Operations and Management Area Working Group
Internet-Draft
Intended status: Informational
Expires: March 30, 2015

Q. Wu
M. Wexler
Huawei
D. Romascanu
AVAYA
T. Taylor, Ed.
PT Taylor Consulting
September 26, 2014

Problem Statement for Layer and Technology Independent OAM in a Multi-Layer Environment

draft-edprop-opsawg-multi-layer-oam-ps-02.txt

Abstract

Operations, Administration, and Maintenance (OAM) mechanisms are critical building blocks in network operations. They used for service fulfillment assurance, and for service diagnosis, troubleshooting, and repair. The current practice is that many technologies rely on their own OAM protocols and procedures that are exclusive to a given layer.

At present, there is little consolidation of OAM in the management plane or well-documented inter-layer OAM operation. Vendors and operators dedicate significant resources and effort through the whole OAM life-cycle each time a new technology is introduced. This is exacerbated when dealing with integration of OAM into overlay networks, which require better OAM visibility since there is no method to exchange OAM information between overlay and underlay.

This document analyzes the problem space for multi-layer OAM in the management plane with a focus on layer and technology independent OAM management considerations. It concludes that an attempt to define an architecture for consolidated management should be undertaken, and if this attempt satisfies key objectives, a gap analysis and a program of standardization should follow.

Status of This Memo

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

This Internet-Draft will expire on March 30, 2015.

Copyright Notice

Table of Contents

1. Introduction

Operations, Administration, and Maintenance (OAM, [RFC6291]) mechanisms are critical tools, used for service assurance, fulfillment, or service diagnosis, troubleshooting, and repair, as well as supporting functions such as accounting and security management. The key foundations of OAM and its functional roles in monitoring and diagnosing the behavior of networks have been studied at OSI layers 1, 2 and 3 for many years.

When operating networks with more than one technology in an overlay network, maintenance and troubleshooting are achieved per technology and per layer. As a result, operational processes can be very cumbersome. Stitching together the OAM of adjacent transport segments (as defined in Section 2 in one administrative domain is often not defined and left to proprietary solutions.

Current practice, which consists in enabling specific OAM techniques for each layer, has shown its limits. Concretely, we see today a large number of layer 1/2/3 OAM protocols being well developed and some of them being successfully deployed, but how these OAM protocols in each layer can be applied to overlay networks that are using different encapsulation protocols so as to provide better OAM visibility is still a challenging issue. When no mechanism is defined to exchange performance and liveliness information between the underlay and overlay(s) by a coordination system, it is hard, for instance, to determine whether a fault originates in higher or lower layer.

Section 1.1 of [RFC7276] makes the point that each layer in a multi-layer architecture has its own OAM protocols. From this follows the basic principle that OAM in the data plane cannot cross layer boundaries. A similar constraint holds for boundaries between different transport technologies in the same layer, barring the stitching mentioned above.

One concludes that to simplify OAM and make it more responsive in a multi-layer network requires further consolidation in the management plane. The work on management consolidation would benefit from at least some new standardization. A detailed examination of the potential scope of the work is left for a gap analysis following successful definition of an architecture.

This document further argues that in addition to the ability to retrieve technology specific information from managed entities when following up on problems, consolidated management requires a technology independent view of the network and supporting layers.

How this view is obtained is a key architectural issue outside the scope of the present document.

1.1. A Vision of Layer and Technology Independent Management

What follows is based on the assumption of a network supported by a strict hierarchy of underlying layers in the data plane. There may be multiple layers at a given level of the OSI layer 1-2-3 hierarchy, but that is irrelevant to the vision.

A management application presents to an user a view of this network and its supporting layers that is strictly topological, free of any technology specific information. The user notes a defect along a path serving a particular customer. Looking at the next lower path, the user also sees a defect. Looking the next lower path again, there is also a defect. No lower defect is noted.

At this point it is appropriate to indicate what the user can see along a given path. The path is divided into one or more segments, each spanned by a specific transport technology. However, as already stated, the user does not see any technology specific information. Instead, as well as distinguishing the segments, the user can identify the managed elements at the beginning and end of each segment.

To clarify the situation, the user issues an abstract Continuity Check command, directed toward the initial managed element of the segment in which a fault appears to lie (i.e., in the lowest layer where a defect was observed). By means to be determined by architectural choice, this command is converted into a technology-specific request which is executed across the selected segment. Possible outcomes include:

1. The fault could come clear as a result of the test. The immediate problem is solved (and may have affected multiple upper paths besides the one of initial interest) and the point at which it occurred could be flagged for follow-up maintenance.

2. Local craft action to clear the fault is available in timely fashion.

3. Timely local craft action is not possible, and capacity is reallocated on other paths to ensure that service levels are maintained. Note that capacity reallocation can be done based on the topological view of the network, still on a layer and technology independent basis.

In case (2), technology specific management capabilities are likely to be required by the craftperson following up on the problem.

1.2. Looking Forward

The remainder of this document develops the ideas just stated at a greater level of detail. Section 2 provides terminology that is important to the understanding of the rest of the document. Section 3 establishes preliminary objectives that are key to determining whether a complete program of standardization of consolidated management should be undertaken. Section 4 provides the problem analysis. It is divided into three parts: an argument for consolidated management (Section 4.1), an argument for layer and technology independent management (Section 4.2), and an examination of some more detailed issues. Section 5 provides the problem statement, and Section 6 provides some considerations that should be taken into account in the proposed work on architecture.

2. Terminology

[RFC6291], cited above, provides the official IETF description of Operations, Administration, and Maintenance (OAM) terminology. For a more extensive description of OAM and related terms, see the opening sections, but particularly Sections 2.2.1 through 2.2.3, of [RFC7276].

Section 2.2.4 of [RFC7276] introduces the terms data plane, control plane, and management plane.

This document introduces its own interpretation of the following terms, which are in wide use but in that general usage present ambiguities:

Management:

A definition of management can be inferred from [RFC6123], which in turn refers to [RFC5706]. Unfortunately the latter chose to divide operations from management, at least from a documentation point of view. The present document chooses to define management as a function that is concerned with all three of operations, administration, and maintenance.

Layer:

The word "layer" has two potential meanings. In the first instance, it is a topological concept, representing a position in a hierarchy of layers. In the second instance, it refers to OSI layers 1, 2 and 3. Within this document, "layer independent OAM management" as defined below emphasizes the latter meaning when talking about independence, but is intended to extend to all layers of the hierarchy supporting a given network or overlay (the topological view of "layer").

This document makes use of the following additional terms:

Layer independent OAM management:

In a multi-layer network, layer independent OAM management refers to OAM in the management plane that can be deployed independently of media, data protocols, and routing protocols. It denotes the ability to gather OAM information at the different layers, correlate it with layer-specific identifiers and expose it to the management application through a unified interface.

Managed entity:

An architectural concept, an instance of what the management function manages. By definition, a managed entity is capable of communicating with the management function in the management plane.

Local Management Entity (LMgmtE):

An instance of a management function that is restricted in scope to communication with the managed entities associated with a specific transport segment in a specific layer. This term includes legacy management entities in an existing network, and may include entities of a similar scope if they are defined in a consolidated management architecture.

Consolidated Management Entity (CMgmtE):

An instance of the management function that is capable of communicating with all of the LMgmtEs and/or managed entities in a scoped part of the network in order to achieve end-to-end and service-level views of network performance and status and initiate actions when required. The phrase "LMgmtEs and/or managed entities" allows for the possibility that the target architecture allows for direct communication between the CMgmtE and the managed entities or alternatively chooses to assume a distributed management architecture. In any case, as discussed in Section 6, the CMgmtE will have to communicate with legacy LMgmtEs during the transition from the existing to the target architecture.

Management subsystem:

The implementation of the management function in a given network.

Managed device:

A network element associated with at least one technology layer and one managed entity.

Transport segment:

Refers to the portion of a path at a given layer bounded by two points between which a specific transport technology is used and beyond which either a different technology is used or the path is terminated.

Three-dimensional topology:

Refers to a three-dimensional view of the topology of the network and supporting layers. The view of paths along a layer comprises two dimensions. The third dimension is provided by the ordered hierarchy of layers from bottom to top at any point along a path. The three-dimensional topology includes per-path capacity and flow information, permitting layer and technology independent reallocation of capacity as required.

3. A Preliminary Set Of Objectives

Before going further, it is possible to state a preliminary set of objectives for this work. If it does not appear that these can be satisfied, there is no point in undertaking further effort.

As a first objective, the outcome of the work must reduce the time required to respond to and mitigate service-affecting events. The ideal result is that the system be able to do so before the customer notices a service degradation. It is possible that satisfaction of this objective alone is sufficient to carry on.

A second objective relates to the business case for the work and is more difficult for the IETF to judge but crucial for operators attempting to justify changes in their network infrastructure. It should be possible to expect a reduction in life cycle capex and opex as a result of making those changes, even taking account of the potential costs of abandoning or upgrading existing equipment. This objective may influence work on architecture for consolidated management toward minimizing those latter costs (capex). On the positive side, likely savings in craftsperson time implied by the first objective are helpful to the business case (opex).

At a more detailed level, the outcome of the work must allow management to have end-to-end and service-level views of network performance, down to the granularity of service instance. Pre-supposing the arguments made in Section 4.2, it must also allow management to have a layer and technology independent view of the network, at least in the form of the three-dimensional topology, as defined in Section 2.

4. Analysis of the Problem

4.1. Argument For Consolidated Management

Multi-layer OAM actually presents two separate but inter-related issues. The first is technology dependency, at the same or different layers. The second is correlation of events between layers.

OAM mechanisms have a strong technology dependency because each technology (or layer) has its best suited OAM tools. Some of them provide rich functionality with one protocol, while the others provide each function with a different protocol. Today a variety of OAM tools have been developed by different Standards Development Organizations (SDOs) for Optical Transport Network (OTN), Synchronous Digital Hierarchy (SDH), Ethernet, MPLS, and IP networks.

However, orchestrating and coordinating OAM in multi-layer networks to provide better network visibility and efficient OAM operations is still a challenging issue since no mechanisms are defined, for example, to exchange performance and liveliness information between different layers. This means that the required coordination has to happen in the management function through communication with the managed entities.

The development of overlay networks, where one network is the client of another, adds to the magnitude of the problem. To take a specific example, in the Service Function Chaining (SFC) [I.D-ietf-sfc-problem-statement] environment, every Service Function (SF) may operate at a different layer and may use a different encapsulation scheme. When taking into account overlay technologies, the number of encapsulation options increases even more.

At this point, it is useful to recall the preliminary objectives stated in Section 3. To achieve end-to-end and service-level views of network performance requires that the management function be capable of receiving and reacting to related information from every transport segment at every layer in the network. This is a working definition of consolidated management.

A key issue with "management consolidation" is that it may include a requirement for management to interact with every technology used in the network on a per-technology basis either initially or when it has to follow up on detected problems by collecting detailed information. It is an architectural challenge beyond the scope of this document to determine whether consolidated management then becomes an aggregation of local managers of legacy type tied together by a coordination function, or whether simplifications are possible.

4.2. Argument For Layer and Technology Independent Management

The argument for consolidated management to have a layer and technology independent view of the network and supporting layers is two-pronged. The first argument is fairly straightforward and initially independent of architectural considerations. Some management functions are concerned solely with the topology of the network and supporting layers as represented by the three-dimensional topology defined in Section 2. These include network optimization, efficient enforcement of Traffic Engineering (TE) techniques including assurance of path diversity in one layer and over the complete hierarchy of layers, and fine-grained tweaking. Even in this case management action may require interaction with the managed elements at a technology-specific level, barring an alternative architectural solution.

The second argument for a layer and technology independent view involves considerably more substance than the first one. The three-dimensional topology would be a starting point for this view, but in addition it would include an abstracted view of service-affecting or potentially service-affecting events, identified by layer and reporting managed device. This allows management to correlate events in different layers and identify the devices from which it must seek further information or to which it must direct other requests, without being burdened with excess information. The intention is to ease root cause analysis and improve the ability to maintain end-to-end and service-level visibility.

Where this second version of a technology independent view is created is an architectural issue, beyond the scope of the present document. One possibility is that the work is all done in the "consolidated management" function, in which case the latter just becomes an aggregation of legacy technology-specific managers tied together by a coordination function, as mentioned above. A contrasting possibility is that the managed devices also support the abstraction, with a view to minimizing the amount of technology specific information and management actions the management function has to support.

4.3. Detailed Issues

4.3.1. Strong Technology Dependency For MIB Modules

OAM protocols rely heavily on the specific network technology they are associated with. For example, ICMPv6 [RFC4443] and LSP Ping [RFC4379] provide the same OAM functionality, path discovery, for IPv6 and MPLS Label Switched Path (LSP) technologies respectively.

