Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 27, 2017

D. Kumar
Cisco
M. Wang
Q. Wu
Huawei
R. Rahman
S. Raghavan
Cisco

February 23, 2017

Generic YANG Data Model for Connectionless Operations, Administration, and Maintenance(OAM) protocols
draft-ietf-lime-yang-connectionless-oam-04

Abstract

This document presents a base YANG Data model for connectionless Operations Administration, and Maintenance(OAM) protocols. It provides a technology-independent abstraction of key OAM constructs for connectionless protocols. The base model presented here can be extended to include technology specific details. This is leading to uniformity between OAM protocols and support both nested OAM workflows (i.e., performing OAM functions at different or same levels through a unified interface).

Status of This Memo

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

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

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

This Internet-Draft will expire on August 27, 2017.

Table of Contents

1. Introduction

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

1. Monitor networks connections (Reachability 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.

The different OAM tools may support connection-oriented technologies or connectionless technologies. In connection-oriented technologies, a connection is established prior to the transmission of data. In connectionless technologies, data is typically sent between end points without prior arrangement [RFC7276]. Note that the Connection-Oriented OAM YANG DATA model is defined in [I-D.ietf-lime-yang-oam-model].

In this document, we presents a base YANG Data model for connectionless OAM protocols. The generic YANG model for connectionless OAM only includes configuration data and state data. It can be used in conjunction with data retrieval method model [I-D.ietf-lime-yang-connectionless-oam-methods], which focuses on data retrieval procedures like RPC. However it also can be used independently of data retrieval method model.

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 terms are defined in [RFC6241] and are not redefined here:

- client
- configuration data
- server
- state data

The following terms are defined in [RFC6020] and are not redefined here:

- augment

- data model
- data node

The terminology for describing YANG data models is found in [RFC6020].

2.1. Terminology

TP - Test Point

MAC - Media Access Control

BFD - Bidirectional Forwarding Detection

RPC - A Remote Procedure Call, as used within the NETCONF protocol

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

2.2. Tree Diagrams

A simplified graphical representation of the data model is used in this document. The meaning of the symbols in these diagrams is as follows:

Each node is printed as:

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

<status> is one of:
+ for current

<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

3. Overview of the Connectionless OAM Model

At the top of the model, there is an 'cc-oper-data' container for session statistics. Grouping is also defined for common session statistics and these are applicable for proactive OAM sessions. Multiple 'test-point-locations' keyed using technology specific keys (eg., IPv4 address for IPv4 locations) are possible by augmented network nodes which are defined in [I-D.ietf-i2rs-yang-network-topo] to describe the network hierarchies and the inventory of nodes contained in a network. Each 'test-point-location' is chosen based on 'location-type' which when chosen, leads to a container that includes a list of 'test-point-locations' keyed by technology specific keys. Each test point location includes a 'test-point-location-info'. The 'test-point-location-info' includes 'tp-technology', 'tp-tools', and 'connectionless-oam-layers'. The groupings of 'tp-address' and 'tp-address-vrf' are kept out of 'test-point-location-info' to make it addressing agnostic and allow varied composition. Depending upon the choice of the 'location-type' (determined by the 'tp-address-vrf'), the containers differ in its composition of 'test-point-locations' while the 'test-point-location-info', is a common aspect of every 'test-point-location'. The vrf is used to describe the corresponding network instance. The 'tp-technology' indicate OAM technology details. The 'tp-tools' describe the OAM tools supported. The 'connectionless-oam-layers' is used to describe the relationship of one test point with other test points. The level in 'oam-layers' indicate whether related OAM test point is The level in oam-layers indicate whether related oam test point is in client layer(lower layer described in section 3.3), server layer (upper layer described in section 3.3) or the same layer as the current test point under Test point Locations. The model is augmented to "/nd:networks/nd:network/nd:node" using 'test-point-locations' defined below.

3.1. TP Address

In connectionless OAM, the tp address is defined with the following type:

o MAC address [RFC6136]

- IPv4 or IPv6 address
- TP-attribute
- System-id to represent the device or node.[I-D.ietf-spring-sr-yang]

To define a forwarding treatment of a test packet, the 'tp-address' needs to be associated with additional parameters, e.g. DSCP for IP or TC for MPLS. In generic connectionless OAM YANG model, these parameters are not explicit configured. The model user can add corresponding parameters according to their requirements.

3.2. Tools

The different OAM tools may be used in one of two basic types of activation: proactive and on-demand. The proactive OAM refers to OAM actions which are carried out continuously to permit proactive reporting of fault. The on-demand OAM refers to OAM actions which are initiated via manual intervention for a limited time to carry out diagnostics [RFC7276] [G.8013]. In connectionless OAM, 'session-type' is defined to indicate which kind of activation will be used by the current session.

In connectionless OAM, the tools attribute is used to describe a toolset for fault detection and isolation. And it can serve as a constraint condition when the base model be extended to specific OAM technology. For example, to fulfill the ICMP PING configuration, the "../coam:continuity-check" should be set to "true", and then the lime base model should be augmented with ICMP PING specific details.

3.3. OAM-layers

As typical networks have a multi-layer architecture, the set of OAM protocols similarly take a multi-layer structure; each layer may has its own OAM protocol [RFC7276] and is corresponding to specific network portion or path and has associated test points. OAM-layers is referred to a list of upper layer, lower layer that are related to current test point. This allows users to easily navigate up and down to efficiently troubleshoot a "loss of continuity defect" at different layer. In this model, we have kept level default as 0, when all test points are located at the same layer. 'Level' defines the relative technology level in a sequence of network portions, and is provided to allow correlation of faults in related OAM domains. For example, there is a network in which data traffic between two customer edges is transported over three consecutive network portions, the current test point is located in the second network portion. If there is a defect in the first network portion is

located at the upstream of the second network portion, the level of the first network portion is set to "-1". If the third network portion is located at the downstream of the second network portion and the level is set to "1". In another case, if the first network portion and the third network portion is in the same level of the second network portion, the level is set to "0". The snippet below depicts an example of OAM layers.

```
list oam-layers {
  key "index";
  leaf index {
     type uint16 {
        range "0..65535";
     }
  }
  leaf level {
      type int32 {
           range "-1..1";
      }
      description
        "Level";
  }
  ordered-by user;
  description
     "List of related oam layers.";
}
```

3.4. Test Point Locations Information

This is a generic grouping for Test Point Locations Information. It Provide details of Test Point Location using Tools, 'OAM-Layers' grouping defined above.

3.5. Test Point Locations

This is a generic grouping for Test Point Locations. Choice statement is used to define locations types, for example 'ipv4-location-type', 'ipv6-location-type', etc. Container is defined under each location type containing list keyed to test point address, Test Point Location Information defined in section above, and routing instance VRF name if required.

3.6. Path Discovery Data

This is a generic grouping for path discovery data model that can be retrieved by any data retrieval methods including RPCs. Path discovery data output from methods, includes 'src-test-point', 'dst-test-point', 'sequence-number', 'hop-cnt', session statistics of

various kinds, path verification and path trace related information. Path discovery includes data to be retrieved on a 'per-hop' basis via a list of 'path-trace-info-list' which includes information like 'timestamps', 'ingress-interface', 'egress-interface' and 'app-meta-data'. The path discovery data model is made generic enough to allow different methods of data retrieval. None of the fields are made mandatory for that reason. Noted that the retrieval methods are defined in [I-D.ietf-lime-yang-connectionless-oam-methods].

3.7. Continuity Check Data

This is a generic grouping for continuity check data model that can be retrieved by any data retrieval methods including RPCs. Continuity check data output from methods, includes 'src-test-point', 'dst-test-point', 'sequence-number', 'hop-cnt' and session statistics of various kinds. The continuity check data model is made generic enough to allow different methods of data retrieval. None of the fields are made mandatory for that reason. Noted that the retrieval methods are defined in [I-D.ietf-lime-yang-connectionless-oam-methods].

3.8. OAM data hierarchy

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

```
module: ietf-connectionless-oam
    +--ro cc-oper-data {continuity-check}?
       +--ro cc-ipv4-sessions-statistics
       |  +--ro cc-session-statistics
       |     +--ro session-count?              uint32
       |     +--ro session-up-count?           uint32
       |     +--ro session-down-count?         uint32
       |     +--ro session-admin-down-count?   uint32
       +--ro cc-ipv6-sessions-statistics
          +--ro cc-session-statistics
             +--ro session-count?              uint32
             +--ro session-up-count?           uint32
             +--ro session-down-count?         uint32
             +--ro session-admin-down-count?   uint32
  augment /nd:networks/nd:network/nd:node:
    +--rw tp-location-type-value?               identityref
    +--rw (location-type)?
       +--:(ipv4-location-type)
       |  +--rw test-point-ipv4-location-list
       |     +--rw test-point-locations* [ipv4-location]
       |        +--rw ipv4-location    inet:ipv4-address
       |        +--rw vrf?             routing-instance-ref
```

```
|        +--rw (technology)?
|        |  +--:(technology-null)
|        |  |  +--rw tech-null?       empty
|        |  +--:(technology-string)
|        |     +--rw ipv4-icmp?       string
|        +--rw tp-tools
|        |  +--rw connectivity-verification?   boolean
|        |  +--rw continuity-check?            boolean
|        |  +--rw path-discovery?              boolean
|        +--rw root?
|        +--rw oam-layers* [index]
|           +--rw index                        uint16
|           +--rw level?                       int32
|           +--rw (tp-location)?
|              +--:(mac-address)
|              |  +--rw mac-address-location? yang:mac-address
|              +--:(ipv4-address)
|              |  +--rw ipv4-location?        inet:ipv4-address
|              +--:(ipv6-location)
|              |  +--rw ipv6-address          inet:ipv6-address
|              +--:(tunnel-location)
|              |  +--rw tunnel-location?      uint32
|              +--:(ip-prefix-location)
|              |  +--rw ip-prefix-location?   inet:ip-prefix
|              +--:(route-dist-location)
|              |  +--rw route-dist-location?  uint32
|              +--:(group-ip-address-location)
|              |  +--rw group-ip-address-location?   IP-Multicast
                                                      -Group-Address
|              +--:(as-number-location)
|              |  +--rw as-number-location    inet:as-number
|              +--:(lsp-id-location)
|              |  +--rw lsp-id-location?             string
|              +--:(system-id-location)
|                 +--rw system-id-location?          router-id
+--:(ipv6-location-type)
|  +--rw test-point-ipv6-location-list
|     +--rw test-point-locations* [ipv6-location]
|        +--rw ipv6-location    inet:ipv6-address
|        +--rw vrf?             routing-instance-ref
|        +--rw (technology)?
|        |  +--:(technology-null)
|        |  |  +--rw tech-null?       empty
|        |  +--:(technology-string)
|        |     +--rw ipv4-icmp?       string
|        +--rw tp-tools
|        |  +--rw connectivity-verification?   boolean
|        |  +--rw continuity-check?            boolean
```

```
|        |  +--rw path-discovery?              boolean
|        +--rw root?
|        +--rw oam-layers* [index]
|           +--rw index                       uint16
|           +--rw level?                      int32
|           +--rw (tp-location)?
|              +--:(mac-address)
|              |  +--rw mac-address-location?  yang:mac-address
|              +--:(ipv4-address)
|              |  +--rw ipv4-location?         inet:ipv4-address
|              +--:(ipv6-location)
|              |  +--rw ipv6-address?          inet:ipv6-address
|              +--:(tunnel-location)
|              |  +--rw tunnel-location?       uint32
|              +--:(ip-prefix-location)
|              |  +--rw ip-prefix-location?    inet:ip-prefix
|              +--:(route-dist-location)
|              |  +--rw route-dist-location?   uint32
|              +--:(group-ip-address-location)
|              |  +--rw group-ip-address-location?   IP-Multicast
                                                      -Group-Address
|              +--:(as-number-location)
|              |  +--rw as-number-location?        inet:as-number
|              +--:(lsp-id-location)
|              |  +--rw lsp-id-location?             string
|              +--:(system-id-location)
|                 +--rw system-id-location?          router-id
+--:(mac-location-type)
|  +--rw test-point-mac-address-location-list
|     +--rw test-point-locations* [mac-address-location]
|        +--rw mac-address-location    yang:mac-address
|        +--rw (technology)?
|        |  +--:(technology-null)
|        |  |  +--rw tech-null?              empty
|        |  +--:(technology-string)
|        |     +--rw ipv4-icmp?              string
|        +--rw tp-tools
|        |  +--rw connectivity-verification?   boolean
|        |  +--rw continuity-check?            boolean
|        |  +--rw path-discovery?              boolean
|        +--rw root?
|        +--rw oam-layers* [index]
|           +--rw index                       uint16
|           +--rw level?                      int32
|           +--rw (tp-location)?
|              +--:(mac-address)
|              |  +--rw mac-address-location?  yang:mac-address
|              +--:(ipv4-address)
```

```
|              |  +--rw ipv4-location?         inet:ipv4-address
|              +--:(ipv6-location)
|              |  +--rw ipv6-address?          inet:ipv6-address
|              +--:(tunnel-location)
|              |  +--rw tunnel-location?       uint32
|              +--:(ip-prefix-location)
|              |  +--rw ip-prefix-location     inet:ip-prefix
|              +--:(route-dist-location)
|              |  +--rw route-dist-location?   uint32
|              +--:(group-ip-address-location)
|              |  +--rw group-ip-address-location?   IP-Multicast
                                                      -Group-Address
|              +--:(as-number-location)
|              |  +--rw as-number-location?        inet:as-number
|              +--:(lsp-id-location)
|              |  +--rw lsp-id-location?           string
|              +--:(system-id-location)
|                 +--rw system-id-location?        router-id
+--:(tunnel-location-type)
|  +--rw test-point-tunnel-location-list
|     +--rw test-point-locations* [tunnel-location]
|        +--rw tunnel-location    uint32
|        +--rw vrf?               routing-instance-ref
|        +--rw (technology)?
|        |  +--:(technology-null)
|        |  |  +--rw tech-null?         empty
|        |  +--:(technology-string)
|        |     +--rw ipv4-icmp?         string
|        +--rw tp-tools
|        |  +--rw connectivity-verification?   boolean
|        |  +--rw continuity-check?            boolean
|        |  +--rw path-discovery?              boolean
|        +--rw root?
|        +--rw oam-layers* [index]
|           +--rw index                        uint16
|           +--rw level?                       int32
|           +--rw (tp-location)?
|              +--:(mac-address)
|              |  +--rw mac-address-location?  yang:mac-address
|              +--:(ipv4-address)
|              |  +--rw ipv4-location?         inet:ipv4-address
|              +--:(ipv6-location)
|              |  +--rw ipv6-address?          inet:ipv6-address
|              +--:(tunnel-location)
|              |  +--rw tunnel-location?       uint32
|              +--:(ip-prefix-location)
|              |  +--rw ip-prefix-location     inet:ip-prefix
|              +--:(route-dist-location)
```

```
|              |  +--rw route-dist-location?   uint32
|              +--:(group-ip-address-location)
|              |  +--rw group-ip-address-location?   IP-Multicast
                                                      -Group-Address
|              +--:(as-number-location)
|              |  +--rw as-number-location?        inet:as-number
|              +--:(lsp-id-location)
|              |  +--rw lsp-id-location?           string
|              +--:(system-id-location)
|                 +--rw system-id-location?        router-id
+--:(ip-prefix-location-type)
|  +--rw test-point-ip-prefix-location-list
|     +--rw test-point-locations* [ip-prefix-location]
|        +--rw ip-prefix-location    inet:ip-prefix
|        +--rw vrf?                  routing-instance-ref
|        +--rw (technology)?
|        |  +--:(technology-null)
|        |  |  +--rw tech-null?            empty
|        |  +--:(technology-string)
|        |     +--rw ipv4-icmp?            string
|        +--rw tp-tools
|        |  +--rw connectivity-verification?   boolean
|        |  +--rw continuity-check?            boolean
|        |  +--rw path-discovery?              boolean
|        +--rw root?
|        +--rw oam-layers* [index]
|           +--rw index                        uint16
|           +--rw level?                       int32
|           +--rw (tp-location)?
|              +--:(mac-address)
|              |  +--rw mac-address-location?  yang:mac-address
|              +--:(ipv4-address)
|              |  +--rw ipv4-location          inet:ipv4-address
|              +--:(ipv6-location)
|              |  +--rw ipv6-address?          inet:ipv6-address
|              +--:(tunnel-location)
|              |  +--rw tunnel-location?       uint32
|              +--:(ip-prefix-location)
|              |  +--rw ip-prefix-location?    inet:ip-prefix
|              +--:(route-dist-location)
|              |  +--rw route-dist-location?         uint32
|              +--:(group-ip-address-location)
|              |  +--rw group-ip-address-location?   IP-Multicast
                                                      -Group-Address
|              +--:(as-number-location)
|              |  +--rw as-number-location?        inet:as-number
|              +--:(lsp-id-location)
|              |  +--rw lsp-id-location?           string
```

```
|              +--:(system-id-location)
|                 +--rw system-id-location?        router-id
+--:(route-distinguisher-location-type)
|  +--rw test-point-route-dist-location-list
|     +--rw test-point-locations* [route-dist-location]
|        +--rw route-dist-location    uint32
|        +--rw vrf?                   routing-instance-ref
|        +--rw (technology)?
|        |  +--:(technology-null)
|        |  |  +--rw tech-null?             empty
|        |  +--:(technology-string)
|        |     +--rw ipv4-icmp?             string
|        +--rw tp-tools
|        |  +--rw connectivity-verification?   boolean
|        |  +--rw continuity-check?            boolean
|        |  +--rw path-discovery?              boolean
|        +--rw root?
|        +--rw oam-layers* [index]
|           +--rw index                        uint16
|           +--rw level?                       int32
|           +--rw (tp-location)?
|              +--:(mac-address)
|              |  +--rw mac-address-location?  yang:mac-address
|              +--:(ipv4-address)
|              |  +--rw ipv4-location          inet:ipv4-address
|              +--:(ipv6-location)
|              |  +--rw ipv6-address           inet:ipv6-address
|              +--:(tunnel-location)
|              |  +--rw tunnel-location?       uint32
|              +--:(ip-prefix-location)
|              |  +--rw ip-prefix-location?    inet:ip-prefix
|              +--:(route-dist-location)
|              |  +--rw route-dist-location?   uint32
|              +--:(group-ip-address-location)
|              |  +--rw group-ip-address-location?   IP-Multicast
|                                                      -Group-Address
|              +--:(as-number-location)
|              |  +--rw as-number-location?        inet:as-number
|              +--:(lsp-id-location)
|              |  +--rw lsp-id-location?           string
|              +--:(system-id-location)
|                 +--rw system-id-location?        router-id
+--:(group-ip-address-location-type)
|  +--rw test-point-group-ip-address-location-list
|     +--rw test-point-locations* [group-ip-address-location]
|        +--rw group-ip-address-location    IP-Multicast
                                                -Group-Address
|        +--rw vrf?                         routing-instance-ref
```

```
|        +--rw (technology)?
|        |  +--:(technology-null)
|        |  |  +--rw tech-null?                    empty
|        |  +--:(technology-string)
|        |     +--rw ipv4-icmp?                    string
|        +--rw tp-tools
|        |  +--rw connectivity-verification?   boolean
|        |  +--rw continuity-check?            boolean
|        |  +--rw path-discovery?              boolean
|        +--rw root?
|        +--rw oam-layers* [index]
|           +--rw index                        uint16
|           +--rw level?                       int32
|           +--rw (tp-location)?
|              +--:(mac-address)
|              |  +--rw mac-address-location?  yang:mac-address
|              +--:(ipv4-address)
|              |  +--rw ipv4-location?         inet:ipv4-address
|              +--:(ipv6-location)
|              |  +--rw ipv6-address?          inet:ipv6-address
|              +--:(tunnel-location)
|              |  +--rw tunnel-location?       uint32
|              +--:(ip-prefix-location)
|              |  +--rw ip-prefix-location?    inet:ip-prefix
|              +--:(route-dist-location)
|              |  +--rw route-dist-location?   uint32
|              +--:(group-ip-address-location)
|              |  +--rw group-ip-address-location?   IP-Multicast
                                                      -Group-Address
|              +--:(as-number-location)
|              |  +--rw as-number-location?        inet:as-number
|              +--:(lsp-id-location)
|              |  +--rw lsp-id-location?             string
|              +--:(system-id-location)
|                 +--rw system-id-location?          router-id
|     +--:(group-as-number-location-type)
|     |  +--rw test-point-as-number-location-list
|     |     +--rw test-point-locations* [as-number-location]
|     |        +--rw as-number-location    inet:as-number
|     |        +--rw vrf?                  routing-instance-ref
|     |        +--rw (technology)?
|     |        |  +--:(technology-null)
|     |        |  |  +--rw tech-null?            empty
|     |        |  +--:(technology-string)
|     |        |     +--rw ipv4-icmp?            string
|     |        +--rw tp-tools
|     |        |  +--rw connectivity-verification?   boolean
|     |        |  +--rw continuity-check?            boolean
```

```
|        |  +--rw path-discovery?              boolean
|        +--rw root?
|        +--rw oam-layers* [index]
|           +--rw index                       uint16
|           +--rw level?                      int32
|           +--rw (tp-location)?
|              +--:(mac-address)
|              |  +--rw mac-address-location?  yang:mac-address
|              +--:(ipv4-address)
|              |  +--rw ipv4-location?         inet:ipv4-address
|              +--:(ipv6-location)
|              |  +--rw ipv6-address?          inet:ipv6-address
|              +--:(tunnel-location)
|              |  +--rw tunnel-location        uint32
|              +--:(ip-prefix-location)
|              |  +--rw ip-prefix-location?    inet:ip-prefix
|              +--:(route-dist-location)
|              |  +--rw route-dist-location?         uint32
|              +--:(group-ip-address-location)
|              |  +--rw group-ip-address-location?   IP-Multicast
                                                      -Group-Address
|              +--:(as-number-location)
|              |  +--rw as-number-location?        inet:as-number
|              +--:(lsp-id-location)
|              |  +--rw lsp-id-location?           string
|              +--:(system-id-location)
|                 +--rw system-id-location?        router-id
+--:(group-lsp-id-location-type)
|  +--rw test-point-lsp-id-location-list
|     +--rw test-point-locations* [lsp-id-location]
|        +--rw lsp-id-location    string
|        +--rw vrf?               routing-instance-ref
|        +--rw (technology)?
|        |  +--:(technology-null)
|        |  |  +--rw tech-null?         empty
|        |  +--:(technology-string)
|        |     +--rw ipv4-icmp?         string
|        +--rw tp-tools
|        |  +--rw connectivity-verification?   boolean
|        |  +--rw continuity-check?            boolean
|        |  +--rw path-discovery?              boolean
|        +--rw root?
|        +--rw oam-layers* [index]
|           +--rw index                       uint16
|           +--rw level?                      int32
|           +--rw (tp-location)?
|              +--:(mac-address)
|              |  +--rw mac-address-location?  yang:mac-address
```

```
|              +--:(ipv4-address)
|              |  +--rw ipv4-location          inet:ipv4-address
|              +--:(ipv6-location)
|              |  +--rw ipv6-address?          inet:ipv6-address
|              +--:(tunnel-location)
|              |  +--rw tunnel-location?       uint32
|              +--:(ip-prefix-location)
|              |  +--rw ip-prefix-location?    inet:ip-prefix
|              +--:(route-dist-location)
|              |  +--rw route-dist-location?         uint32
|              +--:(group-ip-address-location)
|              |  +--rw group-ip-address-location?   IP-Multicast
|                                                     -Group-Address
|              +--:(as-number-location)
|              |  +--rw as-number-location?        inet:as-number
|              +--:(lsp-id-location)
|              |  +--rw lsp-id-location?            string
|              +--:(system-id-location)
|                 +--rw system-id-location?         router-id
|           +--:(group-system-id-location-type)
|              +--rw test-point-system-info-location-list
|                 +--rw test-point-locations* [system-id-location]
|                    +--rw system-id-location    inet:uri
|                    +--rw vrf?                  routing-instance-ref
|                    +--rw (technology)?
|                    |  +--:(technology-null)
|                    |  |  +--rw tech-null?            empty
|                    |  +--:(technology-string)
|                    |     +--rw ipv4-icmp?            string
|                    +--rw tp-tools
|                    |  +--rw connectivity-verification?   boolean
|                    |  +--rw continuity-check?            boolean
|                    |  +--rw path-discovery?              boolean
|                    +--rw root?
|                    +--rw oam-layers* [index]
|                       +--rw index                      uint16
|                       +--rw level?                     int32
|                       +--rw (tp-location)?
|                          +--:(mac-address)
|                          |  +--rw mac-address-location?  yang:mac-address
|                          +--:(ipv4-address)
|                          |  +--rw ipv4-location?         inet:ipv4-address
|                          +--:(ipv6-location)
|                          |  +--rw ipv6-address?          inet:ipv6-address
|                          +--:(tunnel-location)
|                          |  +--rw tunnel-location?       uint32
|                          +--:(ip-prefix-location)
|                          |  +--rw ip-prefix-location     inet:ip-prefix
```

```
                      +--:(route-dist-location)
                      |  +--rw route-dist-location?         uint32
                      +--:(group-ip-address-location)
                      |  +--rw group-ip-address-location?   IP-Multicast
                                                          -Group-Address
                      +--:(as-number-location)
                      |  +--rw as-number-location?        inet:as-number
                      +--:(lsp-id-location)
                      |  +--rw lsp-id-location?           string
                      +--:(system-id-location)
                         +--rw system-id-location?        router-id

                           data hierarchy of OAM
```

