClass.8 ==      port(10)
entPhysicalParentRelPos.8 == 1
entPhysicalName.8 ==       'Ethernet-A'
entPhysicalHardwareRev.8 == 'A(1.04F)'
entPhysicalSoftwareRev.8 == ''
entPhysicalFirmwareRev.8 == '1.4'
entPhysicalSerialNum.8 ==   ''
entPhysicalMfgName.8 ==     'Acme'
entPhysicalModelName.8 ==   'RB'
entPhysicalAlias.8 ==       ''
entPhysicalAssetID.8 ==     ''
entPhysicalIsFRU.8 ==       false(2)

entPhysicalDescr.9 ==      'Acme 10Base-T Port RB'
entPhysicalVendorType.9 == acmeProducts.portTypes.10
entPhysicalContainedIn.9 == 7
entPhysicalClass.9 ==      port(10)
entPhysicalParentRelPos.9 == 2
entPhysicalName.9 ==       'Ethernet-B'
entPhysicalHardwareRev.9 == 'A(1.04F)'
entPhysicalSoftwareRev.9 == ''
entPhysicalFirmwareRev.9 == '1.4'
entPhysicalSerialNum.9 ==   ''
entPhysicalMfgName.9 ==     'Acme'
entPhysicalModelName.9 ==   'RB'
entPhysicalAlias.9 ==       ''
entPhysicalAssetID.9 ==     ''
entPhysicalIsFRU.9 ==       false(2)
```

```
entPhysicalDescr.10 ==      'Acme 10Base-T Port RB'
entPhysicalVendorType.10 == acmeProducts.portTypes.10
entPhysicalContainedIn.10 == 7
entPhysicalClass.10 ==      port(10)
entPhysicalParentRelPos.10 == 3
entPhysicalName.10 ==       'Ethernet-C'
entPhysicalHardwareRev.10 == 'B(1.02.07)'
entPhysicalSoftwareRev.10 == ''
entPhysicalFirmwareRev.10 == '1.4'
entPhysicalSerialNum.10 ==   ''
entPhysicalMfgName.10 ==     'Acme'
entPhysicalModelName.10 ==   'RB'
entPhysicalAlias.10 ==       ''
entPhysicalAssetID.10 ==     ''
entPhysicalIsFRU.10 ==      false(2)
```

```
entPhysicalDescr.11 ==      'Acme 10Base-T Port RB'
entPhysicalVendorType.11 == acmeProducts.portTypes.10
entPhysicalContainedIn.11 == 7
entPhysicalClass.11 ==      port(10)
entPhysicalParentRelPos.11 == 4
entPhysicalName.11 ==       'Ethernet-D'
entPhysicalHardwareRev.11 == 'B(1.02.07)'
entPhysicalSoftwareRev.11 == ''
entPhysicalFirmwareRev.11 == '1.4'
entPhysicalSerialNum.11 ==   ''
entPhysicalMfgName.11 ==     'Acme'
entPhysicalModelName.11 ==   'RB'
entPhysicalAlias.11 ==       ''
entPhysicalAssetID.11 ==     ''
entPhysicalIsFRU.11 ==      false(2)
```

Slot 2 contains another ethernet module with 2 ports.

```
entPhysicalDescr.12 ==      'Acme 10Base-T Module Model 4'
entPhysicalVendorType.12 == acmeProducts.moduleTypes.30
entPhysicalContainedIn.12 == 5
entPhysicalClass.12 ==      module(9)
entPhysicalParentRelPos.12 == 1
entPhysicalName.12 ==       'M2'
entPhysicalHardwareRev.12 == 'A(1.01.07)'
entPhysicalSoftwareRev.12 == '1.8.4'
entPhysicalFirmwareRev.12 == 'A(1.8)'
entPhysicalSerialNum.12 ==   'C100102384'
entPhysicalMfgName.12 ==     'Acme'
entPhysicalModelName.12 ==   '4'
entPhysicalAlias.12 ==       'bldg09:floor1:devtest'
entPhysicalAssetID.12 ==     '0007968462'
entPhysicalIsFRU.12 ==      true(1)
```

```
entPhysicalDescr.13 ==      'Acme 802.3 AUI Port'
entPhysicalVendorType.13 == acmeProducts.portTypes.11
entPhysicalContainedIn.13 == 12
entPhysicalClass.13 ==      port(10)
entPhysicalParentRelPos.13 == 1
entPhysicalName.13 ==       'AUI'
entPhysicalHardwareRev.13 == 'A(1.06F)'
entPhysicalSoftwareRev.13 == ''
entPhysicalFirmwareRev.13 == '1.5'
entPhysicalSerialNum.13 ==  ''
entPhysicalMfgName.13 ==    'Acme'
entPhysicalModelName.13 ==  ''
entPhysicalAlias.13 ==      ''
entPhysicalAssetID.13 ==    ''
entPhysicalIsFRU.13 ==      false(2)

entPhysicalDescr.14 ==      'Acme 10Base-T Port RD'
entPhysicalVendorType.14 == acmeProducts.portTypes.14
entPhysicalContainedIn.14 == 12
entPhysicalClass.14 ==      port(10)
entPhysicalParentRelPos.14 == 2
entPhysicalName.14 ==       'E2'
entPhysicalHardwareRev.14 == 'B(1.01.02)'
entPhysicalSoftwareRev.14 == ''
entPhysicalFirmwareRev.14 == '2.1'
entPhysicalSerialNum.14 ==  ''
entPhysicalMfgName.14 ==    'Acme'
entPhysicalModelName.14 ==  ''
entPhysicalAlias.14 ==      ''
entPhysicalAssetID.14 ==    ''
entPhysicalIsFRU.14 ==      false(2)
```

Logical entities -- entLogicalTable; with SNMPv3 support

Repeater 1--comprised of any ports attached to backplane 1

```
entLogicalDescr.1 ==      'Acme repeater v3.1'
entLogicalType.1 ==       snmpDot3RptrMgt
entLogicalCommunity.1 ==  'public-repeater1'
entLogicalTAddress.1 ==   192.0.2.1:161
entLogicalTDomain.1 ==    snmpUDPDomain
entLogicalContextEngineID.1 == '80000777017c7d7e7f'H
entLogicalContextName.1 == 'repeater1'
```

Repeater 2--comprised of any ports attached to backplane 2:

```
entLogicalDescr.2 ==      'Acme repeater v3.1'
entLogicalType.2 ==       snmpDot3RptrMgt
entLogicalCommunity.2 ==  'public-repeater2'
entLogicalTAddress.2 ==   192.0.2.1:161
entLogicalTDomain.2 ==    snmpUDPDomain
```

```
entLogicalContextEngineID.2 == '80000777017c7d7e7f'H
entLogicalContextName.2 ==    'repeater2'
```

Logical to Physical Mappings -- entLPMappingTable:

```
repeater1 uses backplane 1, slot 1-ports 1 & 2, slot 2-port 1
[ed. -- Note that a mapping to the module is not included,
because this example represents a port-switchable hub.
Even though all ports on the module could belong to the
same repeater as a matter of configuration, the LP port
mappings should not be replaced dynamically with a single
mapping for the module (e.g., entLPPhysicalIndex.1.7).
If all ports on the module shared a single backplane connection,
then a single mapping for the module would be more appropriate.]
```

```
entLPPhysicalIndex.1.2 ==      2
entLPPhysicalIndex.1.8 ==      8
entLPPhysicalIndex.1.9 ==      9
entLPPhysicalIndex.1.13 ==     13
```

```
repeater2 uses backplane 2, slot 1-ports 3 & 4, slot 2-port 2
entLPPhysicalIndex.2.3 ==      3
entLPPhysicalIndex.2.10 ==     10
entLPPhysicalIndex.2.11 ==     11
entLPPhysicalIndex.2.14 ==     14
```

Physical to Logical to MIB Alias Mappings -- entAliasMappingTable:

Repeater Port Identifier values are shared by both repeaters:

```
entAliasMappingIdentifier.8.0 == rptrPortGroupIndex.1.1
entAliasMappingIdentifier.9.0 == rptrPortGroupIndex.1.2
entAliasMappingIdentifier.10.0 == rptrPortGroupIndex.1.3
entAliasMappingIdentifier.11.0 == rptrPortGroupIndex.1.4
entAliasMappingIdentifier.13.0 == rptrPortGroupIndex.2.1
entAliasMappingIdentifier.14.0 == rptrPortGroupIndex.2.2
```

Physical Containment Tree -- entPhysicalContainsTable

chassis has two backplanes and three containers:

```
entPhysicalChildIndex.1.2 ==    2
entPhysicalChildIndex.1.3 ==    3
entPhysicalChildIndex.1.4 ==    4
entPhysicalChildIndex.1.5 ==    5
entPhysicalChildIndex.1.6 ==    6
```

container 1 has a module:

```
entPhysicalChildIndex.4.7 ==    7
```

container 2 has a module

```
entPhysicalChildIndex.5.12 ==   12
```

[ed. -- in this example, container 3 is empty.]

```
module 1 has 4 ports:
    entPhysicalChildIndex.7.8 == 8
    entPhysicalChildIndex.7.9 == 9
    entPhysicalChildIndex.7.10 == 10
    entPhysicalChildIndex.7.11 == 11

module 2 has 2 ports:
    entPhysicalChildIndex.12.13 == 13
    entPhysicalChildIndex.12.14 == 14
```

## 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 network operations.

There are a number of managed objects in this MIB that may contain sensitive information. These are:

```
entPhysicalDescr
entPhysicalVendorType
entPhysicalHardwareRev
entPhysicalFirmwareRev
entPhysicalSoftwareRev
entPhysicalSerialNum
entPhysicalMfgName
entPhysicalModelName
```

These objects expose information about the physical entities within a managed system, which may be used to identify the vendor, model, and version information of each system component.

```
entPhysicalAssetID
```

This object can allow asset identifiers for various system components to be exposed, in the event this MIB object is actually configured by an NMS application.