SNMP MIB modules to manage these protocols were developed on a per OAM protocol basis. As a result, there was little reuse of MIB modules for other existing OAM protocols. To the extent that management operations are being redesigned in terms of YANG modules [RFC6020] over NETCONF [RFC6241], the opportunity exists to use the concept of layer and technology independent abstraction to extract the reusable parts, simplifying the work on the remainder.

4.3.2. Issues of Abstraction

In a multi-layer network, OAM functions are enabled at different layers and OAM information needs to be gathered from various layers independently. Without multi-layer OAM in place, it is hard for management applications to understand what information (e.g., Context, OAM functionalities) at different layers stands for and have a unified view of OAM information at different layers. A mechanism is required to provide this information to management.

The challenge is to abstract in a way that retains in the management plane as much useful information as possible while filtering the data that is not needed. An important part of this effort is a clear understanding of what information is actually needed. There is a close relationship between this issue and the issue already identified in the previous section.

4.3.3. OAM Interworking Issues

When multiple layer OAMs are used in the different parts of the network, two layer OAMs interworking at the boundaries need to be considered:

- o How one layer OAM in given part of the network interworks with another layer OAM in another part of the network operated by the same administrative entity through a consolidated management interface? e.g., E-LMI used in UNI interworks with Ethernet link OAM used on an IEEE 802.3 link in the same domain?

- o How one layer OAM interworks with another layer OAM in the same part of the network through a conssolidated management interface?

e.g., Ethernet OAM interworks with MPLS OAM in the same part of the network? In this case, Ethernet OAM and MPLS OAM are both supported by the same two managed devices in communication.

In these cases, mapping and notifications of defect states between different layer OAMs is required at the boundary nodes of the two parts of the network [RFC6310] [RFC7023] [I-D.ietf-l2vpn-vpws-iw-oam]. Management must provide the interworking function to establish dynamic mapping and translation, supervise defects, and suppress alarms. [Issue for debate. The original text from draft-ww-oamwg provides for a separate interworking function. To me, that violates the concept of consolidated management. Maybe this is a case of local versus consolidated management as discussed in Section 6 -- PTT as individual contributor]

4.3.4. Multiple (ECMP) Paths OAM Issue

Network devices typically use fields in the MAC or IP header or MPLS header and perform hash computations (e.g., 5-tuple hash consisting of IP protocol, source address, destination address, source port, and destination port) on these packet header fields to classify packets into flows and select the forwarding path for the flow among multiple equal cost paths, ECMP becomes more important when network overlay, service chain technology are introduced, e.g., in case of multi-instances of the same service function is invoked for a given chain to provide redundancy, how 5-tuple hash is used based on contents in the outer headers and inner encapsulated packet.

Multiple path OAM requires that Connectivity Check and Continuity Check must follow the same path as the data traffic (e.g., TCP traffic and UDP traffic). Overlay encapsulation allows OAM data to piggyback packets, in the way record route is used in IPv4 options. However, there is no standard way to exercise end to end continuity and connectivity verification that covers all of ECMP paths in the IP networks. Such a standard is desirable.

5. Problem Statement

OAM functions are used heavily during service and network life-cycle. Today, OAM management requires expertise due to technology dependency despite the similarity in functions (adding to CAPEX and OPEX). Troubleshooting is cumbersome due to protocol variety and lack of multi- layer OAM. This requires expertise and long troubleshooting cycles (OPEX). Last but not least, today’s various management interfaces make it difficult to accept and introduce new protocols and technologies

There is value in attempting to define an architecture for consolidated management that may reasonably be argued to meet the objectives stated in Section 3. If this attempt succeeds, it can be followed up with a gap analysis, which in turn will define a further program of standardization.

At the detailed level, Section 4.3.1 and Section 4.3.2 deal with the matter of abstraction and its relationship to the specification of YANG modules. This is work beyond the initial definition of architecture and awaits justification and prioritization by the gap analysis. A similar consideration relates to the solution to the ECMP problem.

The remaining issue is the OAM interworking issue identified in Section 4.3.3. This is architectural in nature, and should be addressed by the proposed work on architecture.

6. Considerations For the Work On Architecture

Definition of an architecture for consolidated management is beyond the scope of the present document. This section instead provides considerations that should be taken into account when defining such an architecture.

6.1. What the Architecture Must Define

This section is a discussion in the nature of a very general use case rather than a discussion of functions and entities. However, as a preliminary remark, the architecture must be thought through for all five of the FCAPS areas (fault, configuration, accounting, performance, and security management). RFC 5706 Section 3, while nominally directed to protocol design, reviews operational issues associated with each of these areas.

To begin with, previous analysis (Section 4.2) has indicated that the CMgmtE Section 2 needs to work with a view of network topology that is layer and technology independent in order to achieve the objectives stated in Section 3. Two questions immediately come to mind: where is this view prepared, taking account of the limited processing power of network devices in particular, and what model is used to present the topology to the CMgmtE? Of course, these questions are evaded if the architecture makes the CMgmtE responsible for creating the abstracted topology from data gathered from the LMgmtEs and/or managed entities Section 2 within its scope.

Note that from the end-to-end point of view multiple network topologies will typically exist in the network at one time, possibly down to the granularity of a service instance. The relationship of the scope of a CMgmtE to the set of available topologies is subject to the condition that it has end-to-end and service-level views of all paths between the endpoints within its scope, and is otherwise undefined.

The CMgmtE must be aware of all of the LMgmtEs and/or managed entities within its scope. The architecture must define how the CMgmtE identifies the correct sequence of these entities along a path in a given layer, and similarly, must identify the correct ordering of layers from bottom to top. In effect, the CMgmtE requires a three-dimensional topological view of the data plane maintenance infrastructure. Entity identification may be implicit in this work. Note that management actions may alter this topology (e.g., for routine maintenance or installation of new equipment).

The next issue is how the CMgmtE and the other entities discover each other. Bound up in this is the issue of trust. This bootstrapping problem is a hard one, constantly recurring in IETF work but never yet solved. The architecture work will have to come to its own conclusions on this topic.

Where correlation of events from different layers and transport segments is done is not an issue. By definition it can be done only by the CMgmtE. The architecture must decide whether the necessary data gathering is done as required or continuously.

As a final point, the architecture must specify how an existing network evolves from legacy operation to the target architecture. The existing network will have LMgmtEs in place. The question is whether the CMgmtE simply replaces them or communicates with them. If it simply replaces them, the architecture must define (in an operational considerations section) how testing of the new management configuration takes place before cutover. Considerations of data continuity during cutover should also be addressed.

The above is not an exhaustive list of considerations, but should give a good start to the architectural work.

7. Security Considerations

The architectural work must include work on the security architecture of the whole system. Beyond that, potential future work on individual interfaces must include the appropriate security mechanisms within the architectural framework. The present document cannot be more specific by its nature.

8. IANA Considerations

This document does not require any action from IANA.

9. Contributors

In the understanding of the Editor, the following individuals (listed in alphabetical order by last name) contributed text to or strongly influenced the development of versions of draft-ww-opsawg-multi-layer-oam, from which this document was derived:

- Sam Aldrin mailto:aldrin.ietf@gmail.com, Huawei USA;
- Mohamed Boucadair mailto:mohamed.boucadair@orange.com, France Telecom;
- Huub van Helvoort mailto:huubatwork@gmail.com, Hai Gaoming BV;
- Tom Herbert mailto:therbert@google.com, Google;
- Pradeep Jain mailto:pradeep@nuagenetworks.net, Nuage Networks;
- Daniel King mailto:daniel@olddog.co.uk, Olddog Consulting;
- Greg Mirsky mailto:gregory.mirsky@ericsson.com, Ericsson;
- Ravi Shekhar mailto:rshekhar@juniper.net, Juniper.

10. Acknowledgements

The authors would like to thank Jan Lindblad, Tissa Senevirathne, Yuji Tochio, Ignas Bagdonas, Eric Osborne, Rob Shakir, Georgis Karagiannis, Melinda Shore and Jouni Korhonen for their reviews and suggestions.

11. References

11.1. Normative References

[RFC6291] Andersson, L., van Helvoort, H., Bonica, R., Romascanu, D., and S. Mansfield, "Guidelines for the Use of the "OAM" Acronym in the IETF", BCP 161, RFC 6291, June 2011.

[RFC7276] Mizrahi, T., Sprecher, N., Bellagamba, E., and Y. Weingarten, "An Overview of Operations, Administration, and Maintenance (OAM) Tools", RFC 7276, June 2014.

11.2. Informative References

[I-D.ietf-l2vpn-vpws-iw-oam]
Aissaoui, M., Busschbach, P., Allan, D., Morrow, M., and T. Nadeau, "OAM Procedures for VPWS Interworking", draft-ietf-l2vpn-vpws-iw-oam-04 (work in progress), March 2014.

[I.D-ietf-sfc-problem-statement]
Quinn, P., Guichard, J., and S. Surendra, "Network Service Chaining Problem Statement (Work in progress)", ID draft-ietf-sfc-problem-statement, August 2014.

[RFC4379] Kompella, K. and G. Swallow, "Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures", RFC 4379, February 2006.

[RFC4443] Conta, A., Deering, S., and M. Gupta, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 4443, March 2006.

[RFC5706] Harrington, D., "Guidelines for Considering Operations and Management of New Protocols and Protocol Extensions", RFC 5706, November 2009.

[RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.

[RFC6123] Farrel, A., "Inclusion of Manageability Sections in Path Computation Element (PCE) Working Group Drafts", RFC 6123, February 2011.

[RFC6241] Enns, R., Bjorklund, M., Schoenwaelder, J., and A. Bierman, "Network Configuration Protocol (NETCONF)", RFC 6241, June 2011.

[RFC6310] 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.

[RFC7023] Mohan, D., Bitar, N., Sajassi, A., DeLord, S., Niger, P., and R. Qiu, "MPLS and Ethernet Operations, Administration, and Maintenance (OAM) Interworking", RFC 7023, October 2013.

Authors' Addresses

Qin Wu
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: bill.wu@huawei.com

Mishael Wexler
Huawei
Riesstr. 25
Munich 80992
Germany

Email: mishael.wexler@huawei.com

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

Email: dromasca@avaya.com

T. Taylor (editor)
PT Taylor Consulting
Ottawa
Canada

Email: tom.taylor.stds@gmail.com

Operations and Management Area Working Group D. King
Internet-Draft Lancaster University
Intended status: Informational M. Boucadair
Expires: March 25, 2014 France Telecom
S. Aldrin
Huawei USA
G. Mirsky
Ericsson
Mishael Wexler
Huawei
September 25, 2014

Use Cases and Requirements for Layer Independent OAM Management in multi-layer environments
draft-king-opsawg-lime-multi-layer-oam-use-case-00

Abstract

As operators deploy and operate multi-layer networks and diverse transport technologies, layer independent Operations, Maintenance & Administration Management (OAM) would be beneficial for monitoring and troubleshooting operations of network and service infrastructure.

This document identifies and discusses the key use-cases and high-level requirements for layer independent Operations, Administration, and Maintenance management to facilitate operations and maintenance in multi-layer and multi-domain networks utilizing a wide variety of heterogeneous networking technologies.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document discusses use-cases for layer independent OAM management that would interface to multi-layer or multi-domain networks to cover various heterogeneous networking technologies.

As operators and providers (e.g., network operators, data center operators, service providers, etc.) continue to deploy and operate multi-layer networks using a wide range of transport technologies, layer independent OAM Management for stitching different layer OAMs is desirable to minimise operational complexity and simplifying O&M (OAM and O&M are used as specified in [RFC6291]).

This document discusses Layer Independent OAM in Multi-Layer Environment (LIME), and is intended to:

- outline use cases for layer independent OAM management and highlight the issues encountered with existing OAM protocols;

- o discuss OAM requirements for when designing and deploying new technologies;

- o outline eixsting technologies to facilitate layer independent OAM management, including MEF work, ITU-T work, IETF related work;

- o discuss how OAM might be configured via a unified management interface:

    * Establishment of OAM Entities(e.g., MEG, ME,MIP,MEP) and Functions(e.g.,CC,CV)

    * Adjustment of OAM Parameters

    * Deleting OAM Entities

- o highlight a generic OAM Management model that may be applied to various OAM technologies:

    * Defining common objects and relationships model for various technologies

    * Defining a common set of methods/calls to use for the various functions

    * Defining a common set of attributes per object

    * Defining a common set of alarms and notifications

  Specific OAM technology models will augment the basic OAM management model defined by the LIME Group.

- o detail OAM fault management(e.g., fault location, path discovery) data model using layer independent OAM Management:

    * Propose means to help during service diagnosis; these means may rely on filtering information so that time recovery can be optimized. A typical example would be efficient root cause analysis that is fed with input from various layers.