4. OAM YANG Module

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

```
module ietf-connectionless-oam {
    yang-version 1.1;
        namespace "urn:ietf:params:xml:ns:yang:ietf-connectionless-oam";

    prefix coam;

    import ietf-yang-schema-mount {
     prefix yangmnt;
    }

    import ietf-network{
      prefix nd;
    }
    import ietf-yang-types {
      prefix yang;
    }
    import ietf-interfaces {
      prefix if;
    }
    import ietf-inet-types {
      prefix inet;
    }
    import ietf-network-instance {
      prefix "ni";
    }

    organization "IETF LIME Working Group";
    contact

        "Deepak Kumar dekumar@cisco.com
```

```
          Qin Wu       bill.wu@huawei.com
          S Raghavan   srihari@cisco.com
          Zitao Wang   wangzitao@huawei.com
          R Rahman     rrahman@cisco.com";

    description

      "This YANG module defines the generic configuration,
       data model, statistics for connectionless 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 2017-02-08 {
      description
        "Initial revision. - 10 version";
      reference "";
    }
    /* features */
    feature connection-less {
      description
        "This feature indicates that OAM solution is connection less.";
    }
    feature continuity-check {
      description
        "This feature indicates that the server supports
         executing continuity check OAM command and
         returning a response. Servers that do not advertise
         this feature will not support executing
         continuity check command or rpc model for
         continuity check command.";
    }
    feature path-discovery {
        description
        "This feature indicates that the server supports
         executing path discovery OAM command and
         returning a response. Servers that do not advertise
         this feature will not support executing
         path discovery command or rpc model for
         path discovery command.";
    }

    /* Identities */
    /* typedefs */

         typedef router-id {
```

```
       type yang:dotted-quad;
       description
         "A 32-bit number in the dotted quad format assigned to each
          router. This number uniquely identifies the router within an
          Autonomous System.";
     }

      typedef routing-instance-ref {

        type leafref {
          path "/ni:network-instances/ni:network-instance/ni:name";
        }
        description
        "This type is used for leafs that reference a routing instance
         configuration.";
      }

       typedef IPv4-Multicast-Group-Address {
            type string {
             pattern '(2((2[4-9])|(3[0-9]))\.)'
             +'(([0-9]|[1-9][0-9]|1[0-9][0-9]|'
             +'2[0-4][0-9]|25[0-5])\.){2}'
             +'([0-9]|[1-9][0-9]|1[0-9][0-9]'
             +'|2[0-4][0-9]|25[0-5])';
            }
            description
              "The IPv4-Multicast-Group-Address type
               represents an IPv4 multicast address
               in dotted-quad notation.";
            reference "RFC4607";
       } // typedef IPv4-Multicast-Group-Address
       typedef IPv6-Multicast-Group-Address {
            type string {
            pattern
            '(((FF|ff)[0-9a-fA-F]{2}):)([0-9a-fA-F]'
             +'{0,4}:){0,5}((([0-9a-fA-F]{0,4}:)?'
             +'(:|[0-9a-fA-F]{0,4}))|(((25[0-5]|2[0-4]'
             +'[0-9]|[01]?[0-9]?[0-9])\.){3}(25[0-5]|'
             +'2[0-4][0-9]|[01]?[0-9]?[0-9])))';
            pattern
            '(([^:]+:){6}(([^:]+:[^:]+)|'
             +'(.*\..*))|((([^:]+:)*[^:]+)'
             +'?::(([^:]+:)*[^:]+)?)';
             }
            description
             "The IPv6-Multicast-Group-Address
              type represents an IPv6 address in full,
              mixed, shortened, and shortened-mixed
```

```
              notation.";
            reference "RFC4291 2.7.
            ietf-inet-types:ipv6-address";
      }
      typedef IP-Multicast-Group-Address {
            type union {
              type IPv4-Multicast-Group-Address;
              type IPv6-Multicast-Group-Address;
            }
            description
             "The IP-Multicast-Group-Address type
              represents an IP multicast address and
              is IP version neutral. The format of the
              textual representations implies the IP version.";
      }

    identity address-attribute-types {

      description
      "This is base identity of address
          attribute types which are ip-prefix,
      bgp, tunnel, pwe3, vpls, etc.";
    }

    typedef address-attribute-type {
      type identityref {
        base address-attribute-types;
      }
      description
          "Target address attribute type.";
    }

    typedef opaque-tlv-type {
      type identityref {
        base opaque-tlv-type-id;
      }
      description
          "Opaque TLV type definition.";
    }

    identity time-resolution{
      description
        "Time interval resolution";
    }

    identity hours {
      base time-resolution;
      description
```

```
        "Hours";
    }

    identity minutes {
      base time-resolution;
      description
        "Minutes";
    }

    identity seconds {
      base time-resolution;
      description
        "Seconds";

    }

    identity milliseconds {
      base time-resolution;
      description
        "Milliseconds";
    }

    identity microseconds {

      base time-resolution;
      description
        "Microseconds";
    }

    identity nanoseconds {
      base time-resolution;
      description
        "Nanoseconds";
    }
        identity opaque-tlv-type-id {
      description
          "Base identity for opaque tlv types.";
    }
    identity default-tlv-type-id {
      base opaque-tlv-type-id;
      description
          "Default or unknown TLV type id.";
    }

    /* groupings */
    grouping cc-session-statsitics {
      description "Grouping for session statistics.";
      container cc-session-statistics {
```

```
        description "cc session counters";
        leaf session-count {
           type uint32;
           description
                   "Number of cc sessions.";
        }
        leaf session-up-count {
           type uint32;
           description
                   "Number of sessions which are up.";
        }
        leaf session-down-count {
           type uint32;
           description
                   "Number of sessions which are down.";
        }
        leaf session-admin-down-count {
           type uint32;
           description
                   "Number of sessions which are admin-down.";
        }
      }
    }

    grouping session-packet-statistics {
      description "Grouping for per session packet statistics";
      container session-packet-statistics {

        description "Per session packet statistics.";
        leaf rx-packet-count {
          type uint32;
          description
                  "Total number of received OAM packet count.";
        }
        leaf tx-packet-count {
          type uint32;
          description
                  "Total number of transmitted OAM packet count.";
        }
        leaf rx-bad-packet {
          type uint32;
          description
                  "Total number of received bad OAM packet.";
        }
        leaf tx-packet-failed {
          type uint32;
          description
                  "Total number of send OAM packet failed.";
```

```
        }
      }
    }

    grouping cc-per-session-statistics {
      description "Grouping for per session statistics";
      container cc-per-session-statistics {
        description "per session statistics.";
        leaf create-time {
          type yang:date-and-time;
          description
                  "Time and date when session is created.";
        }
        leaf last-down-time {
          type yang:date-and-time;
          description
                  "Time and date last time session is down.";
        }
        leaf last-up-time {
          type yang:date-and-time;
          description
                  "Time and date last time session is up.";
        }
        leaf down-count {
          type uint32;
          description
                  "Total down count.";
        }
        leaf admin-down-count {
          type uint32;
          description
                  "Total down count.";
        }
        uses session-packet-statistics;
      }

    }

    grouping session-error-statistics {
      description
          "Grouping for per session error statistics";
      container session-error-statistics {
        description "Per session error statistics.";
        leaf packet-drops-count {
          type uint32;
          description
                  "Total received packet drops count.";
        }
```

```
        leaf packet-reorder-count {

          type uint32;
          description
                  "Total received packet reordered count.";
        }
        leaf packets-out-of-seq-count {
          type uint32;
          description
                  "Total received out of sequence count.";
        }
        leaf packets-dup-count {
          type uint32;
          description
                  "Total received packet duplicates count.";
        }
      }
    }

    grouping session-delay-statistics {
      description
          "Grouping for per session delay statistics";
      container session-delay-statistics {
        description
                "Session delay summarised information.";
        leaf time-resolution-value {
          type identityref {
            base time-resolution;
          }
          description
                  "Time units among choice of s,ms,ns etc.";
        }
        leaf min-delay-value {
          type uint32;
          description
                  "Minimum delay value observed.";
        }
        leaf max-delay-value {
          type uint32;
          description
                  "Maximum delay value observed.";
        }
        leaf average-delay-value {
          type uint32;
          description
                  "Average delay value observed.";
        }
```

```
      }
    }

    grouping session-jitter-statistics {
      description
          "Grouping for per session jitter statistics";
      container session-jitter-statistics {
        description
                "Session jitter summarised information.";
        leaf time-resolution-value {
          type identityref {
            base time-resolution;
          }
          description
                  "Time units among choice of s,ms,ns etc.";
        }
        leaf min-jitter-value {
          type uint32;
          description
                  "Minimum jitter value observed.";
        }
        leaf max-jitter-value {
          type uint32;
          description
                  "Maximum jitter value observed.";
        }
        leaf average-jitter-value {
          type uint32;
          description
                  "Average jitter value observed.";
        }
      }
    }

    grouping session-path-verification-statistics {
      description
          "Grouping for per session path verification statistics";
      container session-path-verification-statistics{
        description
                "OAM per session path verification statistics.";
        leaf verified-count {
          type uint32;
          description
                  "Total number of OAM packets that
                  went through a path as intended.";
        }
        leaf failed-count {
          type uint32;
```

```
          description
                  "Total number of OAM packets that
                  went through an unintended path.";
        }
      }
    }

    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.";
          }
        }
        default "on-demand";
        description
          "Session type enum";
      }
    }
        grouping opaque-info-tlv {
      description
          "Opaque information as a TLV.";
      leaf type {
       type opaque-tlv-type;
       description "TLV type.";
      }
      leaf length {
       type uint16;
       description "TLV length.";
      }
      leaf value {
       type yang:hex-string;
       description "TLV value.";
      }
    }

    grouping opaque-tlvs {
      description
          "Opaque TLVs.";
      container opaque-tlvs {
```

```
        description
                "Opaque TLVs container.";
        list opaque-tlvs-list {
          description
                  "Opaque TLVs list.";
          uses opaque-info-tlv;
        }
      }
    }
    identity tp-address-type {
      description
        "Test point address type";
    }

    identity mac-address-type {
      base tp-address-type;
      description
        "MAC address type";
    }

    identity ipv4-address-type {
      base tp-address-type;
      description
        "IPv4 address type";
    }

    identity ipv6-address-type {
      base tp-address-type;
      description
        "IPv6 address type";
    }

    identity tp-attribute-type {

      base tp-address-type;
      description
        "Test point attribute type";
    }

    identity system-id-address-type {
      base tp-address-type;
      description
        "System id address type";
    }

    identity lsp-id-address-type {
      base tp-address-type;
      description
```

```
        "LSP ID address type";
    }

    identity as-number-address-type {
      base tp-address-type;
      description
        "AS number address type";
    }

    identity group-ip-address-type {
      base tp-address-type;
      description
        "Group IP address type";
    }

    identity route-distinguisher-address-type {
      base tp-address-type;
      description
        "Route Distinguisher address type";
    }

    identity ip-prefix-address-type {
      base tp-address-type;
      description
        "IP prefix address type";
    }

    identity tunnel-address-type {
      base tp-address-type;
      description
        "Tunnel address type";
    }

    grouping tp-address {
      leaf tp-location-type-value {
        type identityref {
          base tp-address-type;
        }
        description "Test point address type.";
      }

      choice tp-address {

        case mac-address {
          when "'tp-location-type-value' = 'mac-address-type'" {
            description "MAC address type";
          }
          leaf mac-address {
```

```
            type yang:mac-address;
            description
            "MAC Address";
          }
          description
          "MAC Address based MP Addressing.";
        }
        case ipv4-address {
          when "'tp-location-type-value' = 'ipv4-address-type'" {
            description "IPv4 address type";
          }
          leaf ipv4-address {
            type inet:ipv4-address;
            description
            "IPv4 Address";
          }
          description
          "IP Address based MP Addressing.";
        }
        case ipv6-address {
          when "'tp-location-type-value' = 'ipv6-address-type'" {
            description "IPv6 address type";
          }
          leaf ipv6-address {
            type inet:ipv6-address;
            description
            "IPv6 Address";

          }
          description
          "ipv6 Address based MP Addressing.";
        }

        case tp-attribute {
          when "'tp-location-type-value' = 'tp-attribute-type'" {
            description "Test point attribute type";
          }
          leaf tp-attribute-type {
            type address-attribute-type;
             description
             "Test point type.";
          }
          choice tp-attribute-value {
             description
            "Test point value.";
            case ip-prefix {
              leaf ip-prefix {
                type inet:ip-prefix;
```

```
            description
            "IP prefix.";
          }
        }
        case bgp {
          leaf bgp {
            type inet:ip-prefix;
            description
            "BGP Labeled Prefix ";
          }
        }
        case tunnel {
          leaf tunnel-interface {
            type uint32;
            description
            "VPN Prefix ";
          }
        }

        case pw {
          leaf remote-pe-address{
           type inet:ip-address;
           description
           "Remote pe address.";
          }
          leaf pw-id {
           type uint32;
           description
           "Pseudowire id.";
          }
        }
        case vpls {
          leaf route-distinguisher {
           type uint32;
           description
           "Route Distinguisher(8 octets).";
          }
          leaf sender-ve-id{
           type uint32;
           description
                       "Sender's VE ID.";
          }
          leaf receiver-ve-id{
           type uint32;
           description
           "Receiver's VE ID.";
          }
        }
```

```
            case mpls-mldp{
             choice root-address{
              description
              "Root address choice.";
              case ip-address{
               leaf source-address{
                type inet:ip-address;
                description
                                "IP address.";
               }
               leaf group-ip-address{
                type IP-Multicast-Group-Address;
                description
                                "Group ip address.";
               }
              }
              case vpn{
               leaf as-number{
                type inet:as-number;
                description
                                "AS number.";
               }
              }
              case global-id{
               leaf lsp-id{
                type string;
                description
                "LSP id.";
               }
              }
             }
            }
          }
        }

        case system-info {
          when "’tp-location-type-value’ = ’system-id-address-type’" {
           description "System id address type";
          }
          leaf system-id {
            type router-id;
            description
            "System ID assigned to this node.";
          }

        }
        description
        "TP Addressing.";
```

```
      }
      description
      "TP Address";
    }

    grouping tp-address-vrf {
      description
        "Test point address with VRF.";
      leaf vrf {
        type routing-instance-ref;
         description
         "The vrf is used to describe the
          corresponding network instance";
      }

      uses tp-address;
    }

    grouping connectionless-oam-layers {
      list oam-layers {
         key "index";
         leaf index {
          type uint16 {
            range "0..65535";
          }
                  description
          "Index";
         }
        leaf level {
         type int32 {
         range "-1..1";
         }
         default 0;
         description
                 "Level 0 indicates default level,
                 -1 means server and +1 means client layer.
          In relationship 0 means same layer.";
        }
                choice tp-location {

        case mac-address {
          leaf mac-address-location {
           type yang:mac-address;
           description
           "MAC Address";
          }
          description
          "MAC Address based MP Addressing.";
```

```
        }
        case ipv4-address {
          leaf ipv4-location {
           type inet:ipv4-address;
           description
           "Ipv4 Address";
          }
          description
          "IP Address based MP Addressing.";
        }
        case ipv6-location {
          leaf ipv6-address {
           type inet:ipv6-address;
           description
           "IPv6 Address";
          }
          description
          "IPv6 Address based MP Addressing.";
        }

        case tunnel-location{
         leaf tunnel-location{
                   type uint32;
                   description
                   "VPN Prefix";
                  }
                  description
                  "Tunnel location";
                  }

                  case ip-prefix-location{
                   leaf ip-prefix-location{
                    type inet:ip-prefix;
                    description
                    "IP prefix location";
                   }
                   description
                   "IP prefix location";
                  }

                  case route-dist-location{
                   leaf route-dist-location{
                    type uint32;
                    description
                    "Route Distinguisher (8 octets)";
                   }
                   description
                   "Route distinguisher location";
```

```
                }

                case group-ip-address-location{
                 leaf group-ip-address-location{
                  type IP-Multicast-Group-Address;
                  description
                  "Group IP address location";
                 }
                 description
                 "Group IP address";
                }

                case as-number-location{
                 leaf as-number-location{
                  type inet:as-number;
                  description
                  "AS number location";
                 }
                 description
                 "AS number";
                }

                case lsp-id-location{
                 leaf lsp-id-location{
                  type string;
                  description
                  "LSP id";
                 }
                 description
                 "LSP ID";
                }

                case system-id-location{
                 leaf system-id-location{
                  type router-id;
                  description
                  "System id location";
                 }
                 description
                 "System ID";
                }

        description
          "TP location.";

          }
        ordered-by user;
         description
```

```
         "List of related oam layers.
                 0 means they are in same level, especially
                 interworking scenarios of stiching multiple
                 technology at same layer. -1 means server layer,
                 for eg:- in case of Overlay and Underlay,
                 Underlay is server layer for Overlay Test Point.
                 +1 means client layer, for eg:- in case of
                 Service OAM and Transport OAM, Service OAM is client
                 layer to Transport OAM.";
      }
      description
        "Connectionless related OAM layer";
    }

    grouping tp-technology {
       choice technology {
          default technology-null;
          case technology-null {
            description
            "This is a placeholder when no technology is needed.";
            leaf tech-null {
             type empty;
             description
             "There is no technology define";
            }
          }
          description
          "Technology choice null";
          case technology-string {
            description
            "OAM technology string";
            leaf ipv4-icmp {
              type string;
              description
              "Name to identify oam technology";
            }
        }
       }

       description
        "OAM Technology";
    }

    grouping tp-tools {
      description
        "Test Point OAM Toolset.";
      container tp-tools{
           leaf connectivity-verification{
```

```
            type boolean;
                description
                "A flag indicating whether or not the
                connectivity-verification function is supported.";
                reference
          "RFC 792: INTERNET CONTROL MESSAGE PROTOCOL.
                   RFC 4443: Internet Control Message Protocol (ICMPv6)
           for the Internet Protocol Version 6 (IPv6) Specification.
                   RFC 5085: Pseudowire Virtual Circuit Connectivity Verification
.
                   RFC 5880: Bidirectional Forwarding Detection.
                   RFC 5881: BFD for IPv4 and IPv6.
                   RFC 5883: BFD for Multihop Paths.
                   RFC 5884: BFD for MPLS Label Switched Paths.
                   RFC 5885: BFD for PW VCCV.
                   RFC 4379: LSP-PING.";
            }
            leaf continuity-check{
             type boolean;
                description
                "A flag indicating whether or not the
                continuity check function is supported.";
                reference
          "RFC 792: INTERNET CONTROL MESSAGE PROTOCOL.
                   RFC 4443: Internet Control Message Protocol (ICMPv6)
           for the Internet Protocol Version 6 (IPv6) Specification.
                   RFC 5880: Bidirectional Forwarding Detection.
                   RFC 5881: BFD for IPv4 and IPv6.
                   RFC 5883: BFD for Multihop Paths.
                   RFC 5884: BFD for MPLS Label Switched Paths.
                   RFC 5885: BFD for PW VCCV.
                   RFC 6450: Multicast Ping Protocol.";
            }
            leaf path-discovery{
             type boolean;
                description
                "A flag indicating whether or not the
                path discovery function is supported.";
             reference
          "RFC 792: INTERNET CONTROL MESSAGE PROTOCOL.
                   RFC 4443: Internet Control Message Protocol (ICMPv6)
           for the Internet Protocol Version 6 (IPv6) Specification.
                   RFC 4884: Extended ICMP to Support Multi-part Message.
                   RFC 5837:Extending ICMP for Interface
                   and Next-Hop Identification.
                   RFC 4379: LSP-PING.";
            }
            description
            "Container for test point OAM tools set.";
```

```
        }

    }

    grouping test-point-location-info {
      uses tp-technology;
      uses tp-tools;
      anydata root {
       yangmnt:mount-point root;
       description
           "Root for models supported per
        test point";
      }
      uses connectionless-oam-layers;
      description
        "Test point Location";
    }

    grouping test-point-locations {
      description "Group of test point locations.";
      leaf tp-location-type-value {
        type identityref {
          base tp-address-type;
        }
        description "Test point location type.";

      }
      choice location-type {
        case ipv4-location-type {
          when "'tp-location-type-value' = 'ipv4-address-type'" {
          description
            "When test point location type is equal to ipv4 address.";
          }
          container test-point-ipv4-location-list {
            list test-point-locations {
              key "ipv4-location";
              leaf ipv4-location {
                type inet:ipv4-address;
                description
                  "IPv4 Address.";
              }
              leaf vrf {
                type routing-instance-ref;
                description
                "The vrf is used to describe the
                 corresponding network instance";
              }
```

```
            uses test-point-location-info;

            ordered-by user;
            description
              "List of test point locations.";
          }
          description
          "Serves as top-level container
                      for test point location list.";
        }
      }
      case ipv6-location-type {
        when "'tp-location-type-value' = 'ipv6-address-type'" {
        description
          "when test point location is equal to ipv6 address";
        }
        container test-point-ipv6-location-list {
          list test-point-locations {
            key "ipv6-location";
            leaf ipv6-location {
              type inet:ipv6-address;
              description
                "IPv6 Address.";
            }
            leaf vrf {
              type routing-instance-ref;
              description
                              "The vrf is used to describe the
              corresponding network instance";
            }
            uses test-point-location-info;
            ordered-by user;
            description
              "List of test point locations.";
          }
          description
            "Serves as top-level container
                        for test point location list.";
        }
      }
      case mac-location-type {
        when "'tp-location-type-value' = 'mac-address-type'" {
          description
            "when test point location type is equal to mac address.";
        }
        container test-point-mac-address-location-list {
          list test-point-locations {
          key "mac-address-location";
```

```
            leaf mac-address-location {
              type yang:mac-address;
              description
                "MAC Address";
            }
            uses test-point-location-info;
            ordered-by user;
            description
             "List of test point locations.";
            }
            description
            "Serves as top-level container
                        for test point location list.";
          }
        }
        case tunnel-location-type {
         when "'tp-location-type-value' ="
                 +" 'tunnel-address-type'" {
         description
         "When test point location type
                 is equal to tunnel type.";
         }
          container test-point-tunnel-location-list {
            list test-point-locations {
            key "tunnel-location";
            leaf tunnel-location {
              type uint32;
              description
                "VPN Prefix";

            }
            leaf vrf {
             type routing-instance-ref;
             description
             "The vrf is used to describe the
              corresponding network instance";
            }
            uses test-point-location-info;
            ordered-by user;
            description
             "List of test point locations.";
            }
            description
            "Serves as top-level container
                        for test point location list.";
          }
        }
        case ip-prefix-location-type {
```

```
            when "'tp-location-type-value' = "
                +"'ip-prefix-address-type'" {
          description
          "When test point location
                  type is equal to ip prefix.";
          }
          container test-point-ip-prefix-location-list {
            list test-point-locations {
            key "ip-prefix-location";
            leaf ip-prefix-location {
              type inet:ip-prefix;
              description
              "IP Prefix";
            }
            leaf vrf {
              type routing-instance-ref;
              description
              "The vrf is used to describe the
               corresponding network instance";
            }
            uses test-point-location-info;
            ordered-by user;
            description
             "List of test point locations.";
            }
            description
             "Serves as top-level container
                         for test point location list.";
          }
        }
        case route-distinguisher-location-type {
          when "'tp-location-type-value' = "
                  +"'route-distinguisher-address-type'" {
          description
                      "When test point
                      location type is equal to
          route distinguiher.";
          }

          container test-point-route-dist-location-list {
            list test-point-locations {
            key "route-dist-location";
            leaf route-dist-location {
              type uint32;
              description
              "Route Distinguisher(8 octets).";
            }
            leaf vrf {
```

```
            type routing-instance-ref;
            description
            "The vrf is used to describe the
             corresponding network instance";
          }
          uses test-point-location-info;
          ordered-by user;
          description
           "List of test point locations.";
          }
          description
           "Serves as top-level container
                       for test point location list.";
        }
      }
      case group-ip-address-location-type {
        when "'tp-location-type-value' = "
                +"'group-ip-address-type'" {
          description
                      "When test point location type is equal to
           group ip address.";
        }
        container test-point-group-ip-address-location-list {
          list test-point-locations {
          key "group-ip-address-location";
          leaf group-ip-address-location {
            type IP-Multicast-Group-Address;
            description
            "Group IP address.";
          }
          leaf vrf {
            type routing-instance-ref;
            description
            "The vrf is used to describe the
             corresponding network instance";
          }
          uses test-point-location-info;
          ordered-by user;
          description
           "List of test point locations.";
          }

          description
            "Serves as top-level container for
                        test point location list.";
        }
      }
      case group-as-number-location-type {
```

```
          when "'tp-location-type-value' = "
                  +"'as-number-address-type'" {
            description
                        "When test point location type is equal to
             as-number.";
          }
          container test-point-as-number-location-list {
            list test-point-locations {
            key "as-number-location";
            leaf as-number-location {
              type inet:as-number;
              description
               "AS number.";
            }
            leaf vrf {
              type routing-instance-ref;
              description
              "The vrf is used to describe the
               corresponding network instance";
            }
            uses test-point-location-info;
            ordered-by user;
            description
             "List of test point locations.";
            }
            description
             "Serves as top-level container
                         for test point location list.";
          }
        }
        case group-lsp-id-location-type {
          when "'tp-location-type-value' = "
                  +"'lsp-id-address-type'" {
            description
                        "When test point location
                        type is equal to lspid.";
          }
          container test-point-lsp-id-location-list {
            list test-point-locations {
            key "lsp-id-location";
            leaf lsp-id-location {
              type string;
              description
              "LSP Id.";
            }
            leaf vrf {
              type routing-instance-ref;
               description
```

```
                "The vrf is used to describe the
                 corresponding network instance";
              }
              uses test-point-location-info;
              ordered-by user;
              description
               "List of test point locations.";
              }
              description
               "Serves as top-level container for
                           test point location list.";
            }
          }
          case group-system-id-location-type {
            when "'tp-location-type-value' = "
                    +"'system-id-address-type'" {
              description
                          "When test point location is equal to
               system info.";
            }
            container test-point-system-info-location-list {
              list test-point-locations {
              key "system-id-location";
              leaf system-id-location {
                type inet:uri;
                description
                 "System Id.";
              }
              leaf vrf {
                type routing-instance-ref;
                description
                "The vrf is used to describe the
                corresponding network instance";
              }
              uses test-point-location-info;
              ordered-by user;
              description
              "List of test point locations.";
              }
              description
              "Serves as top-level container for
                        test point location list.";
            }
          }
          description
            "Choice of address types.";
        }
      }
```

```
    augment "/nd:networks/nd:network/nd:node"{
      description
      "Augment test points of connectionless oam.";
      uses test-point-locations;

    }

    grouping path-discovery-data {
      description
          "Path discovery related data output from nodes.";
      container src-test-point {
        description "Source test point.";
        uses tp-address-vrf;
      }
      container dest-test-point {
        description "Destination test point.";
        uses tp-address-vrf;
      }
      leaf sequence-number {
        type uint64;
        description "Sequence number in data packets.";
      }
      leaf hop-cnt {
        type uint8;
        description "Hop count.";
      }

      uses session-packet-statistics;
      uses session-error-statistics;
      uses session-delay-statistics;
      uses session-jitter-statistics;

      container path-verification {
        description
                "Optional path verification related information.";
        leaf flow-info {
          type string;
          description
          "Informations that refers to the flow.";
        }
        uses session-path-verification-statistics;
      }

      container path-trace-info {
        description
                "Optional path trace per-hop test point information.
         The list has typically a single element for per-hop
         cases like path-discovery RPC but allows a list of
```

```
         hop related information for other types of
         data retrieval methods.";
        list path-trace-info-list {
          key "index";
          description
            "Path trace information list.";
          leaf index {
            type uint32;
            description
                        "Trace information index.";
          }

          uses tp-address-vrf;

          leaf timestamp-sec {
            type uint64;
            description
                        "Absolute timestamp in
                        seconds as per IEEE1588v2.";
          }
          leaf timestamp-nanosec {
            type uint32;
            description
                        "Fractional part in
                        nanoseconds as per IEEE1588v2.";
          }

          leaf ingress-intf-name {
            type if:interface-ref;
            description
            "Ingress interface name";
          }
          leaf egress-intf-name {
            type if:interface-ref;
            description
              "Egress interface name";
          }
          leaf queue-depth {
            type uint32;
            description
            "Length of the egress interface
                        queue of the interface.";
          }
          leaf transit-delay {
            type uint32;
            description
            "Time in nano seconds
                        packet spent transiting a node.";
```

```
          }

          leaf app-meta-data {
            type uint64;
            description
             "Application specific
                         data added by node.";
          }
                  uses opaque-tlvs;
        }
      }
    }

    grouping continuity-check-data {
      description
          "Continuity check data output from nodes.";
      container src-test-point {
        description "Source test point.";
        uses tp-address-vrf;

        leaf egress-intf-name {
          type if:interface-ref;
          description
            "Egress interface name";
        }
      }
      container dest-test-point {
        description
                "Destination test point.";
        uses tp-address-vrf;

        leaf ingress-intf-name {
          type if:interface-ref;
          description
            "Ingress interface name";

        }
      }
      leaf sequence-number {
        type uint64;
        description "Sequence number.";
      }
      leaf hop-cnt {
        type uint8;
        description "Hop count.";
      }

      uses session-packet-statistics;
```

```
      uses session-error-statistics;
      uses session-delay-statistics;
      uses session-jitter-statistics;
    }

    container cc-oper-data {
      if-feature continuity-check;
      config "false";
      description "CC operational information.";
      container cc-ipv4-sessions-statistics {
        description "CC ipv4 sessions";
        uses cc-session-statsitics;
      }
      container cc-ipv6-sessions-statistics {
        description "CC ipv6 sessions";
        uses cc-session-statsitics;
      }
    }
}
```

<CODE ENDS>

5. Connectionless model applicability

"ietf-connectionless-oam" model defined in this document provides technology-independent abstraction of key OAM constructs for connectionless protocols. This model can be further extended to include technology specific details, e.g., adding new data nodes with technology specific functions and parameters into proper anchor points of the base model, so as to develop a technology-specific connectionless OAM model.

This section demonstrates the usability of the connectionless YANG OAM data model to various connectionless OAM technologies, e.g., BFD, LSP ping. Note that, in this section, we only present several snippets of technology-specific model extensions for illustrative purposes. The complete model extensions should be worked on in respective protocol working groups.

5.1. BFD Extension

5.1.1. Augment Method

The following sections shows how the "ietf-connectionless-oam" model can be extended to cover BFD technology. For this purpose, a set of extension are introduced such as technology-type extension and test-point attributes extension.

Note that in BFD WG, there is a BFD yang data model [I-D.ietf-bfd-yang] to be produced. Users can choose to use "ietf-connectioless-oam" as basis and augment the "ietf-connectionless-oam" model with bfd specific details. The bfd specific details can be the grouping defined in the BFD model.

5.1.1.1. Technology type extension

No BFD technology type has been defined in the "ietf-connectionless-oam" model. Therefore a technology type extension is required in the model Extension.

The snippet below depicts an example of augmenting "bfd" type into the ietf-connectionless-oam":

```
augment "/nd:networks/nd:network/nd:node/"
+"coam:location-type/coam:ipv4-location-type"
+"/coam:test-point-ipv4-location-list/"
       +"coam:test-point-locations/coam:technology"
+"/coam:technology-string"
{
    leaf bfd{
   type string;
  }
}
```

5.1.1.2. Test point attributes extension

To support bfd technology, the "ietf-connectionless-oam" model can be extended and add bfd specific parameters under "test-point-location" list and/or add new location type such as "bfd over MPLS-TE" under "location-type".

5.1.1.2.1. Define and insert new nodes into corresponding test-point-location

In the "ietf-connectionless-oam" model, multiple "test-point-location" lists are defined under the "location-type" choice node. Therefore, to derive a model for some bfd technologies (such as ip single-hop, ip multi-hops, etc), data nodes for bfd specific details need to be added into corresponding "test-point-locations" list. In this section, we reuse some groupings which are defined in [I-D.ietf-bfd-yang] as following:

The snippet below shows how the "ietf-connectionless-oam" model can be extended to support "BFD IP single-hop":

```
augment "/nd:networks/nd:network/nd:node/"
+"coam:location-type/coam:ipv4-location-type"
+"/coam:test-point-ipv4-location-list/"
        +"coam:test-point-locations"
{
        container session-cfg {
          description "BFD IP single-hop session configuration";
          list sessions {
            key "interface dest-addr";
            description "List of IP single-hop sessions";
            leaf interface {
              type if:interface-ref;
              description
                "Interface on which the BFD session is running.";
            }
            leaf dest-addr {
              type inet:ip-address;
              description "IP address of the peer";
            }
            uses bfd:bfd-grouping-common-cfg-parms;
            uses bfd:bfd-grouping-echo-cfg-parms;
          }
        }
}
```

Similar augmentations can be defined to support other BFD technologies such as BFD IP multi-hop, BFD over MPLS, etc.

5.1.1.2.2. Add new location-type cases

In the "ietf-connectionless-oam" model, If there is no appropriate "location type" case that can be extended, a new "location-type" case can be defined and inserted into the "location-type" choice node.

Therefore, the model user can flexibly add "location-type" to support other type of test point which are not defined in the "ietf-connectionless-oam" model. In this section, we add a new "location-type" case and reuse some groupings which are defined in [I-D.ietf-bfd-yang] as follows:

The snippet below shows how the "ietf-connectionless-oam" model can be extended to support "BFD over MPLS-TE":

```
augment "/nd:networks/nd:network/nd:node/coam:location-type"{
 case te-location{
  list test-point-location-list{
   key "tunnel-name";
   leaf tunnel-name{
    type leafref{
 path "/te:te/te:tunnels/te:tunnel/te:name";
}
description
"point to a te instance.";
   }
    uses bfd:bfd-grouping-common-cfg-parms;
        uses bfd-mpls:bfd-encap-cfg;
  }
 }
}
```

Similar augmentations can be defined to support other BFD technologies such as BFD over LAG, etc.

5.1.2. Schema Mount

And anohter alternative method is using schema mount mechanism [I-D.ietf-netmod-schema-mount] in the "ietf-connectionless-oam". Within the "test-point-location" list, a "root" attribute is defined to provide a mounted point for models mounted per "test-point-location". Therefore, the "ietf-connectionless-oam" model can provide a place in the node hierarchy where other OAM YANG data models can be attached, without any special extension in the "ietf-connectionless-oam" YANG data models [I-D.ietf-netmod-schema-mount]. Note that the limitation of the Schema Mount method is it is not allowed to specify certain modules that are required to be mounted under a mount point.

The snippet below depicts the definition of "root" attribute.

```
      anydata root {
       yangmnt:mount-point root;
       description
      "Root for models supported per
        test point";
      }
```

The following section shows how the "ietf-connectionless-oam" model can use schema mount to support BFD technology.

5.1.2.1. BFD Modules be populated in schema-mount

To support BFD technology, "ietf-bfd-ip-sh" and "ietf-bfd-ip-mh" YANG modules might be populated in the "schema-mounts" container:

```
<schema-mounts
    xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-schema-mount">
  <mount-point>
    <module> ietf-connectionless-oam </module>
    <name>root</name>
    <use-schema>
      <name>root</name>
    </use-schema>
  </mount-point>
  <schema>
    <name>root</name>
    <module>
      <name>ietf-bfd-ip-sh </name>
      <revision>2016-07-04</revision>
      <namespace>
        urn:ietf:params:xml:ns:yang: ietf-bfd-ip-sh
      </namespace>
      <conformance-type>implement</conformance-type>
    </module>
    <module>
      <name>ietf-bfd-ip-mh </name>
      <revision> 2016-07-04</revision>
      <namespace>
        urn:ietf:params:xml:ns:yang: ietf-bfd-ip-mh
      </namespace>
      <conformance-type>implement</conformance-type>
    </module>
  </schema>
</schema-mounts>
```

and the " ietf-connectionless-oam " module might have:

```
<ietf-connectionless-oam
uri="urn:ietf:params:xml:ns:yang:ietf-connectionless-oam">
   ......
 <test-point-locations>
  <ipv4-location> 1.1.1.1</ipv4-location>
   ......
  <root>
   <ietf-bfd-ip-sh uri="urn:ietf:params:xml:ns:yang:ietf-bfd-ip-sh">
    <ip-sh>
     foo
     ......
    </ip-sh>
   </ietf-bfd-ip-sh>
   <ietf-bfd-ip-mh uri="urn:ietf:params:xml:ns:yang:ietf-bfd-ip-mh">
    <ip-mh>
     foo
     ......
    </ip-mh>
   </ietf-bfd-ip-mh>
  </root>
 </test-point-locations>
</ietf-connectionless-oam>
```

5.2. LSP ping extension

The following sections shows how the "ietf-connectionless-oam" model can be extended to support LSP ping technology. For this purpose, a set of extension are introduced such as technology-type extension and test-point attributes extension.

Note that in MPLS WG, there is a LSP Ping yang data model [I-D.draft-zheng-mpls-lsp-ping-yang-cfg] to be produced. Users can choose to use "ietf-connectioless-oam" as basis and augment the "ietf-connectionless-oam" model with LSP Ping specific details in the model extension. The LSP Ping specific details can be the grouping defined in the LSP ping model.

5.2.1. Technology type extension

No lsp-ping technology type has been defined in the "ietf-connectionless-oam" model. Therefore a technology type extension is required in the model extension.

The snippet below depicts an example of augmenting the "ietf-connectionless-oam" with "lsp-ping" type:

```
augment "/nd:networks/nd:network/nd:node/"
+"coam:location-type/coam:ipv4-location-type"
+"/coam:test-point-ipv4-location-list/"
        +"coam:test-point-locations/coam:technology"
+"/coam:technology-string"
{
   leaf lsp-ping{
   type string;
  }
}
```

5.2.2. Test point attributes extension

To support lsp-ping, the "ietf-connectionless-oam" model can be extended and add lsp-ping specific parameters can be defined and under "test-point-location" list.

User can reuse the attributes or groupings which are defined in [I-D.draft-zheng-mpls-lsp-ping-yang-cfg] as follows:

The snippet below depicts an example of augmenting the "test-point-locations" list with lsp ping attributes:

```
augment "/nd:networks/nd:network/nd:node/"
+"coam:location-type/coam:ipv4-location-type"
+"/coam:test-point-ipv4-location-list/"
        +"coam:test-point-locations"
{
list lsp-ping {
         key "lsp-ping-name";
         leaf lsp-ping-name {
          type string {
            length "1..31";
         }
        mandatory "true";
        description "LSP Ping test name.";
        ......
      }
```

6. Security Considerations

The YANG module defined in this memo is designed to be accessed via the NETCONF protocol [RFC6241]. The lowest NETCONF layer is the secure transport layer and the mandatory-to-implement secure transport is SSH [RFC6242]. The NETCONF access control model [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:

/nd:networks/nd:network/nd:node/coam:location-type/coam:ipv4-location-type/coam:test-point-ipv4-location-list/coam:test-point-locations/

/nd:networks/nd:network/nd:node/coam:location-type/coam:ipv6-location-type/coam:test-point-ipv6-location-list/coam:test-point-locations/

/nd:networks/nd:network/nd:node/coam:location-type/coam:mac-location-type/coam:test-point-mac-address-location-list/coam:test-point-locations/

/nd:networks/nd:network/nd:node/coam:location-type/coam:tunnel-location-type/coam:test-point-tunnel-address-location-list/coam:test-point-locations/

/nd:networks/nd:network/nd:node/coam:location-type/coam:ip-prefix-location-type/coam:test-point-ip-prefix-location-list/coam:test-point-locations/

/nd:networks/nd:network/nd:node/coam:location-type/coam:route-distinguisher-location-type/coam:test-point-route-dist-location-list/coam:test-point-locations/

/nd:networks/nd:network/nd:node/coam:location-type/coam:group-ip-address-location-type/coam:test-point-group-ip-address-location-list/coam:test-point-locations/

/nd:networks/nd:network/nd:node/coam:location-type/coam:group-as-number-location-type/coam:test-point-as-number-location-list/coam:test-point-locations/

/nd:networks/nd:network/nd:node/coam:location-type/coam:group-lsp-id-location-type/coam:test-point-lsp-id-location-list/coam:test-point-locations/

/nd:networks/nd:network/nd:node/coam:location-type/coam:group-system-id-location-type/coam:test-point-system-info-location-list/coam:test-point-locations/

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.

7. IANA Considerations

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

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

namespace: urn:ietf:params:xml:ns:yang:ietf-connectionless-oam

prefix: coam

reference: RFC XXXX

8. Acknowlegements

The authors of this document would like to thank Greg Mirsky and others for their sustainable review and comments, proposals to improve and stabilize document.

9. References

9.1. Normative References

[I-D.draft-zheng-mpls-lsp-ping-yang-cfg]
Zheng, L., Aldrin, S., Zheng, G., Mirsky, G., and R. Rahman, "Yang Data Model for LSP-PING", I-D draft-zheng-mpls-lsp-ping-yang-cfg, March 2016.

[I-D.ietf-bfd-yang]
Zheng, L., Rahman, R., Networks, J., Jethanandani, M., and G. Mirsky, "Yang Data Model for Bidirectional Forwarding Detection (BFD)", draft-ietf-bfd-yang-04 (work in progress), January 2017.

[I-D.ietf-i2rs-yang-network-topo]
Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A Data Model for Network Topologies", draft-ietf-i2rs-yang-network-topo-11 (work in progress), February 2017.

[I-D.ietf-lime-yang-connectionless-oam-methods]
Kumar, D., Wang, Z., Wu, Q., Rahman, R., and S. Raghavan, "Retrieval Methods YANG Data Model for Connectionless Operations, Administration, and Maintenance(OAM) protocols", draft-ietf-lime-yang-connectionless-oam-methods-00 (work in progress), October 2016.

[I-D.ietf-lime-yang-oam-model]
Kumar, D., Wu, Q., and Z. Wang, "Generic YANG Data Model for Connection Oriented Operations, Administration, and Maintenance(OAM) protocols", draft-ietf-lime-yang-oam-model-08 (work in progress), December 2016.

[I-D.ietf-netmod-schema-mount]
Bjorklund, M. and L. Lhotka, "YANG Schema Mount", draft-ietf-netmod-schema-mount-03 (work in progress), October 2016.

[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>.

[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>.

[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.

9.2. Informative References

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

[I-D.ietf-spring-sr-yang] Litkowski, S., Qu, Y., Sarkar, P., and J. Tantsura, "YANG Data Model for Segment Routing", draft-ietf-spring-sr-yang-05 (work in progress), October 2016.

[RFC6136] Sajassi, A., Ed. and D. Mohan, Ed., "Layer 2 Virtual Private Network (L2VPN) Operations, Administration, and Maintenance (OAM) Requirements and Framework", RFC 6136, DOI 10.17487/RFC6136, March 2011, <http://www.rfc-editor.org/info/rfc6136>.