```
entLogicalDescr
entLogicalType
```

These objects expose the type of logical entities present in the managed system.

#### entLogicalCommunity

This object exposes community names associated with particular logical entities within the system.

#### entLogicalTAddress entLogicalTDomain

These objects expose network addresses that can be used to communicate with an SNMP agent on behalf of particular logical entities within the system.

#### entLogicalContextEngineID entLogicalContextName

These objects identify the authoritative SNMP engine that contains information on behalf of particular logical entities within the system.

It is thus important to control even GET access to these objects and possibly to even encrypt the values of these object when sending them over the network via SNMP. Not all versions of SNMP provide features for such a secure environment.

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

It is recommended that the implementers consider the security features as provided by the SNMPv3 framework. Specifically, the use of the User-based Security Model RFC 3414 [RFC3414] and the View-based Access Control Model RFC 3415 [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. 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
-----	-----
entityMIB	{ mib-2 47 }

## 7. Acknowledgements

The first three versions of RFCs on the ENTITY MIB were authored by A. Bierman and K. McCloghrie. The authors would like thank A. Bierman and K. McCloghrie for the earlier versions of ENTITY MIB.

The motivation for the extension to RFC 4133 stems from the requirements of the EMAN WG at IETF.

The authors also thank Juergen Schoenwaelder for his review and comments on this draft.

## 8. Open Issues

Open issue 1: entPhysicalUUID object with a compact UUID representation instead of using the URI format.

## 9. References

### 9.1. Normative References

- [RFC2578] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.
- [RFC2579] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "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.
- [RFC3411] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks", STD 62, RFC 3411, December 2002.
- [RFC3417] Presuhn, R., "Transport Mappings for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3417, December 2002.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.
- [RFC2737] McCloghrie, K. and A. Bierman, "Entity MIB (Version



2)", RFC 2737, December 1999.

- [RFC4133] McCloghrie, K. and A. Bierman, "Entity MIB (Version 3)", RFC 4133, August 2005.

## 9.2. Informative References

- [RFC1157] Case, J., Fedor, M., Schoffstall, M., and J. Davin, "Simple Network Management Protocol", STD 15, RFC 1157, May 1990.
- [RFC1493] Decker, E., Langille, P., Rijsinghani, A., and K. McCloghrie, "Definitions of Managed Objects for Bridges", RFC 1493, July 1993.
- [RFC1516] McMaster, D. and K. McCloghrie, "Definitions of Managed Objects for IEEE 802.3 Repeater Devices", RFC 1516, September 1993.
- [RFC4122] Leach, P., Mealling, M., Salz, R., "A Universally Unique IDentifier (UUID) URN Namespace", RFC 4122, July 2005.
- [RFC2108] de Graaf, K., Romascanu, D., McMaster, D., and K. McCloghrie, "Definitions of Managed Objects for IEEE 802.3 Repeater Devices using SMIV2", RFC 2108, February 1997.
- [RFC2737] McCloghrie, K. and A. Bierman, "Entity MIB (Version 2)", RFC 2737, December 1999.
- [RFC4133] McCloghrie, K. and A. Bierman, "Entity MIB (Version 3)", RFC 4133, August 2005.
- [RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, June 2000.
- [RFC3406] Daigle, L., van Gulik, D., Iannella, R., and P. Faltstrom, "Uniform Resource Names (URN) Namespace Definition Mechanisms", BCP 66, RFC 3406, October 2002.
- [RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, 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.
- [RFC4152] Tesink, K. and R. Fox, "A Uniform Resource Name (URN) Namespace for the CLEI Code", RFC 4152, August 2005.
- [EMAN-FMWK] Parello, J. Claise, B. Schoening, B. Quittek, J. and Nordman, B. "Energy Management Framework", draft-ietf-eman-framework-04, March 2012.
- [T1.213] ATIS T1.213-2001, "Coded Identification of Equipment Entities in the North American Telecommunications System for Information Exchange", 2001, [www.ansi.org](http://www.ansi.org).
- [T1.213a] ATIS T1.213a, "Supplement to T1.213-2001, Coded Identification of Equipment Entities in the North American Telecommunications System for Information Exchange, to correct the representation of the Basic Code in Figure B.1", 2001, [www.ansi.org](http://www.ansi.org).

#### Authors' Addresses

Andy Bierman  
Yumaworks  
[andy@yumaworks.com](mailto:andy@yumaworks.com)

Dan Romascanu  
AVAYA  
Park Atidim, Bldg. #3  
Tel Aviv, 61581  
Israel

+972-3-6458414  
[dromasca@avaya.com](mailto:dromasca@avaya.com)

Juergen Quittek  
NEC Europe Ltd.  
Network Research Division  
Kurfuersten-Anlage 36  
Heidelberg 69115  
DE

Phone: +49 6221 4342-115  
Email: quittek@neclab.eu

Mouli Chandramouli  
Cisco Systems, Inc.  
Sarjapur Outer Ring Road  
Bangalore 561013  
IN

Phone: +91 80 4426 3947  
Email: moulchan@cisco.com

Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: August 18, 2013

M. Ersue, Ed.  
Nokia Siemens Networks  
D. Romascanu, Ed.  
Avaya  
J. Schoenwaelder, Ed.  
Jacobs University Bremen  
February 14, 2013

Management of Networks with Constrained Devices: Problem Statement, Use  
Cases and Requirements  
draft-ersue-constrained-mgmt-03

#### Abstract

This document provides a problem statement and discusses the use cases and requirements for the management of networks with constrained devices.

#### Status of this Memo

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

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

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

This Internet-Draft will expire on August 18, 2013.

#### Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	4
1.1. Overview . . . . .	4
1.2. Terminology . . . . .	5
1.3. Class of Networks in Focus . . . . .	7
1.4. Constrained Device Deployment Options . . . . .	10
1.5. Management Topology Options . . . . .	11
1.6. Managing the Constrainedness of a Device or Network . . . . .	11
2. Problem Statement . . . . .	15
3. Use Cases . . . . .	17
3.1. Environmental Monitoring . . . . .	17
3.2. Medical Applications . . . . .	17
3.3. Industrial Applications . . . . .	18
3.4. Home Automation . . . . .	19
3.5. Building Automation . . . . .	20
3.6. Energy Management . . . . .	22
3.7. Transport Applications . . . . .	23
3.8. Infrastructure Monitoring . . . . .	24
3.9. Community Network Applications . . . . .	25
3.10. Mobile Applications . . . . .	27
3.11. Automated Metering Infrastructure (AMI) . . . . .	29
3.12. MANET Concept of Operations (CONOPS) in Military . . . . .	31
4. Requirements on the Management of Networks with Constrained Devices . . . . .	36
4.1. Management Architecture/System . . . . .	36
4.2. Management protocols and data model . . . . .	41
4.3. Configuration management . . . . .	44
4.4. Monitoring functionality . . . . .	46
4.5. Self-management . . . . .	51
4.6. Security and Access Control . . . . .	52
4.7. Energy Management . . . . .	54
4.8. SW Distribution . . . . .	56
4.9. Traffic management . . . . .	56
4.10. Transport Layer . . . . .	57
4.11. Implementation Requirements . . . . .	59
5. IANA Considerations . . . . .	61
6. Security Considerations . . . . .	62
7. Contributors . . . . .	63
8. Acknowledgments . . . . .	64
9. References . . . . .	65
9.1. Normative References . . . . .	65
9.2. Informative References . . . . .	65
Appendix A. Related Development in other Bodies . . . . .	67

A.1.	ETSI TC M2M . . . . .	67
A.2.	OASIS . . . . .	68
A.3.	OMA . . . . .	69
A.4.	IPSO Alliance . . . . .	69
Appendix B.	Related Research Projects . . . . .	71
Appendix C.	Open issues . . . . .	72
Appendix D.	Change Log . . . . .	73
D.1.	02-03 . . . . .	73
D.2.	01-02 . . . . .	74
D.3.	00-01 . . . . .	74
Authors' Addresses	. . . . .	76

## 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 becoming 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 raises the questions on and aims to understand the use cases and requirements for the management of a network with constrained devices. The document especially aims to avoid recommending any particular solutions. Section 1.3 and Section 1.5 describe different topology options for the networking and management of constrained devices. Section 1.4 explains different deployment options for the networking of constrained devices. Section 2 provides a problem statement on the issue of the management of networked constrained devices. Section 3 lists diverse use cases and scenarios for the management from the network as well as from the application point of view. Section 4 lists requirements on the management of applications and networks with constrained devices. Note that the requirements in Section 4 need to be seen as standalone requirements. As of today this document does not recommend the realization of a profile of requirements.

## 1.2. Terminology

Concerning constrained devices and networks this document generally builds on the terminology defined in [LWIG-TERMS]. As such the terms like Constrained Device, Constrained Network, etc. are defined in [LWIG-TERMS].

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 [LWIG-TERMS].

C1: Class 1 constrained device as defined in Section 3. of [LWIG-TERMS].

C2: Class 2 constrained device as defined in Section 3. of [LWIG-TERMS].

Client: The originating endpoint of a request; the destination endpoint of a response.

Intermediary entity: As defined in the CoAP document an intermediary entity can be a CoAP endpoint that acts both as a server and as a client towards (possibly via further intermediaries) an origin server. An intermediary entity can be used to support hierarchical management.

Network of Constrained Devices: A network to which constrained devices are connected. It may or may not be a Constrained Network (see [LWIG-TERMS] 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.

Mote: A sensor node in a wireless network that is capable of performing some limited processing, gathering sensory information and communicating with other connected nodes in the network.

Server: The destination endpoint of a request; the originating endpoint of a response.

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) that records consumption of electric energy in intervals of an hour or less and communicates that information at least daily back to the utility network for monitoring and billing purposes.

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

### 1.3. Class of Networks 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.)

- o Wireline non-constrained networks (CN0), e.g. an Ethernet-LAN with non-constrained and constrained devices involved.
- o A combination of wireline and wireless networks (CN1), 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. CN1 usually support highly distributed applications with many nodes (e.g. environmental monitoring). CN1 tend to deal with large-scale multipoint-to-point systems with massive data flows. Wireless Mesh Networks (WMN), as a specific type of CN1 networks, 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.
- o A combination of wireline and wireless networks with point-to-point or point-to-multipoint communication (CN2) generally with single-hop connectivity to constrained devices, utilizing static routing over the wireless network. CN2 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. CN2 usually 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 CN2 networks.
- o 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.

A CN0 is used for specific applications like Building Automation or Infrastructure Monitoring. However, CN1 and CN2 networks 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 (as described in Section 3.9, the topology is not pre-planned, and thus may be unknown for management purposes. In other use cases, such as industrial applications (as described in Section 3.3, 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 Ethernet networks (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 Section 3.5).

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 Section 3.5).

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. The use case in Section 3.4 has been provided 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 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, failures detected by the transport protocol and other self-monitoring mechanisms 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 might become necessary to design as much as possible simplified and less complex.

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 Section 3 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 4.

### 3. Use Cases

This section discusses some application scenarios where networks of constrained devices are expected to be deployed. For each application scenario, we first briefly describe the characteristics followed by a discussion 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.

#### 3.1. 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).

#### 3.2. 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.

### 3.3. 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.

### 3.4. 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.

### 3.5. 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 all devices are 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.



### 3.6. 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 [I-D.ietf-eman-requirements] 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 3.11. The management needs for Smart Home and Building Automation are discussed in Section 3.4 and Section 3.5.

### 3.7. 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.

### 3.8. 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.

### 3.9. 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.

### 3.10. 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.

### 3.11. 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.

### 3.12. 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.

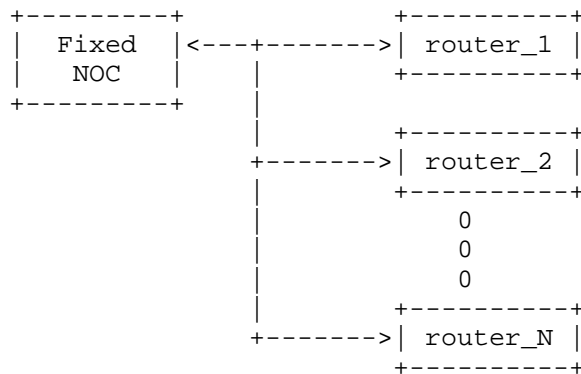


Figure 1: Parking Lot Staging Area

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 scenario, the same concept can be applied to mobile units.

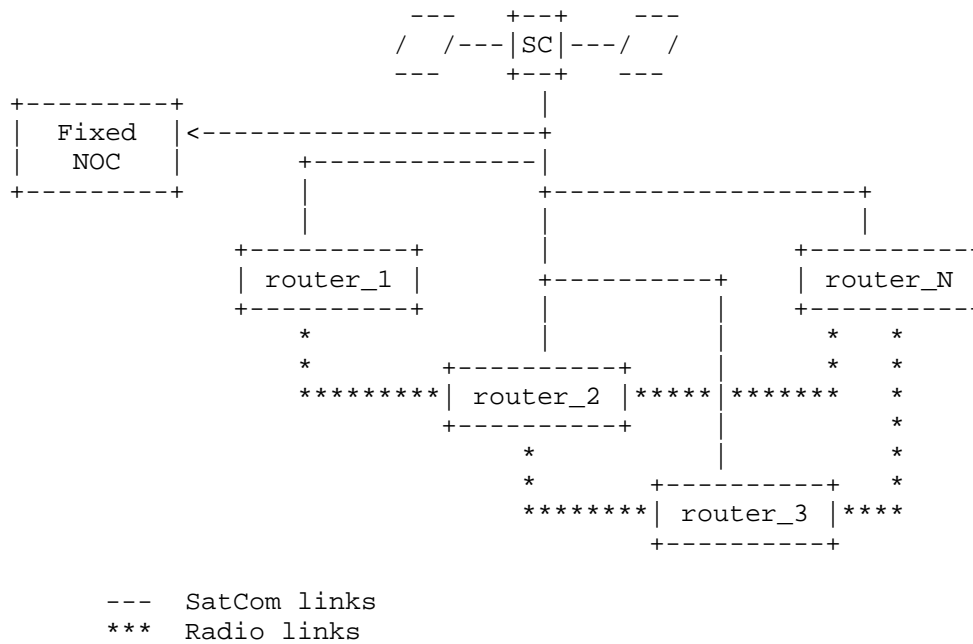
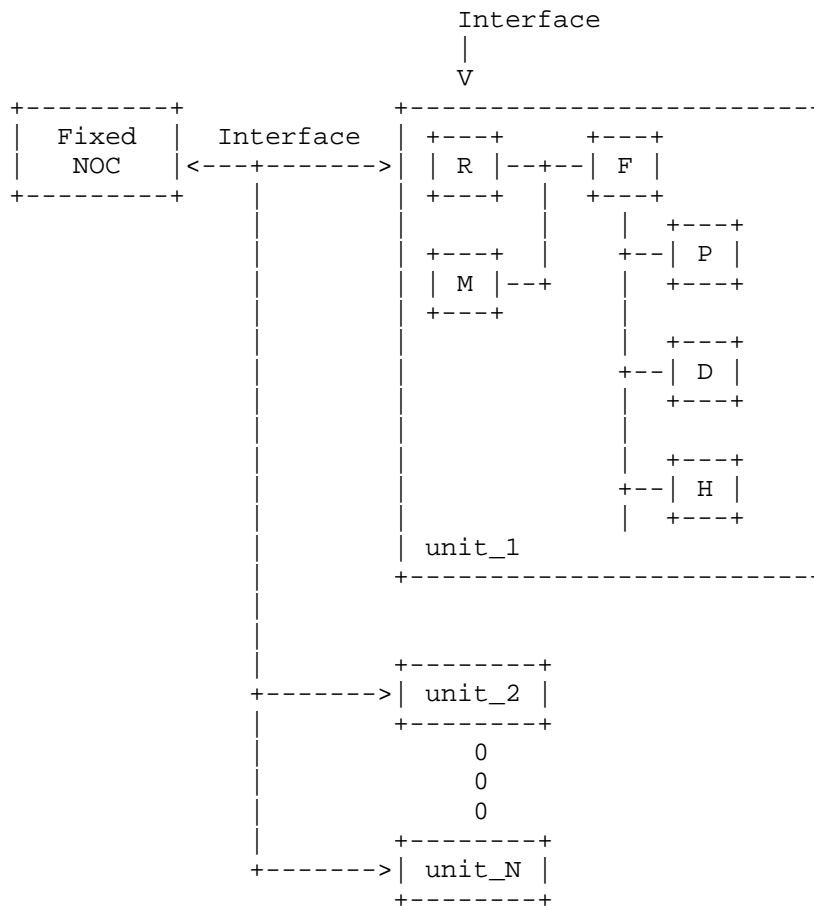


Figure 2: Monitoring with one-hop SatCom Reachback network

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.

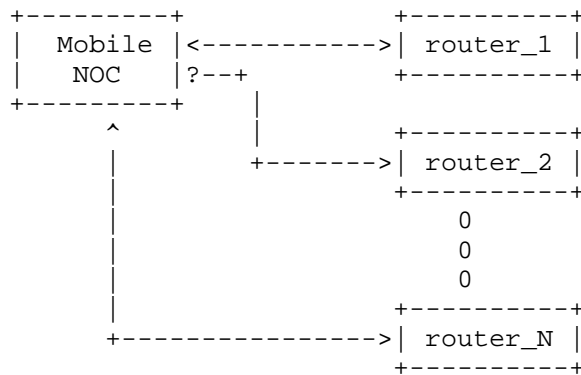


Figure 4: Management over Lossy/intermittent Links

#### 4. 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 selected requirements but might not be capable to provide all requirements at once. 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.

Requirement Type: Functional Requirement, Non-Functional Requirement, Design Constraint

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

Priority: The priority of the requirement showing the importance: Mandatory (M), Optional (O), Conditional (C).

##### 4.1. 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 [LWIG-TERMS] for the definition of Constrained Device Classes.

Source: All use cases.

Requirement Type: Non-Functional Requirement

Device type: Managing and intermediary entities.

Priority: Mandatory

---