    * Propose means that would help to optimize a network as a whole instead of the monolithic approach that is specific to a given layer. For example, investigate means that would help in computing diverse and completely disjoint paths, not only at the overlay but also at the underlay.

- o discuss the security model for layer independent OAM management:

    * Propose means to avoid leaking OAM information to no authorized entities, and to avoid altering OAM information exposed by each

layer.

* Propose means to ensure reliability of OAM information exposed by each layer.

These requirements are not frozen; further discussion is required to target key issues and scope the work to be conducted within IETF accordingly.

The problem statement and architecture is discussed in [LIME-PS].

2. Terminology

2.1. Conventions used in this document

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

2.2. Acronyms and Abbreviations

LIME - Layer Independent OAM in Multi-Layer Environment

OAM - Operations, Administration, and Maintenance

O&M - OAM and Management

ABNO -Application Based Network Operation

MEG - Maintenance Entity Group [RFC6371]

MEP - Maintenance Entity Group End Point [RFC6371]

MIP - Maintenance Entity Group Intermediate Point [RFC6371]

CC -Continuity Check [RFC7276]

CV - Connectivity Verification [RFC7276]

CFM -Connectivity Fault Management

EFM - Ethernet In the First Mile

BFD -Bidirectional Forwarding Detect

LBM -Loopback Message

LBR -Loopback Reply

LTM -Linktrace Message

LTR -Linktrace Reply

OSS -Operation Support System

NMS -Network Management System

SFC -Service Function Chain [I-D.ietf-sfc-problem-statement]

SF - Service Function [I-D.ietf-sfc-problem-statement]

TES -Tenant End System [I-D.ietf-nvo3-framework]

NVO3 -Network Virtualization Overlay [I-D.ietf-nvo3-overlay-problem-statement]

WAN -Wide Area Network

VXLAN -Virtual eXtensible Local Area Network [RFC7348]

NVGRE -Network Virtualization using Generic Routing Encapsulation [I-D.sridharan-virtualization-nvgre]

PW -Pseudowire

LSP -Label Switching Path

MPLS -Multi-Protocol Label Switching

3. Layer Independent OAM Management Use Cases

3.1. Multi-layer multi-region OAM Consolidation in the Management Plane

A multi-layer multi-region network will often require data traffic between two customer edges to be transported across two regions, as illustrated in figure 2. In this scenario the same domain and multiple layer OAMs(i.e.,PW OAM,end to end LSP OAM, segment LSP OAM) are used.

For PW OAM is used at the customer level for monitoring the end-to -end connection between the two Provider edges(PEs), while end to end LSP OAM and segment LSP OAM is used at the provider level for monitoring the segment LSP and end to end LSP respectively. A segment is between MEPs and The OAM in each segment is independent of any other segment.

```
           +--------------------------------------------------+
           |                Carrier  1                        |
           |+---------------------+    +---------------------+|
           || Region1             |    | Region2             ||
   +----+  || +----+ +-+  +----+  |    | +----+   +-+  +----+||   +----+
   | PE ------| PE --|P|--| PE ----------| PE ----|P|--| PE ------| PE |
   +----+  || +----+ +-+  +----+  |    | +----+   +-+  +----+||   +----+
           ||                     |    |                     ||
           |+---------------------+    +---------------------+|
           +--------------------------------------------------+

                        End to end LSP OAM
     o----------D----------------------------------------D----------D
                                                           Segment LSP
                     Carrier 1 LSP OAM segment                 OAM
                o------------D-------------D-------------o-o--------o
               Carrier1 Region 1          Carrier1 Region 2
                LSP OAM segment            LSP OAM segment

                o----D-------o             o-------D------o
                     o  Maintenance Endpoint(MEP)
                     D  Maintenance Intermediary point (MIP)
```

Figure 1: Multi-Region Multi-Layer OAM

With single OSS/NMS in the management plane, customized service diagnose can be provided, e.g.,

- o initiating tests on any layer in the multi-layer network

- o initiating test on any segment of the end to end path

- o initiate test on end to end path

- o check end-to-end connectivity test results across a multi-layer network even when each layer runs a different technology.

3.2. Multiple layer OAMs stitching in different part of the network

Figure 2 illustrates a multi-layer network in which data traffic between two access nodes is transported through access section between access node(AN) and aggregation node(AGG Node) and aggregation section between aggregation node and edge node and even core section from edge node to Internet or WAN. EFM OAM is used at the access section for monitoring the access connection between the access node and aggregation node, CFM OAM and IP OAM is used at the aggregation section for monitoring end to end connection between aggregation node and edge node. BFD is used at the core section for monitoring end to end connection from edge node to Internet or WAN.

```
        |          |             |                   |
        ----EFM---------CFM/IP-----------BFD---------|
        |          |             |                   |
    +---|----------|---+         |                   |
    | +----+    +-----+|         |                   |
    | | AN |    |AGG  +---+      |                   |
    | +----+    |Node ||  |      |                   |
    |           +-----+|  |  +------+   +-----+   /-----\
    +-----------------+  |  | Edge +---+Core |++|Internet|
                          ---| Node |   |Node |  |       |
    +-----------------+  |  +------+   +-----+   \-----/
    | ------    |AGG  ||  |
    | | AN |    |Node +---+
    | +----+    +-----+|  |
    | +----+    +-----+|  |
    | | AN |    |AGG  ||  |
    | +----+    |Node +---- +-------+  +-----+    /----\
    |           +-----+|  | | Edge  +--+Core |+-+| WAN  ||
    +-----------------+  --- Node  |  |Node |   |      |
    +-----------------+  | +-------+  +-----+    \----/
    |           +-----+|  |       |                   |
    | +----+    |AGG  +---+      |                   |
    | | AN |    |Node ||         |                   |
    | +----+    +-----+|         |                   |
    +---|----------|---+         |                   |
        |          |             |                   |
        ----Access---Aggregation-| -----Core----------
        |          |             |                   |
```

Figure 2

With single OSS/NMS in the management plane, different layer OAM at the different part of the network can be stitching together to provide unified view for network problem reporting by consuming all status reports from the network, aggregating them, correlating them.

In addition, a user who wishes to issue a IP Ping Command or use connectivity verification command in the Ethernet layer can do so in the same manner regardless of the underlying protocol or transport technology. This can be achieved by invoking IP Ping Command or connectivity verification command through uniform interface between management plane and data plane. Consider a scenario where an IP ping to Edge node B from Aggregation node A failed. Between AGG node A and Edge Node B there are IEEE 802.1 [IEEE-802.1Q] bridges a,b and c. Let’s assume a,b and c are using [IEEE-802.1ag] CFM. IP layer Ping can be invoked using uniform interface between single OSS/NMS and AGG node A. Upon detecting IP layer ping failure, the user may wish to "go down" to the Ethernet layer and issue the corresponding fault verification (LBM/LBR) and fault isolation (LTM/LTR) tools, using the same uniform interface used by IP Layer Ping.

3.3. Stitching OAM at layer requiring L4 to L7 service

In Service Function Chain ([I-D.ietf-sfc-problem-statement]), the service packets are steered through a set of Service Function distributed in the network. Overlay technologies and other tunneling techniques can be used to stitch these Service Function Nodes in order to form end to end path (see Figure 3).

```
                              +---------+
                              | Single  |
       |----------------------| OSS/NMS -------------------------
       |                      +---------+                        |
       |                                                         |
       |                 ----------------------                  |
       |          /////--                      --\\\\\           |
       |     /////                                    \\\\\      |
       | ////  |->SF1 SF2-+     SF3----+  +->Proxy-SF4    \\\\ |
       |/      |   |  ^   |       ^     |  |                   \|
               |   V  |   |       |     |  |
   +-------+   |  +---+-+ |     +-+---+ |  |  +-----+        +-----+
   | SFC   |   |--|     +<+     |     |<-  |->|     |        |     |
   |Ingress|------ NE-a  -------+NE-b +-------  NE-c --------+NE-d |
   | Node  |      |     |       |     |       |     |        |     |
   +-------+      +-----+       +-----+       +-----+        +-----+

         \\\\                                                 ////
             \\\\\                                      /////
                  \\\\\--                         --/////
                         ----------------------
    SF OAM
                  o   o           o             o             D
    SFC  OAM
    o-------------D---D-----------D-------------D

    NVO3 OAM
    o---------------o----------------D----------D----------------o

    Link OAM
              o--o---o--o-       o---o     o--o--o--o

                   o  Maintenance Endpoint(MEP)
                   D  Maintenance Intermediary point (MIP)
      Layer7-  SF1  ---------------------------------------------------
      Layer6-----------------------F4 --------------------------------
      Layer5------------ SF3------------------------------------------
      Layer4---SF2 --------------------------------- ----------------
```

Figure 3: Stitching OAM at layer requiring L4 to L7 service

In figure 3, Link OAM is used between any two adjacent SFs hosted in the same service node in the SFC layer or between a SF and the service node hosting that SF. NVO3 OAM is used between SFC ingress node and NVO3-enabled Network element or any two NVO3-enabled network element in the SFC domain. SFC OAM is used between a set of Service Functions belong to the same service function chain in the SFC domain. SF OAM is used between SF and SF Controller.

When the service packet enters into the network, OAM information needs to be imposed by ingress node of the network into the OAM packet (e.g., packet header extension or TLV extension in the overlay header) and pass through the network in the same path as the service traffic and processed by a set of Service Functions that are hosted in Service Nodes and located in different layers requiring L4-L7 service.

When any Service Nodes or any service segment between two Service Nodes fails to deliver user traffic, there is a need to provide a tool that would enable users to detect such failures (e.g., fault element in the path), and a mechanism to isolate faults.

In case of several SFs co-located in the same Service Node, the packet is processed by all SFs in the Service Node, Once the packet is successfully handled by one SF, the packet is forwarded to the next SF that is in the same Service Node.

When the packet leaves the network, the OAM information needs to be stripped out from the packet.

To provide unified view of OAM information from different layers and different segment of the Service Function Path, these OAM information needs to gathered from various layer using different encapsulation and tunneling techniques and abstracted and provided to the management application via the uniform management interface.

As indicated in [I-D.boucadair-sfc-requirements], the following OAM functions are to be supported:

o Support means to verify the completion of the forwarding actions until the SFC Border Node is reached (see Section 3.4.1 of [RFC5706]).

o Support means to ensure coherent classification rules are installed in and enforced by all the Classifiers of the SFC-enabled domain.

o Support means to correlate classification policies with observed forwarding actions.

o Support in-band liveliness and functionality checking mechanisms for the instantiated Service Function Chains and the Service Functions that belong to these chains.

Other service diagnosis and troubleshooting requirements are discussed in [I-D.boucadair-sfc-requirements].

3.4. Multi-Operator OAM Stitching

Multi-operator networks can be abstracted, virtualized and shared by several tenants. A tenant has an end-to-end view of its virtual network. Figure 5 illustrates an example of multi-layer multi-operator network in which data traffic between two tenant systems is transported across three operators.

```
                                +--------+
   -----------------------------+ Tenant +------------------------------
  |                             |        |                             |
  |                             +--------+                             |
  |                                |                                   |
  |                          + - - - - - - +                           |
  |                          | Abstraction |                           |
  |        - - - --- - - - - | Layer (opt.)| - - - - - - - -           |
  |       |                  + - - - - - - +                |          |
  |       |                        |                       |          |
  |     +--------+              +--------+             +--------+      |
  |     + Single +              + Single +             + Single +      |
  |     |OSS/NMS |              |OSS/NMS |             |OSS/NMS |      |
  |     +--------+              +--------+             +--------+      |
  |      ---------               ---------              ---------      |
  |   ///         \\\         ///        \\\        ///          \\\   |
  |  /               \       /              \      /                \  |
  | |                |     |                  |   |                  | |
 +-----+           +-----+ +-----+       +----+  +-----+         +-----+
 | A1  |           | A2  | | B1  |       | B2 |  | C1  |         |  C2 |
 +-----+           +-----+ +-----+       +----+  +-----+         +-----+
    |                 |     |                 |    |                 |
     \               /       \               /      \               /
      \\\         ///         \\\         ///        \\\         ///
         ---------               ---------              ---------


     Oper A           Oper B     Oper C
     o-------------D----------D---------------D----D-----------------o
```

Figure 4: Multi-Operator OAM Stitching

Each operator is using a different management system and is handling only a segment of the whole end-to-end service. Each operator can also use a different technology to transport the clients over its segment. Within one operator region, multi-layers acn be used, e.g. IP over WDM to transport L3VPN service.

The tenants has to view an end-to-end OAM model via abstraction of each region and/or using abstraction layer between tenants and the OSS/NMSs.

4. Requirements

This section identifies high-level requirements to fulfill layer independent OAM management in Multi-layer Environment to support various use cases discussed in the previous sections.

- o The interfaces between the management entity and each Managed device in one administrative domain SHOULD support standards-based abstraction with a common information/data model.