[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>.

Authors' Addresses

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

Email: dekumar@cisco.com

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

Email: wangzitao@huawei.com

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

Email: bill.wu@huawei.com

Reshad Rahman
CISCO Systems
2000 Innovation Drive
KANATA, ONTARIO K2K 3E8
CANADA

Email: rrahman@cisco.com

Srihari Raghavan
CISCO Systems
TRIL INFOPARK SEZ, Ramanujan IT City
NEVILLE BLOCK, 2nd floor, Old Mahabalipuram Road
CHENNAI, TAMIL NADU 600113
INDIA

Email: srihari@cisco.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: November 16, 2017

D. Kumar
Cisco
M. Wang
Q. Wu
Huawei
R. Rahman
S. Raghavan
Cisco

May 15, 2017

Generic YANG Data Model for Connectionless Operations, Administration, and Maintenance(OAM) protocols
draft-ietf-lime-yang-connectionless-oam-05

Abstract

This document presents a base YANG Data model for connectionless Operations Administration, and Maintenance(OAM) protocols. It provides a technology-independent abstraction of key OAM constructs for connectionless protocols. The base model presented here can be extended to include technology specific details. This is leading to uniformity between OAM protocols and support both nested OAM workflows (i.e., performing OAM functions at different or same levels through a unified interface).

Status of This Memo

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

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

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

This Internet-Draft will expire on November 16, 2017.

Copyright Notice

Table of Contents

1. Introduction

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

1. Monitor networks connections (Reachability 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.

The different OAM tools may support connection-oriented technologies or connectionless technologies. In connection-oriented technologies, a connection is established prior to the transmission of data. In connectionless technologies, data is typically sent between end points without prior arrangement [RFC7276]. Note that the Connection-Oriented OAM YANG DATA model is defined in [I-D.ietf-lime-yang-oam-model].

In this document, we presents a base YANG Data model for connectionless OAM protocols. The generic YANG model for connectionless OAM only includes configuration data and state data. It can be used in conjunction with data retrieval method model [I-D.ietf-lime-yang-connectionless-oam-methods], which focuses on data retrieval procedures like RPC. However it also can be used independently of data retrieval method model.

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 terms are defined in [RFC6241] and are not redefined here:

- client
- configuration data
- server
- state data

The following terms are defined in [RFC6020] and are not redefined here:

- augment
- data model

o data node

The terminology for describing YANG data models is found in [RFC6020].

2.1. Terminology

TP - Test Point

MAC - Media Access Control

BFD - Bidirectional Forwarding Detection

RPC - A Remote Procedure Call, as used within the NETCONF protocol

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

3. Overview of the Connectionless OAM Model

At the top of the model, there is an 'cc-oper-data' container for session statistics. Grouping is also defined for common session statistics and these are applicable for proactive OAM sessions. Multiple 'test-point-locations' keyed using technology specific keys (eg., IPv4 address for IPv4 locations) are possible by augmented network nodes which are defined in [I-D.ietf-i2rs-yang-network-topo] to describe the network hierarchies and the inventory of nodes contained in a network. Each 'test-point-location' is chosen based on 'location-type' which when chosen, leads to a container that includes a list of 'test-point-locations' keyed by technology specific keys. Each test point location includes a 'test-point-location-info'. The 'test-point-location-info' includes 'tp-technology', 'tp-tools', and 'connectionless-oam-layers'. The groupings of 'tp-address' and 'tp-address-vrf' are kept out of 'test-point-location-info' to make it addressing agnostic and allow varied composition. Depending upon the choice of the 'location-type' (determined by the 'tp-address-vrf'), the containers differ in its composition of 'test-point-locations' while the 'test-point-location-info', is a common aspect of every 'test-point-location'. The vrf is used to describe the corresponding network instance. The 'tp-technology' indicate OAM technology details. The 'tp-tools' describe the OAM tools supported. The 'connectionless-oam-layers' is used to describe the relationship of one test point with other test points. The level in 'oam-layers' indicate whether related OAM test point is The level in oam-layers indicate whether related oam test point is in client layer(lower layer described in section 3.3), server layer (upper layer described in section 3.3) or the same layer as the

current test point under Test point Locations. The model is augmented to "/nd:networks/nd:network/nd:node" using 'test-point-locations' defined below.

3.1. TP Address

In connectionless OAM, the tp address is defined with the following type:

- MAC address [RFC6136]
- IPv4 or IPv6 address
- TP-attribute
- System-id to represent the device or node.[I-D.ietf-spring-sr-yang]

To define a forwarding treatment of a test packet, the 'tp-address' needs to be associated with additional parameters, e.g. DSCP for IP or TC for MPLS. In generic connectionless OAM YANG model, these parameters are not explicit configured. The model user can add corresponding parameters according to their requirements.

3.2. Tools

The different OAM tools may be used in one of two basic types of activation: proactive and on-demand. The proactive OAM refers to OAM actions which are carried out continuously to permit proactive reporting of fault. The proactive OAM method requires persistent configuration. The on-demand OAM refers to OAM actions which are initiated via manual intervention for a limited time to carry out diagnostics. The on-demand OAM method requires only transient configuration.[RFC7276] [G.8013]. In connectionless OAM, 'session-type' is defined to indicate which kind of activation will be used by the current session.

In connectionless OAM, the tools attribute is used to describe a toolset for fault detection and isolation. And it can serve as a constraint condition when the base model be extended to specific OAM technology. For example, to fulfill the ICMP PING configuration, the "../coam:continuity-check" should be set to "true", and then the lime base model should be augmented with ICMP PING specific details.

3.3. OAM-layers

As typical networks have a multi-layer architecture, the set of OAM protocols similarly take a multi-layer structure; each layer may has its own OAM protocol [RFC7276] and is corresponding to specific administrative domain or path and has associated test points. OAM-layers is referred to a list of server layer, client layer that are related to current test point. This allows users to easily navigate between related layer to efficiently troubleshoot a "loss of continuity defect". In this model, we have kept level default as 0, when all test points are located at the same layer. 'Level' defines the relative technology level in a sequence of administrative domains, and is provided to allow correlation of faults in related OAM domains. For example, there is a network in which data traffic between two customer edges is transported over three consecutive administrative domains, where the current test point is located in the second administrative domain. In this scenario second administrative domain is acting as client to first administrative domain and server to third administrative domain. For Test Point at second administrative domain, client level is "-1", i.e. third administrative domain and server level is "1", i.e. first administrative domain. In another example, if the first administrative domain and the second are in same level then it's upstream or downstream administrative domain scenario and thus second administrative domain level is set to "0".

```
list oam-layers {
  key "index";
  leaf index {
     type uint16 {
        range "0..65535";
     }
  }
  leaf level {
      type int32 {
           range "-1..1";
      }
      description
        "Level";
  }
  ordered-by system;
  description
     "List of related oam layers.";
}
```

3.4. Test Point Locations Information

This is a generic grouping for Test Point Locations Information. It Provide details of Test Point Location using Tools, 'OAM-Layers' grouping defined above.

3.5. Test Point Locations

This is a generic grouping for Test Point Locations. Choice statement is used to define locations types, for example 'ipv4-location-type', 'ipv6-location-type', etc. Container is defined under each location type containing list keyed to test point address, Test Point Location Information defined in section above, and routing instance VRF name if required.

3.6. Path Discovery Data

This is a generic grouping for path discovery data model that can be retrieved by any data retrieval methods including RPCs. Path discovery data output from methods, includes 'src-test-point', 'dst-test-point', 'sequence-number', 'hop-cnt', session statistics of various kinds, path verification and path trace related information. Path discovery includes data to be retrieved on a 'per-hop' basis via a list of 'path-trace-info-list' which includes information like 'timestamps', 'ingress-interface', 'egress-interface' and 'app-meta-data'. The path discovery data model is made generic enough to allow different methods of data retrieval. None of the fields are made mandatory for that reason. Noted that the retrieval methods are defined in [I-D.ietf-lime-yang-connectionless-oam-methods].

3.7. Continuity Check Data

This is a generic grouping for continuity check data model that can be retrieved by any data retrieval methods including RPCs. Continuity check data output from methods, includes 'src-test-point', 'dst-test-point', 'sequence-number', 'hop-cnt' and session statistics of various kinds. The continuity check data model is made generic enough to allow different methods of data retrieval. None of the fields are made mandatory for that reason. Noted that the retrieval methods are defined in [I-D.ietf-lime-yang-connectionless-oam-methods].

4. OAM YANG Module

<CODE BEGINS> file "ietf-connectionless-oam@2017-04-25.yang"

```
module ietf-connectionless-oam {
  yang-version 1.1;
```

```
  namespace "urn:ietf:params:xml:ns:yang:ietf-connectionless-oam";
  prefix coam;

  import ietf-yang-schema-mount {
    prefix yangmnt;
  }
  import ietf-network {
    prefix nd;
  }
  import ietf-yang-types {
    prefix yang;
  }
  import ietf-interfaces {
    prefix if;
  }
  import ietf-inet-types {
    prefix inet;
  }
  import ietf-network-instance {
    prefix ni;
  }

  organization
    "IETF LIME Working Group";
  contact
    "Deepak Kumar dekumar@cisco.com
     Qin Wu       bill.wu@huawei.com
     S Raghavan   srihari@cisco.com
     Zitao Wang   wangzitao@huawei.com
     R Rahman     rrahman@cisco.com";
  description
    "This YANG module defines the generic configuration,
     data model, statistics for connectionless 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 2017-04-25 {
    description
      " Base model for Connectionless
       Operations, Administration,
       and Maintenance(OAM) ";
    reference
      " RFC XXXX: Connectionless
       Operations, Administration, and
```

```
       Maintenance(OAM)YANG Data Model";
  }

  feature connection-less {
    description
      "This feature indicates that OAM solution is connection less.";
  }

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

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

  typedef router-id {
    type yang:dotted-quad;
    description
      "A 32-bit number in the dotted quad format assigned to each
       router. This number uniquely identifies the router within an
       Autonomous System.";
  }

  typedef routing-instance-ref {
    type leafref {
      path "/ni:network-instances/ni:network-instance/ni:name";
    }
    description
      "This type is used for leafs that reference a routing instance
       configuration.";
  }

  typedef ipv4-multicast-group-address {
    type string {
      pattern "(2((2[4-9])|(3[0-9]))\\.)(([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-
9]|25[0-5])\\.){2}([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])";
```

```
    }
    description
      "The ipv4-multicast-group-address type
       represents an IPv4 multicast address
       in dotted-quad notation.";
    reference "RFC4607";
  }

  typedef ipv6-multicast-group-address {
    type string {
      pattern "(((FF|ff)[0-9a-fA-F]{2}):)([0-9a-fA-F]{0,4}:){0,5}((([0-9a-fA-F]{
0,4}:)?(:|[0-9a-fA-F]{0,4}))|(((25[0-5]|2[0-4][0-9]|[01]?[0-9]?[0-9])\\.){3}(25[
0-5]|2[0-4][0-9]|[01]?[0-9]?[0-9])))";
      pattern "(([^:]+:){6}(([^:]+:[^:]+)|(.*\\..*))|((([^:]+:)*[^:]+)?::(([^:]
+:)*[^:]+)?)";
    }
    description
      "The ipv6-multicast-group-address
       type represents an IPv6 address in full,
       mixed, shortened, and shortened-mixed
       notation.";
    reference
      "RFC4291 2.7.
       ietf-inet-types:ipv6-address";
  }

  typedef ip-multicast-group-address {
    type union {
      type ipv4-multicast-group-address;
      type ipv6-multicast-group-address;
    }
    description
      "The ip-multicast-group-address type
       represents an IP multicast address and
       is IP version neutral. The format of the
       textual representations implies the IP version.";
  }

  identity address-attribute-types {
    description
      "This is base identity of address
       attribute types which are ip-prefix,
       bgp, tunnel, pwe3, vpls, etc.";
  }

  typedef address-attribute-type {
    type identityref {
      base address-attribute-types;
    }
    description
      "Target address attribute type.";
```

```
  }

  identity time-resolution {
    description
      "Time interval resolution";
  }

  identity hours {
    base time-resolution;
    description
      "Time resolution in Hours";
  }

  identity minutes {
    base time-resolution;
    description
      "Time resolution in Minutes";
  }

  identity seconds {
    base time-resolution;
    description
      "Time resolution in Seconds";
  }

  identity milliseconds {
    base time-resolution;
    description
      "Time resolution in Milliseconds";
  }

  identity microseconds {
    base time-resolution;
    description
      "Time resolution in Microseconds";
  }

  identity nanoseconds {
    base time-resolution;
    description
      "Time resolution in Nanoseconds";
  }

  grouping cc-session-statistics {
    description
      "Grouping for session statistics.";
    container cc-session-statistics {
      description
```

```
        "cc session counters";
      leaf session-count {
        type uint32;
        description
          "Number of Continuity Check sessions.";
      }
      leaf session-up-count {
        type uint32;
        description
          "Number of sessions which are up.";
      }
      leaf session-down-count {
        type uint32;
        description
          "Number of sessions which are down.";
      }
      leaf session-admin-down-count {
        type uint32;
        description
          "Number of sessions which are admin-down.";
      }
    }
  }

  grouping session-packet-statistics {
    description
      "Grouping for per session packet statistics";
    container session-packet-statistics {
      description
        "Per session packet statistics.";
      leaf rx-packet-count {
        type uint32;
        description
          "Total number of received OAM packet count.";
      }
      leaf tx-packet-count {
        type uint32;
        description
          "Total number of transmitted OAM packet count.";
      }
      leaf rx-bad-packet {
        type uint32;
        description
          "Total number of received bad OAM packet.";
      }
      leaf tx-packet-failed {
        type uint32;
        description
```

```
          "Total number of send OAM packet failed.";
      }
    }
  }

  grouping cc-per-session-statistics {
    description
      "Grouping for per session statistics";
    container cc-per-session-statistics {
      description
        "per session statistics.";
      leaf create-time {
        type yang:date-and-time;
        description
          "Time and date when session is created.";
      }
      leaf last-down-time {
        type yang:date-and-time;
        description
          "Time and date last time session is down.";
      }
      leaf last-up-time {
        type yang:date-and-time;
        description
          "Time and date last time session is up.";
      }
      leaf down-count {
        type uint32;
        description
          "Total Continuity Check sessions down count.";
      }
      leaf admin-down-count {
        type uint32;
        description
          "Total Continuity Check sessions admin down count.";
      }
      uses session-packet-statistics;
    }
  }

  grouping session-error-statistics {
    description
      "Grouping for per session error statistics";
    container session-error-statistics {
      description
        "Per session error statistics.";
      leaf packet-drops-count {
        type uint32;
```

```
        description
          "Total received packet drops count.";
      }
      leaf packet-reorder-count {
        type uint32;
        description
          "Total received packet reordered count.";
      }
      leaf packets-out-of-seq-count {
        type uint32;
        description
          "Total received out of sequence count.";
      }
      leaf packets-dup-count {
        type uint32;
        description
          "Total received packet duplicates count.";
      }
    }
  }

  grouping session-delay-statistics {
    description
      "Grouping for per session delay statistics";
    container session-delay-statistics {
      description
        "Session delay summarised information.";
      leaf time-resolution-value {
        type identityref {
          base time-resolution;
        }
        description
          "Time units among choice of s,ms,ns etc.";
      }
      leaf min-delay-value {
        type uint32;
        description
          "Minimum delay value observed.";
      }
      leaf max-delay-value {
        type uint32;
        description
          "Maximum delay value observed.";
      }
      leaf average-delay-value {
        type uint32;
        description
          "Average delay value observed.";
```

```
      }
    }
  }

  grouping session-jitter-statistics {
    description
      "Grouping for per session jitter statistics";
    container session-jitter-statistics {
      description
        "Session jitter summarised information.";
      leaf time-resolution-value {
        type identityref {
          base time-resolution;
        }
        description
          "Time units among choice of s,ms,ns etc.";
      }
      leaf min-jitter-value {
        type uint32;
        description
          "Minimum jitter value observed.";
      }
      leaf max-jitter-value {
        type uint32;
        description
          "Maximum jitter value observed.";
      }
      leaf average-jitter-value {
        type uint32;
        description
          "Average jitter value observed.";
      }
    }
  }

  grouping session-path-verification-statistics {
    description
      "Grouping for per session path verification statistics";
    container session-path-verification-statistics {
      description
        "OAM per session path verification statistics.";
      leaf verified-count {
        type uint32;
        description
          "Total number of OAM packets that
           went through a path as intended.";
      }
      leaf failed-count {
```

```
        type uint32;
        description
          "Total number of OAM packets that
           went through an unintended path.";
      }
    }
  }

  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.";
        }
      }
      default "on-demand";
      description
        "Session type enum";
    }
  }

  identity tp-address-technology-type {
    description
      "Test point address type";
  }

  identity mac-address-type {
    base tp-address-technology-type;
    description
      "MAC address type";
  }

  identity ipv4-address-type {
    base tp-address-technology-type;
    description
      "IPv4 address type";
  }

  identity ipv6-address-type {
    base tp-address-technology-type;
```

```
    description
      "IPv6 address type";
  }

  identity tp-attribute-type {
    base tp-address-technology-type;
    description
      "Test point attribute type";
  }

  identity system-id-address-type {
    base tp-address-technology-type;
    description
      "System id address type";
  }

  identity as-number-address-type {
    base tp-address-technology-type;
    description
      "AS number address type";
  }

  identity group-ip-address-type {
    base tp-address-technology-type;
    description
      "Group IP address type";
  }

  identity route-distinguisher-address-type {
    base tp-address-technology-type;
    description
      "Route Distinguisher address type";
  }

  identity ip-prefix-address-type {
    base tp-address-technology-type;
    description
      "IP prefix address type";
  }

  identity tunnel-address-type {
    base tp-address-technology-type;
    description
      "Tunnel address type";
  }

  grouping tp-address {
    leaf tp-location-type-value {
```

```
      type identityref {
        base tp-address-technology-type;
      }
      description
        "Test point address type.";
    }
    choice tp-address {
      case mac-address {
        when "’tp-location-type-value’ = ’mac-address-type’" {
          description
            "MAC address type";
        }
        leaf mac-address {
          type yang:mac-address;
          description
            "MAC Address";
        }
        description
          "MAC Address based MP Addressing.";
      }
      case ipv4-address {
        when "’tp-location-type-value’ = ’ipv4-address-type’" {
          description
            "IPv4 address type";
        }
        leaf ipv4-address {
          type inet:ipv4-address;
          description
            "IPv4 Address";
        }
        description
          "IP Address based MP Addressing.";
      }
      case ipv6-address {
        when "’tp-location-type-value’ = ’ipv6-address-type’" {
          description
            "IPv6 address type";
        }
        leaf ipv6-address {
          type inet:ipv6-address;
          description
            "IPv6 Address";
        }
        description
          "ipv6 Address based MP Addressing.";
      }
      case tp-attribute {
        when "’tp-location-type-value’ = ’tp-attribute-type’" {
```

```
          description
            "Test point attribute type";
        }
        leaf tp-attribute-type {
          type address-attribute-type;
          description
            "Test point type.";
        }
        choice tp-attribute-value {
          description
            "Test point value.";
          case ip-prefix {
            leaf ip-prefix {
              type inet:ip-prefix;
              description
                "IP prefix.";
            }
          }
          case bgp {
            leaf bgp {
              type inet:ip-prefix;
              description
                "BGP Labeled Prefix ";
            }
          }
          case tunnel {
            leaf tunnel-interface {
              type uint32;
              description
                "VPN Prefix ";
            }
          }
          case pw {
            leaf remote-pe-address {
              type inet:ip-address;
              description
                "Remote pe address.";
            }
            leaf pw-id {
              type uint32;
              description
                "Pseudowire id.";
            }
          }
          case vpls {
            leaf route-distinguisher {
              type uint32;
              description
```

```
                "Route Distinguisher(8 octets).";
            }
            leaf sender-ve-id {
              type uint32;
              description
                "Sender's VE ID.";
            }
            leaf receiver-ve-id {
              type uint32;
              description
                "Receiver's VE ID.";
            }
          }
          case mpls-mldp {
            choice root-address {
              description
                "Root address choice.";
              case ip-address {
                leaf source-address {
                  type inet:ip-address;
                  description
                    "IP address.";
                }
                leaf group-ip-address {
                  type ip-multicast-group-address;
                  description
                    "Group ip address.";
                }
              }
              case vpn {
                leaf as-number {
                  type inet:as-number;
                  description
                    "AS number.";
                }
              }
              case global-id {
                leaf lsp-id {
                  type string;
                  description
                    "LSP id.";
                }
              }
            }
          }
        }
        case system-info {
```

```
        when "'tp-location-type-value' = 'system-id-address-type'" {
          description
            "System id address type";
        }
        leaf system-id {
          type router-id;
          description
            "System ID assigned to this node.";
        }
      }
      description
        "TP Addressing.";
    }
    description
      "TP Address";
  }

  grouping tp-address-vrf {
    description
      "Test point address with VRF.";
    leaf vrf {
      type routing-instance-ref;
      description
        "The vrf is used to describe the
         corresponding network instance";
    }
    uses tp-address;
  }

  grouping connectionless-oam-layers {
    list oam-layers {
      key "index";
      leaf index {
        type uint16 {
          range "0..65535";
        }
        description
          "Index";
      }
      leaf level {
        type int32 {
          range "-1..1";
        }
        default "0";
        description
          "Level 0 indicates default level,
           -1 means server and +1 means client layer.
           In relationship 0 means same layer.";
```

```
      }
      choice tp-location {
        case mac-address {
          leaf mac-address-location {
            type yang:mac-address;
            description
              "MAC Address";
          }
          description
            "MAC Address based MP Addressing.";
        }
        case ipv4-address {
          leaf ipv4-location {
            type inet:ipv4-address;
            description
              "Ipv4 Address";
          }
          description
            "IP Address based MP Addressing.";
        }
        case ipv6-location {
          leaf ipv6-address {
            type inet:ipv6-address;
            description
              "IPv6 Address";
          }
          description
            "IPv6 Address based MP Addressing.";
        }
        case group-ip-address-location {
          leaf group-ip-address-location {
            type ip-multicast-group-address;
            description
              "Group IP address location";
          }
          description
            "Group IP address";
        }
        case as-number-location {
          leaf as-number-location {
            type inet:as-number;
            description
              "AS number location";
          }
          description
            "AS number for point to multipoint OAM";
        }
        case system-id-location {
```

```
          leaf system-id-location {
            type router-id;
            description
              "System id location";
          }
          description
            "System ID";
        }
        description
          "TP location.";
      }
      ordered-by system;
      description
        "List of related oam layers.
         0 means they are in same level, especially
         interworking scenarios of stitching multiple
         technology at same layer. -1 means server layer,
         for eg:- in case of Overlay and Underlay,
         Underlay is server layer for Overlay Test Point.
         +1 means client layer, for example in case of
         Service OAM and Transport OAM, Service OAM is client
         layer to Transport OAM.";
    }
    description
      "Connectionless related OAM layer";
  }

  grouping tp-technology {
    choice technology {
      default "technology-null";
      case technology-null {
        description
          "This is a placeholder when no technology is needed.";
        leaf tech-null {
          type empty;
          description
            "There is no technology define";
        }
      }
      description
        "Technology choice.";
    }
    description
      "OAM Technology";
  }

  grouping tp-tools {
    description
```

```
      "Test Point OAM Toolset.";
    container tp-tools {
      leaf continuity-check {
        type boolean;
        description
          "A flag indicating whether or not the
           continuity check function is supported.";
        reference
          "RFC 792: INTERNET CONTROL MESSAGE PROTOCOL.
           RFC 4443: Internet Control Message Protocol (ICMPv6)
           for the Internet Protocol Version 6 (IPv6) Specification.
           RFC 5880: Bidirectional Forwarding Detection.
           RFC 5881: BFD for IPv4 and IPv6.
           RFC 5883: BFD for Multihop Paths.
           RFC 5884: BFD for MPLS Label Switched Paths.
           RFC 5885: BFD for PW VCCV.
           RFC 6450: Multicast Ping Protocol.";
      }
      leaf path-discovery {
        type boolean;
        description
          "A flag indicating whether or not the
           path discovery function is supported.";
        reference
          "RFC 792: INTERNET CONTROL MESSAGE PROTOCOL.
           RFC 4443: Internet Control Message Protocol (ICMPv6)
           for the Internet Protocol Version 6 (IPv6) Specification.
           RFC 4884: Extended ICMP to Support Multi-part Message.
           RFC 5837:Extending ICMP for Interface
           and Next-Hop Identification.
           RFC 4379: LSP-PING.";
      }
      description
        "Container for test point OAM tools set.";
    }
  }

  grouping test-point-location-info {
    uses tp-technology;
    uses tp-tools;
    anydata root {
      yangmnt:mount-point "root";
      description
        "Root for models supported per
         test point";
    }
    uses connectionless-oam-layers;
    description
```

```
      "Test point Location";
  }

  grouping test-point-locations {
    description
      "Group of test point locations.";
    leaf tp-location-type-value {
      type identityref {
        base tp-address-technology-type;
      }
      description
        "Test point location type.";
    }
    choice location-type {
      case ipv4-location-type {
        when "'tp-location-type-value' = 'ipv4-address-type'" {
          description
            "When test point location type is equal to ipv4 address.";
        }
        container test-point-ipv4-location-list {
          list test-point-locations {
            key "ipv4-location";
            leaf ipv4-location {
              type inet:ipv4-address;
              description
                "IPv4 Address.";
            }
            leaf vrf {
              type routing-instance-ref;
              description
                "The vrf is used to describe the
                 corresponding network instance";
            }
            uses test-point-location-info;
            ordered-by system;
            description
              "List of test point locations.";
          }
          description
            "Serves as top-level container
             for test point location list.";
        }
      }
      case ipv6-location-type {
        when "'tp-location-type-value' = 'ipv6-address-type'" {
          description
            "when test point location is equal to ipv6 address";
        }
```

```
        container test-point-ipv6-location-list {
          list test-point-locations {
            key "ipv6-location";
            leaf ipv6-location {
              type inet:ipv6-address;
              description
                "IPv6 Address.";
            }
            leaf vrf {
              type routing-instance-ref;
              description
                "The vrf is used to describe the
                 corresponding network instance";
            }
            uses test-point-location-info;
            ordered-by system;
            description
              "List of test point locations.";
          }
          description
            "Serves as top-level container
             for test point location list.";
        }
      }
      case mac-location-type {
        when "'tp-location-type-value' = 'mac-address-type'" {
          description
            "when test point location type is equal to mac address.";
        }
        container test-point-mac-address-location-list {
          list test-point-locations {
            key "mac-address-location";
            leaf mac-address-location {
              type yang:mac-address;
              description
                "MAC Address";
            }
            uses test-point-location-info;
            ordered-by system;
            description
              "List of test point locations.";
          }
          description
            "Serves as top-level container
             for test point location list.";
        }
      }
      case group-ip-address-location-type {
```

```
        when "'tp-location-type-value' = 'group-ip-address-type'" {
          description
            "When test point location type is equal to
             group ip address.";
        }
        container test-point-group-ip-address-location-list {
          list test-point-locations {
            key "group-ip-address-location";
            leaf group-ip-address-location {
              type ip-multicast-group-address;
              description
                "Group IP address.";
            }
            leaf vrf {
              type routing-instance-ref;
              description
                "The vrf is used to describe the
                 corresponding network instance";
            }
            uses test-point-location-info;
            ordered-by system;
            description
              "List of test point locations.";
          }
          description
            "Serves as top-level container for
             test point location list.";
        }
      }
      case group-as-number-location-type {
        when "'tp-location-type-value' = 'as-number-address-type'" {
          description
            "When test point location type is equal to
             as-number.";
        }
        container test-point-as-number-location-list {
          list test-point-locations {
            key "as-number-location";
            leaf as-number-location {
              type inet:as-number;
              description
                "AS number for point to multi point OAM.";
            }
            leaf vrf {
              type routing-instance-ref;
              description
                "The vrf is used to describe the
                 corresponding network instance";
```

```
            }
            uses test-point-location-info;
            ordered-by system;
            description
              "List of test point locations.";
          }
          description
            "Serves as top-level container
             for test point location list.";
        }
      }
      case group-system-id-location-type {
        when "'tp-location-type-value' = 'system-id-address-type'" {
          description
            "When test point location is equal to
             system info.";
        }
        container test-point-system-info-location-list {
          list test-point-locations {
            key "system-id-location";
            leaf system-id-location {
              type inet:uri;
              description
                "System Id.";
            }
            leaf vrf {
              type routing-instance-ref;
              description
                "The vrf is used to describe the
                 corresponding network instance";
            }
            uses test-point-location-info;
            ordered-by system;
            description
              "List of test point locations.";
          }
          description
            "Serves as top-level container for
             test point location list.";
        }
      }
      description
        "Choice of address types.";
    }
  }

  augment "/nd:networks/nd:network/nd:node" {
    description
```

```
      "Augment test points of connectionless oam.";
    uses test-point-locations;
  }

  grouping uint64-timestamp {
    description
      "Grouping for timestamp.";
    leaf timestamp-sec {
      type uint32;
      description
        "Absolute timestamp in seconds as per IEEE1588v2
         or seconds part in 64-bit NTP timestamp.";
    }
    leaf timestamp-nanosec {
      type uint32;
      description
        "Fractional part in nanoseconds as per IEEE1588v2
         or Fractional part in 64-bit NTP timestamp.";
    }
  }

  grouping timestamp {
    description
      "Grouping for timestamp.";
    leaf timestamp-type {
      type uint32;
      description
        "Truncated PTP = 0, NTP = 1";
    }
    uses uint64-timestamp;
  }

  grouping path-discovery-data {
    description
      "Path discovery related data output from nodes.";
    container src-test-point {
      description
        "Source test point.";
      uses tp-address-vrf;
    }
    container dest-test-point {
      description
        "Destination test point.";
      uses tp-address-vrf;
    }
    leaf sequence-number {
      type uint64;
      description
```

```
        "Sequence number in data packets.";
    }
    leaf hop-cnt {
      type uint8;
      description
        "Hop count.";
    }
    uses session-packet-statistics;
    uses session-error-statistics;
    uses session-delay-statistics;
    uses session-jitter-statistics;
    container path-verification {
      description
        "Optional path verification related information.";
      leaf flow-info {
        type string;
        description
          "Informations that refers to the flow.";
      }
      uses session-path-verification-statistics;
    }
    container path-trace-info {
      description
        "Optional path trace per-hop test point information.
         The list has typically a single element for per-hop
         cases like path-discovery RPC but allows a list of
         hop related information for other types of
         data retrieval methods.";
      list path-trace-info-list {
        key "index";
        description
          "Path trace information list.";
        leaf index {
          type uint32;
          description
            "Trace information index.";
        }
        uses tp-address-vrf;
        uses timestamp;
        leaf ingress-intf-name {
          type if:interface-ref;
          description
            "Ingress interface name";
        }
        leaf egress-intf-name {
          type if:interface-ref;
          description
            "Egress interface name";
```

```
        }
        leaf queue-depth {
          type uint32;
          description
            "Length of the egress interface
             queue of the interface.";
        }
        leaf transit-delay {
          type uint32;
          description
            "Time in nano seconds
             packet spent transiting a node.";
        }
        leaf app-meta-data {
          type uint64;
          description
            "Application specific
             data added by node.";
        }
      }
    }
  }

  grouping continuity-check-data {
    description
      "Continuity check data output from nodes.";
    container src-test-point {
      description
        "Source test point.";
      uses tp-address-vrf;
      leaf egress-intf-name {
        type if:interface-ref;
        description
          "Egress interface name";
      }
    }
    container dest-test-point {
      description
        "Destination test point.";
      uses tp-address-vrf;
      leaf ingress-intf-name {
        type if:interface-ref;
        description
          "Ingress interface name";
      }
    }
    leaf sequence-number {
      type uint64;
```

```
      description
        "Sequence number.";
    }
    leaf hop-cnt {
      type uint8;
      description
        "Hop count.";
    }
    uses session-packet-statistics;
    uses session-error-statistics;
    uses session-delay-statistics;
    uses session-jitter-statistics;
  }

  container cc-session-statistics-data {
    if-feature "continuity-check";
    config false;
    description
      "CC operational information.";
    container cc-ipv4-sessions-statistics {
      description
        "CC ipv4 sessions";
      uses cc-session-statistics;
    }
    container cc-ipv6-sessions-statistics {
      description
        "CC ipv6 sessions";
      uses cc-session-statistics;
    }
  }
}
```

<CODE ENDS>

5. Connectionless model applicability

"ietf-connectionless-oam" model defined in this document provides technology-independent abstraction of key OAM constructs for connectionless protocols. This model can be further extended to include technology specific details, e.g., adding new data nodes with technology specific functions and parameters into proper anchor points of the base model, so as to develop a technology-specific connectionless OAM model.

This section demonstrates the usability of the connectionless YANG OAM data model to various connectionless OAM technologies, e.g., BFD, LSP ping. Note that, in this section, we only present several snippets of technology-specific model extensions for illustrative

purposes. The complete model extensions should be worked on in respective protocol working groups.

5.1. BFD Extension

5.1.1. Augment Method

The following sections shows how the "ietf-connectionless-oam" model can be extended to cover BFD technology. For this purpose, a set of extension are introduced such as technology-type extension and test-point attributes extension.

Note that in BFD WG, there is a BFD yang data model [I-D.ietf-bfd-yang] to be produced. Users can choose to use "ietf-connectioless-oam" as basis and augment the "ietf-connectionless-oam" model with bfd specific details. The bfd specific details can be the grouping defined in the BFD model.

5.1.1.1. Technology type extension

No BFD technology type has been defined in the "ietf-connectionless-oam" model. Therefore a technology type extension is required in the model Extension.

The snippet below depicts an example of augmenting "bfd" type into the ietf-connectionless-oam":

```
augment "/nd:networks/nd:network/nd:node/"
+"coam:location-type/coam:ipv4-location-type"
+"/coam:test-point-ipv4-location-list/"
       +"coam:test-point-locations/coam:technology"
+"/coam:technology-string"
{
    leaf bfd{
   type string;
  }
}
```

5.1.1.2. Test point attributes extension

To support bfd technology, the "ietf-connectionless-oam" model can be extended and add bfd specific parameters under "test-point-location" list and/or add new location type such as "bfd over MPLS-TE" under "location-type".