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: Design Constraint

Device type: C0, C1, and C2

Priority: Mandatory

---

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: Optional

---

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: Non-Functional Requirement

Device type: C0, C1, and C2

Priority: Mandatory

---

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:    Mandatory

---

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:    Design Constraint

Device type:    C0, C1, and C2

Priority:    Mandatory

---

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: Optional

---

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: Optional

#### 4.2. 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: Mandatory

---

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: Mandatory

---

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: Optional

---

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: Optional

---

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: Mandatory

---

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:    Optional

---

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:    Mandatory

#### 4.3.    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:    Mandatory for C0 and C1, Optional 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:    Optional

---

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:    Conditional

---



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: Mandatory, Conditional if the network has a hierarchical topology.

#### 4.4. 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: Mandatory, Conditional 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: Mandatory for energy reporting devices, Optional for the rest

---

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: Optional

---

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: Optional

---

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: Mandatory for C2, Optional 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 TBD performance 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: Optional

---

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: Optional

---

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: Optional

---

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: Optional

---

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: Optional

---

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: Mandatory for C2, Optional 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: Mandatory for some medical or industrial applications, Optional otherwise

#### 4.5. 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: Optional

#### 4.6. 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: Mandatory, Optional 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:    Mandatory

---

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:    Mandatory

---

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: Mandatory, Optional for hardware-supported algorithms.

#### 4.7. 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: Mandatory for the use case Energy Management, Optional 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:    Functional Requirement

Device type:    C2

Priority:    Optional

---

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:    Functional Requirement

Device type:    C0, C1, and C2

Priority:    Optional

---

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:    Optional

#### 4.8.    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:    Functional Requirement

Device type:    C0, C1, and C2

Priority:    Optional

#### 4.9.    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:    Design Constraint

Device type:   C1 and C2

Priority:   Optional

---

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:   Design Constraint

Device type:   Intermediary entity in the network.

Priority:   Optional

---

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:   Design Constraint

Device type:   Intermediary entity in the network.

Priority:   Optional

#### 4.10.   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: Design Constraint

Device type: C0, C1 and C2

Priority: Conditional, in case such scalability is a prerequisite.

---

Req-ID: 4.10.002

Title: Reliable unicast transport.

Description: Provide reliable unicast transport of messages.

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

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: Mandatory

---

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: Optional

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: Mandatory

#### 4.11. 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:    Design Constraint

Device type:    C0, C1, and C2

Priority:    Mandatory

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:    Design Constraint

Device type:    C0, C1, and C2

Priority:    Mandatory

## 5. IANA Considerations

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

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



## 6. Security Considerations

This document discusses the use cases 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.

## 7. Contributors

Following persons made significant contributions to and reviewed this document:

- o Ulrich Herberg (Fujitsu Laboratories of America) contributed the Section 3.9 on Community Network Applications and to the Section 1.3 on Class of Networks in Focus.
- o Peter van der Stok contributed to Section 3.5 on Building Automation.
- o Zhen Cao contributed to Section 3.10 on Mobile Applications.
- o Gilman Tolle contributed the Section 3.11 on Automated Metering Infrastructure.
- o James Nguyen and Ulrich Herberg contributed the Section 3.12 on MANET Concept of Operations (CONOPS) in Military.

## 8. Acknowledgments

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

## 9. References

### 9.1. Normative References

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

### 9.2. Informative References

- [RFC6632] Ersue, M. and B. Claise, "An Overview of the IETF Network Management Standards", RFC 6632, June 2012.
- [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.
- [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-olsrv2-17 (work in progress), October 2012.
- [I-D.ietf-lwig-guidance] Bormann, C., "Guidance for Light-Weight Implementations of the Internet Protocol Suite", draft-ietf-lwig-guidance-02 (work in progress), August 2012.
- [I-D.ietf-core-coap] Shelby, Z., Hartke, K., Bormann, C., and B. Frank, "Constrained Application Protocol (CoAP)", draft-ietf-core-coap-13 (work in progress), December 2012.
- [I-D.ietf-eman-framework] Claise, B., Parello, J., Schoening, B., Quittek, J., and B. Nordman, "Energy Management Framework", draft-ietf-eman-framework-06 (work in progress), October 2012.
- [I-D.ietf-eman-requirements] Quittek, J., Chandramouli, M., Winter, R., Dietz, T., and B. Claise, "Requirements for Energy Management", draft-ietf-eman-requirements-11 (work in progress), January 2013.

[I-D.ietf-roll-terminology]

Vasseur, J., "Terminology in Low power And Lossy Networks", draft-ietf-roll-terminology-10 (work in progress), January 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/>>.

[LWIG-TERMS]

Bormann, C., "Terminology for Constrained Node Networks", draft-bormann-lwig-terms (work in progress), November 2012.

## 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 discussion and consolidation.

## Appendix D.   Change Log

### D.1.   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.2.    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 4 need to be seen as standalone requirements and the current document does not recommend any profile of requirements.
- o    Added Section 4 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.3.    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 Siemens Networks

Email: mehmet.ersue@nsn.com

Dan Romascanu (editor)  
Avaya

Email: dromasca@avaya.com

Juergen Schoenwaelder (editor)  
Jacobs University Bremen

Email: j.schoenwaelder@jacobs-university.de





Operations and Management Area Working Group  
Internet-Draft  
Intended status: Informational  
Expires: January 16, 2014

P. Fan  
L. Li  
China Mobile  
July 15, 2013

Requirements for IP/MPLS network transmission interruption duration  
draft-fan-opsawg-transmission-interruption-03

## Abstract

The transmission performance of IP/MPLS network affects upper layer services and networks, but there is no consensus in the industry on transmission interruption for IP/MPLS network up to now. This memo studies requirements for the interruption duration criteria in several service scenarios.

## 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 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 . . . . .	2
2. Services and Performance Criteria . . . . .	3
2.1. Softswitch . . . . .	3
2.2. SS7 transport . . . . .	5
2.3. LTE Backhaul . . . . .	6
2.4. Ethernet VPN . . . . .	6
2.5. IPTV . . . . .	7
3. Other considerations . . . . .	7
4. Security Considerations . . . . .	7
5. IANA Considerations . . . . .	7
6. Appendix: Impact Analysis on Transmission Quality of IP Carried Softswitch Voice . . . . .	7
7. Acknowledgements . . . . .	10
8. Informative References . . . . .	10
Authors' Addresses . . . . .	11

## 1. Introduction

Today's IP/MPLS network is widely used as a bearer network to carry diversified packet switched services. The transmission qualities of these services are closely related to the performance of bearer layers, as network failure, delay, congestion and other abnormalities will inevitably bring about service interruption and user perception degradation. However, there is no consensus in the industry on transmission interruption for IP/MPLS network up to now. This memo studies relationships between service performance and transmission interruption duration in several scenarios, and is intended to reach a list of requirements for these interruption duration criteria.

For a long time the industry has been aspiring for the so-called golden standard for network resilience, that is the 50-millisecond recovery threshold. [HeavyReading] gives us a basic introduction to the origin of this fast protection legacy which can date back to 1980s. The 50ms threshold was established informally in the early 1980s, and then formally through standardization of [G.841] recommendation on SDH network protection architects. The specific requirement shows a maximum threshold for detecting and restoring a fault of 60ms, which adds up fault detection duration of less than 10ms and protection switching time of less than 50ms. The report also mentions original concerns that the threshold results from. The voice channel banks deployed in early 1980s had limited fault tolerance. Failures that lasted longer than 200ms would generate a Carrier Group Alarm (CGA) which caused the channel bank to terminate all connections over that given TDM line. So an outage budget was developed by carriers and the 50ms standard was employed to protect voice services. However newer channel banks at that time had started

to implement a CGA timer of 2s, so the 50ms protection was adopted to protect a small and diminishing fraction of digital network.

Historically this 50ms fast protection speed has been achieved by SDH network. Using various fast convergence technics, IP/MPLS is also able to react within 50ms. As for network applications that are carried by optical or packet core, changes have been made through the past decades, accompanied by the continuing questions about needs for 50ms protection. Here we list three basic considerations about services and their requirement for IP/MPLS: for services like TDM over IP/MPLS, the traditional 50ms guarantee should be kept and met; for current IP services (e.g. voice, internet), experiences or experiments are to be provided for guidance; for services in future, we are supposed to propose requirement early and give consideration to IP/MPLS.

## 2. Services and Performance Criteria

Services delivered by IP/MPLS network have different transmission quality requirements, thus introduce different performance criteria for the bearing IP/MPLS network. We believe there are two principles that need to be considered during network and service design, configuration and operation. The IP/MPLS bearer should satisfy quality requirements of upper level services and applications, while services and applications should also take into account the intrinsic IP capabilities. In this section we will describe concerns on IP/MPLS and service mutual adaptation from aspects of several kinds of service scenarios.

### 2.1. Softswitch

From the softswitch point of view, the IP carrying nature imposes certain influence to the service quality. Especially when speech is delivered by IP, the communication quality of voice is impaired, and in turn makes higher requirements for the transmission performance of IP. The following table gives a list of criteria regarding transmission quality of a typical GSM network as well as impacting factors brought by IP bearer.

Criteria of GSM Transmission Quality		Impacting Factors Brought by IP Bearer
Call Loss	Call loss of wireless channel	None
	Call loss between switches (typical value: <=1%)	Failure of Nc/Mc interface carried by IP

	Call loss between switch and BSC (typical value: $\leq 0.5\%$ )	None
Call Cut-off	Call cut-off rate (typical value: $< 1\%$ )	Failure of Nc/Mc interface carried by IP
	Service providing delay	None
Connection Delay	Calling party connection delay (typical value: $\leq 4s$ )	IP carried signaling delay
	Called party connection delay (typical value: $\leq 4s$ )	None

If voice is carried by IP, communication quality criteria of call loss, call cut-off and connection delay are likely to be influenced. This subsection focuses on the three criteria and their impacting factors to give requirements for softswitch and IP bearer networks, with detailed analysis described in the appendix. Note that the current discussion on softswitch is focused on quality of transmission while not on quality of voice. In another word, the scope of discussion is limited to network related QoS aspect, while subjective QoE criteria such as PESQ (Perceptual Evaluation of Speech Quality) and MOS (Mean Opinion Score) are left to later revisions.

Call loss related requirement: The duration of SCTP interface association timer should be shorter than that of the state machine message timer of upper layer protocols, and this duration is further recommended to be no longer than 6 seconds in order to maintain detection sensitivity; the interruption duration of IP bearer network should be as short as possible to avoid call loss, and this duration is further recommended to be no longer than 5 seconds.

Call cut-off related requirement: The SCTP association should be guaranteed during IP layer interruption to avoid interface breakoff alert. The requirements are the same as those related to call loss.

Connection delay related requirement: The IP convergence time should be no longer than 3 seconds to ensure that connection delay is shorter than 4 seconds.

The overall requirement for IP/MPLS interruption duration is no longer than 3 seconds.