- o The management entity should be able to create a single unified view of OAM information that is common to various layers, various segment of the same domain.

- o The management entity should provide an unified management interface for multiple OAM technologies that will expose a common set of management interface capabilities for different OAM technologies (e.g. Continuity Check(CC),Connectivity Verification(CV)). The management interface implementation will convert the defined common management capabilities to the OAM technology specific operations.

- o The management entity should Model OAM operations management and represent OAM information and mechanisms in the same way using YANG at the management plane to provide consistent configuration, reporting, and presentation for the OAM mechanisms. Specific OAM technology models will augment the basic OAM management model defined by the LIME Group.

- o The following capability for layer independent OAM management entity should be supported:

  * Support customized service diagnostic.

  * Support diagnose the availability of a end-to-end path.

  * Support diagnose the availability of a segment Path that is sub-path of end to end path.

  * Support verification on the correct value of Path ID between any two pair of overlay nodes or any two pair of service nodes.

* Support out-band liveliness and functionality checking mechanisms for the overlay node or service node.

5. IANA Considerations

This memo includes no request to IANA.

6. Security Considerations

TBD.

7. References

7.1. Normative References

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

7.2. Informative References

[I-D.boucadair-sfc-requirements]
Boucadair, M., Jacquenet, C., Jiang, Y., Parker, R., Pignataro, C., and K. Kengo, "Requirements for Service Function Chaining (SFC)", draft-boucadair-sfc-requirements-05 (work in progress), July 2014.

[I-D.ietf-sfc-problem-statement]
Quinn, P. and T. Nadeau, "Service Function Chaining Problem Statement", draft-ietf-sfc-problem-statement-10 (work in progress), August 2014.

[IEEE-802.1Q]
IEEE 802.1Q-2011, "IEEE standard for local and metropolitan area networks: Media access control (MAC) bridges and virtual bridged local area networks", August 2011.

[IEEE-802.1ag]
IEEE 802.1ag-2007, "IEEE Standard for Local and Metropolitan Area Networks Virtual Bridged Local Area Networks Amendment 5: Connectivity Fault Management", December 2007.

[LIME-PS] Taylor, T. and Q. Wu, "Problem Statement and Architecture for Layer-Independent OAM in Multi-Layer Environment", ID draft-edprop-opsawg-multi-layer-oam-ps-01, (work in progress).

[RFC5706] Harrington, D., "Guidelines for Considering Operations and Management of New Protocols and Protocol Extensions", RFC 5706, November 2009.

[RFC6291] Andersson, L., Helvoort, H., Bonica, R., Romascanu, D., and S. Mansfield, "Guidelines for the Use of the "OAM" Acronym in the IETF", RFC 6291, June 2011.

[RFC6371] Busi, I. and D. Allan, "Operations, Administration, and Maintenance Framework for MPLS-Based Transport Networks", RFC 6371, September 2011.

[RFC7276] Mizrahi, T. et al., "An Overview of Operations, Administration, and Maintenance (OAM) Tools," June 2014

[RFC7348]

[I-D.sridharan-virtualization-nvgre]
Sridharan, M., Greenberg, A., Venkataramaiah, N., Wang, Y., Duda, K., Ganga, I., Lin, G., Pearson, M., Thaler, P., and C. Tumuluri, "NVGRE: Network Virtualization using Generic Routing Encapsulation", draft-sridharan-virtualization-nvgre-05 (work in progress).

Authors' Addresses

Daniel King
Lancaster University
UK

Email: d.king@lancaster.ac.uk

Mohamed Boucadair
France Telecom
Rennes 35000
France

Email: mohamed.boucadair@orange.com

Sam Aldrin
Huawei Technologies USA
2330 Central Expressway
NSanta Clara, CA 95051
USA

Email: aldrin.ietf@gmail.com

Greg Mirsky
Ericsson

Email: gregory.mirsky@ericsson.com

Mishael Wexler
Huawei
Riesstr. 25
Munich 80992
Germany
Email: mishael.wexler@huawei.com

Contributors' Addresses

Qin Wu
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China
Email: bill.wu@huawei.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: February 5, 2016

T. Senevirathne
N. Finn
D. Kumar, Ed.
S. Salam
Cisco
Q. Wu, Ed.
M. Wang
Huawei

August 4, 2015

Generic YANG Data Model for Operations, Administration, and Maintenance (OAM)

draft-tissa-lime-yang-oam-model-06

Abstract

This document presents base YANG Data model for OAM. It provides a protocol-independent and technology-independent abstraction of key OAM constructs. Based model presented here can be extended to include technology specific details. This is leading to uniformity between OAM technologies and support nested OAM workflows (i.e., performing OAM functions at different layers through a unified interface).

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Operations, Administration, and Maintenance (OAM) are important networking functions that allow operators to:

1. Monitor networks connections (Connectivity Verification, Continuity Check).

2. Troubleshoot failures (Fault verification and localization).

3. Monitor Performance

An overview of OAM tools is presented at [RFC7276].

Ping and Traceroute [RFC792], [RFC4443] are well-known fault verification and isolation tools, respectively, for IP networks. Over the years, different technologies have developed similar tools for similar purposes.

[IEEE802.1Q] Connectivity Fault Management is a well-established OAM standard that is widely adopted for Ethernet networks. ITU-T [Y.1731][Y.1731], MEF Service OAM, MPLS-TP [RFC6371], TRILL [RFC7455][RFC7455] all define OAM methods based on manageability frame work of [IEEE802.1Q] [IEEE802.1Q]CFM.

Given the wide adoption of the underlying OAM concepts defined in [IEEE802.1Q][IEEE802.1Q] CFM, it is a reasonable choice to develop the unified management framework based on those concepts. In this document, we take the [IEEE802.1Q][IEEE802.1Q] CFM model and extend it to a technology independent framework and build the corresponding YANG model accordingly. The YANG model presented in this document is the base model and supports generic continuity check, connectivity verification and path discovery. The generic YANG model for OAM is designed such that it can be extended to cover various technologies. Technology dependent nodes and RPC (remote process call) commands are defined in technology specific YANG models, which use and extend the base model defined here. As an example, VXLAN uses source UDP port number for flow entropy, while MPLS [RFC4379] uses IP addresses or the label stack for flow entropy in the hashing for multipath selection. To capture this variation, corresponding YANG models would define the applicable structures as augmentation to the generic base model presented here. This accomplishes three purposes: first it keeps each YANG model smaller and manageable. Second, it allows independent development of corresponding YANG models. Third, implementations can limit support to only the applicable set of YANG models. (e.g. TRILL RBridge may only need to implement Generic model and the TRILL YANG model).

All implementations that follow the YANG framework presented in this document MUST implement the generic YANG model presented here.

The YANG data model presented in this document is generated at the management layer. Encapsulations and state machines may differ according to each OAM protocol. A user who wishes to issues a Ping command or a Traceroute or initiate a performance monitoring session can do so in the same manner regardless of the underlying protocol or technology or specific vendor implementation.

As an example, consider a scenario where an IP ping from device A to Device B failed. Between device A and B there are IEEE 802.1 bridges a,b and c. Let’s assume a,b and c are using [IEEE802.1Q] CFM. A user upon detecting the IP layer ping failures may decide to drill down to the Ethernet layer and issue the corresponding fault verification (LBM) and fault isolation (LTM) tools, using the same API. This ability to go down to the same portion of path atlower layer for Fault localization and troubleshooting is referred to as "nested OAM workflow" and is a useful concept that leads to efficient network troubleshooting and maintenance. The OAM YANG model presented in this document facilitates that without needing changes to the underlying protocols.

2. Conventions used in this document

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

The following notations are used within the data tree and carry the meaning as below.

Each node is printed as:

```
<status> <flags> <name> <opts> <type>

<status> is one of:
     +  for current
     x  for deprecated
     o  for obsolete


<flags> is one of:

    rw for configuration data
    ro for non-configuration data
    -x for rpcs
    -n for notifications


<name> is the name of the node
```

If the node is augmented into the tree from another module, its name is printed as <prefix>:<name>.

<opts> is one of:

```
? for an optional leaf or choice
! for a presence container
* for a leaf-list or list
[<keys>] for a list's keys
```

<type> is the name of the type for leafs and leaf-lists

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying RFC-2119 significance.

2.1. Terminology

CCM - Continuity Check Message [IEEE802.1Q].

ECMP - Equal Cost Multipath.

LBM - Loopback Message [IEEE802.1Q].

MP - Maintenance Point [IEEE802.1Q].

MEP - Maintenance End Point [RFC7174] [IEEE802.1Q] [RFC6371].

MIP - Maintenance Intermediate Point [RFC7174] [IEEE802.1Q] [RFC6371].

MA - Maintenance Association [IEEE802.1Q] [RFC7174].

MD - Maintenance Domain [IEEE802.1Q]

MTV - Multi-destination Tree Verification Message.

OAM - Operations, Administration, and Maintenance [RFC6291].

TRILL - Transparent Interconnection of Lots of Links [RFC6325].

CFM - Connectivity Fault Management [RFC7174] [IEEE802.1Q].

RPC - Remote Process Call.

CC - Continuity Check [RFC7276]. Continuity Checks are used to verify that a destination is reachable and therefore also referred to as reachability verification.

CV - Connectivity Verification [RFC7276].Connectivity Verifications are also referred to as path verification and used to verify not only that the two MPs are connected, but also that they are connected through the expected path, allowing detection of unexpected topology changes.

3. Architecture of Generic YANG Model for OAM

In this document we define a generic YANG model for OAM. The YANG model defined here is generic such that other technologies can extend it for technology specific needs. The Generic YANG model acts as the root for other OAM YANG models. This allows users to traverse between OAM of different technologies at ease through a uniform API set. This is also provides a nested OAM workflow. Figure 1 depicts the relationship of different OAM YANG models to the Generic YANG Model for OAM. Some technologies may have different sub-technologies. As an example, consider Network Virtualization Overlays. These could employ either VXLAN or NVGRE as encapsulation. The Generic YANG model for OAM provides a framework where technology-specific YANG models can inherit constructs from the base YANG models without needing to redefine them within the sub-technology.

Figure 1 depicts relationship of different YANG modules.

```
                            +-+-+-+-+-+
                            |  gen    |
                            |OAM YANG |
                            +-+-+-+-+-+
                                 |
                                 O
                                 |
      +---------------------------------------------------------+
      |              |               |         |                |
  +-+-+-+-+-+-+   +-+-+-+-+-+-+   +-+-+-+-+-+-+ +-+-+-+-+-+-+      +-+-+-+-+-+-+
  | TRILL    |   | NVO3     |   | MPLS     | | IP       | . . .|  foo     |
  |OAM YANG |   |OAM YANG |   |OAM YANG | |OAM YANG |      |OAM YANG |
  +-+-+-+-+-+-+   +-+-+-+-+-+-+   +-+-+-+-+-+-+ +-+-+-+-+-+-+      +-+-+-+-+-+-+
        |             |             |            |              |
        |       +-+-+-+-+-+-+   +-+-+-+-+-+-+          |       +-+-+-+-+-+-+
        |       | NVO3     |   | MPLS     |          |  . . .|  foo     |
        |       |sub tech |   |sub tech |          |       |sub tech |
        |       +-+-+-+-+-+-+   +-+-+-+-+-+-+          |       +-+-+-+-+-+-+
        |             |             |            |              |
        |             |             |            |              |
      +-------------------------------------------------------------+
      |                       Uniform API                           |
      +-------------------------------------------------------------+
```

Relationship of OAM YANG model to generic (base) YANG model

4. Overview of the OAM Model

In this document we adopt the concepts of the [IEEE802.1Q] CFM model and structure it such that it can be adapted to different technologies.

At the top of the Model is the Maintenance Domain. Each Maintenance Domain is associated with a Maintenance Name and a Domain Level.

Under each Maintenance Domain there is one or more Maintenance Association (MA). In IP, the MA can be per IP Subnet, in NVO3 this can be per VNI and for TRILL this can be per Fine-Grained Label or for VPLS this can be per VPLS instance.

Under each MA, there can be two or more MEPs (Maintenance Association End Points). MEPs are addressed by their respective technology specific address identifiers. The YANG model presented here provides flexibility to accommodate different addressing schemes.

In the vertical direction orthogonal to the Maintenance Domain, presented are the commands. Those, in YANG terms, are the rpc commands. These rpc commands provide uniform APIs for continuity check,connectivity verification, path discovery and their equivalents as well as other OAM commands.