5.1.1.2.1. Define and insert new nodes into corresponding test-point-location

In the "ietf-connectionless-oam" model, multiple "test-point-location" lists are defined under the "location-type" choice node. Therefore, to derive a model for some bfd technologies (such as ip single-hop, ip multi-hops, etc), data nodes for bfd specific details need to be added into corresponding "test-point-locations" list. In this section, we reuse some groupings which are defined in [I-D.ietf-bfd-yang] as following:

The snippet below shows how the "ietf-connectionless-oam" model can be extended to support "BFD IP single-hop":

```
augment "/nd:networks/nd:network/nd:node/"
+"coam:location-type/coam:ipv4-location-type"
+"/coam:test-point-ipv4-location-list/"
        +"coam:test-point-locations"
{
        container session-cfg {
          description "BFD IP single-hop session configuration";
          list sessions {
            key "interface dest-addr";
            description "List of IP single-hop sessions";
            leaf interface {
              type if:interface-ref;
              description
                "Interface on which the BFD session is running.";
            }
            leaf dest-addr {
              type inet:ip-address;
              description "IP address of the peer";
            }
            uses bfd:bfd-grouping-common-cfg-parms;
            uses bfd:bfd-grouping-echo-cfg-parms;
          }
        }
}
```

Similar augmentations can be defined to support other BFD technologies such as BFD IP multi-hop, BFD over MPLS, etc.

5.1.1.2.2. Add new location-type cases

In the "ietf-connectionless-oam" model, If there is no appropriate "location type" case that can be extended, a new "location-type" case can be defined and inserted into the "location-type" choice node.

Therefore, the model user can flexibly add "location-type" to support other type of test point which are not defined in the "ietf-connectionless-oam" model. In this section, we add a new "location-type" case and reuse some groupings which are defined in [I-D.ietf-bfd-yang] as follows:

The snippet below shows how the "ietf-connectionless-oam" model can be extended to support "BFD over MPLS-TE":

```
augment "/nd:networks/nd:network/nd:node/coam:location-type"{
 case te-location{
  list test-point-location-list{
   key "tunnel-name";
   leaf tunnel-name{
    type leafref{
 path "/te:te/te:tunnels/te:tunnel/te:name";
}
description
"point to a te instance.";
   }
    uses bfd:bfd-grouping-common-cfg-parms;
        uses bfd-mpls:bfd-encap-cfg;
  }
 }
}
```

Similar augmentations can be defined to support other BFD technologies such as BFD over LAG, etc.

5.1.2. Schema Mount

And another alternative method is using schema mount mechanism [I-D.ietf-netmod-schema-mount] in the "ietf-connectionless-oam". Within the "test-point-location" list, a "root" attribute is defined to provide a mounted point for models mounted per "test-point-location". Therefore, the "ietf-connectionless-oam" model can provide a place in the node hierarchy where other OAM YANG data models can be attached, without any special extension in the "ietf-connectionless-oam" YANG data models [I-D.ietf-netmod-schema-mount]. Note that the limitation of the Schema Mount method is it is not allowed to specify certain modules that are required to be mounted under a mount point.

The snippet below depicts the definition of "root" attribute.

```
        anydata root {
         yangmnt:mount-point root;
         description
        "Root for models supported per
          test point";
        }
```

The following section shows how the "ietf-connectionless-oam" model can use schema mount to support BFD technology.

5.1.2.1. BFD Modules be populated in schema-mount

To support BFD technology, "ietf-bfd-ip-sh" and "ietf-bfd-ip-mh" YANG modules might be populated in the "schema-mounts" container:

```
   <schema-mounts
       xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-schema-mount">
     <mount-point>
       <module> ietf-connectionless-oam </module>
       <name>root</name>
       <use-schema>
         <name>root</name>
       </use-schema>
     </mount-point>
     <schema>
       <name>root</name>
       <module>
         <name>ietf-bfd-ip-sh </name>
         <revision>2016-07-04</revision>
         <namespace>
           urn:ietf:params:xml:ns:yang: ietf-bfd-ip-sh
         </namespace>
         <conformance-type>implement</conformance-type>
       </module>
       <module>
         <name>ietf-bfd-ip-mh </name>
         <revision> 2016-07-04</revision>
         <namespace>
           urn:ietf:params:xml:ns:yang: ietf-bfd-ip-mh
         </namespace>
         <conformance-type>implement</conformance-type>
       </module>
     </schema>
   </schema-mounts>
```

and the " ietf-connectionless-oam " module might have:

```
<ietf-connectionless-oam
uri="urn:ietf:params:xml:ns:yang:ietf-connectionless-oam">
   ......
 <test-point-locations>
  <ipv4-location> 1.1.1.1</ipv4-location>
   ......
  <root>
   <ietf-bfd-ip-sh uri="urn:ietf:params:xml:ns:yang:ietf-bfd-ip-sh">
    <ip-sh>
     foo
     ......
    </ip-sh>
   </ietf-bfd-ip-sh>
   <ietf-bfd-ip-mh uri="urn:ietf:params:xml:ns:yang:ietf-bfd-ip-mh">
    <ip-mh>
     foo
     ......
    </ip-mh>
   </ietf-bfd-ip-mh>
  </root>
 </test-point-locations>
</ietf-connectionless-oam>
```

5.2. LSP ping extension

5.2.1. Augment Method

The following sections shows how the "ietf-connectionless-oam" model can be extended to support LSP ping technology. For this purpose, a set of extension are introduced such as technology-type extension and test-point attributes extension.

Note that in MPLS WG, there is a LSP Ping yang data model [I-D.zheng-mpls-lsp-ping-yang-cfg] to be produced. Users can choose to use "ietf-connectioless-oam" as basis and augment the "ietf-connectionless-oam" model with LSP Ping specific details in the model extension. The LSP Ping specific details can be the grouping defined in the LSP ping model.

5.2.1.1. Technology type extension

No lsp-ping technology type has been defined in the "ietf-connectionless-oam" model. Therefore a technology type extension is required in the model extension.

The snippet below depicts an example of augmenting the "ietf-connectionless-oam" with "lsp-ping" type:

```
augment "/nd:networks/nd:network/nd:node/"
+"coam:location-type/coam:ipv4-location-type"
+"/coam:test-point-ipv4-location-list/"
        +"coam:test-point-locations/coam:technology"
+"/coam:technology-string"
{
   leaf lsp-ping{
   type string;
  }
}
```

5.2.1.2. Test point attributes extension

To support lsp-ping, the "ietf-connectionless-oam" model can be extended and add lsp-ping specific parameters can be defined and under "test-point-location" list.

User can reuse the attributes or groupings which are defined in [I-D.zheng-mpls-lsp-ping-yang-cfg] as follows:

The snippet below depicts an example of augmenting the "test-point-locations" list with lsp ping attributes:

```
augment "/nd:networks/nd:network/nd:node/"
+"coam:location-type/coam:ipv4-location-type"
+"/coam:test-point-ipv4-location-list/"
        +"coam:test-point-locations"
{
list lsp-ping {
         key "lsp-ping-name";
         leaf lsp-ping-name {
          type string {
            length "1..31";
         }
        mandatory "true";
        description "LSP Ping test name.";
        ......
      }
```

5.2.2. Schema Mount

And another alternative method is using schema mount mechanism [I-D.ietf-netmod-schema-mount] in the "ietf-connectionless-oam". Within the "test-point-location" list, a "root" attribute is defined to provide a mounted point for models mounted per "test-point-location". Therefore, the "ietf-connectionless-oam" model can provide a place in the node hierarchy where other OAM YANG data models can be attached, without any special extension in the "ietf-

connectionless-oam" YANG data models [I-D.ietf-netmod-schema-mount]. Note that the limitation of the Schema Mount method is it is not allowed to specify certain modules that are required to be mounted under a mount point.

The snippet below depicts the definition of "root" attribute.

```
anydata root {
 yangmnt:mount-point root;
 description
"Root for models supported per
  test point";
}
```

The following section shows how the "ietf-connectionless-oam" model can use schema mount to support LSP-PING technology.

5.2.2.1. LSP-PING Modules be populated in schema-mount

To support LSP-PING technology, "ietf-lspping" YANG module [I-D.zheng-mpls-lsp-ping-yang-cfg] might be populated in the "schema-mounts" container:

```
<schema-mounts
    xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-schema-mount">
  <mount-point>
    <module> ietf-connectionless-oam </module>
    <name>root</name>
    <use-schema>
      <name>root</name>
    </use-schema>
  </mount-point>
  <schema>
    <name>root</name>
    <module>
      <name>ietf-lspping </name>
      <revision>2016-03-18</revision>
      <namespace>
        urn:ietf:params:xml:ns:yang: ietf-lspping
      </namespace>
      <conformance-type>implement</conformance-type>
    </module>
  </schema>
</schema-mounts>
```

and the " ietf-connectionless-oam " module might have:

```
<ietf-connectionless-oam
uri="urn:ietf:params:xml:ns:yang:ietf-connectionless-oam">
   ......
 <test-point-locations>
  <ipv4-location> 1.1.1.1</ipv4-location>
   ......
  <root>
   <ietf-lspping uri="urn:ietf:params:xml:ns:yang:ietf-lspping">
    <lsp-pings>
     foo
     ......
    </lsp-pings>
   </ietf-lspping>
  </root>
 </test-point-locations>
</ietf-connectionless-oam>
```

6. Security Considerations

The YANG module defined in this memo is designed to be accessed via the NETCONF protocol [RFC6241]. The lowest NETCONF layer is the secure transport layer and the mandatory-to-implement secure transport is SSH [RFC6242]. The NETCONF access control model [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:

/nd:networks/nd:network/nd:node/coam:location-type/coam:ipv4-location-type/coam:test-point-ipv4-location-list/coam:test-point-locations/

/nd:networks/nd:network/nd:node/coam:location-type/coam:ipv6-location-type/coam:test-point-ipv6-location-list/coam:test-point-locations/

/nd:networks/nd:network/nd:node/coam:location-type/coam:mac-location-type/coam:test-point-mac-address-location-list/coam:test-point-locations/

/nd:networks/nd:network/nd:node/coam:location-type/coam:tunnel-location-type/coam:test-point-tunnel-address-location-list/coam:test-point-locations/

/nd:networks/nd:network/nd:node/coam:location-type/coam:ip-prefix-location-type/coam:test-point-ip-prefix-location-list/coam:test-point-locations/

/nd:networks/nd:network/nd:node/coam:location-type/coam:route-distinguisher-location-type/coam:test-point-route-dist-location-list/coam:test-point-locations/

/nd:networks/nd:network/nd:node/coam:location-type/coam:group-ip-address-location-type/coam:test-point-group-ip-address-location-list/coam:test-point-locations/

/nd:networks/nd:network/nd:node/coam:location-type/coam:group-as-number-location-type/coam:test-point-as-number-location-list/coam:test-point-locations/

/nd:networks/nd:network/nd:node/coam:location-type/coam:group-lsp-id-location-type/coam:test-point-lsp-id-location-list/coam:test-point-locations/

/nd:networks/nd:network/nd:node/coam:location-type/coam:group-system-id-location-type/coam:test-point-system-info-location-list/coam:test-point-locations/

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.

7. IANA Considerations

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

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

namespace: urn:ietf:params:xml:ns:yang:ietf-connectionless-oam

prefix: coam

reference: RFC XXXX

8. Acknowlegements

The authors of this document would like to thank Greg Mirsky and others for their sustainable review and comments, proposals to improve and stabilize document.

9. References

9.1. Normative References

[I-D.ietf-bfd-yang]
Rahman, R., Zheng, L., Networks, J., Jethanandani, M., and G. Mirsky, "Yang Data Model for Bidirectional Forwarding Detection (BFD)", draft-ietf-bfd-yang-05 (work in progress), March 2017.

[I-D.ietf-i2rs-yang-network-topo]
Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A Data Model for Network Topologies", draft-ietf-i2rs-yang-network-topo-12 (work in progress), March 2017.

[I-D.ietf-lime-yang-connectionless-oam-methods]
Kumar, D., Wang, Z., Wu, Q., Rahman, R., and S. Raghavan, "Retrieval Methods YANG Data Model for Connectionless Operations, Administration, and Maintenance(OAM) protocols", draft-ietf-lime-yang-connectionless-oam-methods-01 (work in progress), February 2017.

[I-D.ietf-lime-yang-oam-model]
Kumar, D., Wu, Q., and Z. Wang, "Generic YANG Data Model for Connection Oriented Operations, Administration, and Maintenance(OAM) protocols", draft-ietf-lime-yang-oam-model-10 (work in progress), April 2017.

[I-D.ietf-netmod-schema-mount]
Bjorklund, M. and L. Lhotka, "YANG Schema Mount", draft-ietf-netmod-schema-mount-04 (work in progress), March 2017.

[I-D.ietf-rtgwg-ni-model]
Berger, L., Hopps, C., Lindem, A., and D. Bogdanovic, "YANG Network Instances", draft-ietf-rtgwg-ni-model-02 (work in progress), March 2017.

[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>.

[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>.

[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>.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <http://www.rfc-editor.org/info/rfc6991>.

[RFC7223] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 7223, DOI 10.17487/RFC7223, May 2014, <http://www.rfc-editor.org/info/rfc7223>.

[RFC7224] Bjorklund, M., "IANA Interface Type YANG Module", RFC 7224, DOI 10.17487/RFC7224, May 2014, <http://www.rfc-editor.org/info/rfc7224>.

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

9.2. Informative References

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

[I-D.ietf-spring-sr-yang] Litkowski, S., Qu, Y., Sarkar, P., and J. Tantsura, "YANG Data Model for Segment Routing", draft-ietf-spring-sr-yang-06 (work in progress), March 2017.

[I-D.zheng-mpls-lsp-ping-yang-cfg] Zheng, L., Aldrin, S., Zheng, G., Mirsky, G., and R. Rahman, "Yang Data Model for LSP-PING", draft-zheng-mpls-lsp-ping-yang-cfg-04 (work in progress), October 2016.

[RFC6136] Sajassi, A., Ed. and D. Mohan, Ed., "Layer 2 Virtual Private Network (L2VPN) Operations, Administration, and Maintenance (OAM) Requirements and Framework", RFC 6136, DOI 10.17487/RFC6136, March 2011, <http://www.rfc-editor.org/info/rfc6136>.

[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>.

Authors' Addresses

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

Email: dekumar@cisco.com

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

Email: wangzitao@huawei.com

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

Email: bill.wu@huawei.com

Reshad Rahman
CISCO Systems
2000 Innovation Drive
KANATA, ONTARIO K2K 3E8
CANADA

Email: rrahman@cisco.com

Srihari Raghavan
CISCO Systems
TRIL INFOPARK SEZ, Ramanujan IT City
NEVILLE BLOCK, 2nd floor, Old Mahabalipuram Road
CHENNAI, TAMIL NADU 600113
INDIA

Email: srihari@cisco.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 27, 2017

D. Kumar
Cisco
M. Wang
Q. Wu
Huawei
R. Rahman
S. Raghavan
Cisco

February 23, 2017

Retrieval Methods YANG Data Model for Connectionless Operations, Administration, and Maintenance(OAM) protocols draft-ietf-lime-yang-connectionless-oam-methods-01

Abstract

This document presents a retrieval method YANG Data model for connectionless OAM protocols. It provides a technology-independent RPC commands for connectionless OAM protocols. The retrieval methods model presented here can be extended to include technology specific details. This is leading to uniformity between OAM protocols and support both nested OAM workflows (i.e., performing OAM functions at different levels through a unified interface) and interactive OAM workflows (i.e., performing OAM functions at same levels through a unified interface).

Status of This Memo

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

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

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

This Internet-Draft will expire on August 27, 2017.

Table of Contents

1. Introduction

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

1. Monitor reachability of destinations (Reachability 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] and BFD [RFC5880] are well-known fault verification and isolation tools, respectively, for IP

networks. Over the years, different technologies have developed similar tools for similar purposes.

In this document, we present a retrieval method YANG Data model for connectionless OAM protocols. This module provides technology-independent RPC commands for connectionless OAM protocols. It is separated from the generic YANG model for connectionless OAM [I-D.ietf-lime-yang-connectionless-oam] and can avoid mixing the models for the retrieved-data from the retrieval procedures. It is expected that retrieval procedures would evolve faster than the data model [I-D.ietf-lime-yang-connectionless-oam] and will allow new procedures to be defined for retrieval of the same data defined by the base data model.

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 terms are defined in [RFC6241] and are not redefined here:

- client
- configuration data
- server
- state data

The following terms are defined in [RFC6020] and are not redefined here:

- augment
- data model
- data node

The terminology for describing YANG data models is found in [RFC6020].

2.1. Terminology

TP - Test Point

MAC - Media Access Control

RPC - A Remote Procedure Call, as used within the NETCONF protocol

2.2. Tree Diagrams

A simplified graphical representation of the data model is used in this document. The meaning of the symbols in these diagrams is as follows:

Each node is printed as:

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

<status> is one of:
+ for current

<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

3. Overview of the Connectionless OAM retrieval methods Model

In this document, we present a retrieval method YANG Data model for connectionless OAM protocols. This module provides technology-independent retrieval procedures (RPC commands) for connectionless OAM protocols. It provides a flexible way to retrieve the retrieved-data which defined by the "ietf-connectionless-oam.yang" [I-D.ietf-lime-yang-connectionless-oam].

3.1. 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 obtaining a response.

Under 'connectionless-oam-methods' module, we summarize the common OAM functions and define the generic RPC commands: 'continuity-check' and 'path-discovery'. In practice, these commands are supported by corresponding technology-specific OAM tools [RFC7276]. For example, for the IP OAM model, the continuity-check RPC corresponds to the IP Ping [RFC792] [RFC4443], while the path-discovery RPC command corresponds to IP Traceroute [RFC792] [RFC4443].

Note that the RPC command presented in this document is the base building block, which is used to derive a model for a technology-specific OAM (i.e., ICMP ping [RFC792] [RFC4443], LSP ping [RFC4379]), the base building block should be extended with corresponding technology specific parameters. To facilitate this and for future enhancements to data retrieval methods, the RPCs are captured under a separate module.

The generic 'path-discovery-data' and 'continuity-check-data' are used as data outputs from the different RPCs described in the document. Similar methods including other RPCs can retrieve the data using the same data model.

The persistent method RPCs are commands to trigger persistent continuity check or path discovery OAM while specifying the options for data-export from the device. Internet Protocol Flow Information Export (IPFIX) [RFC7011] or yang-push [I-D.ietf-netconf-yang-push]. are currently outlined here as data export options and more can be added in future. It should be noted that the persistent methods are used to trigger create, modify and delete the persistent state associated with the data export options. The data export specific configurations are beyond the scope of this document.

```
    rpc continuity-check {
      if-feature coam:continuity-check;
      description
        "Generates continuity-check as per RFC7276.";
      input {
        container destination-tp {
          uses coam:tp-address;
                  description
                  "destination test point.";
        }
        uses coam:session-type;
```

```
            leaf source-interface {
              type if:interface-ref;
                      description
                      "source interface.";
            }
            leaf outbound-interface {
              type if:interface-ref;

                      description
                      "outbound interface.";
            }
            leaf count {
              type uint32;
              default "5";
              description
                "Specifies the number of packets that will be sent.";
            }
            leaf vrf {
              type coam:routing-instance-ref;
              description
                "vrf instance.";
            }
            leaf ttl {
              type uint8;
              default "255";
              description
                "Time to live (TTL).";
            }
            leaf packet-size {
              type uint32 {
                range "64..10000";
              }
              default "64";
              description
                "Size of ping echo request packets, in octets";
            }
          }
          output {
            list error-code-list {
              key "response-index";
              leaf response-index {
                type uint32;
                        description
                        "response index.";
              }
              leaf status-code {
                type int32;
                description
```

```
                "error code is ";
            }
            leaf status-sub-code {
              type uint8;
                      description
                      "sub code.";
            }
                    description
                    "error code list.";
          }

          uses coam:continuity-check-data;
        }
      }

      rpc path-discovery {
        description
          "Generates path discovery as per RFC7276.";
        input {
          container destination-tp {
            uses coam:tp-address;
                    description
                    "destination test point.";
          }
          uses coam:session-type;
          leaf source-interface {
            type if:interface-ref;
            description
              "source interface.";
          }
          leaf outbound-interface {
            type if:interface-ref;
            description
              "outbound interface.";
          }
          leaf vrf {
            type coam:routing-instance-ref;
            description
              "vrf";
          }
          leaf max-ttl {
            type uint8;
            default "255";
            description
              "max ttl.";
          }
        }
        output {
```

```
           list response-list {
             key "response-index";
             description
               "path discovery response list.";
             leaf response-index {
               type uint32;
               description
                 "response index.";
             }
             leaf status-code {
               type int32;
               description
                 "error code is ";
             }
             leaf status-sub-code {
               type uint8;

                       description
                       "sub code is ";
             }
           }

           uses coam:path-discovery-data;
         }
       }
```

Snippet of data hierarchy related to RPC calls

3.2. OAM Retrieval Methods Hierarchy

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

```
module: ietf-connectionless-oam-methods

  rpcs:
    +---x continuity-check {coam:continuity-check}?
    |  +---w input
    |  |  +---w destination-tp
    |  |  |  +---w tp-location-type-value?   identityref
    |  |  |  +---w (tp-address)?
    |  |  |     +--:(mac-address)
    |  |  |     |  +---w mac-address?              yang:mac-address
    |  |  |     +--:(ipv4-address)
    |  |  |     |  +---w ipv4-address?             inet:ipv4-address
    |  |  |     +--:(ipv6-address)
    |  |  |     |  +---w ipv6-address?             inet:ipv6-address
    |  |  |     +--:(tp-attribute)
```

```
|  |  |     |  +---w tp-attribute-type?        fec-type
|  |  |     |  +---w (tp-attribute-value)?
|  |  |     |     +--:(ip-prefix)
|  |  |     |     |  +---w ip-prefix?          inet:ip-prefix
|  |  |     |     +--:(bgp)
|  |  |     |     |  +---w bgp                 inet:ip-prefix
|  |  |     |     +--:(tunnel)
|  |  |     |     |  +---w tunnel-interface?         uint32
|  |  |     |     +--:(pw)
|  |  |     |     |  +---w remote-pe-address?  inet:ip-address
|  |  |     |     |  +---w pw-id?                    uint32
|  |  |     |     +--:(vpls)
|  |  |     |     |  +---w route-distinguisher?      uint32
|  |  |     |     |  +---w sender-ve-id?             uint32
|  |  |     |     |  +---w receiver-ve-id?           uint32
|  |  |     |     +--:(mpls-mldp)
|  |  |     |        +---w (root-address)?
|  |  |     |           +--:(ip-address)
|  |  |     |           |  +---w source-address?   inet:ip-address
|  |  |     |           |  +---w group-ip-address? IP-Multicast
     -Group-Address
|  |  |     |           +--:(vpn)
|  |  |     |           |  +---w as-number         inet:as-number
|  |  |     |           +--:(global-id)
|  |  |     |              +---w lsp-id?                  string
|  |  |     +--:(system-info)
|  |  |        +---w system-id?               router-id
|  |  +---w session-type-enum?    enumeration
|  |  +---w source-interface?     if:interface-ref
|  |  +---w outbound-interface?   if:interface-ref
|  |  +---w count?                uint32
|  |  +---w vrf?                  coam:routing-instance-ref
|  |  +---w ttl?                  uint8
|  |  +---w packet-size?          uint32
|  +--ro output
|     +--ro error-code-list* [response-index]
|     |  +--ro response-index     uint32
|     |  +--ro status-code?       int32
|     |  +--ro status-sub-code?   uint8
|     +--ro src-test-point
|     |  +--ro vrf?                       routing-instance-ref
|     |  +--ro tp-location-type-value?    identityref
|     |  +--ro (tp-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
|     |  |  +--:(tp-attribute)
|     |  |  |  +--ro tp-attribute-type?       fec-type
|     |  |  |  +--ro (tp-attribute-value)?
|     |  |  |     +--:(ip-prefix)
|     |  |  |     |  +--ro ip-prefix?         inet:ip-prefix
|     |  |  |     +--:(bgp)
|     |  |  |     |  +--ro bgp?               inet:ip-prefix
|     |  |  |     +--:(tunnel)
|     |  |  |     |  +--ro tunnel-interface?        uint32
|     |  |  |     +--:(pw)
|     |  |  |     |  +--ro remote-pe-address?  inet:ip-address
|     |  |  |     |  +--ro pw-id?                   uint32
|     |  |  |     +--:(vpls)
|     |  |  |     |  +--ro route-distinguisher?     uint32
|     |  |  |     |  +--ro sender-ve-id?            uint32
|     |  |  |     |  +--ro receiver-ve-id?          uint32
|     |  |  |     +--:(mpls-mldp)
|     |  |  |        +--ro (root-address)?
|     |  |  |           +--:(ip-address)
|     |  |  |           |  +--ro source-address?  inet:ip-address
|     |  |  |           |  +--ro group-ip-address? IP-Multicast
    -Group-Address
|     |  |  |           +--:(vpn)
|     |  |  |           |  +--ro as-number?        inet:as-number
|     |  |  |           +--:(global-id)
|     |  |  |              +--ro lsp-id?           string
|     |  |  +--:(system-info)
|     |  |     +--ro system-id?               router-id
|     |  +--ro egress-intf-name?         if:interface-ref
|     +--ro dest-test-point
|     |  +--ro vrf?                        routing-instance-ref
|     |  +--ro tp-location-type-value?   identityref
|     |  +--ro (tp-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
|     |  |  +--:(tp-attribute)
|     |  |  |  +--ro tp-attribute-type?       fec-type
|     |  |  |  +--ro (tp-attribute-value)?
|     |  |  |     +--:(ip-prefix)
|     |  |  |     |  +--ro ip-prefix          inet:ip-prefix
|     |  |  |     +--:(bgp)
|     |  |  |     |  +--ro bgp?               inet:ip-prefix
|     |  |  |     +--:(tunnel)
```

```
|     |  |  |     |  +--ro tunnel-interface?         uint32
|     |  |  |     +--:(pw)
|     |  |  |     |  +--ro remote-pe-address?  inet:ip-address
|     |  |  |     |  +--ro pw-id?                    uint32
|     |  |  |     +--:(vpls)
|     |  |  |     |  +--ro route-distinguisher?      uint32
|     |  |  |     |  +--ro sender-ve-id?             uint32
|     |  |  |     |  +--ro receiver-ve-id?           uint32
|     |  |  |     +--:(mpls-mldp)
|     |  |  |        +--ro (root-address)?
|     |  |  |           +--:(ip-address)
|     |  |  |           |  +--ro source-address?  inet:ip-address
|     |  |  |           |  +--ro group-ip-address? IP-Multicast-
    Group-Address
|     |  |  |           +--:(vpn)
|     |  |  |           |  +--ro as-number?       inet:as-number
|     |  |  |           +--:(global-id)
|     |  |  |              +--ro lsp-id?          string
|     |  |  +--:(system-info)
|     |  |     +--ro system-id?               router-id
|     |  +--ro ingress-intf-name?        if:interface-ref
|     +--ro sequence-number?             uint64
|     +--ro hop-cnt?                     uint8
|     +--ro session-packet-statistics
|     |  +--ro rx-packet-count?    uint32
|     |  +--ro tx-packet-count?    uint32
|     |  +--ro rx-bad-packet?      uint32
|     |  +--ro tx-packet-failed?   uint32
|     +--ro session-error-statistics
|     |  +--ro packet-drops-count?         uint32
|     |  +--ro packet-reorder-count?       uint32
|     |  +--ro packets-out-of-seq-count?   uint32
|     |  +--ro packets-dup-count?          uint32
|     +--ro session-delay-statistics
|     |  +--ro time-resolution-value?   identityref
|     |  +--ro min-delay-value?         uint32
|     |  +--ro max-delay-value?         uint32
|     |  +--ro average-delay-value?     uint32
|     +--ro session-jitter-statistics
|        +--ro time-resolution-value?   identityref
|        +--ro min-jitter-value?        uint32
|        +--ro max-jitter-value?        uint32
|        +--ro average-jitter-value?    uint32
+---x cc-persistent-create {coam:continuity-check}?
|  +---w input
|  |  +---w destination-tp
|  |  |  +---w tp-location-type-value?   identityref
|  |  |  +---w (tp-address)?
```

```
|  |  |     +--:(mac-address)
|  |  |     |  +---w mac-address?              yang:mac-address
|  |  |     +--:(ipv4-address)
|  |  |     |  +---w ipv4-address?             inet:ipv4-address
|  |  |     +--:(ipv6-address)
|  |  |     |  +---w ipv6-address?             inet:ipv6-address
|  |  |     +--:(tp-attribute)
|  |  |     |  +---w tp-attribute-type?        fec-type
|  |  |     |  +---w (tp-attribute-value)?
|  |  |     |     +--:(ip-prefix)
|  |  |     |     |  +---w ip-prefix?                inet:ip-prefix
|  |  |     |     +--:(bgp)
|  |  |     |     |  +---w bgp?                      inet:ip-prefix
|  |  |     |     +--:(tunnel)
|  |  |     |     |  +---w tunnel-interface?         uint32
|  |  |     |     +--:(pw)
|  |  |     |     |  +---w remote-pe-address?        inet:ip-address
|  |  |     |     |  +---w pw-id?                    uint32
|  |  |     |     +--:(vpls)
|  |  |     |     |  +---w route-distinguisher?      uint32
|  |  |     |     |  +---w sender-ve-id?             uint32
|  |  |     |     |  +---w receiver-ve-id?           uint32
|  |  |     |     +--:(mpls-mldp)
|  |  |     |        +---w (root-address)?
|  |  |     |           +--:(ip-address)
|  |  |     |           |  +---w source-address?     inet:ip-address
|  |  |     |           |  +---w group-ip-address?   IP-Multicast-
    Group-Address
|  |  |     |           +--:(vpn)
|  |  |     |           |  +---w as-number?         inet:as-number
|  |  |     |           +--:(global-id)
|  |  |     |              +---w lsp-id?                   string
|  |  |     +--:(system-info)
|  |  |        +---w system-id?               router-id
|  |  +---w session-type-enum?    enumeration
|  |  +---w source-interface?     if:interface-ref
|  |  +---w outbound-interface?   if:interface-ref
|  |  +---w vrf?                  coam:routing-instance-ref
|  |  +---w ttl?                  uint8
|  |  +---w data-export-method?   export-method
|  |  +---w (cc-trigger)?
|  |     +--:(periodic)
|  |     |  +---w period?              yang:timeticks
|  |     |  +---w start-time?          yang:date-and-time
|  |     +--:(on-change)
|  |        +---w all-data-on-start?   boolean
|  |        +---w excluded-change*     change-type
|  +--ro output
```

```
|     +--ro error-code-list* [response-index]
|     |  +--ro response-index     uint32
|     |  +--ro status-code?       int32
|     |  +--ro status-sub-code?   uint8
|     +--ro cc-persistent-id?    string
+---x cc-persistent-modify {coam:continuity-check}?
|  +---w input
|  |  +---w cc-persistent-id?       string
|  |  +---w data-export-method?   export-method
|  |  +---w (cc-trigger)?
|  |     +--:(periodic)
|  |     |  +---w period?               yang:timeticks
|  |     |  +---w start-time?           yang:date-and-time
|  |     +--:(on-change)
|  |        +---w all-data-on-start?    boolean
|  |        +---w excluded-change*      change-type
|  +--ro output
|     +--ro error-code-list* [response-index]
|     |  +--ro response-index     uint32
|     |  +--ro status-code?       int32
|     |  +--ro status-sub-code?   uint8
|     +--ro cc-persistent-id?    string
+---x cc-persistent-delete {coam:continuity-check}?
|  +---w input
|  |  +---w cc-persistent-id?   string
|  +--ro output
|     +--ro error-code-list* [response-index]
|        +--ro response-index     uint32
|        +--ro status-code?       int32
|        +--ro status-sub-code?   uint8
+---x path-discovery
|  +---w input
|  |  +---w destination-tp
|  |  |  +---w tp-location-type-value?   identityref
|  |  |  +---w (tp-address)?
|  |  |     +--:(mac-address)
|  |  |     |  +---w mac-address?              yang:mac-address
|  |  |     +--:(ipv4-address)
|  |  |     |  +---w ipv4-address?             inet:ipv4-address
|  |  |     +--:(ipv6-address)
|  |  |     |  +---w ipv6-address?             inet:ipv6-address
|  |  |     +--:(tp-attribute)
|  |  |     |  +---w tp-attribute-type?        fec-type
|  |  |     |  +---w (tp-attribute-value)?
|  |  |     |     +--:(ip-prefix)
|  |  |     |     |  +---w ip-prefix?          inet:ip-prefix
|  |  |     |     +--:(bgp)
|  |  |     |     |  +---w bgp                 inet:ip-prefix
```

```
|  |  |     |     +--:(tunnel)
|  |  |     |     |  +---w tunnel-interface?   uint32
|  |  |     |     +--:(pw)
|  |  |     |     |  +---w remote-pe-address?  inet:ip-address
|  |  |     |     |  +---w pw-id?                    uint32
|  |  |     |     +--:(vpls)
|  |  |     |     |  +---w route-distinguisher?      uint32
|  |  |     |     |  +---w sender-ve-id?             uint32
|  |  |     |     |  +---w receiver-ve-id?           uint32
|  |  |     |     +--:(mpls-mldp)
|  |  |     |        +---w (root-address)?
|  |  |     |           +--:(ip-address)
|  |  |     |           |  +---w source-address? inet:ip-address
|  |  |     |           |  +---w group-ip-address? IP-Multicast
    -Group-Address
|  |  |     |           +--:(vpn)
|  |  |     |           |  +---w as-number?        inet:as-number
|  |  |     |           +--:(global-id)
|  |  |     |              +---w lsp-id?                   string
|  |  |     +--:(system-info)
|  |  |        +---w system-id?                router-id
|  |  +---w session-type-enum?    enumeration
|  |  +---w source-interface?     if:interface-ref
|  |  +---w outbound-interface?   if:interface-ref
|  |  +---w vrf?                  coam:routing-instance-ref
|  |  +---w max-ttl?              uint8
|  +--ro output
|     +--ro response-list* [response-index]
|     |  +--ro response-index     uint32
|     |  +--ro status-code?       int32
|     |  +--ro status-sub-code?   uint8
|     +--ro src-test-point
|     |  +--ro vrf?                      routing-instance-ref
|     |  +--ro tp-location-type-value?   identityref
|     |  +--ro (tp-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
|     |     +--:(tp-attribute)
|     |     |  +--ro tp-attribute-type?        fec-type
|     |     |  +--ro (tp-attribute-value)?
|     |     |     +--:(ip-prefix)
|     |     |     |  +--ro ip-prefix?          inet:ip-prefix
|     |     |     +--:(bgp)
|     |     |     |  +--ro bgp?                inet:ip-prefix
```

```
|     |     |     +--:(tunnel)
|     |     |     |  +--ro tunnel-interface?         uint32
|     |     |     +--:(pw)
|     |     |     |  +--ro remote-pe-address?  inet:ip-address
|     |     |     |  +--ro pw-id?                    uint32
|     |     |     +--:(vpls)
|     |     |     |  +--ro route-distinguisher?      uint32
|     |     |     |  +--ro sender-ve-id?             uint32
|     |     |     |  +--ro receiver-ve-id?           uint32
|     |     |     +--:(mpls-mldp)
|     |     |        +--ro (root-address)?
|     |     |           +--:(ip-address)
|     |     |           |  +--ro source-address? inet:ip-address
|     |     |           |  +--ro group-ip-address? IP-Multicast
    -Group-Address
|     |     |           +--:(vpn)
|     |     |           |  +--ro as-number?        inet:as-number
|     |     |           +--:(global-id)
|     |     |              +--ro lsp-id?           string
|     |     +--:(system-info)
|     |        +--ro system-id?               router-id
|     +--ro dest-test-point
|     |  +--ro vrf?                       routing-instance-ref
|     |  +--ro tp-location-type-value?    identityref
|     |  +--ro (tp-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
|     |     +--:(tp-attribute)
|     |     |  +--ro tp-attribute-type?       fec-type
|     |     |  +--ro (tp-attribute-value)?
|     |     |     +--:(ip-prefix)
|     |     |     |  +--ro ip-prefix          inet:ip-prefix
|     |     |     +--:(bgp)
|     |     |     |  +--ro bgp?               inet:ip-prefix
|     |     |     +--:(tunnel)
|     |     |     |  +--ro tunnel-interface?        uint32
|     |     |     +--:(pw)
|     |     |     |  +--ro remote-pe-address?  inet:ip-address
|     |     |     |  +--ro pw-id?                   uint32
|     |     |     +--:(vpls)
|     |     |     |  +--ro route-distinguisher?     uint32
|     |     |     |  +--ro sender-ve-id?            uint32
|     |     |     |  +--ro receiver-ve-id?          uint32
|     |     |     +--:(mpls-mldp)
```

```
|     |     |        +--ro (root-address)?
|     |     |           +--:(ip-address)
|     |     |           |  +--ro source-address? inet:ip-address
|     |     |           |  +--ro group-ip-address? IP-Multicast
      -Group-Address
|     |     |           +--:(vpn)
|     |     |           |  +--ro as-number?        inet:as-number
|     |     |           +--:(global-id)
|     |     |              +--ro lsp-id?                   string
|     |     +--:(system-info)
|     |        +--ro system-id?                router-id
|     +--ro sequence-number?             uint64
|     +--ro hop-cnt?                     uint8
|     +--ro session-packet-statistics
|     |  +--ro rx-packet-count?    uint32
|     |  +--ro tx-packet-count?    uint32
|     |  +--ro rx-bad-packet?      uint32
|     |  +--ro tx-packet-failed?   uint32
|     +--ro session-error-statistics
|     |  +--ro packet-drops-count?         uint32
|     |  +--ro packet-reorder-count?       uint32
|     |  +--ro packets-out-of-seq-count?   uint32
|     |  +--ro packets-dup-count?          uint32
|     +--ro session-delay-statistics
|     |  +--ro time-resolution-value?   identityref
|     |  +--ro min-delay-value?         uint32
|     |  +--ro max-delay-value?         uint32
|     |  +--ro average-delay-value?     uint32
|     +--ro session-jitter-statistics
|     |  +--ro time-resolution-value?   identityref
|     |  +--ro min-jitter-value?        uint32
|     |  +--ro max-jitter-value?        uint32
|     |  +--ro average-jitter-value?    uint32
|     +--ro path-verification
|     |  +--ro flow-info?                              string
|     |  +--ro session-path-verification-statistics
|     |     +--ro verified-count?   uint32
|     |     +--ro failed-count?     uint32
|     +--ro path-trace-info
|        +--ro path-trace-info-list* [index]
|           +--ro index                     uint32
|           +--ro vrf?                      routing-instance-ref
|           +--ro tp-location-type-value?   identityref
|           +--ro (tp-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
|           |  +--:(tp-attribute)
|           |  |  +--ro tp-attribute-type?       fec-type
|           |  |  +--ro (tp-attribute-value)?
|           |  |     +--:(ip-prefix)
|           |  |     |  +--ro ip-prefix?         inet:ip-prefix
|           |  |     +--:(bgp)
|           |  |     |  +--ro bgp?               inet:ip-prefix
|           |  |     +--:(tunnel)
|           |  |     |  +--ro tunnel-interface?        uint32
|           |  |     +--:(pw)
|           |  |     |  +--ro remote-pe-address?  inet:ip-address
|           |  |     |  +--ro pw-id?                   uint32
|           |  |     +--:(vpls)
|           |  |     |  +--ro route-distinguisher?     uint32
|           |  |     |  +--ro sender-ve-id?            uint32
|           |  |     |  +--ro receiver-ve-id?          uint32
|           |  |     +--:(mpls-mldp)
|           |  |        +--ro (root-address)?
|           |  |           +--:(ip-address)
|           |  |           |  +--ro source-address? inet:ip-address
|           |  |           |  +--ro group-ip-address? IP-Multicast
    -Group-Address
|           |  |           +--:(vpn)
|           |  |           |  +--ro as-number?       inet:as-number
|           |  |           +--:(global-id)
|           |  |              +--ro lsp-id?                  string
|           |  +--:(system-info)
|           |     +--ro system-id?              router-id
|           +--ro timestamp-sec?            yang:timestamp
|           +--ro timestamp-nanosec?        yang:timestamp
|           +--ro ingress-intf-name?        if:interface-ref
|           +--ro egress-intf-name?         if:interface-ref
|           +--ro queue-depth?              uint32
|           +--ro transit-delay?            uint32
|           +--ro app-meta-data?            uint64
|           +--ro opaque-tlvs
|              +--ro opaque-tlvs-list*
|                 +--ro type?     opaque-tlv-type
|                 +--ro length?   uint16
|                 +--ro value?    yang:hex-string
|
+---x pd-persistent-create
|  +---w input
|  |  +---w destination-tp
|  |  |  +---w tp-location-type-value?   identityref
|  |  |  +---w (tp-address)?
|  |  |     +--:(mac-address)
```

```
|  |  |     |  +---w mac-address?              yang:mac-address
|  |  |     +--:(ipv4-address)
|  |  |     |  +---w ipv4-address?             inet:ipv4-address
|  |  |     +--:(ipv6-address)
|  |  |     |  +---w ipv6-address?             inet:ipv6-address
|  |  |     +--:(tp-attribute)
|  |  |     |  +---w tp-attribute-type?        fec-type
|  |  |     |  +---w (tp-attribute-value)?
|  |  |     |     +--:(ip-prefix)
|  |  |     |     |  +---w ip-prefix?          inet:ip-prefix
|  |  |     |     +--:(bgp)
|  |  |     |     |  +---w bgp?                inet:ip-prefix
|  |  |     |     +--:(tunnel)
|  |  |     |     |  +---w tunnel-interface?         uint32
|  |  |     |     +--:(pw)
|  |  |     |     |  +---w remote-pe-address?  inet:ip-address
|  |  |     |     |  +---w pw-id?                    uint32
|  |  |     |     +--:(vpls)
|  |  |     |     |  +---w route-distinguisher?      uint32
|  |  |     |     |  +---w sender-ve-id?             uint32
|  |  |     |     |  +---w receiver-ve-id?           uint32
|  |  |     |     +--:(mpls-mldp)
|  |  |     |        +---w (root-address)?
|  |  |     |           +--:(ip-address)
|  |  |     |           |  +---w source-address inet:ip-address
|  |  |     |           |  +---w group-ip-address? IP-Multicast
    -Group-Address
|  |  |     |           +--:(vpn)
|  |  |     |           |  +---w as-number?      inet:as-number
|  |  |     |           +--:(global-id)
|  |  |     |              +---w lsp-id?                 string
|  |  |     +--:(system-info)
|  |  |        +---w system-id?               router-id
|  |  +---w session-type-enum?    enumeration
|  |  +---w source-interface?     if:interface-ref
|  |  +---w outbound-interface?   if:interface-ref
|  |  +---w vrf?                  coam:routing-instance-ref
|  |  +---w max-ttl?              uint8
|  |  +---w data-export-method?   export-method
|  |  +---w (pd-trigger)?
|  |     +--:(periodic)
|  |     |  +---w period?              yang:timeticks
|  |     |  +---w start-time?          yang:date-and-time
|  |     +--:(on-change)
|  |        +---w all-data-on-start?   boolean
|  |        +---w excluded-change*     change-type
|  +--ro output
|     +--ro response-list* [response-index]
```

```
    |        +--ro response-index     uint32
    |        +--ro status-code?       int32
    |        +--ro status-sub-code?   uint8
    |        +--ro pd-persistent-id?    string
    +---x pd-persistent-modify
    |  +---w input
    |  |  +---w pd-persistent-id?       string
    |  |  +---w data-export-method?   export-method
    |  |  +---w (pd-trigger)?
    |  |     +--:(periodic)
    |  |     |  +---w period?               yang:timeticks
    |  |     |  +---w start-time?           yang:date-and-time
    |  |     +--:(on-change)
    |  |        +---w all-data-on-start?    boolean
    |  |        +---w excluded-change*      change-type
    |  +--ro output
    |     +--ro response-list* [response-index]
    |        +--ro response-index     uint32
    |        +--ro status-code?       int32
    |        +--ro status-sub-code?   uint8
    |        +--ro pd-persistent-id?    string
    +---x pd-persistent-delete
       +---w input
       |  +---w pd-persistent-id?   string
       +--ro output
          +--ro response-list* [response-index]
             +--ro response-index     uint32
             +--ro status-code?       int32
             +--ro status-sub-code?   uint8
```