## 2.2. SS7 transport

The Signaling System No. 7 (SS7/C7) network is one of the examples of the principle that services should take into account the ability of IP. The bearer of SS7 protocol stack has been experiencing evolution from TDM to IP. Traditionally the user parts of SS7 (including MAP, CAP, BSSAP+, ISUP, etc.) are carried by MTP layers, but the bearer has gradually been evolved into a packetized form with SIGTRAN (including M2PA, M2UA, M3UA, etc.) using SCTP associations over IP. The change requires transport layer to take mechanisms to meet demand of SCN signaling, and more importantly it requires protocols to make adaption to the "best effort" fact of IP.

The SIGTRAN uses an architecture that can be described as standard IP plus unified transport plus diversified adaption units. It introduces SCTP to realize reliable signaling transport over IP. The SCTP itself provides reliable transmission mechanisms, such as path selection and monitoring, validation and acknowledgment mechanisms, and retransmission timing management.

The unreliable nature of IP makes it necessary for the upper-level protocols to be more tolerable to the possible instability of bearer. Once a service request from a UE is accepted, the system allocates resources and establishes paths for the user. A breakoff caused by IP will result in signaling disconnection or rerouting. Signaling transmission path may also be switched back after IP layer restores. Frequent switchovers and disconnections lead to unnecessary system cost and service interruption, so parameters should be configured a little bit "insensitive" to try to sustain connections on control plane.

One of the examples of parameter configuration is the timer value. The following gives two cases about SCTP on transport layer and M2PA on adaption layer. The values should not be set very small to prevent unnecessary disconnection caused by IP instability. However, because upper services of SS7 may also have timeout rules, values should not be set very large too to avoid violating the rules.

### 1) SCTP

SCTP uses RTO to manage timeout duration for retransmission in case of feedback missing. The RTO is given an initial, a max and a min value, and is calculated instantaneously with a set of management rules. Many other parameters are used for fault detection in SCTP. Association.Max.Retrans is used to indicate the upper limit of number of possible retransmission without considering endpoint down. Path.Max.Retrans is a similar value to detect path failure. The parameters together characterize the ability of SCTP to tolerate

bearer downwards and provide reliable SS7 transport upwards. The typical values of the parameters are RTO.Initial = 0.5 sec, RTO.MIN = 0.5 sec, RTO.MAX = 1.5 sec, Path.Max.Retrans = 5, Assoc.Max.Retrans = 10.

## 2) M2PA

Although protocols like H.248 and BICC can be carried directly upon SCTP, the user part protocols of SS7 usually have to be carried by SCTP/IP with the help of different adaption layers. In this case, the attributes of adaption layers, e.g. M2PA used between STPs, are more important to SS7. M2PA uses a T7 timer to indicate the maximum delay of acknowledgement and start T7 at the time of data transmission. If no message is acknowledged after the maximum waiting time, T7 expires and M2PA sends a message of out of service to the peer end. Because propagation delays in IP networks are more variable than in traditional SS7 networks, the value of T7 should be set considering IP propagation delays, as well as acknowledgement time, SCTP slow-start algorithms, upper service timers and other factors. Typical value of T7 is 7~10 sec.

Parameter configuration induced tolerance to bearer may have some influence on service, but it avoids service cut-off or severe user perception degradation. For services like SMS or route lookup, possible latency may be introduced, but operations can still be completed after short delay. Because SMS has no strict requirement for instantaneity, impact on service is limited. If route lookup takes more time due to IP interruption and convergence, user may experience longer setup delay when dialing. For service of location update, even if operation fails because bearer is interrupted for too long, UE has the mechanism to initiate request again.

## 2.3. LTE Backhaul

To be further analyzed.

## 2.4. Ethernet VPN

Ethernet VPNs (e.g. VPLS) are used to provide transparent Ethernet type layer 2 connections for customers. Ethernet frames are treated as service payload and encapsulated and transported in providers MPLS network. The interruption criteria of IP/MPLS bearer should guarantee continuity of Ethernet service, and IP/MPLS failover is not supposed to generate outage of Ethernet service.

[Y.1731] and [IEEE802.lag] describe in detail OAM functions and mechanisms for Ethernet, with specific recommendation on connectivity fault management. Ethernet uses continuity check function to detect

loss of continuity between any pair of MEPs in a MEG, and this function is realized by sending CCMs (connectivity check messages) between peer MEPs. When a MEP does not receive CCM from a peer MEP within a certain interval, it detects loss of continuity to that peer MEP. The threshold interval is specified as 3.5 times the CCM transmission period, which corresponds to a loss of three consecutive CCMs from the peer MEP, and the CCM transmission period is recommended to be the default value of 1 second. So the interruption duration of IP/MPLS for Ethernet VPN services should be less than 3 seconds.

## 2.5. IPTV

To be further analyzed.

## 3. Other considerations

So far this document has focused on use cases and their requirement for IP/MPLS, and other practical issues are not included in this version. For example, an IP/MPLS packet core is expected to carry a variety of services, so the requirement for IP/MPLS may have to include additional concerns on this multi-service co-existence scenario. A simple and straight-forward way may be to satisfy the most critical need for protection time required by the services. Another issue is related to service awareness. Whether service type is or can be known by IP/MPLS would influence the ability of IP/MPLS to provide reliability guarantee accordingly. It seems to be easier to perform service identification on edge devices than network core. We believe these kinds of issues need to be taken into account, and currently we will just leave them to be updated in future revisions.

## 4. Security Considerations

TBD

## 5. IANA Considerations

This memo includes no request to IANA.

## 6. Appendix: Impact Analysis on Transmission Quality of IP Carried Softswitch Voice

This section describes impact on transmission quality of softswitch voice when carried by IP and requirements for IP bearer convergence time.

### 1) Call Loss

Call loss is used to describe the circumstance where a phone call fails to establish after initiated by a subscriber due to network faults. In the practical network, the call loss rate is mainly associated by the factors as follows:

1. Interfaces, including Nc, Mc and interface between MSS and SG.
2. State machine message timer. If a timeout takes place, the state machine releases signaling messages, producing a call loss. Typical value of BICC timer is 10~15 seconds and value of DTAP timer about 15 seconds.
3. Interface association timer. Associations breaks off at the expiration of timer.
4. Bearer network convergence time.

If the configured timer duration of a state machine is shorter than the timer duration of interface association, then although interface association may not be broken off, call loss is still possible to occur due to message timer expiration. If the association timer duration is shorter than IP routing convergence time, the association is considered broken off by SCTP, hence message loss at interface between MSS and SG as well as interface Nc results in massive call loss, and new calling request cannot be satisfied because of interface Mc breakoff. In this case, the call loss rate can be calculated as

$$\text{Call Loss Rate} = ( \text{IP Convergence Time} + \text{Association Restoration Time} ) * \text{CAPS} / \text{BHCA}.$$

However, if the association timer duration is longer than IP routing convergence time, then the association is considered normal by SCTP, and data will be retransmitted. Although this may cause buffer overflow leading to call loss, the call loss rate is possible to achieve approximately zero if buffer is big enough.

From the analysis above and practical operation experience, the requirements for softswitch and IP bearer are as follows: the duration of SCTP interface association timer should be shorter than that of the state machine message timer, and this duration is further recommended to be no longer than 6 seconds in order to maintain detection sensitivity; the interruption duration of IP bearer network should be as short as possible to avoid call loss during the IP layer interruption period, and this duration is further recommended to be no longer than 5 seconds.



## 2) Call Cut-off

Call cut-off is referred to the abnormal release during a phone call due to reasons other than intentional release by any of the parties involved in the call. The call cut-off rate is related with:

1. Interfaces, including Nc and interface between MSS and SG.
2. Interface association timer.
3. Bearer network convergence time.

If the association timer duration is shorter than IP routing convergence time, established phone calls will be released once interruption of interface Nc or interface connecting MSS and SG is detected. In the case of association breakoff, call cut-off rate can be calculated as

$$\text{Call Cut-off Rate} = ( \text{CAPS} * \text{Call Duration} ) * \text{Busy Hour Association Breakoffs} / \text{BHCA}.$$

While if the association is not interrupted, the call cut-off rate can be approximately zero.

In conclusion, the SCTP association should be guaranteed during IP layer interruption to avoid interface breakoff alert. The requirements for softswitch and IP bearer are the same as those related to call loss.

## 3) Connection Delay

The connection delay from a call initiation by a calling party to PLMN should be no longer than 4 seconds. This delay is affected by factors below:

1. RRC connection setup delay (irrelevant to whether service is carried by IP or not).
2. Core network signaling interaction delay. The message number at interface Nc/Nb is 6, and is 8 (calling side) or 16 (called side, in case of IP-IP) at interface Mc. Each message is with a delay of no longer than 50 milliseconds. Calling message delay at interface Nc is no longer than 300 milliseconds. If long distance call is made through CMN, the message delay is to be increased by transmission delay of 5 msec/km and CMN process delay. So the message delay is likely to be 400 milliseconds.

### 3. IP bearer network QoS and load.

The connection delay is influenced by the delay criterion defined in the IP bearer network QoS, and is raised by delay, jitter, packet loss caused by network overload. In addition, if the configured timer duration of interface association is too long, the SCTP sensitivity to the retransmitted messages after packet loss will be decreased, which increases connection delay.

Connection delay is generally expressed as

$$\text{Connection Delay} = \text{IP convergence time} + \text{RRC connection setup delay} \\ + \text{Signaling Interaction Delay},$$

and is no longer than 4 seconds. So the IP network in normal working state should be constrained within a certain range of load to ensure that delay is shorter than 50 milliseconds, while in interruption state the IP convergence time should be no longer than 3 seconds to ensure that connection delay is shorter than 4 seconds.

From the analysis of IP/MPLS performance according to the three criteria above, we suggest the transmission interruption duration of IP/MPLS network for softswitch service should be no longer than 3 seconds.

### 7. Acknowledgements

The authors would like to thank Chris Donley and Melinda Shore for their kind help in content enrichment, and Christopher Liljenstolpe, Andrew Malis and Adrian Farrel for their helpful comments on the document.

### 8. Informative References

[G.841] ITU-T Recommendation G.841, ., "Types and characteristics of SDH network protection architectures", October 1998.

[HeavyReading]

Bennett, G., "Resilience Reliability and OAM in Converged Network", Heavy Reading, Vol. 2, No. 6, February 2004.

[IEEE802.1ag]

IEEE Std 802.1ag-2007, ., "IEEE Standard for Local and metropolitan area networks, Virtual Bridged Local Area Networks, Amendment 5: Connectivity Fault Management", December 2007.

[Y.1731] ITU-T Recommendation Y.1731, ., "OAM Functions and Mechanisms for Ethernet based Networks", July 2011.

Authors' Addresses

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

Email: fanpeng@chinamobile.com

Lianyuan Li  
China Mobile  
32 Xuanwumen West Street, Xicheng District  
Beijing 100053  
P.R. China

Email: lilianyuan@chinamobile.com

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

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

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

Abstract

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

Status of this Memo

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

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

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

This Internet-Draft will expire on April 21, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

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

## Table of Contents

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

## 1. Introduction

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

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

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

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

Modern firewalls apply perimeters at three layers:

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

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

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

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

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

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

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

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

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

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

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

### 1.2. Terminology

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

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

## 2. High-Level Firewall Concepts

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

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

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

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

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

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

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

## 2.2. Building a Communication

Any communication requires at least three components:



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

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

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