The generic YANG model defined here does not require explicit configuration of OAM entities prior to using any of the OAM tools.The OAM tools used here are limited to OAM toolset specified in section 5.1 of [RFC7276]. Users of Ping and Traceroute tools within IP devices are expecting ability to use OAM tools with no explicit configuration. In order to facilitate zero- touch experience, this document defines a default mode of OAM. The default mode of OAM is referred to as the Base Mode and specifies default values for each of model parameters, such as Maintenance Domain Level, Name of the Maintenance Association and Addresses of MEP and so on. The default values of these depend on the technology. Base Mode for TRILL is defined in [RFC7455]. Base mode for other technologies such as NVO3, MPLS and future extensions will be defined in their corresponding documents.

It is important to note that, no specific enhancements are needed in the YANG model to support Base Mode. Implementations that comply with this document, by default implement the data nodes of the applicable technology. Data nodes of the Base Mode are read-only nodes.

4.1. Maintenance Domain (MD) configuration

The container "domains" is the top level container within the gen-oam module. Within the container "domains", separate list is maintained per MD. The MD list uses the key MD-name-string for indexing. MD-name-string is a leaf and derived from type string. Additional name formats as defined in [IEEE802.1Q] or other standards can be included by association of the MD-name-format with an identity-ref. MD-name-format indicates the format of the augmented MD-names. MD-name is presented as choice/case construct. Thus, it is easily augmentable by derivative work.

```
module: ietf-gen-oam
  +--rw domains
     +--rw domain* [technology MD-name-string]
        +--rw technology        identityref
        +--rw MD-name-string    MD-name-string
        +--rw MD-name-format?   identityref
        +--rw (MD-name)?
        |  +--:(MD-name-null)
        |     +--rw MD-name-null?     empty
        +--rw md-level          MD-level  .
```

Snippet of data hierarchy related to OAM domains

4.2. Maintenance Association (MA) configuration

Within a given Maintenance Domain there can be one or more Maintenance Associations (MA). MAs are represented as a list and indexed by the MA-name-string. Similar to MD-name defined previously, additional name formats can be added by augmenting the name-format identity-ref and adding applicable case statements to MA-name.

```
module: ietf-gen-oam
   +--rw domains
      +--rw domain* [technology MD-name-string]
         .
         .
         +--rw MAs
            +--rw MA* [MA-name-string]
               +--rw MA-name-string       MA-name-string
               +--rw MA-name-format?      identityref
               +--rw (MA-name)?
               |  +--:(MA-name-null)
               |     +--rw MA-name-null?        empty
```

Snippet of data hierarchy related to Maintenance Associations (MA)

4.3. Maintenance Endpoint (MEP) configuration

Within a given Maintenance Association (MA), there can be one or more Maintenance End Points (MEP). MEPs are represented as a list within the data hierarchy and indexed by the key MEP-name.

```
module: ietf-gen-oam
   +--rw domains
      +--rw domain* [technology MD-name-string]
         +--rw technology        identityref
         .
         .
         +--rw MAs
            +--rw MA* [MA-name-string]
               +--rw MA-name-string       MA-name-string
               .
               .
               +--rw MEP* [mep-name]
               |  +--rw mep-name             MEP-name
               |  +--rw (MEP-ID)?
               |  |  +--:(MEP-ID-int)
               |  |     +--rw MEP-ID-int?          int32
               |  |  +--:(MEP-ID-tlv)
               |  |     +--rw MEP-ID-type?         int16
               |  |     +--rw MEP-ID-len?          int16
               |  |     +--rw MEP-ID-value?        binary
               |  +--rw MEP-ID-format?       identityref
               |  +--rw (mp-address)?
               |  |  +--:(mac-address)
               |  |  |  +--rw mac-address?   yang:mac-address
               |  |  +--:(ipv4-address)
               |  |  |  +--rw ipv4-address?  inet:ipv4-address
               |  |  +--:(ipv6-address)
               |  |     +--rw ipv6-address?  inet:ipv6-address
    .          .
    .          .
    .          .
```

Snippet of data hierarchy related to Maintenance Endpoint (MEP)

4.4. rpc definitions

The rpc model facilitates issuing commands to a NETCONF server (in this case to the device that need to execute the OAM command) and obtain a response. rpc model defined here abstracts OAM specific commands in a technology independent manner.

There are several rpc commands defined for the purpose of OAM. In this section we present a snippet of the continuity check command for illustration purposes. Please refer to Section 4 for the complete data hierarchy and Section 5 for the YANG model.

```
module: ietf-gen-oam
   +--rw domains
         +--rw domain* [technology MD-name-string]
         +--rw technology        identityref
   .
   .
rpcs:
   +---x continuity-check
   |  +--ro input
   |  |  +--ro technology             identityref
   |  |  +--ro MD-name-string         MD-name-string
   |  |  +--ro MA-name-string?        MA-name-string
   |  |  +--ro (flow-entropy)?
   |  |  |  +--:(flow-entropy-null)
   |  |  |     +--ro flow-entropy-null?     empty
   |  |  +--ro priority?              uint8
   |  |  +--ro ttl?                   uint8
   |  |  +--ro session-type           enumeration
   |  |  +--ro ecmp-choice?           ecmp-choices
   |  |  +--ro sub-type?              identityref
   |  |  +--ro outgoing-interfaces* [interface]
   |  |  |  +--ro interface          if:interface-ref
   |  |  +--ro source-mep?            MEP-name
   |  |  +--ro destination-mp
   |  |  |  +--ro (mp-address)?
   |  |  |  |  +--:(mac-address)
   |  |  |  |  |  +--ro mac-address?     yang:mac-address
   |  |  |  |  +--:(ipv4-address)
   |  |  |  |  |  +--ro ipv4-address?    inet:ipv4-address
   |  |  |  |  +--:(ipv6-address)
   |  |  |  |     +--ro ipv6-address?    inet:ipv6-address
   |  |  |  +--ro (MEP-ID)?
   |  |  |  |  +--:(MEP-ID-int)
   |  |  |  |     +--ro MEP-ID-int?      int32
   |  |  |  +--ro MEP-ID-format?   identityref
   |  |  +--ro count?                 uint32
   |  |  +--ro interval?              Interval
   |  |  +--ro packet-size?           uint32
   |  +--ro output
   |     +--ro tx-packt-count?    oam-counter32
   |     +--ro rx-packet-count?   oam-counter32
   |     +--ro min-delay?         oam-counter32
   |     +--ro average-delay?     oam-counter32
   |     +--ro max-delay?         oam-counter32
```

Snippet of data hierarchy related to rpc call continuity-check

4.5. OAM data hierarchy

The complete data hierarchy related to the OAM YANG model is presented below.

```
module: ietf-gen-oam
   +--rw domains
      +--rw domain* [technology MD-name-string]
         +--rw technology        identityref
         +--rw MD-name-string    MD-name-string
         +--rw MD-name-format?   identityref
         +--rw (MD-name)?
         |  +--:(MD-name-null)
         |     +--rw MD-name-null?     empty
         +--rw md-level?         MD-level
         +--rw MAs
            +--rw MA*  [MA-name-string]
               +--rw MA-name-string       MA-name-string
               +--rw MA-name-format?      identityref
               +--rw (MA-name)?
               |  +--:(MA-name-null)
               |     +--rw MA-name-null?        empty
               +--rw (connectivity-context)?
               |  +--:(context-null)
               |     +--rw context-null?        empty
               +--rw mep-direction        MEP-direction
               +--rw interval?            Interval
               +--rw loss-threshold?      uint32
               +--rw ttl?                 uint8
               +--rw (flow-entropy)?
               |  +--:(flow-entropy-null)
               |     +--rw flow-entropy-null?   empty
               +--rw priority?            uint8
               +--rw MEP* [mep-name]
               |  +--rw mep-name             MEP-name
               |  +--rw (MEP-ID)?
               |  |  +--:(MEP-ID-int)
               |  |  |  +--rw MEP-ID-int?          int32
               |  |  +--:(MEP-ID-tlv)
               |  |     +--rw MEP-ID-type?         int16
               |  |     +--rw MEP-ID-len?          int16
               |  |     +--rw MEP-ID-value?        binary
               |  +--rw MEP-ID-format?       identityref
               |  +--rw (mp-address)?
               |  |  +--:(mac-address)
               |  |  |  +--rw mac-address?         yang:mac-address
               |  |  +--:(ipv4-address)
               |  |  |  +--rw ipv4-address?        inet:ipv4-address
```

```
                     |  |  +--:(ipv6-address)
                     |  |     +--rw ipv6-address?        inet:ipv6-address
                     |  +--rw (connectivity-context)?
                     |  |  +--:(context-null)
                     |  |     +--rw context-null?        empty
                     |  +--rw Interface?                 if:interface-ref
                     |  +--rw (topology)?
                     |  |  +--:(topo-null)
                     |  |     +--rw topo-null?           empty
                     |  +--ro admin-status?        leafref
                     |  +--ro oper-status?         leafref
                     |  +--rw (flow-entropy)?
                     |  |  +--:(flow-entropy-null)
                     |  |     +--rw flow-entropy-null?   empty
                     |  +--rw priority?            uint8
                     |  +--rw session* [session-cookie]
                     |     +--rw session-cookie             uint32
                     |     +--rw ttl?                       uint8
                     |     +--rw interval?                  Interval
                     |     +--rw enable?                    boolean
                     |     +--rw ecmp-choice?               ecmp-choices
                     |     +--rw source-mep?                MEP-name
                     |     +--rw destination-mep
                     |     |  +--rw (MEP-ID)?
                     |     |  |  +--:(MEP-ID-int)
                     |     |  |  |  +--rw MEP-ID-int?      int32
                     |     |  |  +--:(MEP-ID-tlv)
                     |     |  |     +--rw MEP-ID-type?     int16
                     |     |  |     +--rw MEP-ID-len?      int16
                     |     |  |     +--rw MEP-ID-value?    binary
                     |     |  +--rw MEP-ID-format?         identityref
                     |     +--rw destination-mep-address
                     |     |  +--rw (mp-address)?
                     |     |     +--:(mac-address)
                     |     |     |  +--rw mac-address?    yang:mac-address
                     |     |     +--:(ipv4-address)
                     |     |     |  +--rw ipv4-address?   inet:ipv4-address
                     |     |     +--:(ipv6-address)
                     |     |        +--rw ipv6-address?   inet:ipv6-address
                     |     +--rw (connectivity-context)?
                     |     |  +--:(context-null)
                     |     |     +--rw context-null?              empty
                     |     +--rw (flow-entropy)?
                     |     |  +--:(flow-entropy-null)
                     |     |     +--rw flow-entropy-null?         empty
                     |     +--rw priority?                  uint8
                     |     +--rw outgoing-interface* [interface]
                     |        +--rw interface         leafref
```

```
                  +--rw MIP* [interface]
                  |  +--rw interface               if:interface-ref
                  +--rw related-oam-layer* [offset]
                     +--rw offset            int32
                     +--rw technology        identityref
                     +--rw MD-name-string    MD-name-string
                     +--rw MA-name-string?   MA-name-string
   rpcs:
      +---x continuity-check
      |  +--ro input
      |  |  +--ro technology             identityref
      |  |  +--ro MD-name-string         MD-name-string
      |  |  +--ro MA-name-string?        MA-name-string
      |  |  +--ro (flow-entropy)?
      |  |  |  +--:(flow-entropy-null)
      |  |  |     +--ro flow-entropy-null?     empty
      |  |  +--ro priority?              uint8
      |  |  +--ro ttl?                   uint8
      |  |  +--ro session-type-enum?     enumeration
      |  |  +--ro ecmp-choice?           ecmp-choices
      |  |  +--ro sub-type?              identityref
      |  |  +--ro outgoing-interfaces* [interface]
      |  |  |  +--ro interface           if:interface-ref
      |  |  +--ro source-mep?            MEP-name
      |  |  +--ro destination-mp
      |  |  |  +--ro (mp-address)?
      |  |  |  |  +--:(mac-address)
      |  |  |  |  |  +--ro mac-address?     yang:mac-address
      |  |  |  |  +--:(ipv4-address)
      |  |  |  |  |  +--ro ipv4-address?    inet:ipv4-address
      |  |  |  |  +--:(ipv6-address)
      |  |  |  |     +--ro ipv6-address?    inet:ipv6-address
      |  |  |  +--ro (MEP-ID)?
      |  |  |  |  +--:(MEP-ID-int)
      |  |  |  |  |  +--ro MEP-ID-int?      int32
      |  |  |  |  +--:(MEP-ID-tlv)
      |  |  |  |     +--ro MEP-ID-type?     int16
      |  |  |  |     +--ro MEP-ID-len?      int16
      |  |  |  |     +--ro MEP-ID-value?    binary
      |  |  |  +--ro MEP-ID-format?      identityref
      |  |  +--ro count?                 uint32
      |  |  +--ro interval?              Interval
      |  |  +--ro packet-size?           uint32
      |  +--ro output
      |     +--ro tx-packt-count?    oam-counter32
      |     +--ro rx-packet-count?   oam-counter32
      |     +--ro min-delay?         oam-counter32
      |     +--ro average-delay?     oam-counter32
```

```
      |     +--ro max-delay?         oam-counter32
      +---x continuity-verification     {connectivity-verification}?
      |  +--ro input
      |  |  +--ro technology             identityref
      |  |  +--ro MD-name-string         MD-name-string
      |  |  +--ro MA-name-string?        MA-name-string
      |  |  +--ro (flow-entropy)?
      |  |  |  +--:(flow-entropy-null)
      |  |  |     +--ro flow-entropy-null?     empty
      |  |  +--ro priority?              uint8
      |  |  +--ro ttl?                   uint8
      |  |  +--ro session-type-enum?     enumeration
      |  |  +--ro ecmp-choice?           ecmp-choices
      |  |  +--ro sub-type?              identityref
      |  |  +--ro outgoing-interfaces* [interface]
      |  |  |  +--ro interface           if:interface-ref
      |  |  +--ro source-mep?            MEP-name
      |  |  +--ro destination-mp
      |  |  |  +--ro (mp-address)?
      |  |  |  |  +--:(mac-address)
      |  |  |  |  |  +--ro mac-address?     yang:mac-address
      |  |  |  |  +--:(ipv4-address)
      |  |  |  |  |  +--ro ipv4-address?    inet:ipv4-address
      |  |  |  |  +--:(ipv6-address)
      |  |  |  |     +--ro ipv6-address?    inet:ipv6-address
      |  |  |  +--ro (MEP-ID)?
      |  |  |  |  +--:(MEP-ID-int)
      |  |  |  |  |  +--ro MEP-ID-int?      int32
      |  |  |  |  +--:(MEP-ID-tlv)
      |  |  |  |     +--ro MEP-ID-type?     int16
      |  |  |  |     +--ro MEP-ID-len?      int16
      |  |  |  |     +--ro MEP-ID-value?    binary
      |  |  |  +--ro MEP-ID-format?      identityref
      |  |  +--ro count?                 uint32
      |  |  +--ro interval?              Interval
      |  |  +--ro packet-size?           uint32
      |  +--ro output
      |     +--ro tx-packt-count?    oam-counter32
      |     +--ro rx-packet-count?   oam-counter32
      |     +--ro min-delay?         oam-counter32
      |     +--ro average-delay?     oam-counter32
      |     +--ro max-delay?         oam-counter32
      +---x path-discovery
         +--ro input
         |  +--ro technology             identityref
         |  +--ro MD-name-string         MD-name-string
         |  +--ro MA-name-string?        MA-name-string
         |  +--ro (flow-entropy)?
```

```
         |  |  +--:(flow-entropy-null)
         |  |     +--ro flow-entropy-null?     empty
         |  +--ro priority?             uint8
         |  +--ro ttl?                  uint8
         |  +--ro session-type-enum?    enumeration
         |  +--ro command-sub-type?     identityref
         |  +--ro ecmp-choice?          ecmp-choices
         |  +--ro outgoing-interfaces* [interface]
         |  |  +--ro interface          if:interface-ref
         |  +--ro source-mep?           MEP-name
         |  +--ro destination-mp
         |  |  +--ro (mp-address)?
         |  |  |  +--:(mac-address)
         |  |  |  |  +--ro mac-address?     yang:mac-address
         |  |  |  +--:(ipv4-address)
         |  |  |  |  +--ro ipv4-address?    inet:ipv4-address
         |  |  |  +--:(ipv6-address)
         |  |  |     +--ro ipv6-address?    inet:ipv6-address
         |  |  +--ro (MEP-ID)?
         |  |  |  +--:(MEP-ID-int)
         |  |  |  |  +--ro MEP-ID-int?      int32
         |  |  |  +--:(MEP-ID-tlv)
         |  |  |     +--ro MEP-ID-type?     int16
         |  |  |     +--ro MEP-ID-len?      int16
         |  |  |     +--ro MEP-ID-value?    binary
         |  |  +--ro MEP-ID-format?      identityref
         |  +--ro count?                uint32
         |  +--ro interval?             Interval
         +--ro output
            +--ro response* [response-index]
               +--ro response-index     uint8
               +--ro ttl?               uint8
               +--ro destination-mp
               |  +--ro (mp-address)?
               |  |  +--:(mac-address)
               |  |  |  +--ro mac-address?     yang:mac-address
               |  |  +--:(ipv4-address)
               |  |  |  +--ro ipv4-address?    inet:ipv4-address
               |  |  +--:(ipv6-address)
               |  |     +--ro ipv6-address?    inet:ipv6-address
               |  +--ro (MEP-ID)?
               |  |  +--:(MEP-ID-int)
               |  |  |  +--ro MEP-ID-int?      int32
               |  |  +--:(MEP-ID-tlv)
               |  |     +--ro MEP-ID-type?     int16
               |  |     +--ro MEP-ID-len?      int16
               |  |     +--ro MEP-ID-value?    binary
               |  +--ro MEP-ID-format?   identityref
```

```
              +--ro tx-packt-count?    oam-counter32
              +--ro rx-packet-count?   oam-counter32
              +--ro min-delay?         oam-counter32
              +--ro average-delay?     oam-counter32
              +--ro max-delay?         oam-counter32
   notifications:
      +---n defect-condition-notification
         +--ro technology          identityref
         +--ro MD-name-string      MD-name-string
         +--ro MA-name-string?     MA-name-string
         +--ro mep-name?           MEP-name
         +--ro defect-type?        identityref
         +--ro generating-mepid
         |  +--ro (MEP-ID)?
         |  |  +--:(MEP-ID-int)
         |  |  |  +--ro MEP-ID-int?      int32
         |  |  +--:(MEP-ID-tlv)
         |  |     +--ro MEP-ID-type?     int16
         |  |     +--ro MEP-ID-len?      int16
         |  |     +--ro MEP-ID-value?    binary
         |  +--ro MEP-ID-format?         identityref
         +--ro (error)?
            +--:(error-null)
            |  +--ro error-null?         empty
            +--:(error-code)
               +--ro error-code?         int3
               +--ro error-code?         int32
```