data hierarchy of OAM Retrieval Methods

4. OAM Retrieval Methods YANG Module

<CODE BEGINS> file "ietf-connectionless-oam-methods.yang"

```
module ietf-connectionless-oam-methods {
namespace "urn:ietf:params:xml:ns:yang:ietf-connectionless-oam-methods";
  prefix coam-methods;

  import ietf-interfaces {
    prefix if;
  }
  import ietf-connectionless-oam {
    prefix coam;
  }
  import ietf-yang-types {
```

```
    prefix yang;
  }

    organization "IETF LIME Working Group";
    contact
        "Deepak Kumar dekumar@cisco.com
         Qin Wu       bill.wu@huawei.com
         S Raghavan   srihari@cisco.com
         Zitao Wang   wangzitao@huawei.com
         R Rahman     rrahman@cisco.com";

    description
      "This YANG module defines the RPCs for ,
       connectionless OAM to be used within IETF
       in a protocol Independent 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 2017-02-15 {
      description
        "02 version";
      reference
          "draft-ietf-lime-yang-connectionless-oam-methods";
    }

    rpc continuity-check {
      if-feature coam:continuity-check;
      description
        "Generates continuity-check as per RFC7276.";
      input {
        container destination-tp {
          uses coam:tp-address;
          description
            "Destination test point.";
        }
        uses coam:session-type;
        leaf source-interface {
          type if:interface-ref;
                  description
                  "Source interface.";
        }
        leaf outbound-interface {
          type if:interface-ref;

                  description
                  "Outbound interface.";
```

```
        }
        leaf count {
          type uint32;
          default "5";
          description
            "Specifies the number of
                        packets that will be sent.";
        }
        leaf vrf {
          type coam:routing-instance-ref;
          description
            "VRF instance.";
        }

        leaf ttl {
          type uint8;
          default "255";
          description
            "Time to live (TTL).";
        }
        leaf packet-size {
          type uint32 {
            range "64..10000";
          }
          default "64";
          description
            "Size of ping echo request
                        packets, in octets";
        }
      }
      output {
        list error-code-list {
          key "response-index";
          leaf response-index {
            type uint32;
            description
              "Response index.";
          }
          leaf status-code {
            type int32;
            description
              "Error code is ";
          }
          leaf status-sub-code {
            type uint8;
            description
              "Sub code.";
          }
```

```
          description
            "Error code list.";
        }

        uses coam:continuity-check-data;
      }
    }

    identity export-method {
      description
        "Base identity to represent a conceptual export-method.";
    }

    identity ipfix-export {
      base export-method;
      description
        "IPFIX based export. Configuration provided separately.";
    }

    identity yang-push-export {
      base export-method;
      description
          "Yang-push from draft-ietf-netconf-yang-push";
    }

    typedef export-method {
      type identityref {
        base export-method;
      }
      description
          "Export method type.";
    }

    typedef change-type {
      type enumeration {
        enum "create" {
        description
          "Change due to a create.";
        }
        enum "delete" {
        description
          "Change due to a delete.";
        }
        enum "modify" {
        description
          "Change due to an update.";
        }
      }
```

```
      description
        "Different types of changes that may occur.";
    }

    rpc cc-persistent-create {
      if-feature coam:continuity-check;
      description
        "Generates persistent continuity-check.";
      input {
        container destination-tp {
          uses coam:tp-address;
          description
            "Destination test point.";
        }

        uses coam:session-type;

        leaf source-interface {
          type if:interface-ref;
            description
                        "Source interface.";
        }

        leaf outbound-interface {
          type if:interface-ref;
            description
                        "Outbound interface.";
        }

        leaf vrf {
          type coam:routing-instance-ref;
          description
                  "VRF instance.";
        }

        leaf ttl {
          type uint8;
          default "255";
          description
            "Time to live (TTL).";
        }

        leaf data-export-method {
          type export-method;
          description
                  "Type of export.";
        }
```

```
        choice cc-trigger {
          description
            "Defines necessary conditions for
                        periodic or on-change trigger.";
          case periodic {
            description "Periodic reports.";
            leaf period {
              type yang:timeticks;
              description "Time interval between reports.";
            }
            leaf start-time {
              type yang:date-and-time;
              description
                          "Timestamp from which reports are started.";
            }
          }
          case on-change {
            description
                        "On-change trigger and not periodic.";
            leaf all-data-on-start {
              type boolean;
              description
                          "Full update needed on start or not.";
            }
            leaf-list excluded-change {
              type change-type;
              description
                          "Restrict which changes trigger an update.";
            }
          }
        }
      }
      output {
        list error-code-list {
          key "response-index";
          leaf response-index {
            type uint32;
            description
              "Response index.";
          }

          leaf status-code {
            type int32;
            description
              "Error code.";
          }

          leaf status-sub-code {
```

```
            type uint8;
            description
              "Sub code.";
          }

          description
            "Error code list.";
        }
        leaf cc-persistent-id {
          type string;
          description
                  "Id to act as a cookie.";
        }
      }
    }

    rpc cc-persistent-modify {
      if-feature coam:continuity-check;
      description
        "Modifies persistent continuity-check
                as per RFC7276.";
      input {
        leaf cc-persistent-id {
          type string;
          description
                  "Cookie Id to be used for modifications.";
        }

        leaf data-export-method {
          type export-method;
          description
                  "Type of export to use.";
        }

        choice cc-trigger {
          description
            "Defines necessary conditions for
                        periodic or on-change trigger.";
          case periodic {
            description "Periodic reports.";
            leaf period {
              type yang:timeticks;
              description
                          "Time interval between reports.";
            }
            leaf start-time {
              type yang:date-and-time;
              description
```

```
                        "Timestamp from which reports are started.";
          }
        }
        case on-change {
          description
                      "On-change trigger and not periodic.";
          leaf all-data-on-start {
            type boolean;
            description
                        "Full update needed on start or not.";
          }
          leaf-list excluded-change {
            type change-type;
            description
                        "Restrict which changes trigger an update.";
          }
        }
      }
    }
    output {
      list error-code-list {
        key "response-index";
        leaf response-index {
          type uint32;
          description
                      "Response index.";
        }

        leaf status-code {
          type int32;
          description
            "Error code";
        }

        leaf status-sub-code {
          type uint8;
          description
            "Sub code.";
        }

        description
          "Error code list.";
      }

      leaf cc-persistent-id {
        type string;
        description
                "Id to represent a cookie.";
```

```
        }
      }
    }

    rpc cc-persistent-delete {
      if-feature coam:continuity-check;
      description
        "Deletes persistent continuity-check as per RFC7276.";
      input {
        leaf cc-persistent-id {
          type string;
          description
                  "Cookie Id to be used in deletion.";
        }
      }
      output {
        list error-code-list {
          key "response-index";
          leaf response-index {
            type uint32;
            description
                        "Response index.";
          }

          leaf status-code {
            type int32;
            description
              "Error code.";
          }

          leaf status-sub-code {
            type uint8;
            description
                        "Sub code.";
          }

          description
            "Error code list.";
        }
      }
    }

    rpc path-discovery {
      description
        "Generates path discovery as per RFC7276.";
      input {
        container destination-tp {
          uses coam:tp-address;
```

```
          description
          "Destination test point.";

        }
        uses coam:session-type;
        leaf source-interface {
          type if:interface-ref;
          description
            "Source interface.";
        }

        leaf outbound-interface {
          type if:interface-ref;
          description
            "Outbound interface.";
        }

        leaf vrf {
          type coam:routing-instance-ref;
          description
            "VRF";
        }

        leaf max-ttl {
          type uint8;
          default "255";
          description
            "Max ttl.";
        }
      }
      output {
        list response-list {
          key "response-index";
          description
            "Path discovery response list.";
          leaf response-index {
            type uint32;
            description
              "Response index.";
          }

          leaf status-code {
            type int32;
            description
              "Error code is ";
          }

          leaf status-sub-code {
```

```
            type uint8;
            description
              "Sub code is ";
          }
        }

        uses coam:path-discovery-data;

       }
    }

    rpc pd-persistent-create {
      description
        "Generates persistent path discovery.";
      input {
        container destination-tp {
          uses coam:tp-address;
          description
            "Destination test point.";

        }

        uses coam:session-type;

        leaf source-interface {
          type if:interface-ref;
          description
            "Source interface.";
        }

        leaf outbound-interface {
          type if:interface-ref;
          description
            "Outbound interface.";
        }

        leaf vrf {
          type coam:routing-instance-ref;
          description
            "VRF";
        }

        leaf max-ttl {
          type uint8;
          default "255";
          description
            "Max ttl.";
        }
```

```
        leaf data-export-method {
          type export-method;
          description
                  "Type of export.";
        }

        choice pd-trigger {
          description
            "Defines necessary conditions
                        for periodic or on-change
             trigger.";
          case periodic {
            description
                        "Periodic reports.";
            leaf period {
              type yang:timeticks;
              description
                          "Time interval between reports.";
            }
            leaf start-time {
              type yang:date-and-time;
              description
                          "Timestamp from which reports are started.";
            }
          }
          case on-change {
            description
                        "On-change trigger and not periodic.";
            leaf all-data-on-start {
              type boolean;
              description
                          "Full update needed on start or not.";
            }
            leaf-list excluded-change {
              type change-type;
              description
                          "Restrict which changes trigger an update.";
            }
          }
        }
      }
      output {
        list response-list {
          key "response-index";
          description
            "Path discovery response list.";
          leaf response-index {
            type uint32;
```

```
            description
            "Response index.";
          }

          leaf status-code {
            type int32;
            description
            "Error code is ";
          }

          leaf status-sub-code {
            type uint8;
            description
            "Sub code is ";
          }

          leaf pd-persistent-id {
            type string;
            description
                        "Id to act as a cookie.";
          }
        }
      }
    }

    rpc pd-persistent-modify {
      description
        "Modifies persistent path discovery.";
      input {
        leaf pd-persistent-id {
          type string;
          description "Cookie Id to be used for modifications.";
        }

        leaf data-export-method {
          type export-method;
          description "Type of export.";
        }

        choice pd-trigger {
          description
            "Defines necessary conditions for periodic or on-change
             trigger.";
          case periodic {
            description "Periodic reports.";
            leaf period {
              type yang:timeticks;
              description
```

```
                        "Time interval between reports.";
            }
            leaf start-time {
              type yang:date-and-time;
              description
                        "Timestamp from which reports are started.";
            }
          }
          case on-change {
            description
                      "On-change trigger and not periodic.";
            leaf all-data-on-start {
              type boolean;
              description
                        "Full update needed on start or not.";
            }
            leaf-list excluded-change {
              type change-type;
              description
                        "Restrict which changes trigger an update.";
            }
          }
        }
      }
      output {
        list response-list {
          key "response-index";
          description
            "path discovery response list.";
          leaf response-index {
            type uint32;
            description
              "response index.";
          }

          leaf status-code {
            type int32;
            description
              "error code is ";
          }

          leaf status-sub-code {
            type uint8;
            description
              "sub code is ";
          }

          leaf pd-persistent-id {
```

```
            type string;
            description "Id to act as a cookie.";
          }
        }
      }
    }

    rpc pd-persistent-delete  {
      description
        "Deletes persistent path discovery.";
      input {
        leaf pd-persistent-id {
          type string;
          description "Cookie Id to be used in deletion.";
        }
      }

      output {
        list response-list {
          key "response-index";
          description
            "path discovery response list.";
          leaf response-index {
            type uint32;
            description
              "response index.";
          }

          leaf status-code {
            type int32;
            description
              "error code is ";
          }

          leaf status-sub-code {
            type uint8;
            description
              "sub code is ";
          }
        }
      }
    }
}
```

<CODE ENDS>

5. Security Considerations

The YANG modules defined in this memo are designed to be accessed via the NETCONF protocol [RFC6241]. The lowest NETCONF layer is the secure transport layer and the mandatory to implement secure transport is SSH [RFC6242]. The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF users to a pre-configured subset of all available NETCONF protocol operations and content. Some of the RPC operations in the "ietf-connectionless-oam-methods" YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control access to these operations.

These are the operations and their sensitivity/vulnerability:

o continuity-check: Generates continuity check.

o path-discovery: Generates path discovery.

6. IANA Considerations

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

URI: urn:ietf:params:xml:ns:yang:ietf-connectionless-oam-methods

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-connectionless-oam-methods

namespace: urn:ietf:params:xml:ns:yang:ietf-connectionless-oam-methods

prefix: coam-methods

reference: RFC XXXX

7. References

7.1. Normative References

[I-D.ietf-lime-yang-connectionless-oam]
Kumar, D., Wang, Z., Wu, Q., Rahman, R., and S. Raghavan, "Generic YANG Data Model for Connectionless Operations, Administration, and Maintenance(OAM) protocols", draft-ietf-lime-yang-connectionless-oam-03 (work in progress), December 2016.

[I-D.ietf-netconf-yang-push]
Clemm, A., Voit, E., Prieto, A., Tripathy, A., Nilsen-Nygaard, E., Bierman, A., and B. Lengyel, "Subscribing to YANG datastore push updates", draft-ietf-netconf-yang-push-04 (work in progress), October 2016.

[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>.

[RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, DOI 10.17487/RFC7011, September 2013, <http://www.rfc-editor.org/info/rfc7011>.

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

7.2. Informative References

[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>.

[RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <http://www.rfc-editor.org/info/rfc5880>.

[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>.

Authors' Addresses

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

Email: dekumar@cisco.com

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

Email: wangzitao@huawei.com

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

Email: bill.wu@huawei.com

Reshad Rahman
CISCO Systems
2000 Innovation Drive
KANATA, ONTARIO K2K 3E8
CANADA

Email: rrahman@cisco.com

Srihari Raghavan
CISCO Systems
TRIL INFOPARK SEZ, Ramanujan IT City
NEVILLE BLOCK, 2nd floor, Old Mahabalipuram Road
CHENNAI, TAMIL NADU 600113
INDIA

Email: srihari@cisco.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: November 16, 2017

D. Kumar
Cisco
M. Wang
Q. Wu
Huawei
R. Rahman
S. Raghavan
Cisco

May 15, 2017

Retrieval Methods YANG Data Model for Connectionless Operations, Administration, and Maintenance(OAM) protocols
draft-ietf-lime-yang-connectionless-oam-methods-02

Abstract

This document presents a retrieval method YANG Data model for connectionless OAM protocols. It provides a technology-independent RPC commands for connectionless OAM protocols. The retrieval methods model presented here can be extended to include technology specific details. This is leading to uniformity between OAM protocols and support both nested OAM workflows (i.e., performing OAM functions at different levels through a unified interface) and interactive OAM workflows (i.e., performing OAM functions at same levels through a unified interface).

Status of This Memo

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

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

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

This Internet-Draft will expire on November 16, 2017.

Table of Contents

1. Introduction

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

1. Monitor reachability of destinations (Reachability 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] and BFD [RFC5880] are well-known fault verification and isolation tools, respectively, for IP networks. Over the years, different technologies have developed similar tools for similar purposes.

In this document, we present a retrieval method YANG Data model for connectionless OAM protocols. This module provides technology-independent RPC commands for connectionless OAM protocols. It is separated from the generic YANG model for connectionless OAM [I-D.ietf-lime-yang-connectionless-oam] and can avoid mixing the models for the retrieved-data from the retrieval procedures. It is expected that retrieval procedures would evolve faster than the data model [I-D.ietf-lime-yang-connectionless-oam] and will allow new procedures to be defined for retrieval of the same data defined by the base data model.

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 terms are defined in [RFC6241] and are not redefined here:

- client
- configuration data
- server
- state data

The following terms are defined in [RFC6020] and are not redefined here:

- augment
- data model
- data node

The terminology for describing YANG data models is found in [RFC6020].

2.1. Terminology

TP - Test Point

MAC - Media Access Control

RPC - A Remote Procedure Call, as used within the NETCONF protocol

2.2. Tree Diagrams

A simplified graphical representation of the data model is used in this document. The meaning of the symbols in these diagrams is as follows:

Each node is printed as:

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

<status> is one of:
+ for current

<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

3. Overview of the Connectionless OAM retrieval methods Model

In this document, we present a retrieval method YANG Data model for connectionless OAM protocols. This module provides technology-independent retrieval procedures (RPC commands) for connectionless OAM protocols. It provides a flexible way to retrieve the retrieved-data which defined by the "ietf-connectionless-oam.yang" [I-D.ietf-lime-yang-connectionless-oam].

3.1. 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 obtaining a response.

Under ’connectionless-oam-methods’ module, we summarize the common OAM functions and define the generic RPC commands: ’continuity-check’ and ’path-discovery’. In practice, these commands are supported by corresponding technology-specific OAM tools [RFC7276]. For example, for the IP OAM model, the continuity-check RPC corresponds to the IP Ping [RFC792] [RFC4443], while the path-discovery RPC command corresponds to IP Traceroute [RFC792] [RFC4443].

Note that the RPC command presented in this document is the base building block, which is used to derive a model for a technology-specific OAM (i.e., ICMP ping [RFC792] [RFC4443], LSP ping [RFC4379]), the base building block should be extended with corresponding technology specific parameters. To facilitate this and for future enhancements to data retrieval methods, the RPCs are captured under a separate module.

The generic ’path-discovery-data’ and ’continuity-check-data’ are used as data outputs from the different RPCs described in the document. Similar methods including other RPCs can retrieve the data using the same data model.

```
    rpc continuity-check {
      if-feature coam:continuity-check;
      description
        "Generates continuity-check as per RFC7276.";
      input {
        container destination-tp {
          uses coam:tp-address;
                  description
                  "destination test point.";
        }
        uses coam:session-type;
        leaf source-interface {
```

```
            type if:interface-ref;
                    description
                    "source interface.";
          }
          leaf outbound-interface {
            type if:interface-ref;

                    description
                    "outbound interface.";
          }
          leaf count {
            type uint32;
            default "5";
            description
              "Specifies the number of packets that will be sent.";
          }
          leaf vrf {
            type coam:routing-instance-ref;
            description
              "vrf instance.";
          }
          leaf ttl {
            type uint8;
            default "255";
            description
              "Time to live (TTL).";
          }
          leaf packet-size {
            type uint32 {
              range "64..10000";
            }
            default "64";
            description
              "Size of ping echo request packets, in octets";
          }
        }
        output {
          list error-code-list {
            key "response-index";
            leaf response-index {
              type uint32;
                      description
                      "response index.";
            }
            leaf status-code {
              type int32;
              description
                "error code is ";
```

```
            }
            leaf status-sub-code {
              type uint8;
                      description
                      "sub code.";
            }
                    description
                    "error code list.";
          }

          uses coam:continuity-check-data;
        }
      }

      rpc path-discovery {
        description
          "Generates path discovery as per RFC7276.";
        input {
          container destination-tp {
            uses coam:tp-address;
                    description
                    "destination test point.";
          }
          uses coam:session-type;
          leaf source-interface {
            type if:interface-ref;
            description
              "source interface.";
          }
          leaf outbound-interface {
            type if:interface-ref;
            description
              "outbound interface.";
          }
          leaf vrf {
            type coam:routing-instance-ref;
            description
              "vrf";
          }
          leaf max-ttl {
            type uint8;
            default "255";
            description
              "max ttl.";
          }
        }
        output {
          list response-list {
```

```
            key "response-index";
            description
              "path discovery response list.";
            leaf response-index {
              type uint32;
              description
                "response index.";
            }
            leaf status-code {
              type int32;
              description
                "error code is ";
            }
            leaf status-sub-code {
              type uint8;

                      description
                      "sub code is ";
            }
          }

          uses coam:path-discovery-data;
        }
      }
```

Snippet of data hierarchy related to RPC calls

3.2. OAM Retrieval Methods Hierarchy

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

```
   module: ietf-connectionless-oam-methods
   rpcs:
      +---x continuity-check {coam:continuity-check}?
      |  +---w input
      |  |  +---w destination-tp
      |  |  +---w source-interface?     if:interface-ref
      |  |  +---w outbound-interface?   if:interface-ref
      |  |  +---w count?                uint32
      |  |  +---w vrf?                  coam:routing-instance-ref
      |  |  +---w ttl?                  uint8
      |  |  +---w packet-size?          uint32
      |  +--ro output
      |     +--ro error-code-list* [response-index]
      |        +--ro response-index     uint32
      |        +--ro status-code?       int32
      |        +--ro status-sub-code?   uint8
      +---x path-discovery {coam:path-discovery}?
         +---w input
         |  +---w destination-tp
         |  +---w source-interface?     if:interface-ref
         |  +---w outbound-interface?   if:interface-ref
         |  +---w vrf?                  coam:routing-instance-ref
         |  +---w max-ttl?              uint8
         +--ro output
            +--ro response-list* [response-index]
               +--ro response-index     uint32
               +--ro status-code?       int32
               +--ro status-sub-code?   uint8
```

data hierarchy of OAM Retrieval Methods

4. OAM Retrieval Methods YANG Module

<CODE BEGINS> file "ietf-connectionless-oam-methods.yang"

```
module ietf-connectionless-oam-methods {
  namespace "urn:ietf:params:xml:ns:yang:ietf-connectionless-oam-methods";
  prefix coam-methods;

  import ietf-interfaces {
    prefix if;
  }
  import ietf-connectionless-oam {
    prefix coam;
  }
  import ietf-yang-types {
    prefix yang;
  }
```

```
  organization
    "IETF LIME Working Group";
  contact
    "Deepak Kumar dekumar@cisco.com
     Qin Wu       bill.wu@huawei.com
     S Raghavan   srihari@cisco.com
     Zitao Wang   wangzitao@huawei.com
     R Rahman     rrahman@cisco.com";
  description
    "This YANG module defines the RPCs for ,
     connectionless OAM to be used within IETF
     in a protocol Independent 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 2017-02-15 {
    description
      "02 version";
    reference "draft-ietf-lime-yang-connectionless-oam-methods";
  }

  rpc continuity-check {
    if-feature "coam:continuity-check";
    description
      "Generates continuity-check as per RFC7276.";
    input {
      container destination-tp {
        uses coam:tp-address;
        description
          "Destination test point.";
      }
      uses coam:session-type;
      leaf source-interface {
        type if:interface-ref;
        description
          "Source interface.";
      }
      leaf outbound-interface {
        type if:interface-ref;
        description
          "Outbound interface.";
      }
      leaf count {
        type uint32;
        default "5";
```

```
        description
          "Specifies the number of
           packets that will be sent.";
      }
      leaf vrf {
        type coam:routing-instance-ref;
        description
          "VRF instance.";
      }
      leaf ttl {
        type uint8;
        default "255";
        description
          "Time to live (TTL).";
      }
      leaf packet-size {
        type uint32 {
          range "64..10000";
        }
        default "64";
        description
          "Size of ping echo request
           packets, in octets";
      }
    }
    output {
      list error-code-list {
        key "response-index";
        leaf response-index {
          type uint32;
          description
            "Response index.";
        }
        leaf status-code {
          type int32;
          description
            "Error code is ";
        }
        leaf status-sub-code {
          type uint8;
          description
            "Sub code.";
        }
        description
          "Error code list.";
      }
      uses coam:continuity-check-data;
    }
```

```
  }

  rpc path-discovery {
    if-feature "coam:path-discovery";
    description
      "Generates path discovery as per RFC7276.";
    input {
      container destination-tp {
        uses coam:tp-address;
        description
          "Destination test point.";
      }
      uses coam:session-type;
      leaf source-interface {
        type if:interface-ref;
        description
          "Source interface.";
      }
      leaf outbound-interface {
        type if:interface-ref;
        description
          "Outbound interface.";
      }
      leaf vrf {
        type coam:routing-instance-ref;
        description
          "VRF";
      }
      leaf max-ttl {
        type uint8;
        default "255";
        description
          "Max ttl.";
      }
    }
    output {
      list response-list {
        key "response-index";
        description
          "Path discovery response list.";
        leaf response-index {
          type uint32;
          description
            "Response index.";
        }
        leaf status-code {
          type int32;
          description
```

```
            "Error code is ";
        }
        leaf status-sub-code {
          type uint8;
          description
            "Sub code is ";
        }
      }
      uses coam:path-discovery-data;
    }
  }
}
```

<CODE ENDS>

5. Security Considerations

The YANG modules defined in this memo are designed to be accessed via the NETCONF protocol [RFC6241]. The lowest NETCONF layer is the secure transport layer and the mandatory to implement secure transport is SSH [RFC6242]. The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF users to a pre-configured subset of all available NETCONF protocol operations and content. Some of the RPC operations in the "ietf-connectionless-oam-methods" YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control access to these operations.