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

### 3. Firewalling Strategies

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

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

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

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

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

### 3.2. Typical Firewall Categories

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

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

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

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

Firewalls that understand IPv6 may have a fourth category:

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

others ]]]]

### 3.3. Newer categories of firewalling

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

## 4. Recommendations for Operators

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

Firewalling strategies

None. This is really the operator's choice.

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

Don't enforce protocol semantics in the firewall

Applications are easier to change than firewalls

Avoid using application-layer gateways for firewalling

Use the security in the applications servers instead

Servers are easier to change than firewalls

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

Allow fragments

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

Document your intended firewall strategy and settings

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

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

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

## 5. Recommendations for Firewall Vendors

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

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

Interface for pinholing of IPv4 NATs needs clearly identify security issues

Follow the BEHAVE RFC rules for binding timeouts on NATs

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

Make leaving notes about the firewalling rules easy and useful

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

## 6. IANA Considerations

None.

## 7. Security Considerations

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

## 8. Acknowledgements

Warren Kumari commented on this document.

## 9. References

### 9.1. Normative References

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

### 9.2. Informative References

[EndToEnd]

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

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

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

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

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

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

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

## Appendix A. IPv4 NATs Are Not Security Devices

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

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

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

Recommendations for deploying NATs in firewalls include:

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

## Appendix B. Origin Reputation and Firewalls

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

Letting someone else curate your security policy  
Different types of reputation for different layers  
draft-ietf-repute-model  
draft-vyncke-advanced-ipv6-security  
draft-hallambaker-omnibroker  
Recommendations  
    Check logs to be sure updates are happening  
    Check vendors' policies

## Authors' Addresses

Fred Baker  
Cisco Systems

Email: fred@cisco.com

Paul Hoffman  
VPN Consortium

Email: paul.hoffman@vpnc.org



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

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

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

Abstract

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

Status of This Memo

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

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

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

This Internet-Draft will expire on October 15, 2014.

## Copyright Notice

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

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

## Table of Contents

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



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

## 1. Introduction

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

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

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

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

### 1.1. Terms

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

CGN - Carrier Grade NAT

DOCSIS - Data Over Cable Service Interface Specification

CMTS - Cable Modem Termination System

DSL -Digital subscriber line

BRAS - Broadband Remote Access Server

GGSN - Gateway GPRS Support Node

GPRS - General Packet Radio Service

ASN-GW - Access Service Network Gateway

GRT - Global Routing Table

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

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

## 2. Existing Network Considerations

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

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

## 3. CGN Network Deployment Requirements

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

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

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

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

### 3.1. Centralized versus Distributed Deployment

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

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

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

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

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

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

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

### 3.3. CGN By-Pass

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

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

### 3.4. Routing Plane Separation

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

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

### 3.5. Flexible Deployment Options

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

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

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

### 3.6. IPv4 Overlap Space

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

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

### 3.7. Transactional Logging for CGN Systems

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

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

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

### 3.8. Base CGN Requirements

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

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

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

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

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

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

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

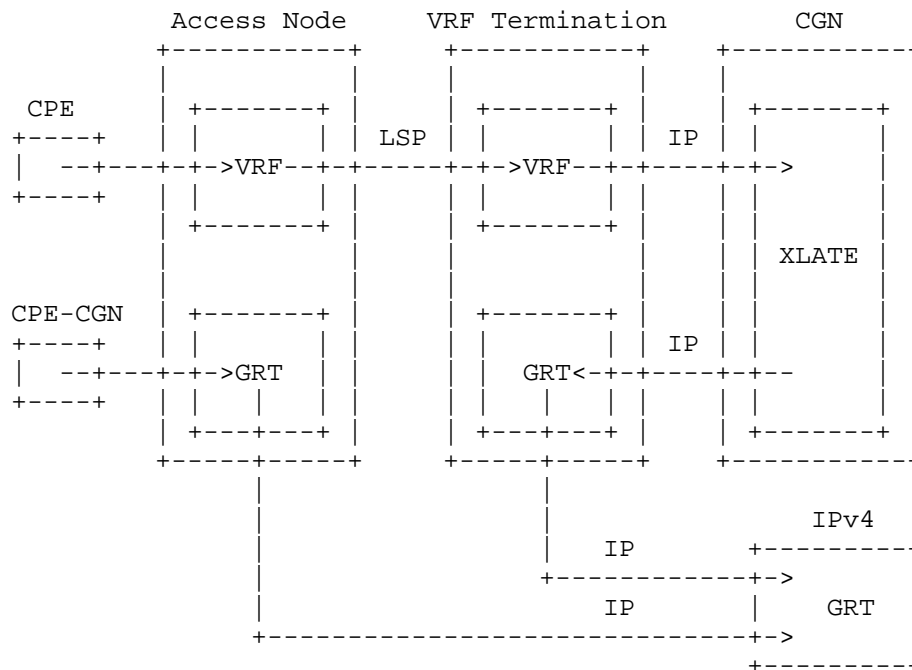


Figure 1: Basic BGP/MPLS IP VPN CGN Model

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

#### 4.1. Service Separation

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



#### 4.2. Internal Service Delivery

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

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

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

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

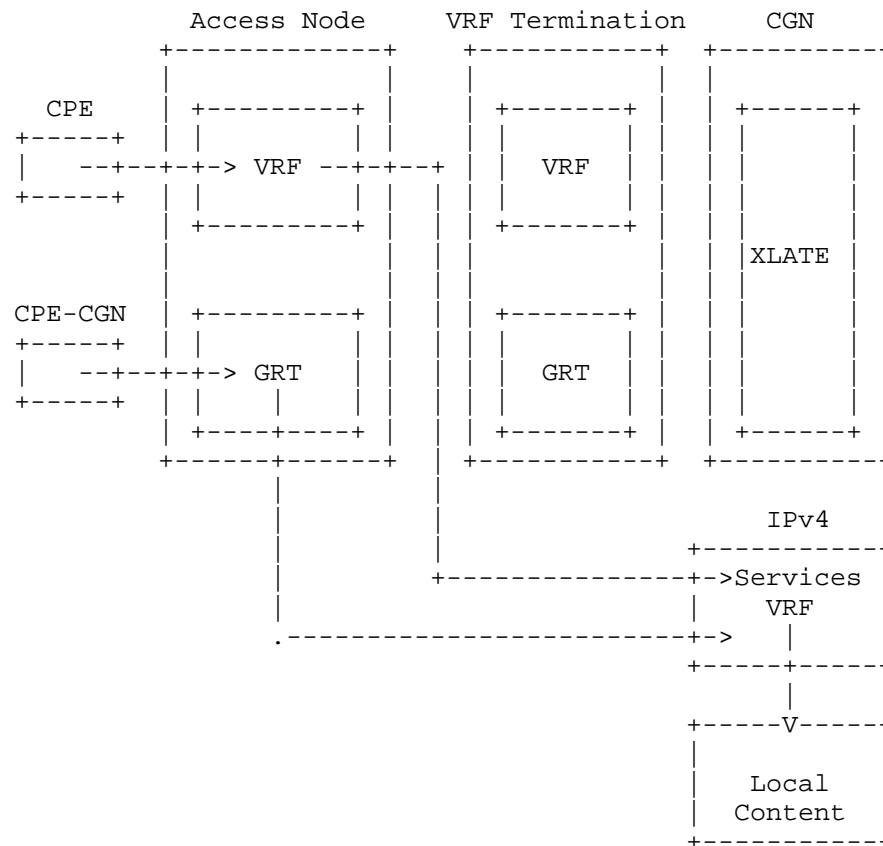
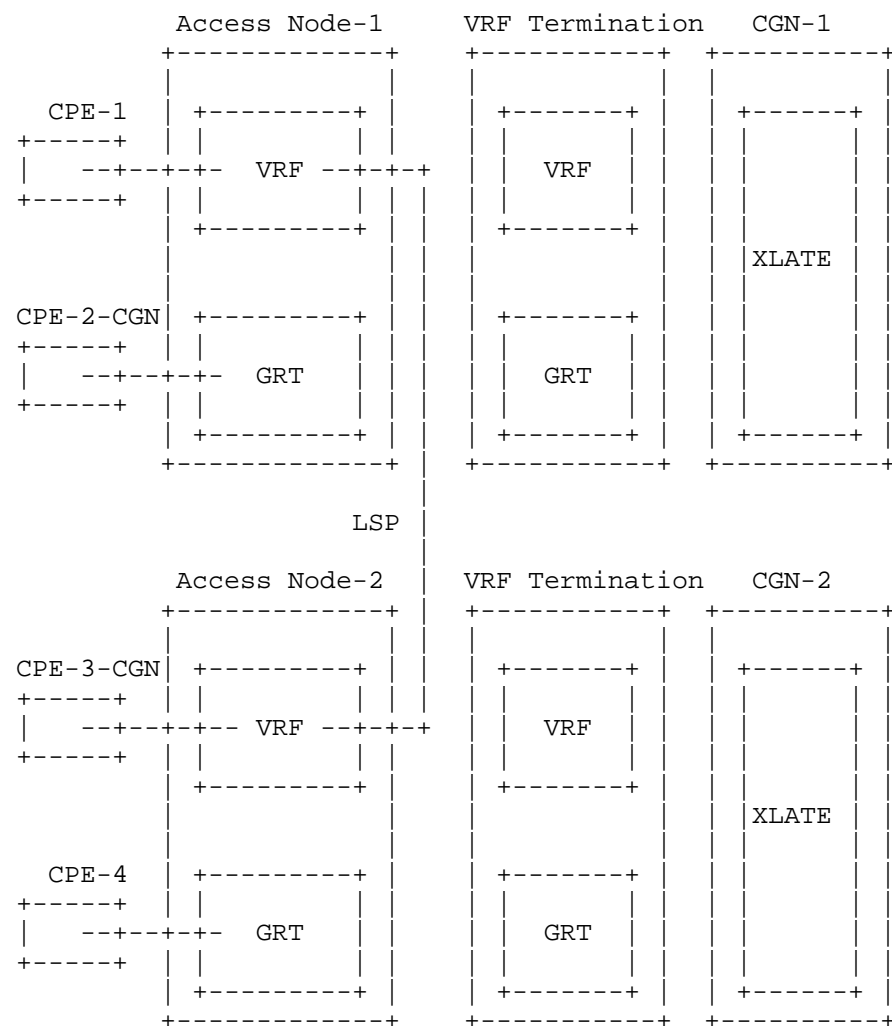


Figure 2: Internal Services and CGN By-Pass

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



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

#### 4.2.1. Dual Stack Operation

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

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

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

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

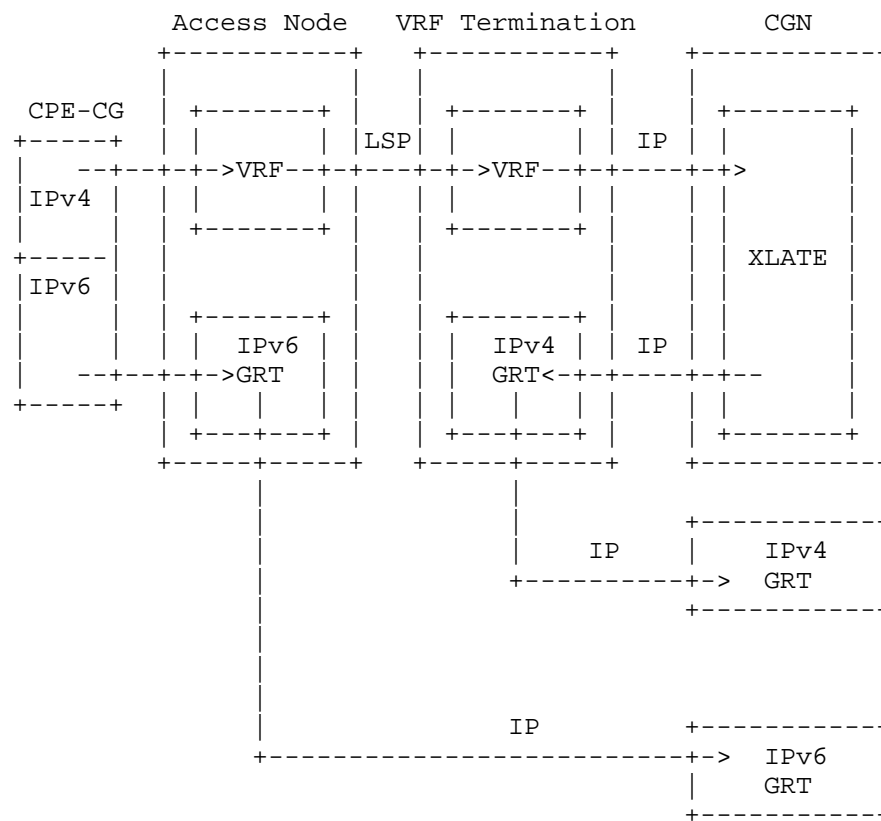


Figure 4: CGN with IPv6 Dual Stack Operation

#### 4.3. Deployment Flexibility

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

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

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

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

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

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

##### 4.4.1. Policy Based Routing

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

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

#### 4.4.2. Traffic Engineering

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

#### 4.4.3. Multiple Routing Topologies

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

#### 4.5. Multicast Considerations

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

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

### 5. Experiences

#### 5.1. Basic Integration and Requirements Support

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

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

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

## 5.2. Performance

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

## 6. IANA Considerations

This document has no IANA actions.

## 7. Security Considerations

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

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

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

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

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

## 9. Acknowledgements

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

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

## 10. References

### 10.1. Normative References

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

### 10.2. Informative References

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



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

#### Authors' Addresses

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

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

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

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

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.

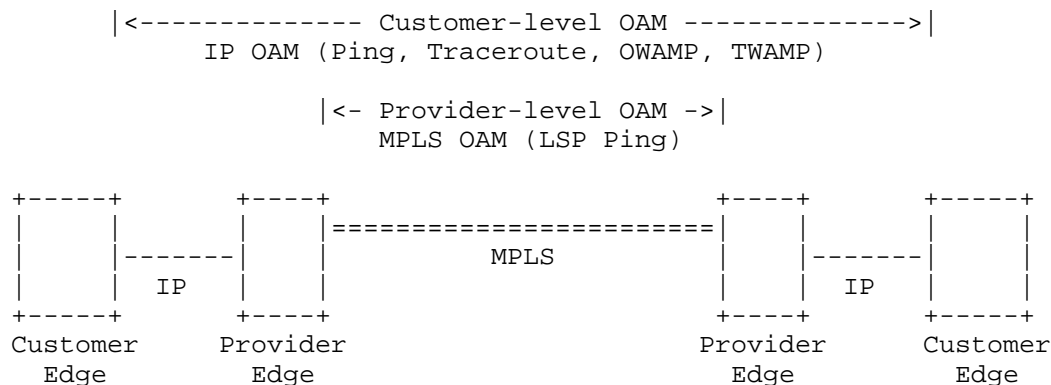


Figure 1 Example: Multi-layer OAM

## 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/proactive 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	

	TRAMP	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

	ICMP Extensions for Multiprotocol Label Switching [ICMP-Ext]	RFC 4950
	Bidirectional Forwarding Detection for MPLS Label Switched Paths (LSPs) [BFD-LSP]	RFC 5884
MPLS-TP OAM	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)



OPSAWG Working Group  
INTERNET-DRAFT  
Intended Status: Proposed Standard  
Expires: November 2012

Janardhanan Pathangi  
Balaji Venkat Venkataswami  
DELL  
Richard Groves  
MICROSOFT  
Peter Hoose  
FACEBOOK  
May 8, 2012

Requirements for OAM tools that enable flow Analysis  
draft-janapath-opsawg-flowoam-req-00

Abstract

This document specifies Operations and Management (OAM) requirements that improve on the traditional OAM tools like Ping and Traceroute. These requirements have arisen from the fact that more details than given by Ping and Traceroute are required while troubleshooting or doing performance and network planning. These requirements have been gathered from network operators especially from data centers where the networks have slightly different characteristics compared to regular campus/carrier networks.

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/lid-abstracts.html>

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

Copyright and License Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

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

## Table of Contents

1	Introduction	4
1.1	Terminology	4
1.2	Acronyms	4
2	Requirements	5
2.1	Flow tracing	5
2.2	Fate sharing and actual flow interference	5
2.2.1	Side effects of requirements 2.1 and 2.2	5
2.3	Capability to send the response to a monitoring station	6
2.4	Terminating the trace on a transit device	6
2.5	Flow monitoring	6
2.5.1	Parameters for monitoring	6
2.5.1.1	Time-based and Monitor-till-stop monitoring	6
2.6	Loop Detection	7
2.7	Additional Information	7
2.7.1	Link statistics	7
2.7.2	Packet drops and their reasons	7
2.8	Additional enhancements	7
2.8.1	Fat-Tree traversal	7
2.8.2	Hash Algorithm Parameters	8
2.9	Future Requirements	8
3	Security Considerations	8
3.1	Securing Requests and Responses	8
3.2	Information hiding	8
3.3	Rate limiting obviating attack vectors	9
4	IANA Considerations	9
4.1	Acknowledgements	9
5	References	10
5.1	Normative References	10
5.2	Informative References	10
	Authors' Addresses	10





## 1 Introduction

Network operators have traditionally managed IP networks with classic OAM tools like Ping and Traceroute. Operators typically use Ping to perform end-2-end connectivity checks, and Traceroute to trace hop-by-hop path to a given destination. Traceroute is also used to isolate the point of failure along the path to a given destination. Also, while these are useful for basic connectivity checks, they are unable to provide sufficient information about the performance aspects of a path (e.g. utilization levels).

In current networks especially data center networks, there are a large number of redundant paths and existing OAM tools are unable to identify flow specific problems and also do not provide sufficient information on the various paths which includes performance characteristics. What is needed is a set of tools that will perform the OAM functions based on header fields of actual user traffic.

### 1.1 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 RFC 2119 [RFC2119].

### 1.2 Acronyms

OAM	Operations and Management
ECMP	Equal Cost Multi-Path
LAG	Link Aggregation Group
TRILL	Transparent Connection of Lots of Links
SPB	Shortest Path Bridging
L2	Layer 2
IP	Internet Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
IDS	Intrusion Detection System
IPS	Intrusion Prevention System
ACL	Access Control List
ARP	Address Resolution Protocol
ICMP	Internet Control Message Protocol

## 2. Requirements

The main requirement is being able to trace the exact path that a particular flow will take through the network and obtain all relevant information about the links along that path which provides the network operator with enough information to troubleshoot a network failure or quickly obtain performance data about that path.

Increasing number of networks are using multi-path configurations to improve load-balancing and redundancy in their networks. These multi-paths could be in the form of ECMP paths offered by the routing protocols, but also Layer 2 ECMP paths (e.g. via TRILL [RFC 5556] or SPB [IEEE SPB]) and LAGs between adjacent routers.