data hierarchy of OAM

5. OAM YANG Module

<CODE BEGINS> file "ietf-gen-oam.yang"

```
      module ietf-gen-oam {
     namespace "urn:ietf:params:xml:ns:yang:ietf-gen-oam";
     prefix goam;

     import ietf-interfaces {
       prefix if;
     }
     import ietf-yang-types {
       prefix yang;
     }
     import ietf-inet-types {
       prefix inet;
     }
```

```
   organization "IETF LIME Working Group";
   contact
     "Tissa Senevirathne tsenevir@cisco.com";
   description
     "This YANG module defines the generic configuration,
      statistics and rpc for OAM to be used within IETF in
      a protocol indpendent manner. Functional level
      abstraction is indendent with YANG modeling. It is
      assumed that each protocol maps corresponding
      abstracts to its native format.
      Each protocol may extend the YANG model defined
      here to include protocol specific extensions";

   revision 2015-04-09 {
     description
       "Initial revision. - 04 version";
     reference "draft-tissa-lime-oam";
   }

   /* features */
   feature connectivity-verification {
     description
       "This feature indicates that the server supports
        executing connectivity verification OAM command and
        returning a response. Servers that do not advertise
        this feature will not support executing
        connectivity verification command or rpc model for
        connectivity verification command.";
   }

   /* Identities */

   identity technology-types {
     description
       "this is the base identy of technology types which are
        vpls, nvo3, TRILL, ipv4, ipv6, mpls, etc";
   }

   identity ipv4 {
     base technology-types;
     description
       "technology of ipv4";
   }


   identity ipv6 {
     base technology-types;
     description
```

```
        "technology of ipv6";
    }

    identity command-sub-type {
      description
        "defines different rpc command subtypes, e.g rfc792 IP
         ping, rfc4379 LSP ping, rfc6905 trill OAM, this is
         optional for most cases";
    }

    identity icmp-rfc792 {
      base command-sub-type;
      description
        "Defines the command subtypes for ICMP ping";
      reference "RFC 792";
    }

    identity name-format {
      description
        "This defines the name format, IEEE 8021Q CFM defines varying
        styles of names. It is expected name format as an identity ref
        to be extended with new types.";
    }

    identity name-format-null {
      base name-format;
      description
        "defines name format as null";
    }

    identity identifier-format {
      description
        "identifier-format identity can be augmented to define other
       format identifiers used in MEPD-ID etc";
    }

    identity identifier-format-integer {
      base identifier-format;
      description
        "defines identifier-format to be integer";
    }

    identity defect-types {
      description
        "defines different defect types, e.g. remote rdi,
         mis-connection defect, loss of continuity";
    }
    identity remote-rdi {
```

```
       base defect-types;
       description
         " Indicates the aggregate health of the remote MEPs. ";
     }

     identity remote-mep-error{
       base defect-types;
       description
         " Indicates that one or more of the remote MEPs is
   reporting a failure ";
     }
     identity invalue-oam-error{
       base defect-types;
       description
    "Indicates that one or more invalid OAM messages has been
   received and that 3.5 times that OAM message transmission
   interval has not yet expired.
   ";
     }
     identity cross-connect-error{
       base defect-types;
       description
   " Indicates that one or more cross-connect oam messages has been
   received and that 3.5 times that OAM message transmission
   interval has not yet expired.
   ";
     }

     /* typedefs */
     typedef MEP-direction {
       type enumeration {
         enum "Up" {
           value 0;
   description
     "Indicates when OAM frames are transmitted towards and
   received from the bridging/routing function.";
         }
         enum "Down" {
           value 1;
   description
     "Indicates when OAM frames are transmitted towards and
   received from the wire.";
         }
       }
       description
         "MEP direction.";
     }
```

```
     typedef MEP-name {
       type string;
       description
         "Generic administrative name for a MEP";
     }

     typedef Interval {
       type uint32;
       units "milliseconds";
       default "1000";
       description
         "Interval between packets in milliseconds.
          0 means no packets are sent.";
     }

     typedef ecmp-choices {
       type enumeration {
         enum "ecmp-use-platform-hash" {
           value 0;
   description
     "Use Platform hashing.";
         }
         enum "ecmp-use-round-robin" {
           value 1;
   description
     "Use round robin hashing.";
         }
       }
       description
         "Equal cost multi Path Choices";
     }

     typedef MD-name-string {
       type string;
       default "";
       description
         "Generic administrative name for an MD";
     }

     typedef MA-name-string {
       type string;
       default "";
       description
         "Generic administrative name for an MA";
     }

     typedef oam-counter32 {
       type yang:zero-based-counter32;
```

```
       description
         "defines 32 bit counter for OAM";
     }

     typedef MD-level {
       type uint32 {
         range "0..255";
       }
       description
         "Maintenance Domain level.  The level may be restricted in
          certain protocols (eg to 0-7)";
     }

     /* groupings */

     grouping topology {
       choice topology {
         case topo-null {
           description
             "this is a placeholder when no topology is needed";
           leaf topo-null {
              type empty;
              description
                "there is no topology define, it will be defined
                 in technology specific model.";
           }
         }
         description
           "Topology choices";
       }
       description
         "Topology";
     }

     grouping error-message {
       choice error {
         case error-null {
           description
             "this is a placeholder when no error status is needed";
           leaf error-null {
              type empty;
              description
                "there is no error define, it will be defined in
                 technology specific model.";
           }
         }
         case error-code {
           description
```

```
             "this is a placeholder to display error code.";
            leaf error-code {
               type int32;
               description
                 "error code is integer value specific to technology.";
            }
          }
          description
            "Error Message choices.";
        }
        description
          "Error Message.";
      }

      grouping mp-address {
        choice mp-address {
          case mac-address {
            leaf mac-address {
              type yang:mac-address;
      description
        "MAC Address";
            }
    description
      "MAC Address based MP Addressing.";
          }
          case ipv4-address {
            leaf ipv4-address {
              type inet:ipv4-address;
      description
        "Ipv4 Address";
            }
    description
      "Ip Address based MP Addressing.";
          }
          case ipv6-address {
            leaf ipv6-address {
              type inet:ipv6-address;
      description
        "Ipv6 Address";
            }
    description
      "ipv6 Address based MP Addressing.";
          }
          description
            "MP Addressing.";
        }
        description
          "MP Address";
```

```
     }

     grouping maintenance-domain-id {
       description
         "Grouping containing leaves sufficient to identify an MD";
       leaf technology {
         type identityref {
           base technology-types;
         }
         mandatory true;

         description
           "Defines the technology";
       }
       leaf MD-name-string {
         type MD-name-string;
         mandatory true;
         description
           "Defines the generic administrative maintenance domain name";
       }
     }

     grouping MD-name {
       leaf MD-name-format {
         type identityref {
           base name-format;
         }
         description
           "Name format.";
       }
       choice MD-name {
         case MD-name-null {
           leaf MD-name-null {
     when "../../../MD-name-format = name-format-null" {
        description
          "MD name format is equal to null format.";
     }
             type empty;
     description
       "MD name Null.";
           }
         }
         description
           "MD name.";
       }
       description
         "MD name";
     }
```

```
     grouping ma-identifier {
       description
         "Grouping containing leaves sufficient to identify an MA";
       leaf MA-name-string {
         type MA-name-string;
         description
           "MA name string.";
       }
     }

     grouping MA-name {
       description
         "MA name";
       leaf MA-name-format {
         type identityref {
           base name-format;
         }
         description
           "Ma name format";
       }
       choice MA-name {
         case MA-name-null {
           leaf MA-name-null {
             when "../../../MA-name-format = name-format-null" {
     description
       "MA";
     }
     type empty;
     description
       "empty";
           }
         }
         description
           "MA name";
       }
     }

     grouping MEP-ID {
       choice MEP-ID {
         default "MEP-ID-int";
         case MEP-ID-int {
           leaf MEP-ID-int {
             type int32;
     description
       "MEP ID in integer format";
           }
         }
         case MEP-ID-tlv {
```

```
            leaf MEP-ID-type {
      type int16;
      description
        "Type of MEP-ID";
            }
    leaf MEP-ID-len {
      type int16;
      description
        "Length of MEP-ID value";
    }
    leaf MEP-ID-value {
       type binary {
         length "12..255";
       }
       description
         "Value please refer RFC6428.";
    }
         }
         description
           "MEP-ID";
       }
       leaf MEP-ID-format {
         type identityref {
           base identifier-format;
         }
         description
           "MEP ID format.";
       }
       description
         "MEP-ID";
     }

     grouping MEP {
       description
         "Defines elements within the MEP";
       leaf mep-name {
         type MEP-name;
         mandatory true;
         description
           "Generic administrative name of the MEP";
       }
       uses MEP-ID;

       uses mp-address;
       uses connectivity-context;
       leaf Interface {
         type if:interface-ref;
         description
```

```
          "Interface name as defined by ietf-interfaces";
      }
      uses topology;
    }

    grouping session-type {
      description
        "This object indicates the current session
         definition.";
      leaf session-type-enum {
        type enumeration {
          enum proactive {
            description
              "The current session is proactive";
          }
          enum on-demand {
            description
              "The current session is on-demand.";
          }
        }
        description
          "session type enum";
      }
    }

    grouping monitor-stats {
      leaf tx-packt-count {
        type oam-counter32;
        description
          "Transmitted Packet count";
      }
      leaf rx-packet-count {
        type oam-counter32;
        description
          "Received packet count";
      }
      leaf min-delay {
        type oam-counter32;
        units milliseconds;
        description
          "Delay is specified in milliseconds";
      }
      leaf average-delay {
        type oam-counter32;
        units millisecond;
        description
          "average delay in milliseconds";
      }
```

```
       leaf max-delay {
         type oam-counter32;
         units millisecond;
         description
           "Maximum delay in milliseconds";
       }
       description
         "Monitor Statistics";
     }

     grouping MIP {
       description
         "defines MIP";
       leaf interface {
         type if:interface-ref;
         description
            "Interface";
       }
     }

     grouping related-oam-layer {
       leaf offset {
         type int32 {
           range "-255..255";
         }
         description
           "defines offset (in MD levels) to a related OAM layer
            +1 is the layer immediately above
            -1 is the layer immediately below";
       }
       uses maintenance-domain-id;
       uses ma-identifier;
       description
         "related OAM layer";
     }

     grouping interface-status {
       description
         "collection of interface related status";
       leaf admin-status {
         type leafref {
           path "/if:interfaces-state/if:interface/if:admin-status";
         }
         config false;
         description
           "oper status from ietf-interface module";
       }
       leaf oper-status {
```

```
         type leafref {
           path "/if:interfaces-state/if:interface/if:oper-status";
         }
         config false;
         description
           "oper status from ietf-interface module";
       }
     }

     grouping connectivity-context {
       description
         "Grouping defining the connectivity context for an MA; for
          example, a VRF for IP, or an LSP for MPLS.  This will be
          augmented by each protocol who use this component";
       choice connectivity-context {
         default "context-null";
         case context-null {
           description
             "this is a place holder when no context is needed";
           leaf context-null {
             type empty;
             description
               "there is no context define";
           }
         }
         description
           "connectivity context";
       }
     }

     grouping priority {
       description
         "Priority used in transmitted packets; for example, in the
          TOS/DSCP field in IP or the Traffic Class field in MPLS";
       leaf priority {
         type uint8;
         description
           "priority";
       }
     }

     grouping flow-entropy {
       description
         "defines the grouping statement for flow-entropy";
       choice flow-entropy {
         default "flow-entropy-null";
         case flow-entropy-null {
           description
```

```
             "this is a place holder when no flow entropy is needed";
           leaf flow-entropy-null {
             type empty;
             description
               "there is no flow entropy defined";
           }
         }
         description
           "Flow entropy";
       }
     }

     grouping measurement-timing-group {
        description
        "This grouping includes objects used for
         proactive and on-demand
         scheduling of PM measurement sessions.";

        container start-time {
           description
   "This container defines the session start time.";
   choice start-time {
     description
     "Measurement sessions tart time can be immediate, relative, or
     absolute.";
     container immediate {
               presence "Start the measurement session immediately.";
               description
               "Start Time of probe immediately.";