These are the operations and their sensitivity/vulnerability:

o continuity-check: Generates continuity check.

o path-discovery: Generates path discovery.

6. IANA Considerations

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

URI: urn:ietf:params:xml:ns:yang:ietf-connectionless-oam-methods

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-connectionless-oam-methods

namespace: urn:ietf:params:xml:ns:yang:ietf-connectionless-oam-methods

prefix: coam-methods

reference: RFC XXXX

7. References

7.1. Normative References

[I-D.ietf-lime-yang-connectionless-oam]
Kumar, D., Wang, Z., Wu, Q., Rahman, R., and S. Raghavan, "Generic YANG Data Model for Connectionless Operations, Administration, and Maintenance(OAM) protocols", draft-ietf-lime-yang-connectionless-oam-04 (work in progress), February 2017.

[I-D.ietf-netconf-yang-push]
Clemm, A., Voit, E., Prieto, A., Tripathy, A., Nilsen-Nygaard, E., Bierman, A., and B. Lengyel, "Subscribing to YANG datastore push updates", draft-ietf-netconf-yang-push-06 (work in progress), April 2017.

[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>.

[RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, DOI 10.17487/RFC7011, September 2013, <http://www.rfc-editor.org/info/rfc7011>.

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

7.2. Informative References

[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>.

[RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <http://www.rfc-editor.org/info/rfc5880>.

[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>.

Appendix A. Appendix

The following are some examples of extensions possible to the yang model. The example discusses persistent methods.

The persistent method RPCs are commands to trigger persistent continuity check or path discovery OAM while specifying the options for data-export from the device. Internet Protocol Flow Information Export (IPFIX) [RFC7011] or yang-push [I-D.ietf-netconf-yang-push]. are currently outlined here as data export options and more can be added in future. It should be noted that the persistent methods are

used to trigger create, modify and delete the persistent state associated with the data export options. The data export specific configurations are beyond the scope of this document.

A.1. OAM Retrieval Persistent Methods YANG Module

<CODE BEGINS> file "ietf-connectionless-oam-persistent-methods.yang"

```
  identity export-method {
    description
      "Base identity to represent a conceptual export-method.";
  }

  identity ipfix-export {
    base export-method;
    description
      "IPFIX based export. Configuration provided separately.";
  }

  identity yang-push-export {
    base export-method;
    description
      "Yang-push from draft-ietf-netconf-yang-push";
  }

  typedef export-method {
    type identityref {
      base export-method;
    }
    description
      "Export method type.";
  }

  typedef change-type {
    type enumeration {
      enum "create" {
        description
          "Change due to a create.";
      }
      enum "delete" {
        description
          "Change due to a delete.";
      }
      enum "modify" {
        description
          "Change due to an update.";
      }
    }
```

```
      description
        "Different types of changes that may occur.";
    }

    rpc cc-persistent-create {
      if-feature "coam:continuity-check";
      description
        "Generates persistent continuity-check.";
      input {
        container destination-tp {
          uses coam:tp-address;
          description
            "Destination test point.";
        }
        uses coam:session-type;
        leaf source-interface {
          type if:interface-ref;
          description
            "Source interface.";
        }
        leaf outbound-interface {
          type if:interface-ref;
          description
            "Outbound interface.";
        }
        leaf vrf {
          type coam:routing-instance-ref;
          description
            "VRF instance.";
        }
        leaf ttl {
          type uint8;
          default "255";
          description
            "Time to live (TTL).";
        }
        leaf data-export-method {
          type export-method;
          description
            "Type of export.";
        }
        choice cc-trigger {
          description
            "Defines necessary conditions for
             periodic or on-change trigger.";
          case periodic {
            description
              "Periodic reports.";
```

```
            leaf period {
              type yang:timeticks;
              description
                "Time interval between reports.";
            }
            leaf start-time {
              type yang:date-and-time;
              description
                "Timestamp from which reports are started.";
            }
          }
          case on-change {
            description
              "On-change trigger and not periodic.";
            leaf all-data-on-start {
              type boolean;
              description
                "Full update needed on start or not.";
            }
            leaf-list excluded-change {
              type change-type;
              description
                "Restrict which changes trigger an update.";
            }
          }
        }
      }
      output {
        list error-code-list {
          key "response-index";
          leaf response-index {
            type uint32;
            description
              "Response index.";
          }
          leaf status-code {
            type int32;
            description
              "Error code.";
          }
          leaf status-sub-code {
            type uint8;
            description
              "Sub code.";
          }
          description
            "Error code list.";
        }
```

```
        leaf cc-persistent-id {
          type string;
          description
            "Id to act as a cookie.";
        }
      }
    }
    rpc cc-persistent-modify {
      if-feature "coam:continuity-check";
      description
        "Modifies persistent continuity-check
         as per RFC7276.";
      input {
        leaf cc-persistent-id {
          type string;
          description
            "Cookie Id to be used for modifications.";
        }
        leaf data-export-method {
          type export-method;
          description
            "Type of export to use.";
        }
        choice cc-trigger {
          description
            "Defines necessary conditions for
             periodic or on-change trigger.";
          case periodic {
            description
              "Periodic reports.";
            leaf period {
              type yang:timeticks;
              description
                "Time interval between reports.";
            }
            leaf start-time {
              type yang:date-and-time;
              description
                "Timestamp from which reports are started.";
            }
          }
          case on-change {
            description
              "On-change trigger and not periodic.";
            leaf all-data-on-start {
              type boolean;
              description
                "Full update needed on start or not.";
```

```
            }
            leaf-list excluded-change {
              type change-type;
              description
                "Restrict which changes trigger an update.";
            }
          }
        }
      }
      output {
        list error-code-list {
          key "response-index";
          leaf response-index {
            type uint32;
            description
              "Response index.";
          }
          leaf status-code {
            type int32;
            description
              "Error code";
          }
          leaf status-sub-code {
            type uint8;
            description
              "Sub code.";
          }
          description
            "Error code list.";
        }
        leaf cc-persistent-id {
          type string;
          description
            "Id to represent a cookie.";
        }
      }
    }
    rpc cc-persistent-delete {
      if-feature "coam:continuity-check";
      description
        "Deletes persistent continuity-check as per RFC7276.";
      input {
        leaf cc-persistent-id {
          type string;
          description
            "Cookie Id to be used in deletion.";
        }
      }
```

```
      output {
        list error-code-list {
          key "response-index";
          leaf response-index {
            type uint32;
            description
              "Response index.";
          }
          leaf status-code {
            type int32;
            description
              "Error code.";
          }
          leaf status-sub-code {
            type uint8;
            description
              "Sub code.";
          }
          description
            "Error code list.";
        }
      }
    }

   rpc pd-persistent-create {
      description
        "Generates persistent path discovery.";
      input {
        container destination-tp {
          uses coam:tp-address;
          description
            "Destination test point.";
        }
        uses coam:session-type;
        leaf source-interface {
          type if:interface-ref;
          description
            "Source interface.";
        }
        leaf outbound-interface {
          type if:interface-ref;
          description
            "Outbound interface.";
        }
        leaf vrf {
          type coam:routing-instance-ref;
          description
            "VRF";
```

```
        }
        leaf max-ttl {
          type uint8;
          default "255";
          description
            "Max ttl.";
        }
        leaf data-export-method {
          type export-method;
          description
            "Type of export.";
        }
        choice pd-trigger {
          description
            "Defines necessary conditions
             for periodic or on-change
             trigger.";
          case periodic {
            description
              "Periodic reports.";
            leaf period {
              type yang:timeticks;
              description
                "Time interval between reports.";
            }
            leaf start-time {
              type yang:date-and-time;
              description
                "Timestamp from which reports are started.";
            }
          }
          case on-change {
            description
              "On-change trigger and not periodic.";
            leaf all-data-on-start {
              type boolean;
              description
                "Full update needed on start or not.";
            }
            leaf-list excluded-change {
              type change-type;
              description
                "Restrict which changes trigger an update.";
            }
          }
        }
      }
      output {
```

```
        list response-list {
          key "response-index";
          description
            "Path discovery response list.";
          leaf response-index {
            type uint32;
            description
              "Response index.";
          }
          leaf status-code {
            type int32;
            description
              "Error code is ";
          }
          leaf status-sub-code {
            type uint8;
            description
              "Sub code is ";
          }
          leaf pd-persistent-id {
            type string;
            description
              "Id to act as a cookie.";
          }
        }
      }
    }
    rpc pd-persistent-modify {
      description
        "Modifies persistent path discovery.";
      input {
        leaf pd-persistent-id {
          type string;
          description
            "Cookie Id to be used for modifications.";
        }
        leaf data-export-method {
          type export-method;
          description
            "Type of export.";
        }
        choice pd-trigger {
          description
            "Defines necessary conditions for periodic or on-change
             trigger.";
          case periodic {
            description
              "Periodic reports.";
```

```
            leaf period {
              type yang:timeticks;
              description
                "Time interval between reports.";
            }
            leaf start-time {
              type yang:date-and-time;
              description
                "Timestamp from which reports are started.";
            }
          }
          case on-change {
            description
              "On-change trigger and not periodic.";
            leaf all-data-on-start {
              type boolean;
              description
                "Full update needed on start or not.";
            }
            leaf-list excluded-change {
              type change-type;
              description
                "Restrict which changes trigger an update.";
            }
          }
        }
      }
      output {
        list response-list {
          key "response-index";
          description
            "path discovery response list.";
          leaf response-index {
            type uint32;
            description
              "response index.";
          }
          leaf status-code {
            type int32;
            description
              "error code is ";
          }
          leaf status-sub-code {
            type uint8;
            description
              "sub code is ";
          }
          leaf pd-persistent-id {
```

```
            type string;
            description
              "Id to act as a cookie.";
          }
        }
      }
    }
    rpc pd-persistent-delete {
      description
        "Deletes persistent path discovery.";
      input {
        leaf pd-persistent-id {
          type string;
          description
            "Cookie Id to be used in deletion.";
        }
      }
      output {
        list response-list {
          key "response-index";
          description
            "path discovery response list.";
          leaf response-index {
            type uint32;
            description
              "response index.";
          }
          leaf status-code {
            type int32;
            description
              "error code is ";
          }
          leaf status-sub-code {
            type uint8;
            description
              "sub code is ";
          }
        }
      }
    }
```

<CODE ENDS>

Authors' Addresses

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

Email: dekumar@cisco.com

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

Email: wangzitao@huawei.com

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

Email: bill.wu@huawei.com

Reshad Rahman
CISCO Systems
2000 Innovation Drive
KANATA, ONTARIO K2K 3E8
CANADA

Email: rrahman@cisco.com

Srihari Raghavan
CISCO Systems
TRIL INFOPARK SEZ, Ramanujan IT City
NEVILLE BLOCK, 2nd floor, Old Mahabalipuram Road
CHENNAI, TAMIL NADU 600113
INDIA

Email: srihari@cisco.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 7, 2017

D. Kumar
Cisco
Q. Wu
M. Wang
Huawei

March 6, 2017

Generic YANG Data Model for Connection Oriented Operations, Administration, and Maintenance(OAM) protocols draft-ietf-lime-yang-oam-model-09

Abstract

This document presents a base YANG Data model for connection oriented OAM protocols. It provides a technology-independent abstraction of key OAM constructs for such protocols. The model presented here can be extended to include technology specific details. This guarantees uniformity in the management of OAM protocols and provides support for nested OAM workflows (i.e., performing OAM functions at different levels through a unified interface)

Status of This Memo

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

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

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

This Internet-Draft will expire on September 7, 2017.

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 in [RFC7276]. Over the years, many technologies have developed similar tools for fault and performance management.

[IEEE802.1ag] Connectivity Fault Management is a well-established OAM standard that is widely adopted for Ethernet networks. ITU-T [G.8013], MEF Service OAM, MPLS-TP [RFC6371], TRILL [RFC7455] all define OAM mechanisms based on the manageability frame work of CFM [IEEE802.1ag].

Given the wide adoption of the underlying OAM concepts defined in CFM [IEEE802.1ag], it is a reasonable choice to develop the unified management framework for connection oriented OAM based on those concepts. In this document, we take the CFM [IEEE802.1ag] model and extend it to a technology independent framework and define the corresponding YANG model accordingly. The YANG model presented in this document is the base model for connection oriented OAM protocols and supports generic continuity check, connectivity verification and path discovery (traceroute). The generic YANG model for connection oriented OAM is designed to be extensible to other connection oriented technologies. Technology dependent nodes and remote process call (RPC) 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 TRILL uses either MAC addresses, the VLAN tag or fine grain label, and/or IP addresses 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 goals: First it keeps each YANG model smaller and more 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 connection oriented 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 Continuity Check command or a Loopback 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 Loopback from device A to Device B fails. Between device A and B there are IEEE 802.1 bridges a, b and c. Let’s assume a,b and c are using CFM [IEEE802.1ag]. Upon detecting the Loopback failures, a user may decide to drill down to the lower level at different segments of the path and issue the corresponding fault verification (LBM) and fault isolation (LTM) tools, using the same API. This ability to drill down to a lower layer of the protocol stack at a specific segment within a path for fault localization and troubleshooting is referred to as "nested OAM workflow". It is a useful concept that leads to efficient network troubleshooting and maintenance workflows. The connection oriented 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]. 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 [RFC2119] significance.

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


   <flags> is one of:

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


   <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
        (choice)/:(case) Parentheses enclose choice and case nodes,
            and case nodes are also marked with a colon (":")
   <type> is the name of the type for leafs and leaf-lists
```

2.1. Terminology

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

ECMP - Equal Cost Multipath.

LBM - Loopback Message [IEEE802.1ag].

MP - Maintenance Point [IEEE802.1ag].

MEP - Maintenance End Point [RFC7174] (Maintenance association End Point [IEEE802.1ag], MEG End Points [RFC6371]).

MIP - Maintenance Intermediate Point [RFC7174] (Maintenance domain Intermediate Point [IEEE802.1ag], MEG Intermediate Point [RFC6371]).

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

MD - Maintenance Domain [IEEE802.1ag]

MEG - Maintenance Entity Group [RFC6371]

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] [IEEE802.1ag].

RPC - Remote Process Call.

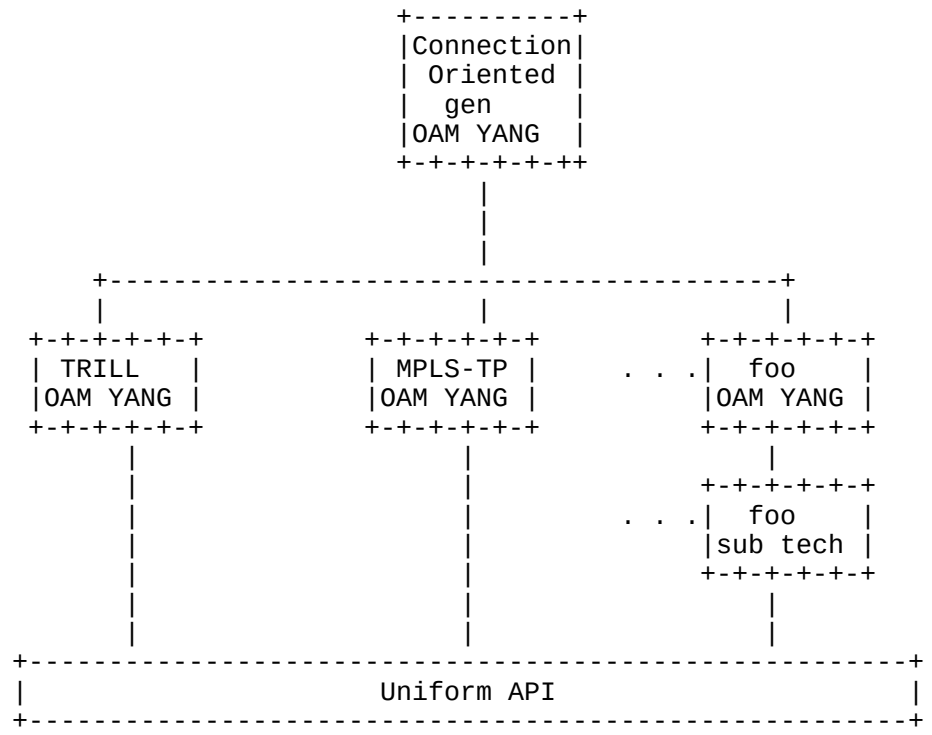
CC - Continuity Check [RFC7276]. Continuity Checks are used to verify that a destination is reachable and therefore also referred to as reachability verification.

CV - Connectivity Verification [RFC7276].Connectivity Verification are used to verify that a destination is connected. It 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 connection oriented OAM protocols. The YANG model defined here is generic in a sense 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 different OAM protocols with ease through a uniform API set. This also enables a nested OAM workflow. Figure 1 depicts the relationship of different OAM YANG models to the Generic YANG Model for connection oriented OAM. 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.



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 CFM [IEEE802.1ag] model and structure it such that it can be adapted to different connection oriented OAM protocols.

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 TRILL this can be per Fine-Grained Label or for VPLS this can be per VPLS instance [RFC6136].

Under each MA, there can be two or more MEPs (Maintenance 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(traceroute) and their equivalents as well as other OAM commands.

The OAM entities in the generic YANG model defined here will be either explicitly or implicitly configured using any of the OAM tools. The OAM tools used here are limited to OAM toolset specified in section 5.1 of [RFC7276]. 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, Addresses of MEPs 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 and future extensions developed in IETF 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 [IEEE802.1ag] 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-conn-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-conn-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-conn-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
               |  +--rw MEP-ID-format?       identityref
               |  +--rw (mep-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 4.5 for the complete data hierarchy and Section 5 for the YANG model.

```
module: ietf-conn-oam
   +--rw domains
         +--rw domain* [technology MD-name-string]
         +--rw technology        identityref
   .
   .
```

```
  rpcs:
    +---x continuity-check {continuity-check}?
    |  +---w input
    |  |  +---w technology?             identityref
    |  |  +---w MD-name-string  -> /domains/domain/MD-name-string
    |  |  +---w MA-name-string  -> /domains/domain/MAs/MA/MA-name-string
    |  |  +---w cos-id?                 uint8
    |  |  +---w (ttl)?
    |  |  |  +--:(ip-ttl)
    |  |  |  |  +---w ip-ttl?                uint8
    |  |  |  +--:(mpls-ttl)
    |  |  |     +---w mpls-ttl?              uint8
    |  |  +---w sub-type?               identityref
    |  |  +---w source-mep?      -> /domains/domain/MAs/MA/MEP/mep-name
    |  |  +---w destination-mep
    |  |  |  +---w (mep-address)?
    |  |  |  |  +--:(mac-address)
    |  |  |  |  |  +---w mac-address?     yang:mac-address
    |  |  |  |  +--:(ipv4-address)
    |  |  |  |  |  +---w ipv4-address?    inet:ipv4-address
    |  |  |  |  +--:(ipv6-address)
    |  |  |  |     +---w ipv6-address?    inet:ipv6-address
    |  |  |  +---w (MEP-ID)?
    |  |  |  |  +--:(MEP-ID-int)
    |  |  |  |     +---w MEP-ID-int?      int32
    |  |  |  +---w MEP-ID-format?   identityref
    |  |  +---w count?                  uint32
    |  |  +---w cc-transmit-interval?   Interval
    |  |  +---w packet-size?            uint32
    |  +--ro output
    |     +--ro (monitor-stats)?
    |        +--:(monitor-null)
    |           +--ro monitor-null?   empty
    +---x continuity-verification {connectivity-verification}?
    |  +---w input
    |  |  +---w MD-name-string  ->/domains/domain/MD-name-string
    |  |  +---w MA-name-string  ->/domains/domain/MAs/MA/MA-name-string
    |  |  +---w cos-id?            uint8
    |  |  +---w (ttl)?
    |  |  |  +--:(ip-ttl)
    |  |  |  |  +---w ip-ttl?            uint8
    |  |  |  +--:(mpls-ttl)
    |  |  |     +---w mpls-ttl?          uint8
    |  |  +---w sub-type?          identityref
    |  |  +---w source-mep?       -> /domains/domain/MAs/MA/MEP/mep-name
    |  |  +---w destination-mep
    |  |  |  +---w (mep-address)?
    |  |  |  |  +--:(mac-address)
```

```
|  |  |  |  |  +---w mac-address?     yang:mac-address
|  |  |  |  +--:(ipv4-address)
|  |  |  |  |  +---w ipv4-address?    inet:ipv4-address
|  |  |  |  +--:(ipv6-address)
|  |  |  |     +---w ipv6-address?    inet:ipv6-address
|  |  |  +---w (MEP-ID)?
|  |  |  |  +--:(MEP-ID-int)
|  |  |  |     +---w MEP-ID-int?      int32
|  |  |  +---w MEP-ID-format?   identityref
|  |  +---w count?             uint32
|  |  +---w interval?          Interval
|  |  +---w packet-size?       uint32
|  +--ro output
|     +--ro (monitor-stats)?
|        +--:(monitor-null)
|           +--ro monitor-null?   empty
+---x traceroute {traceroute}?
   +---w input
   |  +---w MD-name-string  -> /domains/domain/MD-name-string
   |  +---w MA-name-string  -> /domains/domain/MAs/MA/MA-name-string
   |  +---w cos-id?             uint8
   |  +---w (ttl)?
   |  |  +--:(ip-ttl)
   |  |  |  +---w ip-ttl?             uint8
   |  |  +--:(mpls-ttl)
   |  |     +---w mpls-ttl?           uint8
   |  +---w command-sub-type?   identityref
   |  +---w source-mep?      -> /domains/domain/MAs/MA/MEP/mep-name
   |  +---w destination-mep
   |  |  +---w (mep-address)?
   |  |  |  +--:(mac-address)
   |  |  |  |  +---w mac-address?     yang:mac-address
   |  |  |  +--:(ipv4-address)
   |  |  |  |  +---w ipv4-address?    inet:ipv4-address
   |  |  |  +--:(ipv6-address)
   |  |  |     +---w ipv6-address?    inet:ipv6-address
   |  |  +---w (MEP-ID)?
   |  |  |  +--:(MEP-ID-int)
   |  |  |     +---w MEP-ID-int?      int32
   |  |  +---w MEP-ID-format?   identityref
   |  +---w count?             uint32
   |  +---w interval?          Interval
   +--ro output
      +--ro response* [response-index]
         +--ro response-index     uint8
         +--ro (ttl)?
         |  +--:(ip-ttl)
         |  |  +--ro ip-ttl?           uint8
```

```
|  +--:(mpls-ttl)
|     +--ro mpls-ttl?          uint8
+--ro destination-mep
|  +--ro (mep-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 mip {mip}?
|  +--ro interface?      if:interface-ref
|  +--ro (mip-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 (monitor-stats)?
   +--:(monitor-null)
      +--ro monitor-null?      empty
```

Snippet of data hierarchy related to RPC call continuity-check

4.5. Notifications

Notification is sent on defect condition and defect clears with Maintenance Domain Name, MA Name, defect-type (The currently active defects), generating-mepid, and defect-message to indicate more details.

4.6. Monitor statistics

Grouping for monitoring statistics is to be used by Yang modules which Augment Yang to provide statistics due to pro-active OAM like CCM Messages. For example CCM Transmit, CCM Receive, CCM Errors, etc.

4.7. OAM data hierarchy

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

```
module: ietf-conn-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 cos-id?           uint8
                +--rw cc-enable?        boolean
                +--rw MEP* [mep-name]
                |  +--rw mep-name         MEP-name
                |  +--rw (MEP-ID)?
                |  |  +--:(MEP-ID-int)
                |  |     +--rw MEP-ID-int?      int32
                |  +--rw MEP-ID-format?   identityref
                |  +--rw (mep-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 cos-id?          uint8
                |  +--rw cc-enable?       boolean
                |  +--rw session* [session-cookie]
                |     +--rw session-cookie             uint32
                |     +--rw destination-mep
                |     |  +--rw (MEP-ID)?
                |     |  |  +--:(MEP-ID-int)
                |     |  |     +--rw MEP-ID-int?      int32
                |     |  +--rw MEP-ID-format?   identityref
                |     +--rw destination-mep-address
                |     |  +--rw (mep-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 cos-id?                    uint8
                +--rw MIP* [interface] {mip}?
                   +--rw interface       if:interface-ref
                   +--rw (mip-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 level?          MD-level

  rpcs:
    +---x continuity-check {continuity-check}?
    |  +---w input
    |  |  +---w technology?             identityref
    |  |  +---w MD-name-string  -> /domains/domain/MD-name-string
    |  |  +---w MA-name-string  -> /domains/domain/MAs/MA/MA-name-string
    |  |  +---w cos-id?                 uint8
    |  |  +---w (ttl)?
    |  |  |  +--:(ip-ttl)
    |  |  |  |  +---w ip-ttl?                uint8
    |  |  |  +--:(mpls-ttl)
    |  |  |     +---w mpls-ttl?              uint8
    |  |  +---w sub-type?               identityref
    |  |  +---w source-mep?      -> /domains/domain/MAs/MA/MEP/mep-name
    |  |  +---w destination-mep
    |  |  |  +---w (mep-address)?
    |  |  |  |  +--:(mac-address)
    |  |  |  |  |  +---w mac-address?     yang:mac-address
    |  |  |  |  +--:(ipv4-address)
    |  |  |  |  |  +---w ipv4-address?    inet:ipv4-address
    |  |  |  |  +--:(ipv6-address)
    |  |  |  |     +---w ipv6-address?    inet:ipv6-address
    |  |  |  +---w (MEP-ID)?
    |  |  |  |  +--:(MEP-ID-int)
    |  |  |  |     +---w MEP-ID-int?      int32
    |  |  |  +---w MEP-ID-format?   identityref
    |  |  +---w count?                  uint32
    |  |  +---w cc-transmit-interval?   Interval
    |  |  +---w packet-size?            uint32
    |  +--ro output
    |     +--ro (monitor-stats)?
    |        +--:(monitor-null)
```

```
|           +--ro monitor-null?   empty
+---x continuity-verification {connectivity-verification}?
|  +---w input
|  |  +---w MD-name-string  -> /domains/domain/MD-name-string
|  |  +---w MA-name-string  -> /domains/domain/MAs/MA/MA-name-string
|  |  +---w cos-id?            uint8
|  |  +---w (ttl)?
|  |  |  +--:(ip-ttl)
|  |  |  |  +---w ip-ttl?            uint8
|  |  |  +--:(mpls-ttl)
|  |  |     +---w mpls-ttl?          uint8
|  |  +---w sub-type?          identityref
|  |  +---w source-mep?     -> /domains/domain/MAs/MA/MEP/mep-name
|  |  +---w destination-mep
|  |  |  +---w (mep-address)?
|  |  |  |  +--:(mac-address)
|  |  |  |  |  +---w mac-address?     yang:mac-address
|  |  |  |  +--:(ipv4-address)
|  |  |  |  |  +---w ipv4-address?    inet:ipv4-address
|  |  |  |  +--:(ipv6-address)
|  |  |  |     +---w ipv6-address?    inet:ipv6-address
|  |  |  +---w (MEP-ID)?
|  |  |  |  +--:(MEP-ID-int)
|  |  |  |     +---w MEP-ID-int?      int32
|  |  |  +---w MEP-ID-format?   identityref
|  |  +---w count?             uint32
|  |  +---w interval?          Interval
|  |  +---w packet-size?       uint32
|  +--ro output
|     +--ro (monitor-stats)?
|        +--:(monitor-null)
|           +--ro monitor-null?   empty
+---x traceroute {traceroute}?
   +---w input
   |  +---w MD-name-string  -> /domains/domain/MD-name-string
   |  +---w MA-name-string  -> /domains/domain/MAs/MA/MA-name-string
   |  +---w cos-id?             uint8
   |  +---w (ttl)?
   |  |  +--:(ip-ttl)
   |  |  |  +---w ip-ttl?            uint8
   |  |  +--:(mpls-ttl)
   |  |     +---w mpls-ttl?          uint8
   |  +---w command-sub-type?   identityref
   |  +---w source-mep?      -> /domains/domain/MAs/MA/MEP/mep-name
   |  +---w destination-mep
   |  |  +---w (mep-address)?
   |  |  |  +--:(mac-address)
   |  |  |  |  +---w mac-address?     yang:mac-address
```

```
      |  |  |  +--:(ipv4-address)
      |  |  |  |  +---w ipv4-address?    inet:ipv4-address
      |  |  |  +--:(ipv6-address)
      |  |  |     +---w ipv6-address?    inet:ipv6-address
      |  |  +---w (MEP-ID)?
      |  |  |  +--:(MEP-ID-int)
      |  |  |     +---w MEP-ID-int?      int32
      |  |  +---w MEP-ID-format?   identityref
      |  +---w count?              uint32
      |  +---w interval?           Interval
      +--ro output
         +--ro response* [response-index]
            +--ro response-index     uint8
            +--ro (ttl)?
            |  +--:(ip-ttl)
            |  |  +--ro ip-ttl?           uint8
            |  +--:(mpls-ttl)
            |     +--ro mpls-ttl?         uint8
            +--ro destination-mep
            |  +--ro (mep-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 mip {mip}?
            |  +--ro interface?      if:interface-ref
            |  +--ro (mip-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 (monitor-stats)?
               +--:(monitor-null)
                  +--ro monitor-null?      empty

  notifications:
    +---n defect-condition-notification
    |  +--ro technology?         identityref
    |  +--ro MD-name-string  -> /domains/domain/MD-name-string
    |  +--ro MA-name-string  -> /domains/domain/MAs/MA/MA-name-string
```

```
    |  +--ro mep-name?       -> /domains/domain/MAs/MA/MEP/mep-name
    |  +--ro defect-type?        identityref
    |  +--ro generating-mepid
    |  |  +--ro (MEP-ID)?
    |  |  |  +--:(MEP-ID-int)
    |  |  |     +--ro MEP-ID-int?      int32
    |  |  +--ro MEP-ID-format?   identityref
    |  +--ro (defect)?
    |     +--:(defect-null)
    |     |  +--ro defect-null?        empty
    |     +--:(defect-code)
    |        +--ro defect-code?        int32
    +---n defect-cleared-notification
       +--ro technology?         identityref
       +--ro MD-name-string  -> /domains/domain/MD-name-string
       +--ro MA-name-string  -> /domains/domain/MAs/MA/MA-name-string
       +--ro mep-name?       -> /domains/domain/MAs/MA/MEP/mep-name
       +--ro defect-type?        identityref
       +--ro generating-mepid
       |  +--ro (MEP-ID)?
       |  |  +--:(MEP-ID-int)
       |  |     +--ro MEP-ID-int?      int32
       |  +--ro MEP-ID-format?   identityref
       +--ro (defect)?
          +--:(defect-null)
          |  +--ro defect-null?        empty
          +--:(defect-code)
             +--ro defect-code?        int32
```

data hierarchy of OAM

5. OAM YANG Module

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

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

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

```
  organization "IETF LIME Working Group";
  contact
    "WG Web:    http://tools.ietf.org/wg/lime
     WG List:   mailto:lime@ietf.org
     WG Chair:  Carlos Pignataro cpignata@cisco.com
     WG Chair:  Ron Bonica rbonica@juniper.net
     Editor:    Deepak Kumar dekumar@cisco.com
     Editor:    Qin Wu bill.wu@huawei.com
     Editor:    Zitao Wang wangzitao@huawei.com";
  description
    "This YANG module defines the generic configuration,
  statistics and rpc for connection oriented 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 2017-02-10 {
    description
      "Initial revision. - 07 version";

    reference "draft-ietf-lime-yang-oam-model";
  }

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

  feature traceroute{
    description
```

```
      "This feature indicates that the server supports
    executing traceroute OAM command and
    returning a response. Servers that do not advertise
    this feature will not support executing
    traceroute command or rpc model for
    traceroute command.";
   }
  feature mip {
   description
   "This feature indicates that the MIP (Maintenance Intermediate Point)
   need to
   be explicit configured";
  }
  /* Identities */

  identity technology-types {
    description
      "This is the base identy of technology types which are
    TRILL,MPLS-TP,vpls etc";
  }

  identity command-sub-type {
    description
      "Defines different rpc command subtypes,
    e.g rfc6905 trill OAM, this is optional for most cases";
  }

  identity on-demand {
   base command-sub-type;
   description
   "On demand activation - indicates that the tool is activated
    manually to detect a specific anomaly.";
  }

  identity proactive {
   base command-sub-type;
   description
   "Proactive activation - indicates that the tool is activated on a
    continual basis, where messages are sent periodically, and errors
    are detected when a certain number of expected messages are not
    received.";
  }

  identity name-format {

    description
      "This defines the name format, IEEE 8021ag 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 MEP-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 rdi {
    base defect-types;
    description
      "Indicates the aggregate health of the remote MEPs. ";
  }

  identity remote-mep-defect{
    base defect-types;
    description
      "Indicates that one or more of the remote MEPs is
       reporting a failure ";
  }

  identity loss-of-continuity{
    base defect-types;
    description
    "If no proactive CC OAM packets from the source
     MEP (and in the case of CV, this includes the
     requirement to have the expected globally unique
     Source MEP identifier) are received within the interval. ";
   }
```

```
  identity invalid-oam-defect{
    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-defect{
    base defect-types;
    description
    "Indicates that one or more cross-connect defect
    (for example, a service ID does not match the VLAN.)
     messages has been received and that 3.5 times that OAM message
     transmission interval has not yet expired.";
  }