### 2.1 Flow tracing

Ideally, the OAM trace packet should undergo the same processing at each node as would the actual application flow that is being traced. The forwarding process in switches and routers typically includes fields from the L2, IP, and transport (TCP/UDP) headers, to determine which member of an ECMP or LAG to forward the packet on. The OAM packets should also contain the flow entropy allowing for the same processing that a typical data packet would go through.

Current tools like ping and Traceroute do not carry the application flow information, and hence the path that those packets follow through the network could differ from the packets of the specific flow that is of interest.

### 2.2 Fate sharing and actual flow interference

OAM probes while sharing fate with the actual flow, should not affect the real application in progress at the time of troubleshooting. The OAM request originating from the sender should not interfere with the actual application at the target host Likewise, the OAM response should not go back to the real application at the originator of the OAM query.

#### 2.2.1 Side effects of requirements 2.1 and 2.2

To ensure that OAM packets share the same fate as that of the Application's packets, and yet do not get delivered to the application, it would be necessary to have an indication in the packet to distinguish OAM packets from regular application flow packets. The inclusion of such an indication in the packet should still result in the formation of a legitimate packet, and should not trigger security based drops or alarms at intermediate firewalls and IDS/IPS appliances, due to, say, an incorrect checksum or invalid

fragment headers, that regular data packets would not normally experience.

It would be useful for the operator to control which class of service is used by an OAM packet. For example, when measuring one way or round trip delays, it would be useful to send it in the same class of service as regular data.

### 2.3 Capability to send the response to a monitoring station

When tracing the flow from node A to node B, it should be possible to direct all the response packets to a third node C, which could be a management station.

### 2.4 Terminating the trace on a transit device

The tool should have the capability to terminate the trace at a specific hop specified by an IP address or by specifying a limit on the number of hops. This helps in segmented tracing, where portions of the path can be traced.

### 2.5 Flow monitoring

It should be possible to initiate flow monitoring on one or all of the intermediate devices, and should have the following capabilities.

#### 2.5.1 Parameters for monitoring

The tool should provide an extensible mechanism by which the monitoring station can ask for monitoring of certain parameters for the flow like input rate, packet drops, etc at a given network node. It should also be possible to request for packet samples for external monitoring tool to calculate statistics on the flow or interfaces

The requested device may honor the monitoring request based on its policy, authentication of the requester and also the available resources on the device. It should be able to indicate back in the response if and what parts of the monitoring are activated.

The period for which this monitoring is activated could be...

##### 2.5.1.1 Time-based and Monitor-till-stop monitoring

The OAM packet carries a time period and frequency of sampling, and the requested devices send the samples at the specified frequency for the specified time period. This could also be overridden by local policy. In case of the Monitor-till-stop monitoring the OAM packet will initiate the monitoring at a specific sampling rate. The

monitoring will continue till there is a new request for turning off the monitoring. A local policy can also override this behavior and restrict this to a maximum period that is locally defined.

## 2.6 Loop Detection

The tool should be capable of detecting that OAM packets are being looped. If this happens the operation should be aborted. Appropriate heuristics may be considered while implementing this feature.

## 2.7 Additional Information

Apart from reporting the incoming and outgoing interfaces, it would be useful for the tool to report on the following

### 2.7.1 Link statistics

There is a necessity to collect useful information to enable operators to perform more detailed problem analysis or network optimization. The operator may need to know the utilization of the links along the path in addition to the fan-out information. This information could be for example be used by servers to select source ephemeral ports in such a way as to avoid over-utilized links. Also disparities in LAG members with respect to over-utilization of some links and under-utilization of others could help the operator to tweak some of the available parameters or available hash functions for better load distribution.

### 2.7.2 Packet drops and their reasons

Packets may get dropped due to a variety of reasons, and the OAM mechanism should be able to indicate the actual reasons for drop. The response OAM packet should indicate the error code appropriately for various reasons why a packet may have been dropped.

## 2.8 Additional enhancements

Data center networks and applications have specialized needs. To accomplish this the new tools provide certain additional information for the data centers such as the following.

### 2.8.1 Fat-Tree traversal.

The tracing of a fat-tree (i.e. all paths) from the source to the destination is a very important requirement from modern day administrators running say a data-center. This could be done within an administrative boundary and not beyond it.

### 2.8.2 Hash Algorithm Parameters

When choosing from a set of ECMP links or LAG members it is common for a hash function to be performed on select header fields. This hash algorithm is important with respect to which ECMP or LAG member is chosen to forward the packet on. It would be useful to know which fields play a part in the computation of this choice. The actual hash function may be internal to the device and need not be returned since it may be proprietary to the vendor but the header fields accounted for in the hash function would provide enough information for the system / network administrator to vary these parameters in order to figure out a specific path through which the traffic for a flow or a set of flows can be engineered.

### 2.9 Future Requirements

The requirements specified in this document relate to tools for trouble shooting IP layer connectivity with respect to IP nodes in the path from source to destination. This draft deals with tracing the IP nodes such as transit devices and the end target destination. A future version of this set of requirements would look at tracing the intermediate network between IP nodes.

## 3 Security Considerations

This section discusses threats to which these new set of tools might be vulnerable and discusses means by which those threats might be mitigated.

The following are some of the security requirements that need to be adhered to under this framework.

### 3.1 Securing Requests and Responses

Tool developed under this framework should require mechanisms to secure the requests and responses. The security provided for these requests and responses should ensure integrity of these packets and ensure confidentiality if necessary. An administrator should be able to select the information that will be sent in insecure messages, should such secure mechanisms not be available. For example, the set of information exchanged in that case could be limited to the information obtainable via traditional Ping and Traceroute.

### 3.2 Information hiding

There is a concern that tools developed to satisfy the requirements in this document might allow an external user to probe the detailed path that a flow takes through a network. To address this the network

operator could associate multiple security levels with the different types of information that may be included in the response to a discovery packet coming from a legitimate tool. For example only the "Next Hop Router" may be marked as publicly accessible information whereas everything else may be marked as private information. On receiving a flow discovery request packet originating outside the local network, only the publicly accessible information should be included in the response to the originator. However if the request was originated by a legitimate, known source the device could include all of the requested information in the response.

The Result and Additional Information types specified in the section 2.7 provide detailed information about the processing of the request packet and may possibly leak information about the locally configured policies. The amount of information to be included in these sets of data should also depend on whether the request was originated from a legitimate source. The network operator may choose to silently drop the Flow Discovery Request packet without providing any indication of the reason for doing so if the request was originated externally.

### 3.3 Rate limiting obviating attack vectors

Today most network operators throttle conventional OAM traffic (ping and traceroute, and other ICMP messages) that is serviced by the device to protect against Denial-of-Service attacks. Such mechanisms should be employed for OAM packets under this framework for the same reason.

## 4 IANA Considerations

This document does not need any consideration from IANA.

It is likely that tools under this framework may require new IANA assigned protocol ports that signify the specific OAM protocol that is to be implemented to satisfy such requirements. Tools developed to satisfy will require such IANA assignments as the needs arise.

### 4.1 Acknowledgements

The authors would like to thank Ron Bonica for his thorough review and critique of the traceflow proposal [2]. We also would like to thank Melinda Shore for her direction and review of the traceflow proposal which gave rise to this document.

The requirements presented in this document were a result of the traceflow proposal submitted to the IETF by A. Viswanathan, S. Krishnamurthy, R. Manur and V. Zinjuvadia.

## 5 References

### 5.1 Normative References

- [RFC 792] "Internet Control Message Protocol", 2005.
- [RFC 5556] "Transparent Interconnection of Lots of Links", May 2009.
- [IEEE SPB] "IEEE Shortest path Bridging", IEEE 802.1aq

### 5.2 Informative References

- [1] Zinjuvadia et.al, draft-zinjuvadia-traceflow-02.txt, Internet-Draft, August 2008
- [2] Shane Amante et.al, draft-amante-oam-ng-requirements-01.txt, Internet-draft, February 2008.

## Authors' Addresses

Janardhanan Pathangi,  
Dell-Force10,  
Olympia Technology Park,  
Fortius block, 7th & 8th Floor,  
Plot No. 1, SIDCO Industrial Estate,  
Guindy, Chennai - 600032.  
TamilNadu, India.  
Tel: +91 (0) 44 4220 8400  
Fax: +91 (0) 44 2836 2446  
  
Email: Pathangi\_janardhanan@dell.com

Balaji Venkat Venkataswami,  
Dell-Force10,  
Olympia Technology Park,  
Fortius block, 7th & 8th Floor,  
Plot No. 1, SIDCO Industrial Estate,  
Guindy, Chennai - 600032.  
TamilNadu, India.  
Tel: +91 (0) 44 4220 8400  
Fax: +91 (0) 44 2836 2446  
  
Email: BALAJI\_VENKAT\_VENKAT@dell.com



Richard Groves,  
Microsoft Corporation,  
One Microsoft Way,  
Redmond, WA 98052

Email: [rgroves@microsoft.com](mailto:rgroves@microsoft.com)

Peter Hoose,  
Facebook,  
1600, Willow Rd.,  
Menlo Park, CA 94025

Email: [phoose@fb.com](mailto:phoose@fb.com)

Internet Engineering Task Force  
Internet-Draft  
Intended status: Standards Track  
Expires: January 17, 2013

M. MacFaden  
VMware Inc.  
J. Schoenwaelder  
Jacobs University  
T. Tsou  
Huawei Technologies (USA)  
C. Zhou  
Huawei Technologies  
July 16, 2012

Definition of Managed Objects for Virtual Machines Controlled by a  
Hypervisor  
draft-schoenw-opsawg-vm-mib-01

Abstract

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

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 17, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

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

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

## Table of Contents

1. Introduction . . . . .	3
2. The Internet-Standard Management Framework . . . . .	3
3. Conventions . . . . .	3
4. Overview . . . . .	3
5. Relationship to Other MIB Modules . . . . .	4
5.1. Relationship to the HOST-RESOURCES-MIB . . . . .	5
5.2. Relationship to the IF-MIB . . . . .	5
5.3. Relationship to the IEEE8021-BRIDGE-MIB . . . . .	5
5.4. Relationship to the ENTITY-MIB . . . . .	5
6. Definitions . . . . .	6
7. Security Considerations . . . . .	19
8. IANA Considerations . . . . .	20
9. Acknowledgements . . . . .	20
10. References . . . . .	20
10.1. Normative References . . . . .	20
10.2. Informative References . . . . .	21
Appendix A. Open Issues . . . . .	21

## 1. Introduction

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols. In particular, it defines objects for managing virtual machines controlled by a hypervisor.

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.

## 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. Conventions

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].

## 4. Overview

The MIB module is organized into a group of scalars and tables. The scalars below vmHypervisor provide basic information about the hypervisor. The vmGuestTable lists the guests (virtual machines) that are known to the hypervisor. The vmStorageTable and the vmIfTable provide the mapping of logical storage areas and network interfaces to virtual machines.

The GuestState textual convention defines a state model for virtual machines. Events causing transitions between major states will cause the generation of notifications (vmGuestStarted, vmGuestStopped, vmGuestSuspended, vmGuestResumed).

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.XXXX)
+--vmNotifications(0)
|   +--vmGuestStarted(1) [vmGuestName,vmGuestUUID,vmGuestState]
|   +--vmGuestStopped(2) [vmGuestName,vmGuestUUID,vmGuestState]
|   +--vmGuestSuspended(3) [vmGuestName,vmGuestUUID,vmGuestState]
|   +--vmGuestResumed(4) [vmGuestName,vmGuestUUID,vmGuestState]
+--vmObjects(1)
|   +--vmHypervisor(1)
|   |   +-- r-n SnmpAdminString vmHypervisorVersion(1)
|   +--vmGuestTable(2)
|   |   +--vmGuestEntry(1) [vmGuestIndex]
|   |   |   +-- --- GuestIndex          vmGuestIndex(1)
|   |   |   +-- r-n SnmpAdminString vmGuestName(2)
|   |   |   +-- r-n UUIDOrZero      vmGuestUUID(3)
|   |   |   +-- r-n GuestState      vmGuestState(4)
|   |   |   +-- r-n SnmpAdminString vmGuestOS(6)
|   |   |   +-- r-n Unsigned32      vmGuestCurCPUs(7)
|   |   |   +-- rwn Unsigned32      vmGuestMinCPUs(8)
|   |   |   +-- rwn Unsigned32      vmGuestMaxCPUs(9)
|   |   |   +-- r-n KBytes          vmGuestCurMem(10)
|   |   |   +-- rwn KBytes          vmGuestMinMem(11)
|   |   |   +-- rwn KBytes          vmGuestMaxMem(12)
|   |   |   +-- r-n Unsigned32      vmGuestCPUTime(13)
|   +--vmStorageTable(3)
|   |   +--vmStorageEntry(1) [vmGuestIndex,vmStorageIndex]
|   |   |   +-- --- GuestIndexOrZero vmStorageGuest(1)
|   |   |   +-- --- StorageIndex      vmStorageIndex(2)
|   |   |   +-- r-n SnmpAdminString  vmStorageName(3)
|   +--vmIfTable(4)
|   |   +--vmIfEntry(1) [vmGuestIndex,vmIfIndex]
|   |   |   +-- --- GuestIndexOrZero vmIfGuest(1)
|   |   |   +-- --- InterfaceIndex    vmIfIndex(2)
|   |   |   +-- r-n PhysAddress        vmIfPhysAddr(3)
```