     }
     leaf absolute {
       type yang:date-and-time;
       description
       "This objects specifies the scheduled start time
        to perform the on-demand monitoring operations.";
     }
           }
        }

        container stop-time {
   description
   "This container defines the session stop time.";
   choice stop-time {
      description
      "Measurement session stop time can be none, or absolute.";
      container none {
                 presence "Never end the measurement session.";
                 description
```

```
                 "Stop time is never to end.";

      }

      leaf absolute {
        type yang:date-and-time;
        description
          "This objects specifies the scheduled stop time
            to perform the on-demand monitoring operations.";
      }
           }
        }
     }

     container domains {
       description
         "Contains configuration related data. Within the container
          is list of fault domains. Wihin each domian has List of MA.";
       list domain {
         key "technology MD-name-string";
         ordered-by system;
         description
           "Define the list of Domains within the IETF-OAM";
         uses maintenance-domain-id;
         uses MD-name;
         leaf md-level {
           type MD-level;
           description
             "Defines the MD-Level";
         }
         container MAs {
           description
             "This container defines MA, within that have multiple MA
              and within MA have MEP, MIP";
           list MA {
     key "MA-name-string";
             ordered-by system;
             uses ma-identifier;
             uses MA-name;
             uses connectivity-context;
             leaf mep-direction {
               type MEP-direction;
               mandatory true;
               description
                 "Direction for MEPs in this MA";
             }
             leaf interval {
       type Interval;
```

```
               default "0";
               description
                 "Defines default Keepalive/CC Interval.  May be
                  overridden for specific sessions if supported by the
                  protocol.";
             }
             leaf loss-threshold {
       type uint32;
               default "3";
               description
                 "number of consecutive Keepalive/CC messages missed
                  before declaring loss of continuity fault. This is
                  monitored per each remote MEP session";
             }
             leaf ttl {
               type uint8;
               default "255";
       description
         "Time to Live";
             }
             uses flow-entropy {
               description
                 "Default flow entropy in this MA, which may be
                  overridden for particular MEPs, sessions or
                  operations";
             }
             uses priority {
               description
                 "Default priority for this MA, which may be overridden
                  for particular MEPs, sessions or operations.";
             }
             list MEP {
               key "mep-name";
               ordered-by system;
               description
                 "contain list of MEPS";
               uses MEP;
               uses interface-status {
                 description
                   "status of associated interface";
               }
               uses flow-entropy;
               uses priority;
               list session {
                 key "session-cookie";
                 ordered-by user;
                 description
                   "Monitoring session to/from a particular remote MEP.
```

```
                    Depending on the protocol, this could represent CC
                    messages received from a single remote MEP (if the
                    protocol uses multicast CCs) or a target to which
                    unicast echo request CCs are sent and from which
                    responses are received (if the protocol uses a
                    unicast request/response mechanism).";
                 leaf session-cookie {
                   type uint32;
                   description
                     "Cookie to identify different sessions, when there
                      are multiple remote MEPs or multiple sessions to
                      the same remote MEP.";
                 }
                 leaf ttl {
                   type uint8;
                   default "255";
     description
       "Time to Live.";
                 }
                 leaf interval {
                   type Interval;
                   description
                     "Transmission interval for CC packets for this
                      session.";
                 }
                 leaf enable {
            type boolean;
                   default "false";
                   description
                     "enable or disable a monitor session";
                 }
                 leaf ecmp-choice {
                   type ecmp-choices;
                   description
                     "0 means use the specified interface
                      1 means use round robin";
                 }
                 leaf source-mep {
                   type MEP-name;
                   description
                     "Source MEP for this session, if applicable";
                 }
                 container destination-mep {
                   uses MEP-ID;
     description
        "Destination MEP";
                 }
                 container destination-mep-address {
```

```
                 uses mp-address;
     description
        "Destination MEP Address";
               }
               uses connectivity-context;
               uses flow-entropy;
               uses priority;
               list outgoing-interface {
                 key "interface";
                 leaf interface {
                   type if:interface-ref;
       description
         "Outgoing Interface";
                 }
     description
       "outgoing interfaces";
               }
             }
           }
           list MIP {
             key "interface";
             uses MIP;
       description
         "Maintenance Intermediate Point";
           }
           list related-oam-layer {
             key "offset";
             description
               "List of OAM layers above and below that are related to
                current MA. This allow users to easily navigate up and
                down to efficiently troubleshoot a connectivity
                issue";
             uses related-oam-layer;
           }
     description
        "Maintenance Association list";
         }
       }
     }
   }

   notification defect-condition-notification {
     description
       "When defect condition is met this notificiation is sent";
    uses maintenance-domain-id {
       description
         "defines the MD (Maintenance Domain) identifier, which is the
          Generic MD-name-string and the technology.";
```

```
       }
       uses ma-identifier;
       leaf mep-name {
         type MEP-name;
         description
           "Indicate which MEP is seeing the error";
       }
       leaf defect-type {
         type identityref {
           base defect-types;
         }
         description
           "The currently active defects on the specific MEP.";
       }
       container generating-mepid {
         uses MEP-ID;
         description
           "Who is generating the error (if known) if
            unknown make it 0.";
       }
       uses error-message {
         description
            "Error message to indicate more details.";
       }
     }
     rpc continuity-check {
       description
         "Generates continuity-check as per RFC7276 Table 4.";
       input {
         uses maintenance-domain-id {
           description
             "defines the MD (Maintenance Domain) identifier, which is
              the generic
              MD-name-string and the technology.";
         }
         uses ma-identifier {
           description
             "identfies the Maintenance association";
         }
         uses flow-entropy;
         uses priority;
         leaf ttl {
           type uint8;
           default "255";
    description
      "Time to Live";
         }
         uses session-type;
```

```
         leaf ecmp-choice {
           type ecmp-choices;
           description
             "0 means use the specified interface
              1 means use round robin";
         }
         leaf sub-type {
           type identityref {
             base command-sub-type;
           }
           description
             "defines different command types";
         }
         list outgoing-interfaces {
           key "interface";
           leaf interface {
             type if:interface-ref;
     description
       "outgoing interface";
           }
   description
     "outgoing Interfaces";
         }
         leaf source-mep {
           type MEP-name;
   description
     "Source MEP";
         }
         container destination-mp {
           uses mp-address;
           uses MEP-ID {
             description "Only applicable if the destination is a MEP";
           }
   description
     "Destination MEP";
         }
         leaf count {
           type uint32;
           default "3";
           description

             "Number of ping echo request message to send";
         }
         leaf interval {
           type Interval;
           description
             "Interval between echo requests";
```

```
         }
         leaf packet-size {
           type uint32 {
             range "64..10000";
           }
           default "64";
           description
             "Size of ping echo request packets, in octets";
         }
       }
       output {
         uses monitor-stats {
           description
             "Stats of continuity check.";
         }
       }
     }

     rpc continuity-verification {
       if-feature connectivity-verification;
       description
         "Generates continuity-verification as per RFC7276 Table 4.";
       input {
         uses maintenance-domain-id {
           description
             "defines the MD (Maintenance Domain) identifier, which is
              the generic
              MD-name-string and the technology.";
         }
         uses ma-identifier {
           description
             "identfies the Maintenance association";
         }
         uses flow-entropy;
         uses priority;
         leaf ttl {
           type uint8;
           default "255";
   description
     "Time to Live";
         }
         uses session-type;
         leaf ecmp-choice {
           type ecmp-choices;
           description
             "0 means use the specified interface
              1 means use round robin";
         }
```

```
         leaf sub-type {
           type identityref {
             base command-sub-type;
           }
           description
             "defines different command types";
         }
         list outgoing-interfaces {
           key "interface";
           leaf interface {
             type if:interface-ref;
     description
       "outgoing interface";
           }
   description
     "outgoing Interfaces";
         }
         leaf source-mep {
           type MEP-name;
   description
     "Source MEP";
         }
         container destination-mp {
           uses mp-address;
           uses MEP-ID {
             description "Only applicable if the destination is a MEP";
           }
   description
     "Destination MEP";
         }
         leaf count {
           type uint32;
           default "3";
           description
             "Number of ping echo request message to send";
         }
         leaf interval {
           type Interval;
           description
             "Interval between echo requests";
         }
         leaf packet-size {
           type uint32 {
             range "64..10000";
           }
           default "64";
           description
             "Size of ping echo request packets, in octets";
```

```
        }
      }
      output {
        uses monitor-stats {
          description
            "Stats of continuity check.";
        }
      }
    }
    rpc path-discovery {
      description
        "Generates Trace-route or Path Trace and return response.
         Referencing RFC7276 for common Toolset name, for IP it’s
         Traceroute, for MPLS OAM it’s Traceroute mode, for
         MPLS-TP OAM it’s Route Tracing, for Pseudowire OAM it’s
         LSP Ping, and for TRILL OAM It’s Path Tracing tool.
         Starts with TTL
         of one and increment by one at each hop. Untill destination
         reached or TTL reach max valune";
      input {
        uses maintenance-domain-id {
          description
            "defines the MD (Maintenance Domain) identifier, which is
             the generic MD-name-string and the technology.";
        }
        uses ma-identifier {
          description
            "identfies the Maintenance association";
        }
        uses flow-entropy;
        uses priority;
        leaf ttl {
          type uint8;
          default "255";
  description
    "Time to Live";
        }
        uses session-type;
        leaf command-sub-type {
          type identityref {


            base command-sub-type;
          }
          description
            "defines different command types";
        }
        leaf ecmp-choice {
```

```
           type ecmp-choices;
           description
             "0 means use the specified interface
              1 means use round robin";
         }
         list outgoing-interfaces {
           key "interface";
           leaf interface {
             type if:interface-ref;
     description
       "Interface.";
           }
   description
     "Outgoing interface list.";
         }
         leaf source-mep {
           type MEP-name;
   description
     "Source MEP";
         }
         container destination-mp {
           uses mp-address;
           uses MEP-ID {
             description "Only applicable if the destination is a MEP";
           }
   description
     "Destination MEP";
         }
         leaf count {
           type uint32;
           default "1";
           description
             "Number of traceroute probes to send.  In protocols where a
              separate message is sent at each TTL, this is the number
              of packets to send at each TTL.";
         }
         leaf interval {
           type Interval;
           description
             "Interval between echo requests";
         }
       }
       output {
         list response {
           key "response-index";
           leaf response-index {
     type uint8;
             description
```

```
               "Arbitrary index for the response.  In protocols that
                guarantee there is only a single response at each TTL
                (eg IP Traceroute), the TTL can be used as the response
                index.";
           }
           leaf ttl {
             type uint8;
     description
       "Time to Live";
           }
   description
      "Time to Live";
           container destination-mp {
             description "MP from which the response has been received";
             uses mp-address;
             uses MEP-ID {
               description
                 "Only applicable if the destination is a MEP";
             }
           }
           uses monitor-stats {
             description
               "If count is 1, there is a single delay value reported.";
           }
         }
         description
           "List of response.";
       }
     }
   }
```