  /* typedefs */

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

  typedef Interval{
    type decimal64{
    fraction-digits 2;
   }
   units "milliseconds";
    description
    "Interval between packets in milliseconds.
    0 means no packets are sent.";
  }

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

  typedef MA-name-string {
    type string;
    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 maintenance-domain-reference {
  description
     "This grouping uniquely identifies a maintenance domain.";

  leaf maintenance-domain {
     type leafref {
       path "/goam:domains/goam:domain/goam:MD-name-string";
     }
  description
    "A reference to a specific Maintenance Domain.";
}
}

grouping maintenance-association-reference {
  description
    "This grouping uniquely identifies a
        maintenance association. It consists
     of a maintence-domain-reference and
         a maintenance-association leafref";
  uses maintenance-domain-reference;
  leaf maintenance-association {
    type leafref {
      path "/goam:domains/goam:domain"
          +"[goam:MD-name-string = current()/"
          +"../maintenance-domain]/goam:MAs"
          +"/goam:MA/goam:MA-name-string";
    }
  description
      "A reference to a specific Maintenance Association.";
  }
}
```

```
grouping maintenance-association-end-point-reference {
  description
  "This grouping uniquely identifies
  a maintenance association. It consists
   of a maintence-association-reference and
   a maintenance-association-end-point leafref";
  uses maintenance-association-reference;
  leaf maintenance-association-end-point {
    type leafref {
      path "/goam:domains/goam:domain"
          +"[goam:MD-name-string = current()/"
          +"../maintenance-domain]/goam:MAs"
          +"/goam:MA[goam:MA-name-string = "
          +"current()/../maintenance-association]"
          +"/goam:MEP/goam:mep-name";
     }
    description
      "A reference to a specific Maintenance
          association End Point.";
  }
 }

  grouping time-to-live {
    choice ttl{
      case ip-ttl{
        leaf ip-ttl{
        type uint8;
        default "255";
        description
          "Time to live";
        }
      }
     case mpls-ttl{
       leaf mpls-ttl{

       type uint8;
       description
         "Time to live. When an IP packet is imposed with a label,
        the IP TTL value is first decremented then copied into
        the MPLS TTL. As each LSR the MPLS frame's TTL is
        decremented.This behavior can be modified with no
        mpls ip ttl. When a MPLS label is popped, the MPLS
        TTL value is decremented then copied in the IP TTL
        field. If the MPLS TTL value is great than IP TTL,
        that values is not copied over. This is to prevent
        a possible condition of forwarding loop and TTL
        never reaching 0. When two MPLS labels are swapped,
        decrement by 1 and copy over the result into the new label.
```

```
        When a new MPLS labels is pushed, decrement by 1 and copy
        over the result into the new label. When a new MPLS labels
        is popped, decrement by 1 and copy over the result into
        the label below.[RFC3443]";

       }
     }
     description
       "Time to Live.";
   }
   description
     "Time to Live grouping.";
 }
 grouping defect-message {
   choice defect {
     case defect-null {
       description

         "This is a placeholder when no defect status is needed";
       leaf defect-null {
          type empty;
          description
            "there is no defect define, it will be defined in
             technology specific model.";
       }
     }
     case defect-code {
       description
         "This is a placeholder to display defect code.";
       leaf defect-code {
          type int32;
          description
            "Defect code is integer value specific to technology.";
       }
     }

     description
       "Defect Message choices.";
   }

   description
     "Defect Message.";
 }

 grouping mep-address {
   choice mep-address {
     case mac-address {
       leaf mac-address {
```

```
          type yang:mac-address;
          description
          "MAC Address";
        }
      description
      "MAC Address based MEP Addressing.";
      }
      case ipv4-address {
        leaf ipv4-address {
          type inet:ipv4-address;
          description
          "IPv4 Address";
        }
      description
      "IP Address based MEP Addressing.";
      }
      case ipv6-address {
        leaf ipv6-address {
          type inet:ipv6-address;
          description
          "IPv6 Address";
        }
        description
        "IPv6 Address based MEP Addressing.";
      }
      description
       "MEP Addressing.";
    }
    description
     "MEP Address";
  }
  grouping mip-address {
    choice mip-address {
      case mac-address {
        leaf mac-address {
          type yang:mac-address;
          description
          "MAC Address";
        }
      description
      "MAC Address based MIP Addressing.";
      }
      case ipv4-address {
        leaf ipv4-address {
          type inet:ipv4-address;
          description
          "IPv4 Address";
        }
```

```
      description
      "IP Address based MIP Addressing.";
      }
      case ipv6-address {
        leaf ipv6-address {
          type inet:ipv6-address;
          description
          "IPv6 Address";
        }
        description
        "IPv6 Address based MIP Addressing.";
      }
      description
       "MIP Addressing.";
    }
    description
     "MIP 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";
        }
      }
      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 mep-address;
  }
```

```
 grouping monitor-stats {
   description
    "grouping for monitoring statistics, this will be augmented
    by others who use this component";
    choice monitor-stats {

     default "monitor-null";
      case monitor-null {
       description
        "This is a place holder when
         no monitoring statistics is needed";
         leaf monitor-null {
          type empty;
          description
          "There is no monitoring statistics to be defined";
              }
            }
          description
          "Define the monitor stats";
          }
        }


  grouping connectivity-context {
    description
      "Grouping defining the connectivity context for an MA; for
       example, a VRF for VPLS, or an LSP for MPLS-TP.  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 cos {
    description

      "Priority used in transmitted packets; for example, in the
       EXP field in MPLS-TP.";
```

```
    leaf cos-id {
      type uint8;
      description
        "Class of service";
    }
  }
  grouping MIP-grouping {
   uses mip-address;
   leaf level {
    type MD-level;
        description
        "Configure a level of maintenance intermediate point (MIP)
        for the interface. The MIP level range is 0 to 7.";
   }
   description
   "Grouping for MIP configuration";
  }

  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";
        list MA {
          key "MA-name-string";
          ordered-by system;
          uses ma-identifier;
          uses MA-name;
          uses connectivity-context;
          uses cos {
            description
```

```
              "Default class of service for this MA,
                          which may be overridden
               for particular MEPs,
                           sessions or operations.";
          }
             leaf cc-enable{
              type boolean;
              description
              "Indicate whether the CC enable.";
             }
          list MEP {
            key "mep-name";

            ordered-by system;
            description
            "Contain list of MEPS";
            uses MEP;
            uses cos;
                    leaf cc-enable{
                     type boolean;
                     description
                     "Indicate whether the CC enable.";
                    }
            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.";
              }
              container destination-mep {
                uses MEP-ID;
                description
                "Destination MEP";
              }
              container destination-mep-address {
                uses mep-address;
```

```
                description
                "Destination MEP Address";
              }
              uses cos;
            }
          }
                  list MIP {
                   if-feature mip;
                   key "interface";
                   leaf interface {
                    type if:interface-ref;
                        description
                        "Interface";
                   }
                   uses MIP-grouping;
                   description
                   "List for MIP";
                  }

           description
           "Maintenance Association list";
          }
                }
      }

  }

  notification defect-condition-notification {
    description
      "When defect condition is met this notification is sent";
    leaf technology {
      type identityref {
        base technology-types;
      }
      description
        "The technology";
    }
    leaf MD-name-string {
      type leafref{
           path "/domains/domain/MD-name-string";
          }
      mandatory true;
      description
      "Indicate which MD is seeing the defect";
    }
    leaf MA-name-string{
          type leafref{
           path "/domains/domain/MAs/MA/MA-name-string";
```

```
          }
      mandatory true;
      description
      "Indicate which MA is seeing the defect";
        }
    leaf mep-name {
      type leafref{
           path "/domains/domain/MAs/MA/MEP/mep-name";
          }
      description
        "Indicate which MEP is seeing the defect";
    }
    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 defect (if known) if
         unknown make it 0.";
    }
    uses defect-message {
      description
         "Defect message to indicate more details.";
    }
  }

  notification defect-cleared-notification {
    description
      "When defect cleared is met this notification is sent";
    leaf technology {
      type identityref {
        base technology-types;
      }
      description
        "The technology";
    }
    leaf MD-name-string {
      type leafref{
           path "/domains/domain/MD-name-string";
          }
      mandatory true;
```

```
      description
      "Indicate which MD is seeing the defect";
    }
    leaf MA-name-string{
          type leafref{
           path "/domains/domain/MAs/MA/MA-name-string";
          }
      mandatory true;
      description
      "Indicate which MA is seeing the defect";
        }
    leaf mep-name {
      type leafref{
           path "/domains/domain/MAs/MA/MEP/mep-name";
          }
      description
        "Indicate which MEP is seeing the defect";
    }

    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 defect (if known) if
         unknown make it 0.";
    }
    uses defect-message {
      description
         "Defect message to indicate more details.";
    }
  }

  rpc continuity-check {
    if-feature "continuity-check";
    description
      "Generates continuity-check as per RFC7276 Table 4.";
    input {
    leaf technology {
      type identityref {
        base technology-types;
```

```
      }
      description
        "The technology";
    }
    leaf MD-name-string {
      type leafref{
           path "/domains/domain/MD-name-string";
          }
      mandatory true;
      description
      "Indicate which MD is seeing the defect";
    }
    leaf MA-name-string{
          type leafref{
           path "/domains/domain/MAs/MA/MA-name-string";
          }
      mandatory true;
      description
      "Indicate which MA is seeing the defect";
        }
      uses cos;
      uses time-to-live;
      leaf sub-type {

        type identityref {
          base command-sub-type;
        }
        description
          "Defines different command types";
      }
      leaf source-mep {
       type leafref{
           path "/domains/domain/MAs/MA/MEP/mep-name";
           }
       description
       "Source MEP";
      }
      container destination-mep {
        uses mep-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 continuity-check message to send";
      }
      leaf cc-transmit-interval {
        type Interval;
        description

          "Interval between echo requests";
      }
      leaf packet-size {
        type uint32 {
          range "0..10000";
        }
        default "64";
        description
          "Size of continuity-check 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 {
     leaf MD-name-string {
      type leafref{
           path "/domains/domain/MD-name-string";
          }
      mandatory true;
      description
      "Indicate which MD is seeing the defect";
     }
     leaf MA-name-string{
          type leafref{
           path "/domains/domain/MAs/MA/MA-name-string";
          }
      mandatory true;
      description
```

```
      "Indicate which MA is seeing the defect";
         }
      uses cos;
      uses time-to-live;
      leaf sub-type {

        type identityref {
          base command-sub-type;
        }
        description
          "Defines different command types";
      }
      leaf source-mep {
       type leafref{
           path "/domains/domain/MAs/MA/MEP/mep-name";
           }
       description
       "Source MEP";
      }
      container destination-mep {
        uses mep-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 continuity-verification 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 continuity-verification packets, in octets";
      }
    }
```

```
    output {
      uses monitor-stats {
        description
          "Stats of continuity check.";
      }
    }
  }
  rpc traceroute {
    if-feature traceroute;
    description
      "Generates Traceroute or Path Trace and return response.
       Referencing RFC7276 for common Toolset name, for
       MPLS-TP OAM it's Route Tracing, 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 value";
    input {
     leaf MD-name-string {
      type leafref{
           path "/domains/domain/MD-name-string";
          }
      mandatory true;
      description
      "Indicate which MD is seeing the defect";
     }
     leaf MA-name-string{
          type leafref{

           path "/domains/domain/MAs/MA/MA-name-string";
          }
      mandatory true;
      description
      "Indicate which MA is seeing the defect";
         }
      uses cos;
      uses time-to-live;
      leaf command-sub-type {
        type identityref {

          base command-sub-type;
        }
        description
          "Defines different command types";
      }
      leaf source-mep {
       type leafref{
           path "/domains/domain/MAs/MA/MEP/mep-name";
           }
```

```
       description
       "Source MEP";
      }
      container destination-mep {
        uses mep-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
             , the TTL can be used as the response
             index.";
        }
        uses time-to-live;
        container destination-mep {
          description "MEP from which the response has been received";
          uses mep-address;
          uses MEP-ID {
            description
              "Only applicable if the destination is a MEP";
          }
        }
                container mip {
```

```
                if-feature mip;
                leaf interface {
                 type if:interface-ref;
                 description
                 "MIP interface";
                }
         uses mip-address;
                description
                "MIP responding with traceroute";
               }
        uses monitor-stats {
          description
            "Stats of traceroute.";
        }
      description
        "List of response.";
      }
    }
  }
}
```

<CODE ENDS>

6. Base Mode

The Base Mode ('default mode' described in section 4) 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 MEP-ID associated with this MEP is zero (0). The choice of MEP-ID zero is explained below.

MEP-ID is 2 octet field by default. It is never used on the wire except when using CCM. 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.

CFM [IEEE802.1ag] defines MEP ID as an unsigned integer in the range 1 to 8191. In this document we propose 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 Association

The ID of the Maintenance Association (MA-ID) [IEEE802.1ag] 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 [IEEE802.1ag]) and Short MA name set to 65532 (0xFFFC).

7. Connection-oriented OAM YANG model applicability

"ietf-conn-oam" model defined in this document provides technology-independent abstraction of key OAM constructs for connection oriented protocols. This model can be further extended to include technology specific details, e.g., adding new data nodes with technology specific functions and parameters into proper anchor points of the base model, so as to develop a technology-specific connection-oriented OAM model.

This section demonstrates the usability of the connection-oriented YANG OAM data model to various connection-oriented OAM technologies, e.g., TRILL and MPLS-TP. Note that, in this section, we only present several snippets of technology-specific model extensions for illustrative purposes. The complete model extensions should be worked on in respective protocol working groups.

7.1. Generic YANG Model extension for TRILL OAM

The TRILL YANG module is augmenting connection oriented OAM module for both configuration and RPC commands.

The TRILL YANG module requires the base TRILL module ([I-D.ietf-trill-yang]) to be supported as there is a strong relationship between those modules.

The configuration extensions for connection oriented OAM include MD configuration extension, Technology type extension, MA configuration extension, Connectivity-Context Extension, MEP Configuration Extension, ECMP extension. In the RPC extension, the continuity-check and path-discovery RPC are extended with TRILL specific.

7.1.1. MD Configuration Extension

MD level configuration parameters are management information which can be inherited in the TRILL OAM model and set by connection oriented base model as default values. For example domain name can be set to area-ID in the TRILL OAM case. In addition, at the Maintenance Domain level, domain data node at root level can be augmented with technology type.

Note that MD level configuration parameters provides context information for management system to correlate faults, defects, network failures with location information, which helps quickly identify root causes of network failures.

7.1.1.1. Technology Type Extension

No TRILL technology type has been defined in the connection oriented base model. Therefore a technology type extension is required in the TRILL OAM model. The technology type "trill" is defined as an identity that augments the base "technology-types" defined in the connection oriented base model:

```
identity trill{
 base goam:technology-types;
 description
  "trill type";
}
```

7.1.2. MA Configuration Extension

MA level configuration parameters are management information which can be inherited in the TRILL OAM model and set by connection oriented base model as default values. In addition, at the

Maintenance Association(MA) level, MA data node at the second level can be augmented with connectivity-context extension.

Note that MA level configuration parameters provides context information for management system to correlate faults, defects, network failures with location information, which helps quickly identify root causes of network failures.

7.1.2.1. Connectivity-Context Extension

In TRILL OAM, one example of connectivity-context is either a 12 bit VLAN ID or a 24 bit Fine Grain Label. The connection oriented base model defines a placeholder for context-id. This allows other technologies to easily augment that to include technology specific extensions. The snippet below depicts an example of augmenting connectivity-context to include either VLAN ID or Fine Grain Label.

```
augment /goam:domains/goam:domain/goam:MAs
/goam:MA /goam:connectivity-context:
      +--:(connectivity-context-vlan)
      |  +--rw connectivity-context-vlan?   vlan
      +--:(connectivity-context-fgl)
         +--rw connectivity-context-fgl?    fgl
```

7.1.3. MEP Configuration Extension

The MEP configuration definition in the connection oriented base model already supports configuring the interface of MEP with either MAC address or IP address. In addition, the MEP address can be represented using a 2 octet RBridge Nickname in TRILL OAM . Hence, the TRILL OAM model augments the MEP configuration in base model to add a nickname case into the MEP address choice node as follows:

```
augment /goam:domains/goam:domain/goam:MAs
   /goam:MA/ goam:MEP/goam:mep-address:
         +--:( mep-address-trill)
         |  +--rw mep-address-trill?  tril-rb-nickname
```

In addition, at the Maintenance Association Endpoint(MEP) level, MEP data node at the third level can be augmented with ECMP extension.

7.1.3.1. ECMP Extension

Since TRILL supports ECMP path selection, flow-entropy in TRILL is defined as a 96 octet field in the LIME model extension for TRILL OAM. The snippet below illustrates its extension.

```
 augment /goam:domains/goam:domain/goam:MAs/goam:MA/goam:MEP:
           +--rw flow-entropy-trill?   flow-entropy-trill
   augment /goam:domains/goam:domain/goam:MAs/goam:MA/goam:MEP
   /goam:session:
           +--rw flow-entropy-trill?   flow-entropy-trill
```

7.1.4. RPC extension

In the TRILL OAM YANG model, the continuity-check and path-discovery RPC commands are extended with TRILL specific requirements. The snippet below depicts an example of illustrates the TRILL OAM RPC extension.

```
   augment /goam:continuity-check/goam:input:
         +--ro (out-of-band)?
         |  +--:(ipv4-address)
         |  |  +--ro ipv4-address?      inet:ipv4-address
         |  +--:(ipv6-address)
         |  |  +--ro ipv6-address?      inet:ipv6-address
         |  +--:(trill-nickname)
         |     +--ro trill-nickname?    tril-rb-nickname
         +--ro diagnostic-vlan?   boolean
   augment /goam:continuity-check/goam:input:
           +--ro flow-entropy-trill?   flow-entropy-trill
   augment /goam:continuity-check/goam:output:
         +--ro upstream-rbridge?   tril-rb-nickname
         +--ro next-hop-rbridge*   tril-rb-nickname
   augment /goam:path-discovery/goam:input:
         +--ro (out-of-band)?
         |  +--:(ipv4-address)
         |  |  +--ro ipv4-address?      inet:ipv4-address
         |  +--:(ipv6-address)
         |  |  +--ro ipv6-address?      inet:ipv6-address
         |  +--:(trill-nickname)
         |     +--ro trill-nickname?    tril-rb-nickname
         +--ro diagnostic-vlan?   boolean
   augment /goam:path-discovery/goam:input:
           +--ro flow-entropy-trill?   flow-entropy-trill
   augment /goam:path-discovery/goam:output/goam:response:
         +--ro upstream-rbridge?   tril-rb-nickname
         +--ro next-hop-rbridge*   tril-rb-nickname
```

7.2. Generic YANG Model extension for MPLS-TP OAM

The MPLS-TP OAM YANG module can augment connection oriented OAM Module with some technology-specific details. And the [mpls-tp-oam-yang] presents the YANG Data model for MPLS-TP OAM.

The configuration extensions for connection oriented OAM include MD configuration extension, Technology type extension, Sub Technology Type Extension ,MA configuration extension, MEP Configuration Extension.

7.2.1. MD Configuration Extension

MD level configuration parameters are management information which can be inherited in the MPLS-TP OAM model and set by LIME base model as default values. For example domain name can be set to area-ID or the provider's Autonomous System Number(ASN) [RFC6370] in the MPLS-TP OAM case. In addition, at the Maintenance Domain level, domain data node at root level can be augmented with technology type and sub-technology type.

Note that MD level configuration parameters provides context information for management system to correlate faults, defects, network failures with location information, which helps quickly identify root causes of network failures

7.2.1.1. Technology Type Extension

No MPLS-TP technology type has been defined in the connection oriented base model, hence it is required in the MPLS OAM model. The technology type "mpls-tp" is defined as an identity that augments the base "technology-types" defined in the connection oriented base model:

```
identity mpls-tp{
      base goam:technology-types;
      description
       "mpls-tp type";
     }
```

7.2.1.2. Sub Technology Type Extension

In MPLS-TP, since different encapsulation types such as IP/UDP Encapsulation, PW-ACH encapsulation can be employed, the "technology-sub-type" data node is defined and added into the MPLS OAM model to further identify the encapsulation types within the MPLS-TP OAM model. Based on it, we also define a technology sub-type for IP/UDP encapsulation and PW-ACH encapsulation. Other Encapsulation types can be defined in the same way. The snippet below depicts an example of several encapsulation types.

```
   identity technology-sub-type {
         description
         "certain implementations can have different
          encapsulation types such as ip/udp, pw-ach and so on.
          Instead of defining separate models for each
          encapsulation, we define a technology sub-type to
       further identify different encapsulations.
       Technology sub-type is associated at the MA level"; }

             identity technology-sub-type-udp {
               base technology-sub-type;
               description
                 "technology sub-type is IP/UDP encapsulation";
             }

             identity technology-sub-type-ach {
               base technology-sub-type;
               description
                 "technology sub-type is PW-ACH encapsulation";
             }
             }

        augment "/goam:domains/goam:domain/goam:MAs/goam:MA" {
               leaf technology-sub-type {
                 type identityref {
                   base technology-sub-type;
                 }
               }
             }
```

7.2.2. MA Configuration Extension

MA level configuration parameters are management information which can be inherited in the MPLS-TP OAM model and set by Connection Oriented base model as default values. One example of MA Name could be MEG LSP ID or MEG Section ID or MEG PW ID[RFC6370].

Note that MA level configuration parameters provides context information for management system to correlate faults, defects, network failures with location information, which helps quickly identify root causes of network failures.

7.2.3. MEP Configuration Extension

In MPLS-TP, MEP-ID is either a variable length label value in case of G-ACH encapsulation or a 2 octet unsigned integer value in case of IP/UDP encapsulation. One example of MEP-ID is MPLS-TP LSP_MEP_ID [RFC6370]. In the connection-oriented base model, MEP-ID is defined

as a choice/case node which can supports an int32 value, and the same definition can be used for MPLS-TP with no further modification. In addition, at the Maintenance Association Endpoint(MEP) level, MEP data node at the third level can be augmented with Session extension and interface extension.

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,Mishael Wexler, Adi Molkho 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>.

[RFC6370] Bocci, M., Swallow, G., and E. Gray, "MPLS Transport Profile (MPLS-TP) Identifiers", RFC 6370, DOI 10.17487/RFC6370, September 2011, <http://www.rfc-editor.org/info/rfc6370>.

[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>.

11.2. Informative References

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

[IEEE802.1ag] "Connectivity Fault Management", IEEE Std 802.1ag-2011, August 2011.

[mpls-tp-oam-yang] Zhang, L., Zheng, L., Aldrin, S., and G. Mirsky, "YANG Data Model for MPLS-TP Operations, Administration, and Maintenance", draft-zhang-mpls-tp-yang-oam (work in progress), 2016.

[RFC6136] Sajassi, A., Ed. and D. Mohan, Ed., "Layer 2 Virtual Private Network (L2VPN) Operations, Administration, and Maintenance (OAM) Requirements and Framework", RFC 6136, DOI 10.17487/RFC6136, March 2011, <http://www.rfc-editor.org/info/rfc6136>.

[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>.

[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>.

Appendix A. Contributors' Addresses

Tissa Senevirathne
Consultant

Email: tsenevir@gmail.com

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

Email: nfinn@cisco.com

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

Email: ssalam@cisco.com

Authors' Addresses

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

Email: dekumar@cisco.com

Qin Wu
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

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 11, 2017

D. Kumar
Cisco
Q. Wu
M. Wang
Huawei

April 9, 2017

Generic YANG Data Model for Connection Oriented Operations, Administration, and Maintenance(OAM) protocols draft-ietf-lime-yang-oam-model-10

Abstract

This document presents a base YANG Data model for connection oriented OAM protocols. It provides a technology-independent abstraction of key OAM constructs for such protocols. The model presented here can be extended to include technology specific details. This guarantees uniformity in the management of OAM protocols and provides support for nested OAM workflows (i.e., performing OAM functions at different levels through a unified interface)

Status of This Memo

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

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

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

This Internet-Draft will expire on October 11, 2017.

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 in [RFC7276]. Over the years, many technologies have developed similar tools for fault and performance management.

[IEEE802.1ag] Connectivity Fault Management is a well-established OAM standard that is widely adopted for Ethernet networks. ITU-T [G.8013], MEF Service OAM, MPLS-TP [RFC6371], TRILL [RFC7455] all define OAM mechanisms based on the manageability frame work of CFM [IEEE802.1ag].

Given the wide adoption of the underlying OAM concepts defined in CFM [IEEE802.1ag], it is a reasonable choice to develop the unified management framework for connection oriented OAM based on those concepts. In this document, we take the CFM [IEEE802.1ag] model and extend it to a technology independent framework and define the corresponding YANG model accordingly. The YANG model presented in this document is the base model for connection oriented OAM protocols and supports generic continuity check, connectivity verification and path discovery (traceroute). The generic YANG model for connection oriented OAM is designed to be extensible to other connection oriented technologies. Technology dependent nodes and remote process call (RPC) 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 TRILL uses either MAC addresses, the VLAN tag or fine grain label, and/or IP addresses 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 goals: First it keeps each YANG model smaller and more 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 connection oriented 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 Continuity Check command or a Loopback 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 Loopback from device A to Device B fails. Between device A and B there are IEEE 802.1 bridges a, b and c. Let’s assume a,b and c are using CFM [IEEE802.1ag]. Upon detecting the Loopback failures, a user may decide to drill down to the lower level at different segments of the path and issue the corresponding fault verification (LBM) and fault isolation (LTM) tools, using the same API. This ability to drill down to a lower layer of the protocol stack at a specific segment within a path for fault localization and troubleshooting is referred to as "nested OAM workflow". It is a useful concept that leads to efficient network troubleshooting and maintenance workflows. The connection oriented 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]. 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 [RFC2119] significance.

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


   <flags> is one of:

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


   <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
        (choice)/:(case) Parentheses enclose choice and case nodes,
            and case nodes are also marked with a colon (":")
   <type> is the name of the type for leafs and leaf-lists
```

2.1. Terminology

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

ECMP - Equal Cost Multipath.

LBM - Loopback Message [IEEE802.1ag].

MP - Maintenance Point [IEEE802.1ag].

MEP - Maintenance End Point [RFC7174] (Maintenance association End Point [IEEE802.1ag], MEG End Points [RFC6371]).

MIP - Maintenance Intermediate Point [RFC7174] (Maintenance domain Intermediate Point [IEEE802.1ag], MEG Intermediate Point [RFC6371]).

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

MD - Maintenance Domain [IEEE802.1ag]

MEG - Maintenance Entity Group [RFC6371]

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] [IEEE802.1ag].

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 Verification are used to verify that a destination is connected. It 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.

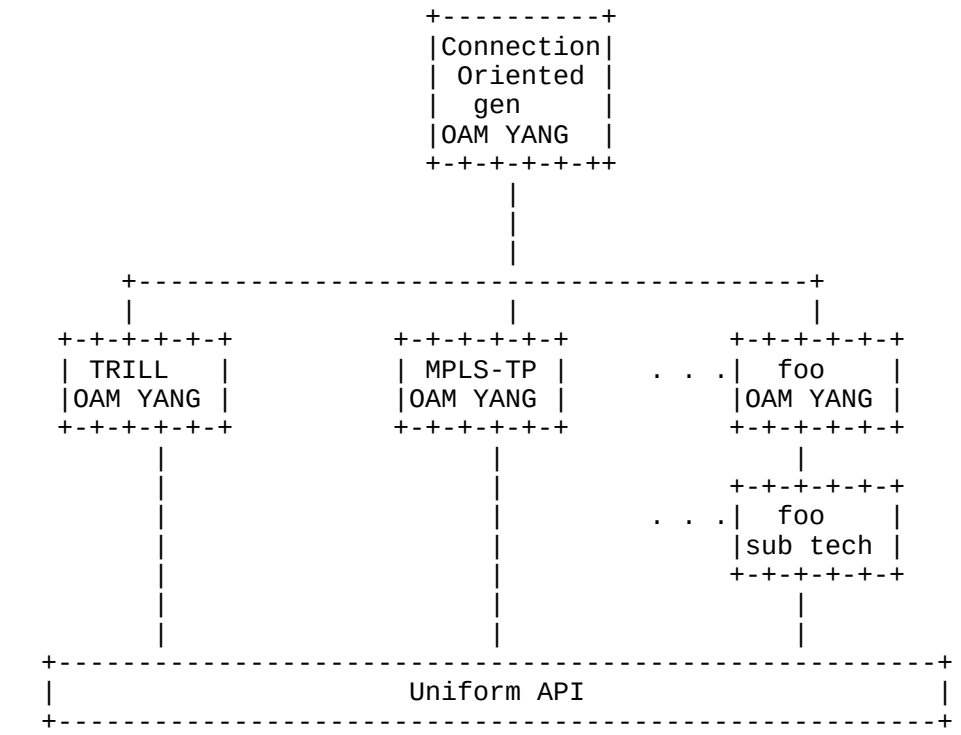
Proactive OAM - The proactive OAM refers to OAM actions which are carried out continuously to permit proactive reporting of fault. Proactive OAM method requires persistent configuration.

On-demand OAM - The on-demand OAM refers to OAM actions which are initiated via manual intervention for a limited time to carry out diagnostics. On-demand OAM method requires only transient configuration.

3. Architecture of Generic YANG Model for OAM

In this document we define a generic YANG model for connection oriented OAM protocols. The YANG model defined here is generic in a sense 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 different OAM protocols with ease through a uniform API set. This also enables a nested OAM workflow. Figure 1 depicts the relationship of different OAM YANG models to the Generic YANG Model for connection oriented OAM. 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.



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 CFM [IEEE802.1ag] model and structure it such that it can be adapted to different connection oriented OAM protocols.

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 TRILL this can be per Fine-Grained Label or for VPLS this can be per VPLS instance [RFC6136].

Under each MA, there can be two or more MEPs (Maintenance 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(traceroute) and their equivalents as well as other OAM commands.

The OAM entities in the generic YANG model defined here will be either explicitly or implicitly configured using any of the OAM tools. The OAM tools used here are limited to OAM toolset specified in section 5.1 of [RFC7276]. 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, Addresses of MEPs 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 and future extensions developed in IETF 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 [IEEE802.1ag] 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-conn-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-conn-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-conn-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
               |  +--rw MEP-ID-format?       identityref
               |  +--rw (mep-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 4.5 for the complete data hierarchy and Section 5 for the YANG model.

```
module: ietf-conn-oam
   +--rw domains
         +--rw domain* [technology MD-name-string]
         +--rw technology        identityref
   .
   .
```

```
  rpcs:
    +---x continuity-check {continuity-check}?
    |  +---w input
    |  |  +---w technology?             identityref
    |  |  +---w MD-name-string  -> /domains/domain/MD-name-string
    |  |  +---w md-level?       -> /domains/domain/md-level
    |  |  +---w MA-name-string  -> /domains/domain/MAs/MA/MA-name-string
    |  |  +---w cos-id?                 uint8
    |  |  +---w ttl?                    uint8
    |  |  +---w sub-type?               identityref
    |  |  +---w source-mep?     -> /domains/domain/MAs/MA/MEP/mep-name
    |  |  +---w destination-mep
    |  |  |  +---w (mep-address)?
    |  |  |  |  +--:(mac-address)
    |  |  |  |  |  +---w mac-address?     yang:mac-address
    |  |  |  |  +--:(ipv4-address)
    |  |  |  |  |  +---w ipv4-address?    inet:ipv4-address
    |  |  |  |  +--:(ipv6-address)
    |  |  |  |     +---w ipv6-address?    inet:ipv6-address
    |  |  |  +---w (MEP-ID)?
    |  |  |  |  +--:(MEP-ID-int)
    |  |  |  |     +---w MEP-ID-int?      int32
    |  |  |  +---w MEP-ID-format?   identityref
    |  |  +---w count?                  uint32
    |  |  +---w cc-transmit-interval?   Interval
    |  |  +---w packet-size?            uint32
    |  +--ro output
    |     +--ro (monitor-stats)?
    |        +--:(monitor-null)
    |           +--ro monitor-null?   empty
    +---x continuity-verification {connectivity-verification}?
    |  +---w input
    |  |  +---w MD-name-string  -> /domains/domain/MD-name-string
    |  |  +---w md-level?       -> /domains/domain/md-level
    |  |  +---w MA-name-string  -> /domains/domain/MAs/MA/MA-name-string
    |  |  +---w cos-id?            uint8
    |  |  +---w ttl?               uint8
    |  |  +---w sub-type?          identityref
    |  |  +---w source-mep?     -> /domains/domain/MAs/MA/MEP/mep-name
    |  |  +---w destination-mep
    |  |  |  +---w (mep-address)?
    |  |  |  |  +--:(mac-address)
    |  |  |  |  |  +---w mac-address?     yang:mac-address
    |  |  |  |  +--:(ipv4-address)
    |  |  |  |  |  +---w ipv4-address?    inet:ipv4-address
    |  |  |  |  +--:(ipv6-address)
    |  |  |  |     +---w ipv6-address?    inet:ipv6-address
    |  |  |  +---w (MEP-ID)?
```

```
|  |  |  |  +--:(MEP-ID-int)
|  |  |  |     +---w MEP-ID-int?      int32
|  |  |  +---w MEP-ID-format?   identityref
|  |  +---w count?             uint32
|  |  +---w interval?          Interval
|  |  +---w packet-size?       uint32
|  +--ro output
|     +--ro (monitor-stats)?
|        +--:(monitor-null)
|           +--ro monitor-null?   empty
+---x traceroute {traceroute}?
   +---w input
   |  +---w MD-name-string  -> /domains/domain/MD-name-string
   |  +---w md-level?       -> /domains/domain/md-level
   |  +---w MA-name-string  -> /domains/domain/MAs/MA/MA-name-string
   |  +---w cos-id?             uint8
   |  +---w ttl?                uint8
   |  +---w command-sub-type?   identityref
   |  +---w source-mep?     -> /domains/domain/MAs/MA/MEP/mep-name
   |  +---w destination-mep
   |  |  +---w (mep-address)?
   |  |  |  +--:(mac-address)
   |  |  |  |  +---w mac-address?     yang:mac-address
   |  |  |  +--:(ipv4-address)
   |  |  |  |  +---w ipv4-address?    inet:ipv4-address
   |  |  |  +--:(ipv6-address)
   |  |  |     +---w ipv6-address?    inet:ipv6-address
   |  |  +---w (MEP-ID)?
   |  |  |  +--:(MEP-ID-int)
   |  |  |     +---w MEP-ID-int?      int32
   |  |  +---w MEP-ID-format?   identityref
   |  +---w count?              uint32
   |  +---w interval?           Interval
   +--ro output
      +--ro response* [response-index]
         +--ro response-index     uint8
         +--ro ttl?               uint8
         +--ro destination-mep
         |  +--ro (mep-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 mip {mip}?
            |  +--ro interface?      if:interface-ref
            |  +--ro (mip-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 (monitor-stats)?
               +--:(monitor-null)
                  +--ro monitor-null?      empty
```

Snippet of data hierarchy related to RPC call continuity-check

4.5. Notifications

Notification is sent on defect condition and defect clears with Maintenance Domain Name, MA Name, defect-type (The currently active defects), generating-mepid, and defect-message to indicate more details.