## 5. Relationship to Other MIB Modules

The MIB module IMPORTS definitions from SNMPv2-SMI [RFC2578], SNMPv2-TC [RFC2579], SNMPv2-CONF [RFC2580], SNMP-FRAMEWORK-MIB [RFC3411], and IF-MIB [RFC2863].

Hypervisors implementing this MIB module should implement the HOST-RESOURCES-MIB [RFC2790] and the IF-MIB [RFC2863] in order to export information about the resources (e.g., processors, memory, logical storage devices, network interfaces) of the physical machine. If the hypervisor emulates a bridge to network virtual machines, then it should implement the IEEE8021-BRIDGE-MIB. (Note that the BRIDGE-MIB defined in [RFC4188] is now further maintained by the IEEE [RFC4663].) Details of the hardware configuration of a physical machine can be made available by implementing the ENTITY-MIB [RFC4133].

#### 5.1. Relationship to the HOST-RESOURCES-MIB

The HOST-RESOURCES-MIB implemented on the physical machine provides information about the number of CPUs and the amount of memory available. Furthermore, the HOST-RESOURCES-MIB provides information about logical storage devices.

The MIB module defined in this memo provides a mapping of logical storage devices to virtual machines. Further details about the storage devices (such as the size and the amount of allocated storage) is provided by the HOST-RESOURCES-MIB. Note that the number of storage types can be extended through the IANA maintained HOST-RESOURCES-TYPES MIB module.

#### 5.2. Relationship to the IF-MIB

The MIB module provides a mapping of network interfaces to virtual machines. Further details about the network interfaces (such as statistics about the number of packets/bytes sent or received) can be obtained from the IF-MIB.

#### 5.3. Relationship to the IEEE8021-BRIDGE-MIB

Hypervisors implementing virtual bridges should export the bridging topologies by implementing the IEEE8021-BRIDGE-MIB. For backwards compatibility with existing management applications, they may also choose to implement the BRIDGE-MIB [RFC4188].

#### 5.4. Relationship to the ENTITY-MIB

The ENTITY-MIB [RFC4133] describes managed objects used for managing multiple logical and physical entities managed by a single SNMP agent. Implementations of the MIB module defined in this document may want to use the ENTITY-MIB to provide the logical to physical entity mapping and if needed to point to the agent in the virtual machine and vice versa.

## 6. Definitions

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

```
IMPORTS
```

```
    MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
    Integer32, Unsigned32, mib-2
        FROM SNMPv2-SMI -- RFC 2578
    TEXTUAL-CONVENTION, PhysAddress
        FROM SNMPv2-TC -- RFC 2579
    OBJECT-GROUP, NOTIFICATION-GROUP, MODULE-COMPLIANCE
        FROM SNMPv2-CONF -- RFC 2580
    SnmpAdminString
        FROM SNMP-FRAMEWORK-MIB -- RFC 3411
    InterfaceIndex
        FROM IF-MIB; -- RFC 2863
```

```
vmMib MODULE-IDENTITY
```

```
    LAST-UPDATED "201203150000Z"
```

```
    ORGANIZATION
```

```
        "Jacobs University Bremen"
```

```
    CONTACT-INFO
```

```
        "Michael MacFaden
```

```
        VMware Inc.
```

```
        Email: mrm@vmware.com
```

```
        Juergen Schoenwaelder
```

```
        Jacobs University Bremen
```

```
        Email: j.schoenwaelder@jacobs-university.de
```

```
        Tina Tsou
```

```
        Huawei Technologies (USA)
```

```
        Email: tina.tsou.zouting@huawei.com
```

```
        Cathy Zhou
```

```
        Huawei Technologies
```

```
        Email: cathyzhou@huawei.com"
```

```
DESCRIPTION
```

```
    "The MIB module for monitoring virtual machines controlled
    by a hypervisor.
```

```
    Copyright (c) 2012 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 "201203150000Z"

DESCRIPTION

"Initial version, published as RFC XXXX."

-- RFC Ed.: replace XXXX with actual RFC number & remove this note  
::= { mib-2 XXXX }

vmNotifications OBJECT IDENTIFIER ::= { vmMib 0 }

vmObjects OBJECT IDENTIFIER ::= { vmMib 1 }

vmConformance OBJECT IDENTIFIER ::= { vmMib 2 }

-- Textual convention definitions:

GuestIndex ::= 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)

GuestIndexOrZero ::= TEXTUAL-CONVENTION

DISPLAY-HINT "d"

STATUS current

DESCRIPTION

"This textual convention is an extension of the VmGuestIndex 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)

StorageIndex ::= TEXTUAL-CONVENTION

DISPLAY-HINT "d"

STATUS current

DESCRIPTION

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

SYNTAX Integer32 (1..2147483647)



UUID ::= TEXTUAL-CONVENTION  
    DISPLAY-HINT "4x-2x-2x-2x-6x"  
    STATUS current  
    DESCRIPTION  
        "The Universally Unique IDentifier (UUID) identifying a  
        virtual machine. The UUID format is defined in RFC 4122."  
    REFERENCE  
        "RFC4122: A Universally Unique IDentifier (UUID) URN Namespace"  
    SYNTAX OCTET STRING (SIZE (16))

UUIDOrZero ::= TEXTUAL-CONVENTION  
    DISPLAY-HINT "4x-2x-2x-2x-6x"  
    STATUS current  
    DESCRIPTION  
        "The Universally Unique IDentifier (UUID) identifying a  
        virtual machine or a zero-length string. The UUID format is  
        defined in RFC 4122. The meaning of the zero-length string is  
        object-specific and must therefore be defined as part of the  
        description of any object which uses this syntax."  
    SYNTAX OCTET STRING (SIZE (0|16))

GuestState ::= TEXTUAL-CONVENTION  
    STATUS current  
    DESCRIPTION  
        "The state of a guest (virtual machine):

unknown(1)	The state is unknown, e.g., because the implementation failed to obtain the state from the hypervisor.
other(2)	The state has been obtained but it is not a known state.
running(3)	The virtual machine is currently running.
blocked(4)	The virtual machine is currently blocked.
paused(5)	The virtual machine is currently paused.
migrating(6)	The virtual machine is currently migrating.
shutdown(7)	The virtual machine is currently in the process of shutting down.
shutoff(8)	The virtual machine is down.
crashed(9)	The virtual machine has crashed."

SYNTAX INTEGER {

```
        unknown(1),
        other(2),
        running(3),
        blocked(4),
        paused(5),
        migrating(6),
        shutdown(7),
        shutoff(8),
        crashed(9)
    }

KBytes ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS      current
    DESCRIPTION
        "Storage size measured in units of 1024 octets (bytes). This
        textual convention allows to represent storage sizes up to
        4096 gigabytes."
    SYNTAX Unsigned32

-- Object definitions

vmHypervisor OBJECT IDENTIFIER ::= { vmObjects 1 }

vmHypervisorVersion OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The version string indicating the version of the hypervisor
        running on the physical host."
    ::= { vmHypervisor 1 }

-- The number of CPUs and the amount of memory can be found
-- in the objects of the HOST-RESOURCES-MIB

vmGuestTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF VmGuestEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A (conceptual) table of all guests (virtual machines)
        on the physical host."
    ::= { vmObjects 2 }

vmGuestEntry OBJECT-TYPE
    SYNTAX      VmGuestEntry
    MAX-ACCESS  not-accessible
```

```

STATUS      current
DESCRIPTION
    "An (conceptual) table entry describing a particular
    guest (virtual machine)."
```

INDEX { vmGuestIndex }

::= { vmGuestTable 1 }

VmGuestEntry ::= SEQUENCE {

```

    vmGuestIndex      GuestIndex,
    vmGuestName       SnmpAdminString,
    vmGuestUUID       UUIDOrZero,
    vmGuestState      GuestState,
-- XXX add information about the CPU type
-- XXX the cpu type may be different from the host CPU
    vmGuestOS         SnmpAdminString,
    vmGuestCurCPUs   Unsigned32,
    vmGuestMinCPUs    Unsigned32,
    vmGuestMaxCPUs    Unsigned32,
    vmGuestCurMem     KBytes,
    vmGuestMinMem      KBytes,
    vmGuestMaxMem      KBytes,
    vmGuestCPUTime     Unsigned32
}
```

vmGuestIndex OBJECT-TYPE

```

SYNTAX      GuestIndex
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "A unique value identifying a guest (virtual machine)."
```

::= { vmGuestEntry 1 }

vmGuestName OBJECT-TYPE

```

SYNTAX      SnmpAdminString
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The name of this guest (virtual machine)."
```

::= { vmGuestEntry 2 }

vmGuestUUID OBJECT-TYPE

```

SYNTAX      UUIDOrZero
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "A UUID identifying this guest (virtual machine). The UUID
    is expected to be a long-term persistent identifier and
    to remain the same across reboots of the virtual machines"
```

and the hypervisor. The zero-length string is returned in case a virtual machine does not have a suitable persistent UUID."  
 ::= { vmGuestEntry 3 }

vmGuestState OBJECT-TYPE  
SYNTAX GuestState  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
"The current operational state of the guest (virtual machine)."  
 ::= { vmGuestEntry 4 }

vmGuestOS OBJECT-TYPE  
SYNTAX SnmpAdminString  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
"The operating system running on this guest (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."  
 ::= { vmGuestEntry 6 }

vmGuestCurCPUs OBJECT-TYPE  
SYNTAX Unsigned32  
UNITS "CPUs"  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
"The number of CPUs currently assigned to this guest (virtual machine). Virtual machines that are not operational typically have 0 CPUs assigned."  
 ::= { vmGuestEntry 7 }

vmGuestMinCPUs OBJECT-TYPE  
SYNTAX Unsigned32  
UNITS "CPUs"  
MAX-ACCESS read-write  
STATUS current  
DESCRIPTION  
"The minimum number of CPUs that are assigned to this guest (virtual machine) when it is in a running state. Changes to this value may not persist across restarts of the hypervisor."

```
 ::= { vmGuestEntry 8 }

vmGuestMaxCPUs OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "CPUs"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The maximum number of CPUs that are assigned to this
        guest (virtual machine) when it is in a running state.
        The value zero denotes that there is no limit. Changes
        to this value may not persist across restarts of the
        hypervisor."
 ::= { vmGuestEntry 9 }

vmGuestCurMem OBJECT-TYPE
    SYNTAX      KBytes
    UNITS       "KBytes"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The amount of main memory currently assigned to this
        guest (virtual machine). Virtual machines that are not
        operational typically have no memory assigned."
 ::= { vmGuestEntry 10 }

vmGuestMinMem OBJECT-TYPE
    SYNTAX      KBytes
    UNITS       "KBytes"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The minimum amount of main memory that is assigned to
        this guest (virtual machine) when it is in a running
        state. Changes to this value may not persist across
        the restart of the hypervisor."
 ::= { vmGuestEntry 11 }

vmGuestMaxMem OBJECT-TYPE
    SYNTAX      KBytes
    UNITS       "KBytes"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The maximum amount of main memory that can be assigned to
        this guest (virtual machine) when it is in a running state.
        The value zero denotes that there is no limit. Changes to
        this value may not persist across the restart of the
```

```

        hypervisor."
 ::= { vmGuestEntry 12 }

vmGuestCPUTime OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "seconds"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of CPU seconds consumed by this guest (virtual
        machine). Note that on a virtual machines with multiple
        CPUs, this value may increment by more than one second
        in a second of real (wall clock) time."
 ::= { vmGuestEntry 13 }

vmStorageTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF VmStorageEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A (conceptual) table of storage devices attached to
        guests (virtual machines)."
```

```

 ::= { vmObjects 3 }

vmStorageEntry OBJECT-TYPE
    SYNTAX      VmStorageEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An (conceptual) table entry describing a particular
        storage device attached to a guest (virtual machine)"
    INDEX       { vmStorageGuest, vmStorageIndex }
 ::= { vmStorageTable 1 }

VmStorageEntry ::= SEQUENCE {
    vmStorageGuest      GuestIndexOrZero,
    vmStorageIndex      StorageIndex,
    vmStorageName       SnmpAdminString
}

vmStorageGuest OBJECT-TYPE
    SYNTAX      GuestIndexOrZero
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Identifies the guest (virtual machine) this storage has
        been allocated to. The value 0 indicates that the storage
        is currently not allocated to a guest (virtual machine)."
```