YANG module of OAM

<CODE ENDS>

6. Base Mode

The Base Mode defines default configuration that MUST be present in the devices that comply with this document. Base Mode allows users to have "zero-touch" experience. Several parameters require technology specific definition.

6.1. MEP Address

In the Base Mode of operation, the MEP Address is by default the IP address of the interface on which the MEP is located.

6.2. MEP ID for Base Mode

In the Base Mode of operation, each device creates a single UP MEP associated with a virtual OAM port with no physical layer (NULL PHY). The MEPID associated with this MEP is zero (0). The choice of MEP-ID zero is explained below.

MEPID is 2 octet field by default. It is never used on the wire except when using CCM. Ping, traceroute and session monitoring does not use the MEPID on its message header. It is important to have method that can derive MEP ID of base mode in an automatic manner with no user intervention. IP address cannot be directly used for this purpose as the MEP ID is much smaller field. For Base Mode of operation we propose to use MEP ID zero (0) as the default MEP-ID.

CCM packet use MEP-ID on the payload. CCM MUST NOT be used in the Base Mode. Hence CCM MUST be disabled on the Maintenance Association of the Base Mode.

If CCM is required, users MUST configure a separate Maintenance association and assign unique value for the corresponding MEP IDs.

[IEEE802.1Q] CFM defines MEP ID as an unsigned integer in the range 1 to 8191. In this document we propose to extend the range to 0 to 65535. Value 0 is reserved for MEP ID of Base Mode operation and MUST NOT be used for other purposes.

6.3. Maintenance Domain

Default MD-LEVEL is set to 3.

6.4. Maintenance Association

MAID [IEEE802.1Q] has a flexible format and includes two parts: Maintenance Domain Name and Short MA name. In the Based Mode of operation, the value of the Maintenance Domain Name must be the character string "GenericBaseMode" (excluding the quotes "). In Base Mode operation Short MA Name format is set to 2-octet integer format (value 3 in Short MA Format field [IEEE802.1Q]) and Short MA name set to 65532 (0xFFFC).

7. Note

This section will be removed or subject to change in the future if any agreement is reached. As per investigation of RFC7276 for performance Monitoring for Loss and Delay are defined for MPLS OAM(RFC6374[RFC6374]), OWAMP (RFC4656[RFC4656]) and TWAMP (RFC5357[RFC5357]) and TRILL OAM (RFC7456[RFC7456]). In case of

Performance Monitoring Statistics are common between these technologies thus generic Yang model for Performance will be worked out through separate draft with Augmentation of Generic LIME model. In case of Other Function, it’s technology specific and thus should be dealt in technology specific Yang model instead of Generic Model.

8. Security Considerations

The YANG module defined in this memo is designed to be accessed via the NETCONF protocol [RFC6241] [RFC6241]. The lowest NETCONF layer is the secure transport layer and the mandatory-to-implement secure transport is SSH [RFC6242] [RFC6242]. The NETCONF access control model [RFC6536] [RFC6536] provides the means to restrict access for particular NETCONF users to a pre-configured subset of all available NETCONF protocol operations and content.

There are a number of data nodes defined in the YANG module which are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., <edit-config>) to these data nodes without proper protection can have a negative effect on network operations.

The vulnerable "config true" subtrees and data nodes are the following:

/goam:domains/goam:domain/

/goam:domains/goam:domain/goam:MAs/goam:MA/

/goam:domains/goam:domain/goam:MAs/goam:MA/goam:MEP

/goam:domains/goam:domain/goam:MAs/goam:MA/goam:MEP/goam:session/

Unauthorized access to any of these lists can adversely affect OAM management system handling of end-to-end OAM and coordination of OAM within underlying network layers This may lead to inconsistent configuration, reporting, and presentation for the OAM mechanisms used to manage the network.

9. IANA Considerations

This document registers a URI in the IETF XML registry [RFC3688] [RFC3688]. Following the format in RFC 3688, the following registration is requested to be made:

URI: urn:ietf:params:xml:ns:yang:ietf-gen-oam

Registrant Contact: The IESG.

XML: N/A, the requested URI is an XML namespace.

This document registers a YANG module in the YANG Module Names registry [RFC6020].

name: ietf-gen-oam namespace: urn:ietf:params:xml:ns:yang:ietf-gen-oam prefix: goam reference: RFC XXXX

10. Acknowledgments

Giles Heron came up with the idea of developing a YANG model as a way of creating a unified OAM API set (interface), work in this document is largely an inspiration of that. Alexander Clemm provided many valuable tips, comments and remarks that helped to refine the YANG model presented in this document.

Carlos Pignataro, David Ball,Mahesh Jethanandani,Benoit Claise,Ladislav Lhotka,GUBALLA JENS,Yuji Tochio,Gregory Mirsky, Huub van Helvoort, Tom Taylor, Dapeng Liu participated and contributed to this document.

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <http://www.rfc-editor.org/info/rfc3688>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <http://www.rfc-editor.org/info/rfc6020>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <http://www.rfc-editor.org/info/rfc6241>.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <http://www.rfc-editor.org/info/rfc6242>.

[RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <http://www.rfc-editor.org/info/rfc6536>.

[RFC792] Postel, J., "Internet Control Message Protocol", RFC 792, September 1981.

11.2. Informative References

[IEEE802.1Q] "Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks", IEEE Std 802.1Q-2011, August 2011.

[RFC4379] Kompella, K. and G. Swallow, "Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures", RFC 4379, DOI 10.17487/RFC4379, February 2006, <http://www.rfc-editor.org/info/rfc4379>.

[RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 4443, DOI 10.17487/RFC4443, March 2006, <http://www.rfc-editor.org/info/rfc4443>.

[RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, DOI 10.17487/RFC4656, September 2006, <http://www.rfc-editor.org/info/rfc4656>.

[RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, DOI 10.17487/RFC5357, October 2008, <http://www.rfc-editor.org/info/rfc5357>.

[RFC6291] Andersson, L., van Helvoort, H., Bonica, R., Romascanu, D., and S. Mansfield, "Guidelines for the Use of the "OAM" Acronym in the IETF", BCP 161, RFC 6291, DOI 10.17487/RFC6291, June 2011, <http://www.rfc-editor.org/info/rfc6291>.

[RFC6325] Perlman, R., Eastlake 3rd, D., Dutt, D., Gai, S., and A. Ghanwani, "Routing Bridges (RBridges): Base Protocol Specification", RFC 6325, DOI 10.17487/RFC6325, July 2011, <http://www.rfc-editor.org/info/rfc6325>.

[RFC6371] Busi, I., Ed. and D. Allan, Ed., "Operations, Administration, and Maintenance Framework for MPLS-Based Transport Networks", RFC 6371, DOI 10.17487/RFC6371, September 2011, <http://www.rfc-editor.org/info/rfc6371>.

[RFC6374] Frost, D. and S. Bryant, "Packet Loss and Delay Measurement for MPLS Networks", RFC 6374, DOI 10.17487/RFC6374, September 2011, <http://www.rfc-editor.org/info/rfc6374>.

[RFC7174] Salam, S., Senevirathne, T., Aldrin, S., and D. Eastlake 3rd, "Transparent Interconnection of Lots of Links (TRILL) Operations, Administration, and Maintenance (OAM) Framework", RFC 7174, DOI 10.17487/RFC7174, May 2014, <http://www.rfc-editor.org/info/rfc7174>.

[RFC7276] Mizrahi, T., Sprecher, N., Bellagamba, E., and Y. Weingarten, "An Overview of Operations, Administration, and Maintenance (OAM) Tools", RFC 7276, DOI 10.17487/RFC7276, June 2014, <http://www.rfc-editor.org/info/rfc7276>.

[RFC7455] Senevirathne, T., Finn, N., Salam, S., Kumar, D., Eastlake 3rd, D., Aldrin, S., and Y. Li, "Transparent Interconnection of Lots of Links (TRILL): Fault Management", RFC 7455, DOI 10.17487/RFC7455, March 2015, <http://www.rfc-editor.org/info/rfc7455>.

[RFC7456] Mizrahi, T., Senevirathne, T., Salam, S., Kumar, D., and D. Eastlake 3rd, "Loss and Delay Measurement in Transparent Interconnection of Lots of Links (TRILL)", RFC 7456, DOI 10.17487/RFC7456, March 2015, <http://www.rfc-editor.org/info/rfc7456>.

[Y.1731] "OAM functions and mechanisms for Ethernet based networks", ITU-T Recommendation G.8013/Y.1731, 2013.

Authors’ Addresses

Tissa Senevirathne
CISCO Systems
375 East Tasman Drive.
San Jose, CA 95134
USA

Phone: 408-853-2291
Email: tsenevir@cisco.com

Norman Finn
CISCO Systems
510 McCarthy Blvd
Milpitas, CA 95035
USA

Email: nfinn@cisco.com

Deepak Kumar (editor)
CISCO Systems
510 McCarthy Blvd
Milpitas, CA 95035
USA

Email: dekumar@cisco.com

Samer Salam
CISCO Systems
595 Burrard St. Suite 2123
Vancouver, BC V7X 1J1
Canada

Email: ssalam@cisco.com

Qin Wu (editor)
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: bill.wu@huawei.com

Michael Wang
Huawei Technologies,Co.,Ltd
101 Software Avenue, Yuhua District
Nanjing 210012
China

Email: wangzitao@huawei.com