4.6. Monitor statistics

Grouping for monitoring statistics is to be used by Yang modules which Augment Yang to provide statistics due to pro-active OAM like CCM Messages. For example CCM Transmit, CCM Receive, CCM Errors, etc.

4.7. OAM data hierarchy

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

```
module: ietf-conn-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 cos-id?           uint8
                +--rw cc-enable?        boolean
                +--rw MEP* [mep-name]
                |  +--rw mep-name         MEP-name
                |  +--rw (MEP-ID)?
                |  |  +--:(MEP-ID-int)
                |  |     +--rw MEP-ID-int?      int32
                |  +--rw MEP-ID-format?   identityref
                |  +--rw (mep-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 cos-id?          uint8
                |  +--rw cc-enable?       boolean
                |  +--rw session* [session-cookie]
                |     +--rw session-cookie             uint32
                |     +--rw destination-mep
                |     |  +--rw (MEP-ID)?
                |     |  |  +--:(MEP-ID-int)
                |     |  |     +--rw MEP-ID-int?      int32
                |     |  +--rw MEP-ID-format?   identityref
                |     +--rw destination-mep-address
                |     |  +--rw (mep-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 cos-id?                    uint8
                +--rw MIP* [interface] {mip}?
                   +--rw interface       if:interface-ref
                   +--rw (mip-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

  rpcs:
    +---x continuity-check {continuity-check}?
    |  +---w input
    |  |  +---w technology?             identityref
    |  |  +---w MD-name-string  -> /domains/domain/MD-name-string
    |  |  +---w md-level?       -> /domains/domain/md-level
    |  |  +---w MA-name-string  -> /domains/domain/MAs/MA/MA-name-string
    |  |  +---w cos-id?                 uint8
    |  |  +---w ttl?                    uint8
    |  |  +---w sub-type?               identityref
    |  |  +---w source-mep?     -> /domains/domain/MAs/MA/MEP/mep-name
    |  |  +---w destination-mep
    |  |  |  +---w (mep-address)?
    |  |  |  |  +--:(mac-address)
    |  |  |  |  |  +---w mac-address?     yang:mac-address
    |  |  |  |  +--:(ipv4-address)
    |  |  |  |  |  +---w ipv4-address?    inet:ipv4-address
    |  |  |  |  +--:(ipv6-address)
    |  |  |  |     +---w ipv6-address?    inet:ipv6-address
    |  |  |  +---w (MEP-ID)?
    |  |  |  |  +--:(MEP-ID-int)
    |  |  |  |     +---w MEP-ID-int?      int32
    |  |  |  +---w MEP-ID-format?   identityref
    |  |  +---w count?                  uint32
    |  |  +---w cc-transmit-interval?   Interval
    |  |  +---w packet-size?            uint32
    |  +--ro output
    |     +--ro (monitor-stats)?
    |        +--:(monitor-null)
    |           +--ro monitor-null?   empty
    +---x continuity-verification {connectivity-verification}?
    |  +---w input
    |  |  +---w MD-name-string  -> /domains/domain/MD-name-string
    |  |  +---w md-level?       -> /domains/domain/md-level
    |  |  +---w MA-name-string  -> /domains/domain/MAs/MA/MA-name-string
    |  |  +---w cos-id?            uint8
    |  |  +---w ttl?               uint8
    |  |  +---w sub-type?          identityref
    |  |  +---w source-mep?     -> /domains/domain/MAs/MA/MEP/mep-name
    |  |  +---w destination-mep
    |  |  |  +---w (mep-address)?
    |  |  |  |  +--:(mac-address)
    |  |  |  |  |  +---w mac-address?     yang:mac-address
    |  |  |  |  +--:(ipv4-address)
    |  |  |  |  |  +---w ipv4-address?    inet:ipv4-address
    |  |  |  |  +--:(ipv6-address)
```

```
|  |  |  |     +---w ipv6-address?    inet:ipv6-address
|  |  |  +---w (MEP-ID)?
|  |  |  |  +--:(MEP-ID-int)
|  |  |  |     +---w MEP-ID-int?      int32
|  |  |  +---w MEP-ID-format?   identityref
|  |  +---w count?             uint32
|  |  +---w interval?          Interval
|  |  +---w packet-size?       uint32
|  +--ro output
|     +--ro (monitor-stats)?
|        +--:(monitor-null)
|           +--ro monitor-null?   empty
+---x traceroute {traceroute}?
   +---w input
   |  +---w MD-name-string -> /domains/domain/MD-name-string
   |  +---w md-level?      -> /domains/domain/md-level
   |  +---w MA-name-string -> /domains/domain/MAs/MA/MA-name-string
   |  +---w cos-id?             uint8
   |  +---w ttl?                uint8
   |  +---w command-sub-type?   identityref
   |  +---w source-mep?    -> /domains/domain/MAs/MA/MEP/mep-name
   |  +---w destination-mep
   |  |  +---w (mep-address)?
   |  |  |  +--:(mac-address)
   |  |  |  |  +---w mac-address?     yang:mac-address
   |  |  |  +--:(ipv4-address)
   |  |  |  |  +---w ipv4-address?    inet:ipv4-address
   |  |  |  +--:(ipv6-address)
   |  |  |     +---w ipv6-address?    inet:ipv6-address
   |  |  +---w (MEP-ID)?
   |  |  |  +--:(MEP-ID-int)
   |  |  |     +---w MEP-ID-int?      int32
   |  |  +---w MEP-ID-format?   identityref
   |  +---w count?             uint32
   |  +---w interval?          Interval
   +--ro output
      +--ro response* [response-index]
         +--ro response-index     uint8
         +--ro ttl?               uint8
         +--ro destination-mep
         |  +--ro (mep-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 mip {mip}?
            |  +--ro interface?      if:interface-ref
            |  +--ro (mip-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 (monitor-stats)?
               +--:(monitor-null)
                  +--ro monitor-null?      empty

  notifications:
    +---n defect-condition-notification
    |  +--ro technology?         identityref
    |  +--ro MD-name-string -> /domains/domain/MD-name-string
    |  +--ro MA-name-string -> /domains/domain/MAs/MA/MA-name-string
    |  +--ro mep-name?      -> /domains/domain/MAs/MA/MEP/mep-name
    |  +--ro defect-type?        identityref
    |  +--ro generating-mepid
    |  |  +--ro (MEP-ID)?
    |  |  |  +--:(MEP-ID-int)
    |  |  |     +--ro MEP-ID-int?      int32
    |  |  +--ro MEP-ID-format?   identityref
    |  +--ro (defect)?
    |     +--:(defect-null)
    |     |  +--ro defect-null?        empty
    |     +--:(defect-code)
    |        +--ro defect-code?        int32
    +---n defect-cleared-notification
       +--ro technology?         identityref
       +--ro MD-name-string -> /domains/domain/MD-name-string
       +--ro MA-name-string -> /domains/domain/MAs/MA/MA-name-string
       +--ro mep-name?      -> /domains/domain/MAs/MA/MEP/mep-name
       +--ro defect-type?        identityref
       +--ro generating-mepid
       |  +--ro (MEP-ID)?
       |  |  +--:(MEP-ID-int)
       |  |     +--ro MEP-ID-int?      int32
       |  +--ro MEP-ID-format?   identityref
       +--ro (defect)?
          +--:(defect-null)
          |  +--ro defect-null?        empty
          +--:(defect-code)
```

```
             +--ro defect-code?        int32
```

data hierarchy of OAM

5. OAM YANG Module

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

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

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

  organization "IETF LIME Working Group";
  contact
    "WG Web:    http://tools.ietf.org/wg/lime
     WG List:   mailto:lime@ietf.org
     WG Chair:  Carlos Pignataro cpignata@cisco.com
     WG Chair:  Ron Bonica rbonica@juniper.net
     Editor:    Deepak Kumar dekumar@cisco.com
     Editor:    Qin Wu bill.wu@huawei.com
     Editor:    Zitao Wang wangzitao@huawei.com";
  description
    "This YANG module defines the generic configuration,
  statistics and rpc for connection oriented 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 2017-04-10 {
    description
      "Initial revision. - 08 version";

    reference "draft-ietf-lime-yang-oam-model";
  }
```

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

  feature traceroute{
    description
      "This feature indicates that the server supports
    executing traceroute OAM command and
    returning a response. Servers that do not advertise
    this feature will not support executing
    traceroute command or rpc model for
    traceroute command.";
   }
  feature mip {
   description
   "This feature indicates that the MIP (Maintenance Intermediate Point)
   need to
   be explicit configured";
  }
  /* Identities */

  identity technology-types {
    description
    "This is the base identy of technology types which are
    TRILL,MPLS-TP,vpls etc";
  }

  identity command-sub-type {
    description
      "Defines different rpc command subtypes,
    e.g rfc6905 trill OAM, this is optional for most cases";
  }
```

```
  identity on-demand {
   base command-sub-type;
   description
   "On demand activation - indicates that the tool is activated
    manually to detect a specific anomaly.
    On-demand OAM method requires only transient configuration.";
  }

  identity proactive {
   base command-sub-type;
   description
   "Proactive activation - indicates that the tool is activated on a
    continual basis, where messages are sent periodically, and errors
    are detected when a certain number of expected messages are not
    received. Proactive OAM method requires persistent configuration.";
  }

  identity name-format {

    description
    "This defines the name format, IEEE 8021ag 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 MEP-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. rdi
    (Remote Defect Indication), loss of continuity";
```

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

  identity remote-mep-defect{
    base defect-types;
    description
        "Indicates that one or more of the remote MEPs is
    reporting a failure ";
  }

  identity loss-of-continuity{
    base defect-types;
    description
    "If no proactive CC OAM packets from the source
    MEP (and in the case of CV, this includes the
    requirement to have the expected unique,
    technology dependent source MEP identifier)
    are received within the interval. ";
   }

   identity cv-defect {
    base defect-types;
    description
    "This function should support monitoring between the MEPs and,
    in addition, between a MEP and MIP.[RFC6371] highlights,
    when performing Connectivity Verification, the need for the
    Continuity Check and Connectivity Verification (CC-V) messages
    to include unique identification of the MEG that is being
    monitored and the MEP that originated the message.";
   }


  identity invalid-oam-defect{
    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-defect{
    base defect-types;
    description
    "Indicates that one or more cross-connect defect
```

```
    (for example, a service ID does not match the VLAN.)
     messages has been received and that 3.5 times that OAM message
     transmission interval has not yet expired.";
  }

  /* typedefs */

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

  typedef Interval{
    type decimal64{
    fraction-digits 2;
   }
   units "milliseconds";
    description
    "Interval between packets in milliseconds.
    0 means no packets are sent.";
  }

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

  typedef MA-name-string {
    type string;
    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 maintenance-domain-reference {
  description
     "This grouping uniquely identifies a maintenance domain.";
  leaf maintenance-domain {
     type leafref {
       path "/goam:domains/goam:domain/goam:MD-name-string";
     }
  description
    "A reference to a specific Maintenance Domain.";
}
}

grouping maintenance-association-reference {
  description
    "This grouping uniquely identifies a
     maintenance association. It consists
     of a maintence-domain-reference and
     a maintenance-association leafref";
  uses maintenance-domain-reference;
  leaf maintenance-association {
    type leafref {
      path "/goam:domains/goam:domain"
          +"[goam:MD-name-string = current()/"
          +"../maintenance-domain]/goam:MAs"
          +"/goam:MA/goam:MA-name-string";
    }
  description
      "A reference to a specific Maintenance Association.";
  }
}

grouping maintenance-association-end-point-reference {
  description
  "This grouping uniquely identifies
   a maintenance association. It consists
   of a maintence-association-reference and
   a maintenance-association-end-point leafref";
  uses maintenance-association-reference;
  leaf maintenance-association-end-point {
    type leafref {
      path "/goam:domains/goam:domain"
          +"[goam:MD-name-string = current()/"
          +"../maintenance-domain]/goam:MAs"
          +"/goam:MA[goam:MA-name-string = "
```

```
          +"current()/../maintenance-association]"
          +"/goam:MEP/goam:mep-name";
     }
    description
      "A reference to a specific Maintenance
          association End Point.";
  }
 }

  grouping time-to-live {
    leaf ttl{
     type uint8;
     description
       "Time to Live.";
    }
    description
      "Time to Live grouping.";
  }
  grouping defect-message {
    choice defect {
      case defect-null {
        description
          "This is a placeholder when no defect status is needed";
        leaf defect-null {
           type empty;
           description
             "there is no defect define, it will be defined in
              technology specific model.";
        }
      }
      case defect-code {
        description
          "This is a placeholder to display defect code.";
        leaf defect-code {
           type int32;
           description
             "Defect code is integer value specific to technology.";
        }
      }

      description
        "Defect Message choices.";
    }

    description
      "Defect Message.";
  }
```

```
  grouping mep-address {
    choice mep-address {
      case mac-address {
        leaf mac-address {
          type yang:mac-address;
          description
          "MAC Address";
        }
      description
      "MAC Address based MEP Addressing.";
      }
      case ipv4-address {
        leaf ipv4-address {
          type inet:ipv4-address;
          description
          "IPv4 Address";
        }
      description
      "IP Address based MEP Addressing.";
      }
      case ipv6-address {
        leaf ipv6-address {
          type inet:ipv6-address;
          description
          "IPv6 Address";
        }
        description
        "IPv6 Address based MEP Addressing.";
      }
      description
       "MEP Addressing.";
    }
    description
     "MEP Address";
  }
  grouping mip-address {
    choice mip-address {
      case mac-address {
        leaf mac-address {
          type yang:mac-address;
          description
          "MAC Address";
        }
      description
      "MAC Address based MIP Addressing.";
      }
      case ipv4-address {
        leaf ipv4-address {
```

```
          type inet:ipv4-address;
          description
          "IPv4 Address";
        }

      description
      "IP Address based MIP Addressing.";
      }
      case ipv6-address {
        leaf ipv6-address {
          type inet:ipv6-address;
          description
          "IPv6 Address";
        }
        description
        "IPv6 Address based MIP Addressing.";
      }
      description
       "MIP Addressing.";
    }
    description
     "MIP 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";
        }
      }
      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 mep-address;
  }

 grouping monitor-stats {
   description
    "grouping for monitoring statistics, this will be augmented
    by others who use this component";
    choice monitor-stats {

     default "monitor-null";
      case monitor-null {
       description
        "This is a place holder when
         no monitoring statistics is needed";
         leaf monitor-null {
          type empty;
          description
          "There is no monitoring statistics to be defined";
              }
            }
          description
          "Define the monitor stats";
          }
        }

  grouping connectivity-context {
    description
      "Grouping defining the connectivity context for an MA; for
       example, a VRF for VPLS, or an LSP for MPLS-TP.  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 cos {
    description
```

```
    "Priority used in transmitted packets; for example, in the
    EXP field in MPLS-TP.";

    leaf cos-id {
     type uint8;
     description
     "Class of service";
    }
  }
  grouping MIP-grouping {
   uses mip-address;
   description
   "Grouping for MIP configuration";
  }

  container domains {
    description
      "Contains configuration related data. Within the container
       is list of fault domains. Within 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";
        list MA {
          key "MA-name-string";
          ordered-by system;
          uses ma-identifier;
          uses MA-name;
          uses connectivity-context;
          uses cos {
          description
          "Default class of service for this MA,
           which may be overridden
           for particular MEPs,
           sessions or operations.";
          }
```

```
             leaf cc-enable{
              type boolean;
              description
              "Indicate whether the CC enable.";
             }
           list MEP {
             key "mep-name";

             ordered-by system;
             description
             "Contain list of MEPS";
             uses MEP;
             uses cos;
             leaf cc-enable{
              type boolean;
              description
              "Indicate whether the CC enable.";
             }
             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.";
               }
               container destination-mep {
                 uses MEP-ID;
                 description
                 "Destination MEP";
               }
               container destination-mep-address {
                 uses mep-address;
                 description
                 "Destination MEP Address";
               }
               uses cos;
             }
```

```
          }
          list MIP {
           if-feature mip;
           key "interface";
           leaf interface {
            type if:interface-ref;
            description
            "Interface";
           }
           uses MIP-grouping;
           description
           "List for MIP";
          }
          description
          "Maintenance Association list";
         }
        }
      }

  }

  notification defect-condition-notification {
    description
      "When defect condition is met this notification is sent";
    leaf technology {
      type identityref {
        base technology-types;
      }
      description
      "The technology";
    }
    leaf MD-name-string {
      type leafref{
      path "/domains/domain/MD-name-string";
      }
      mandatory true;
      description
      "Indicate which MD is seeing the defect";
    }
    leaf MA-name-string{
     type leafref{
     path "/domains/domain/MAs/MA/MA-name-string";
     }
      mandatory true;
      description
      "Indicate which MA is seeing the defect";
        }
    leaf mep-name {
```

```
      type leafref{
       path "/domains/domain/MAs/MA/MEP/mep-name";
      }
      description
       "Indicate which MEP is seeing the defect";
    }
    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 defect (if known) if
       unknown make it 0.";
    }
    uses defect-message {
      description
      "Defect message to indicate more details.";
    }
  }

  notification defect-cleared-notification {
    description
      "When defect cleared is met this notification is sent";
    leaf technology {
      type identityref {
        base technology-types;
      }
      description
        "The technology";
    }
    leaf MD-name-string {
      type leafref{
      path "/domains/domain/MD-name-string";
      }
      mandatory true;
      description
      "Indicate which MD is seeing the defect";
    }
    leaf MA-name-string{
      type leafref{
      path "/domains/domain/MAs/MA/MA-name-string";
      }
```

```
      mandatory true;
      description
      "Indicate which MA is seeing the defect";
        }
    leaf mep-name {
      type leafref{
      path "/domains/domain/MAs/MA/MEP/mep-name";
      }
      description
        "Indicate which MEP is seeing the defect";
    }

    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 defect (if known) if
       unknown make it 0.";
    }
    uses defect-message {
      description
      "Defect message to indicate more details.";
    }
  }

  rpc continuity-check {
    if-feature "continuity-check";
    description
      "Generates continuity-check as per RFC7276 Table 4.";
    input {
    leaf technology {
      type identityref {
        base technology-types;
      }
      description
        "The technology";
    }
    leaf MD-name-string {
      type leafref{
       path "/domains/domain/MD-name-string";
      }
      mandatory true;
```

```
      description
      "Indicate which MD is seeing the defect";
    }
    leaf md-level {
     type leafref {
     path "/domains/domain/md-level";
     }
     description
     "The maintenance domain level.";
    }
    leaf MA-name-string{
     type leafref{
     path "/domains/domain/MAs/MA/MA-name-string";
     }
     mandatory true;
     description
     "Indicate which MA is seeing the defect";
    }
      uses cos;
      uses time-to-live;
      leaf sub-type {
        type identityref {
          base command-sub-type;
        }
        description
        "Defines different command types";
      }
      leaf source-mep {
       type leafref{
       path "/domains/domain/MAs/MA/MEP/mep-name";
       }
       description
       "Source MEP";
      }
      container destination-mep {
        uses mep-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 continuity-check message to send";
        }
        leaf cc-transmit-interval {
          type Interval;
          description
            "Interval between echo requests";
        }
        leaf packet-size {
          type uint32 {
            range "0..10000";
          }
          default "64";
          description
            "Size of continuity-check 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 {
       leaf MD-name-string {
        type leafref{
         path "/domains/domain/MD-name-string";
        }
        mandatory true;
        description
        "Indicate which MD is seeing the defect";
       }

       leaf md-level {
        type leafref {
        path "/domains/domain/md-level";
        }
        description
        "The maintenance domain level.";
       }

       leaf MA-name-string{
```

```
      type leafref{
      path "/domains/domain/MAs/MA/MA-name-string";
      }
      mandatory true;
      description
      "Indicate which MA is seeing the defect";
     }
      uses cos;
      uses time-to-live;
      leaf sub-type {
        type identityref {
          base command-sub-type;
        }
        description
         "Defines different command types";
      }
      leaf source-mep {
       type leafref{
        path "/domains/domain/MAs/MA/MEP/mep-name";
       }
       description
       "Source MEP";
      }
      container destination-mep {
        uses mep-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 continuity-verification 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 continuity-verification packets, in octets";
      }
    }

    output {
      uses monitor-stats {
        description
          "Stats of continuity check.";
      }
    }
  }
  rpc traceroute {
    if-feature traceroute;
    description
      "Generates Traceroute or Path Trace and return response.
       Referencing RFC7276 for common Toolset name, for
       MPLS-TP OAM it's Route Tracing, 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 value";
    input {
     leaf MD-name-string {
      type leafref{
       path "/domains/domain/MD-name-string";
      }
      mandatory true;
      description
      "Indicate which MD is seeing the defect";
     }

     leaf md-level {
      type leafref {
      path "/domains/domain/md-level";
      }
      description
      "The maintenance domain level.";
     }

     leaf MA-name-string{
      type leafref{
       path "/domains/domain/MAs/MA/MA-name-string";
      }
      mandatory true;
      description
      "Indicate which MA is seeing the defect";
         }
      uses cos;
```

```
      uses time-to-live;
      leaf command-sub-type {
        type identityref {
          base command-sub-type;
        }
        description
         "Defines different command types";
      }
      leaf source-mep {
       type leafref{
        path "/domains/domain/MAs/MA/MEP/mep-name";
       }
       description
       "Source MEP";
      }
      container destination-mep {
        uses mep-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,
                   the TTL can be used as the response index.";
```

```
        }
        uses time-to-live;
        container destination-mep {
          description "MEP from which the response has been received";
          uses mep-address;
          uses MEP-ID {
            description
            "Only applicable if the destination is a MEP";
          }
        }
        container mip {
         if-feature mip;
         leaf interface {
         type if:interface-ref;
         description
         "MIP interface";
         }
         uses mip-address;
         description
         "MIP responding with traceroute";
         }
        uses monitor-stats {
         description
         "Stats of traceroute.";
        }
      description
        "List of response.";
      }
    }
  }
}
```

<CODE ENDS>

6. Base Mode

The Base Mode (’default mode’ described in section 4) 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 MEP associated with a virtual OAM port with no physical layer (NULL PHY). The MEP-ID associated with this MEP is zero (0). The choice of MEP-ID zero is explained below.

MEP-ID is 2 octet field by default. It is never used on the wire except when using CCM. 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.

CFM [IEEE802.1ag] defines MEP ID as an unsigned integer in the range 1 to 8191. In this document we propose 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 Association

The ID of the Maintenance Association (MA-ID) [IEEE802.1ag] 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 [IEEE802.1ag]) and Short MA name set to 65532 (0xFFFC).

7. Connection-oriented OAM YANG model applicability

"ietf-conn-oam" model defined in this document provides technology-independent abstraction of key OAM constructs for connection oriented protocols. This model can be further extended to include technology specific details, e.g., adding new data nodes with technology specific functions and parameters into proper anchor points of the base model, so as to develop a technology-specific connection-oriented OAM model.

This section demonstrates the usability of the connection-oriented YANG OAM data model to various connection-oriented OAM technologies, e.g., TRILL and MPLS-TP. Note that, in this section, we only present several snippets of technology-specific model extensions for illustrative purposes. The complete model extensions should be worked on in respective protocol working groups.

7.1. Generic YANG Model extension for TRILL OAM

The TRILL YANG module is augmenting connection oriented OAM module for both configuration and RPC commands.

The TRILL YANG module requires the base TRILL module ([I-D.ietf-trill-yang]) to be supported as there is a strong relationship between those modules.

The configuration extensions for connection oriented OAM include MD configuration extension, Technology type extension, MA configuration extension, Connectivity-Context Extension, MEP Configuration Extension, ECMP extension. In the RPC extension, the continuity-check and path-discovery RPC are extended with TRILL specific.

7.1.1. MD Configuration Extension

MD level configuration parameters are management information which can be inherited in the TRILL OAM model and set by connection oriented base model as default values. For example domain name can be set to area-ID in the TRILL OAM case. In addition, at the Maintenance Domain level, domain data node at root level can be augmented with technology type.

Note that MD level configuration parameters provides context information for management system to correlate faults, defects, network failures with location information, which helps quickly identify root causes of network failures.

7.1.1.1. Technology Type Extension

No TRILL technology type has been defined in the connection oriented base model. Therefore a technology type extension is required in the TRILL OAM model. The technology type "trill" is defined as an identity that augments the base "technology-types" defined in the connection oriented base model:

```
identity trill{
 base goam:technology-types;
 description
  "trill type";
}
```

7.1.2. MA Configuration Extension

MA level configuration parameters are management information which can be inherited in the TRILL OAM model and set by connection oriented base model as default values. In addition, at the Maintenance Association(MA) level, MA data node at the second level can be augmented with connectivity-context extension.

Note that MA level configuration parameters provides context information for management system to correlate faults, defects, network failures with location information, which helps quickly identify root causes of network failures.

7.1.2.1. Connectivity-Context Extension

In TRILL OAM, one example of connectivity-context is either a 12 bit VLAN ID or a 24 bit Fine Grain Label. The connection oriented base model defines a placeholder for context-id. This allows other technologies to easily augment that to include technology specific extensions. The snippet below depicts an example of augmenting connectivity-context to include either VLAN ID or Fine Grain Label.

```
augment /goam:domains/goam:domain/goam:MAs
/goam:MA /goam:connectivity-context:
      +--:(connectivity-context-vlan)
      |  +--rw connectivity-context-vlan?   vlan
      +--:(connectivity-context-fgl)
         +--rw connectivity-context-fgl?    fgl
```

7.1.3. MEP Configuration Extension

The MEP configuration definition in the connection oriented base model already supports configuring the interface of MEP with either MAC address or IP address. In addition, the MEP address can be represented using a 2 octet RBridge Nickname in TRILL OAM . Hence, the TRILL OAM model augments the MEP configuration in base model to add a nickname case into the MEP address choice node as follows:

```
augment /goam:domains/goam:domain/goam:MAs
   /goam:MA/ goam:MEP/goam:mep-address:
         +--:( mep-address-trill)
         |  +--rw mep-address-trill?  tril-rb-nickname
```

In addition, at the Maintenance Association Endpoint(MEP) level, MEP data node at the third level can be augmented with ECMP extension.

7.1.3.1. ECMP Extension

Since TRILL supports ECMP path selection, flow-entropy in TRILL is defined as a 96 octet field in the LIME model extension for TRILL OAM. The snippet below illustrates its extension.

```
 augment /goam:domains/goam:domain/goam:MAs/goam:MA/goam:MEP:
           +--rw flow-entropy-trill?   flow-entropy-trill
   augment /goam:domains/goam:domain/goam:MAs/goam:MA/goam:MEP
   /goam:session:
           +--rw flow-entropy-trill?   flow-entropy-trill
```

7.1.4. RPC extension

In the TRILL OAM YANG model, the continuity-check and path-discovery RPC commands are extended with TRILL specific requirements. The snippet below depicts an example of illustrates the TRILL OAM RPC extension.

```
augment /goam:continuity-check/goam:input:
      +--ro (out-of-band)?
      |  +--:(ipv4-address)
      |  |  +--ro ipv4-address?      inet:ipv4-address
      |  +--:(ipv6-address)
      |  |  +--ro ipv6-address?      inet:ipv6-address
      |  +--:(trill-nickname)
      |     +--ro trill-nickname?    tril-rb-nickname
      +--ro diagnostic-vlan?   boolean
augment /goam:continuity-check/goam:input:
         +--ro flow-entropy-trill?   flow-entropy-trill
augment /goam:continuity-check/goam:output:
      +--ro upstream-rbridge?   tril-rb-nickname
      +--ro next-hop-rbridge*   tril-rb-nickname
augment /goam:path-discovery/goam:input:
      +--ro (out-of-band)?
      |  +--:(ipv4-address)
      |  |  +--ro ipv4-address?      inet:ipv4-address
      |  +--:(ipv6-address)
      |  |  +--ro ipv6-address?      inet:ipv6-address
      |  +--:(trill-nickname)
      |     +--ro trill-nickname?    tril-rb-nickname
      +--ro diagnostic-vlan?   boolean
augment /goam:path-discovery/goam:input:
         +--ro flow-entropy-trill?   flow-entropy-trill
augment /goam:path-discovery/goam:output/goam:response:
      +--ro upstream-rbridge?   tril-rb-nickname
      +--ro next-hop-rbridge*   tril-rb-nickname
```

7.2. Generic YANG Model extension for MPLS-TP OAM

The MPLS-TP OAM YANG module can augment connection oriented OAM Module with some technology-specific details. And the [mpls-tp-oam-yang] presents the YANG Data model for MPLS-TP OAM.

The configuration extensions for connection oriented OAM include MD configuration extension, Technology type extension, Sub Technology Type Extension ,MA configuration extension, MEP Configuration Extension.

7.2.1. MD Configuration Extension

MD level configuration parameters are management information which can be inherited in the MPLS-TP OAM model and set by LIME base model as default values. For example domain name can be set to area-ID or the provider’s Autonomous System Number(ASN) [RFC6370] in the MPLS-TP OAM case. In addition, at the Maintenance Domain level, domain data

node at root level can be augmented with technology type and sub-technology type.

Note that MD level configuration parameters provides context information for management system to correlate faults, defects, network failures with location information, which helps quickly identify root causes of network failures

7.2.1.1. Technology Type Extension

No MPLS-TP technology type has been defined in the connection oriented base model, hence it is required in the MPLS OAM model. The technology type "mpls-tp" is defined as an identity that augments the base "technology-types" defined in the connection oriented base model:

```
identity mpls-tp{
      base goam:technology-types;
      description
       "mpls-tp type";
     }
```

7.2.1.2. Sub Technology Type Extension

In MPLS-TP, since different encapsulation types such as IP/UDP Encapsulation, PW-ACH encapsulation can be employed, the "technology-sub-type" data node is defined and added into the MPLS OAM model to further identify the encapsulation types within the MPLS-TP OAM model. Based on it, we also define a technology sub-type for IP/UDP encapsulation and PW-ACH encapsulation. Other Encapsulation types can be defined in the same way. The snippet below depicts an example of several encapsulation types.

```
   identity technology-sub-type {
         description
         "certain implementations can have different
          encapsulation types such as ip/udp, pw-ach and so on.
          Instead of defining separate models for each
          encapsulation, we define a technology sub-type to
       further identify different encapsulations.
       Technology sub-type is associated at the MA level"; }

             identity technology-sub-type-udp {
               base technology-sub-type;
               description
                 "technology sub-type is IP/UDP encapsulation";
             }

             identity technology-sub-type-ach {
               base technology-sub-type;
               description
                 "technology sub-type is PW-ACH encapsulation";
             }
             }

        augment "/goam:domains/goam:domain/goam:MAs/goam:MA" {
               leaf technology-sub-type {
                 type identityref {
                   base technology-sub-type;
                 }
               }
             }
```

7.2.2. MA Configuration Extension

MA level configuration parameters are management information which can be inherited in the MPLS-TP OAM model and set by Connection Oriented base model as default values. One example of MA Name could be MEG LSP ID or MEG Section ID or MEG PW ID[RFC6370].

Note that MA level configuration parameters provides context information for management system to correlate faults, defects, network failures with location information, which helps quickly identify root causes of network failures.

7.2.3. MEP Configuration Extension

In MPLS-TP, MEP-ID is either a variable length label value in case of G-ACH encapsulation or a 2 octet unsigned integer value in case of IP/UDP encapsulation. One example of MEP-ID is MPLS-TP LSP_MEP_ID [RFC6370]. In the connection-oriented base model, MEP-ID is defined

as a choice/case node which can supports an int32 value, and the same definition can be used for MPLS-TP with no further modification. In addition, at the Maintenance Association Endpoint(MEP) level, MEP data node at the third level can be augmented with Session extension and interface extension.

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,Mishael Wexler, Adi Molkho 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>.

[RFC6370] Bocci, M., Swallow, G., and E. Gray, "MPLS Transport Profile (MPLS-TP) Identifiers", RFC 6370, DOI 10.17487/RFC6370, September 2011, <http://www.rfc-editor.org/info/rfc6370>.

[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>.

11.2. Informative References

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

[IEEE802.1ag] "Connectivity Fault Management", IEEE Std 802.1ag-2011, August 2011.

[mpls-tp-oam-yang] Zhang, L., Zheng, L., Aldrin, S., and G. Mirsky, "YANG Data Model for MPLS-TP Operations, Administration, and Maintenance", draft-zhang-mpls-tp-yang-oam (work in progress), 2016.

[RFC6136] Sajassi, A., Ed. and D. Mohan, Ed., "Layer 2 Virtual Private Network (L2VPN) Operations, Administration, and Maintenance (OAM) Requirements and Framework", RFC 6136, DOI 10.17487/RFC6136, March 2011, <http://www.rfc-editor.org/info/rfc6136>.

[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>.

[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>.

Appendix A. Contributors' Addresses

Tissa Senevirathne
Consultant

Email: tsenevir@gmail.com

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

Email: nfinn@cisco.com

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

Email: ssalam@cisco.com

Authors' Addresses

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

Email: dekumar@cisco.com

Qin Wu
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