```
 ::= { vmStorageEntry 1 }

vmStorageIndex OBJECT-TYPE
    SYNTAX      StorageIndex
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "A unique value identifying a logical storage area. On
        systems implementing the HOST-RESOURCES-MIB, the value
        must be the same value that is used as the index into
        the hrStorageTable (hrStorageIndex)."
```

```
 ::= { vmStorageEntry 2 }

vmStorageName OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The name of the storage area as seen on the hypervisor."
```

```
 ::= { vmStorageEntry 3 }

vmIfTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF VmIfEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "A (conceptual) table of network interfaces attached to
        guests (virtual machines)."
```

```
 ::= { vmObjects 4 }

vmIfEntry OBJECT-TYPE
    SYNTAX      VmIfEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "An (conceptual) table entry describing a particular
        network interface attached to a guest (virtual machine)"
    INDEX       { vmGuestIndex, vmIfIndex }
    ::= { vmIfTable 1 }

VmIfEntry ::= SEQUENCE {
    vmIfGuest      GuestIndexOrZero,
    vmIfIndex      InterfaceIndex,
    vmIfPhysAddr   PhysAddress
}

vmIfGuest OBJECT-TYPE
    SYNTAX      GuestIndexOrZero
```

```
MAX-ACCESS not-accessible
STATUS      current
DESCRIPTION
    "Identifies the guest (virtual machine) this network interface
    has been allocated to. The value 0 indicates that the network
    interface is currently not allocated to a guest (virtual
    machine)."
```

::= { vmIfEntry 1 }

```
vmIfIndex OBJECT-TYPE
    SYNTAX      InterfaceIndex
    MAX-ACCESS not-accessible
    STATUS      current
    DESCRIPTION
        "The interface index of the network interface under which it
        is known on the system running the hypervisor. If the
        interface is a port of a virtual bridge, then the port
        of the virtual bridge should map to this interface index."
```

::= { vmIfEntry 2 }

```
vmIfPhysAddr OBJECT-TYPE
    SYNTAX      PhysAddress
    MAX-ACCESS read-only
    STATUS      current
    DESCRIPTION
        "The physical address used by the interface. For interfaces
        associated to a port of a virtual bridge, this object
        normally contains a MAC address. For interfaces which do not
        have such an address, this object should contain a
        zero-length octet string."
```

::= { vmIfEntry 3 }

-- Notification definitions:

```
vmGuestStarted NOTIFICATION-TYPE
    OBJECTS      {
        vmGuestName,
        vmGuestUUID,
        vmGuestState
    }
    STATUS      current
    DESCRIPTION
        "This notification is generated when a guest (virtual machine)
        has been started and the start process has reached a stable
        state (e.g., running or crashed)."
```

::= { vmNotifications 1 }



```

vmGuestStopped NOTIFICATION-TYPE
    OBJECTS      {
                    vmGuestName,
                    vmGuestUUID,
                    vmGuestState
                }
    STATUS        current
    DESCRIPTION
        "This notification is generated when a guest (virtual machine)
        has been stopped and the shutdown process has reached a stable
        state (e.g., shutdown or shutoff or crashed)."
```

::= { vmNotifications 2 }

```

vmGuestSuspended NOTIFICATION-TYPE
    OBJECTS      {
                    vmGuestName,
                    vmGuestUUID,
                    vmGuestState
                }
    STATUS        current
    DESCRIPTION
        "This notification is generated when a guest (virtual machine)
        has been suspended and the suspension process has reached a
        stable state (e.g., paused or crashed)."
```

::= { vmNotifications 3 }

```

vmGuestResumed NOTIFICATION-TYPE
    OBJECTS      {
                    vmGuestName,
                    vmGuestUUID,
                    vmGuestState
                }
    STATUS        current
    DESCRIPTION
        "This notification is generated when a guest (virtual machine)
        has been resumed and the resumption process has reached a
        stable state (e.g., running or crashed)."
```

::= { vmNotifications 4 }

-- Compliance definitions:

```

vmGroups          OBJECT IDENTIFIER ::= { vmConformance 1 }
vmCompliances     OBJECT IDENTIFIER ::= { vmConformance 2 }
```

```

vmFullCompliance MODULE-COMPLIANCE
    STATUS        current
    DESCRIPTION
        "Compliance statement for implementations supporting
```

```
        read/write access, according to the object definitions."
MODULE      -- this module
MANDATORY-GROUPS {
    vmHypervisorGroup,
    vmGuestGroup,
    vmStorageGroup,
    vmIfGroup,
    vmNotificationGroup
}
 ::= { vmCompliances 1 }

vmReadOnlyCompliance MODULE-COMPLIANCE
STATUS      current
DESCRIPTION
    "Compliance statement for implementations supporting
    only readonly access."
MODULE      -- this module
MANDATORY-GROUPS {
    vmHypervisorGroup,
    vmGuestGroup,
    vmStorageGroup,
    vmIfGroup,
    vmNotificationGroup
}

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

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

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

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

vmHypervisorGroup OBJECT-GROUP
OBJECTS {
```

```
        vmHypervisorVersion
    }
    STATUS          current
    DESCRIPTION
        "A collection of objects providing insight into the
        hypervisor itself."
    ::= { vmGroups 1 }

vmGuestGroup OBJECT-GROUP
    OBJECTS {
        -- vmGuestIndex,
        vmGuestName,
        vmGuestUUID,
        vmGuestState,
        vmGuestOS,
        vmGuestCurCPUs,
        vmGuestMinCPUs,
        vmGuestMaxCPUs,
        vmGuestCurMem,
        vmGuestMinMem,
        vmGuestMaxMem,
        vmGuestCPUTime
    }
    STATUS          current
    DESCRIPTION
        "A collection of objects providing insight into the
        guests (virtual machines) controlled by a hypervisor."
    ::= { vmGroups 2 }

vmStorageGroup OBJECT-GROUP
    OBJECTS {
        -- vmStorageGuest,
        -- vmStorageIndex,
        vmStorageName
    }
    STATUS          current
    DESCRIPTION
        "A collection of objects providing insight into the
        logical storage areas controlled by a hypervisor."
    ::= { vmGroups 3 }

vmIfGroup OBJECT-GROUP
    OBJECTS {
        -- vmIfGuest,
        -- vmIfIndex,
        vmIfPhysAddr
    }
    STATUS          current
```

## DESCRIPTION

"A collection of objects providing insight into the network interfaces controlled by a hypervisor."

::= { vmGroups 4 }

## vmNotificationGroup NOTIFICATION-GROUP

## NOTIFICATIONS {

vmGuestStarted,  
vmGuestStopped,  
vmGuestSuspended,  
vmGuestResumed

}

STATUS current

## DESCRIPTION

"A collection of notifications for virtual machines controlled by a hypervisor."

::= { vmGroups 5 }

END

## 7. Security Considerations

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

- o Unauthorized changes to vmGuestMinCPUs, vmGuestMaxCPUs, vmGuestMinMem, and vmGuestMaxMem can significantly slow down virtual machines or prevent the start of new virtual machines.

Some of the readable objects in this MIB module (i.e., objects with a MAX-ACCESS other than not-accessible) may be considered sensitive or vulnerable in some network environments. It is thus important to control even GET and/or NOTIFY access to these objects and possibly to even encrypt the values of these objects when sending them over the network via SNMP. These are the tables and objects and their sensitivity/vulnerability:

- o The tables vmGuestTable, vmStorageTable, and vmIfTable provide insight into the resources allocated to virtual machines and this knowledge might be exploited for targeted denial of service attacks.

- o The vmGuestStarted, vmGuestStopped, vmGuestSuspended, and vmGuestResumed notifications provides information about state changes of virtual machines and implicitly also on which physical hosts virtual machines are located. Furthermore, the generation of fake notifications might trigger false alarms and subsequent actions in a network management system, which can amplify denial of service attacks or simply lead to less efficient resource usage.

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

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

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

## 8. IANA Considerations

IANA is requested to assign a value for "XXXX" 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 "XXXX" (here and in the MIB module) with the assigned value and to remove this note.

## 9. Acknowledgements

Thanks to David Black and Robert Story for helpful comments during the development of this specification.

## 10. References

### 10.1. 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.
- [RFC3411] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks", STD 62, RFC 3411, December 2002.
- [RFC4133] Bierman, A. and K. McCloghrie, "Entity MIB (Version 3)", RFC 4133, August 2005.

## 10.2. 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.
- [RFC4188] Norseth, K. and E. Bell, "Definitions of Managed Objects for Bridges", RFC 4188, September 2005.
- [RFC4663] Harrington, D., "Transferring MIB Work from IETF Bridge MIB WG to IEEE 802.1 WG", RFC 4663, September 2006.

## Appendix A. Open Issues

This file is used to track issues that were discussed during the development of the SMIV2 to YANG translation in the IETF NETMOD working group. This issues covered here concern major design choices; this file does not attempt to track minor clarification requests etc.

To comment on issues on the mailing list, please include the issue number in the subject line of the email message.

\* vm-mib-01: storage sizes

The MIB does not provide storage sizes, assuming this is provided by the hrStorageTable of the HOST-RESOURCES-MIB. However, some well known implementations of the HOST-RESOURCES-MIB only report about file systems used by the host system and not file systems residing in files used by virtual machines. Furthermore, the hrStorageTable reports sizes "usable by the requesting entity", "excluding loss due to formatting of file system reference information". For storage provided to virtual machines, this information is often not readily available since all you have is the raw block size.

\*\* Solution #01-01

Provide the storage block sizes as part of the VM-MIB. Provide a pointer to the hrStorageTable on systems that can provide this linkage but allow the pointer to be NULL.

\*\* Resolution

TBD

\* vm-mib-02: scaling and caching support

It was mentioned that large data centers are characterized by 100.000 physical hosts running 2.000.000 virtual machines. The NASA is reported with 1.000.000 physical hosts and 60.000.000 virtual machines. Bottom line is that we need to make the MIB module scalable. We can assume up hundreds of VMs running on a single virtual machine.

\*\* Solution #02-01

Add ...LastChange objects to tables so that management applications can easily validate cached information without having to read through potentially larger tables. For the vmGuestTable, we might also provide a ...LastStateChange object so that state changes can be polled with reading a simple scalar.

\*\* Solution #02-02

Make some tables time filtered. Unclear which tables would have to be time filtered.

\*\* Resolution

TBD

\* vm-mib-03: virtual cpu type identification

It is necessary to identify the CPU architecture or type since some virtual machine systems can emulate different CPU types.

\*\* Solution #03-01

Provide an IANA controlled enumeration that provides a CPU classification. The problem will be to provide rules about what constitutes a new CPU type and what not.

\*\* Solution #03-02

Use OBJECT IDENTITIES to identify CPU types. Such a distributed enumeration will not achieve a great deal of interoperability and is likely close to #03-03.

\*\* Solution #03-03

Use a string data type and rely on systems to put meaningful information there, perhaps provide guidelines how to structure the CPU type names, e.g. vendor-arch-model(-features)\* that is amd-x86\_64-opteron or intel-i686-pentium3-vmx-acpi (perhaps using a different separator character since a dash might easily clash). Applications may have to do some normalization across VM-MIB implementations (e.g., regular expression matching) but on the other hand this allows to provide details where necessary.

\*\* Solution #03-04

Following #03-03, we provide ...GuestCpuVendor, ...GuestCpuArch and ...GuestCpuModel objects plus an additional table that provides details about the features of the CPUs used by a certain virtual machine. This essentially breaks the string into a set of separate MIB objects.

\*\* Solution #03-05

Following #03-04, we provide ...GuestCpuVendor, ...GuestCpuArch and ...GuestCpuModel objects plus a string object containing a list of features. This way, things are more compact but still the most important components (vendor, arch, model) are broken out as separate objects.

\*\* Resolution

TBD



\* vm-mib-04: physical CPU type identification

VM migration sometimes requires to match physical CPUs and more important also feature sets of physical CPUs.

\*\* Solution #04-01:

Extend the ENTITY-MIB with a new MIB module, say an ENTITY-CPU-MIB, providing an entPhyCPUTable, sparsely augmenting the entPhysicalTable for physical entities with entPhysicalClass = cpu. The entPhyCPUTable would contain information about CPU vendor, CPU architecture, CPU mode, CPU features, clock speeds, etc. (see also vm-mib-03).

\*\* Resolution

TBD

\* vm-mib-05: per virtual cpu statistics

It seems to be useful to provide statistics for each virtual CPU. However, it remains unclear what can be expected to be provided by a typical hypervisor implementation. There are a number of things to consider:

- a) Reporting the time the virtual CPU has been running (CPU time consumed) seems relatively straight forward.
- b) Reporting the current state of a virtual CPU requires to first define a suitable state model that is coarse grained enough to be useful (otherwise CPU state changes far too quickly to yield meaningful results). Libvirt, for example, has CPU states offline, running, blocked on resource. It is not further defined what blocked on resource really means. Anyway, with a suitable state model, the MIB could provide the time spent in the various CPU states rather than or in addition to the current snapshot state.
- c) Reporting the affinity mapping of virtual CPUs to physical CPUs. This, of course, requires to have a representation of physical CPUs.

\*\* Resolution

TBD

\* representing networks (vmNetTable)

Not yet well enough understood to write up this issue. ;-)

Authors' Addresses

Michael MacFaden  
VMware Inc.

EMail: mrm@vmware.com

Juergen Schoenwaelder  
Jacobs University  
Campus Ring 1  
Bremen 28759  
Germany

EMail: j.schoenwaelder@jacobs-university.de

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

