Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 4, 2017

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

October 31, 2016

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

Abstract

This document presents a base YANG Data model for connectionless OAM protocols. It provides a technology-independent abstraction of key OAM constructs for connectionless protocols. The Based model presented here can be extended to include technology specific details. This is leading to uniformity between OAM protocols and support 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 May 4, 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.

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[lime retrieval methods], which focuses on data retrival 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

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
x for deprecated
o for obsolete

<flags> is one of:

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

<name> is the name of the node

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

<opts> is one of:

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

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

3. Overview of the Connectionless OAM Model

At the top of the Model, there is an oper 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.draft-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 client layer, server layer or same layer. 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
- IPv4 or IPv6 address
- a pair of source, destination addresses, and interface (Useful for BFD)
- FEC
- System-id to represent the device or node.

3.2. Tools

In connectionless OAM, the tools attribute is used to describe a toolset for fault detection and isolation, and for performance measurement. 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:tools-ip/ coam:rfc792" 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 has its own OAM protocols [RFC7276] and is corresponding to specific network portion or pathand has associated test points. OAM-layers is referred to a list of upper layer, lower layer that are related to current test point. This allow users to easily navigate up and down to efficiently troubleshoot a connectivity issue at different layer. In this model, we have kept level default as 0, when all test points are located at the same layer. Level is provided for scenarios where it might be possible to define layering relationship as it can be used to stitching fault at related oam layers. 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 thesecond 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 active, passive and hybrid OAMs to do the retrieval. None of the fields are made mandatory for that reason. Noted that the retrieval methods are defined in [lime retrieval 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 active, passive and hybrid OAMs to do the retrieval. None of the fields are made mandatory for that reason. Noted that the retrieval methods are defined in [lime retrieval methods].

3.8. OAM data hierarchy

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

```
module: ietf-connectionless-oam
   +--ro oper {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-address-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
      |        +--ro (tools)?
      |        |  +--:(tools-empty)
      |        |  |  +--ro tools-null?      empty
      |        |  +--:(tools-ip)
      |        |  |  +--ro rfc792?          boolean
```

```
|        |  |  +--ro rfc4443?         boolean
|        |  |  +--ro rfc4884?         boolean
|        |  |  +--ro rfc5837?         boolean
|        |  +--:(tools-bfd)
|        |  |  +--ro rfc5881?         boolean
|        |  |  +--ro rfc5883?         boolean
|        |  |  +--ro rfc5884?         boolean
|        |  |  +--ro rfc5885?         boolean
|        |  +--:(tools-mpls)
|        |  |  +--ro rfc4379?         boolean
|        |  |  +--ro rfc4687?         boolean
|        |  |  +--ro rfc4950?         boolean
|        |  |  +--ro mpls-rfc5884?    boolean
|        |  +--:(tools-pw)
|        |     +--ro rfc5085?         boolean
|        |     +--ro pw_rfc5885?      boolean
|        |     +--ro rfc6423?         boolean
|        |     +--ro rfc6310?         boolean
|        |     +--ro rfc7023?         boolean
|        +--rw oam-layers* [index]
|           +--rw index    uint16
|           +--rw level?   int32
|     +--:(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
|     |        +--ro (tools)?
|     |        |  +--:(tools-empty)
|     |        |  |  +--ro tools-null?      empty
|     |        |  +--:(tools-ip)
|     |        |  |  +--ro rfc792?          boolean
|     |        |  |  +--ro rfc4443?         boolean
|     |        |  |  +--ro rfc4884?         boolean
|     |        |  |  +--ro rfc5837?         boolean
|     |        |  +--:(tools-bfd)
|     |        |  |  +--ro rfc5881?         boolean
|     |        |  |  +--ro rfc5883?         boolean
|     |        |  |  +--ro rfc5884?         boolean
|     |        |  |  +--ro rfc5885?         boolean
|     |        |  +--:(tools-mpls)
|     |        |  |  +--ro rfc4379?         boolean
|     |        |  |  +--ro rfc4687?         boolean
```

```
|        |  |  +--ro rfc4950?         boolean
|        |  |  +--ro mpls-rfc5884?    boolean
|        |  +--:(tools-pw)
|        |     +--ro rfc5085?         boolean
|        |     +--ro pw_rfc5885?      boolean
|        |     +--ro rfc6423?         boolean
|        |     +--ro rfc6310?         boolean
|        |     +--ro rfc7023?         boolean
|        +--rw oam-layers* [index]
|           +--rw index    uint16
|           +--rw level?   int32
|     +--:(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
|     |        +--ro (tools)?
|     |        |  +--:(tools-empty)
|     |        |  |  +--ro tools-null?             empty
|     |        |  +--:(tools-ip)
|     |        |  |  +--ro rfc792?                 boolean
|     |        |  |  +--ro rfc4443?                boolean
|     |        |  |  +--ro rfc4884?                boolean
|     |        |  |  +--ro rfc5837?                boolean
|     |        |  +--:(tools-bfd)
|     |        |  |  +--ro rfc5881?                boolean
|     |        |  |  +--ro rfc5883?                boolean
|     |        |  |  +--ro rfc5884?                boolean
|     |        |  |  +--ro rfc5885?                boolean
|     |        |  +--:(tools-mpls)
|     |        |  |  +--ro rfc4379?                boolean
|     |        |  |  +--ro rfc4687?                boolean
|     |        |  |  +--ro rfc4950?                boolean
|     |        |  |  +--ro mpls-rfc5884?           boolean
|     |        |  +--:(tools-pw)
|     |        |     +--ro rfc5085?                boolean
|     |        |     +--ro pw_rfc5885?             boolean
|     |        |     +--ro rfc6423?                boolean
|     |        |     +--ro rfc6310?                boolean
|     |        |     +--ro rfc7023?                boolean
|     |        +--rw oam-layers* [index]
|     |           +--rw index    uint16
|     |           +--rw level?   int32
|     +--:(tunnel-location-type)
```

```
|  +--rw test-point-tunnel-address-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
|        +--ro (tools)?
|        |  +--:(tools-empty)
|        |  |  +--ro tools-null?        empty
|        |  +--:(tools-ip)
|        |  |  +--ro rfc792?            boolean
|        |  |  +--ro rfc4443?           boolean
|        |  |  +--ro rfc4884?           boolean
|        |  |  +--ro rfc5837?           boolean
|        |  +--:(tools-bfd)
|        |  |  +--ro rfc5881?           boolean
|        |  |  +--ro rfc5883?           boolean
|        |  |  +--ro rfc5884?           boolean
|        |  |  +--ro rfc5885?           boolean
|        |  +--:(tools-mpls)
|        |  |  +--ro rfc4379?           boolean
|        |  |  +--ro rfc4687?           boolean
|        |  |  +--ro rfc4950?           boolean
|        |  |  +--ro mpls-rfc5884?      boolean
|        |  +--:(tools-pw)
|        |     +--ro rfc5085?           boolean
|        |     +--ro pw_rfc5885?        boolean
|        |     +--ro rfc6423?           boolean
|        |     +--ro rfc6310?           boolean
|        |     +--ro rfc7023?           boolean
|        +--rw oam-layers* [index]
|           +--rw index    uint16
|           +--rw level?   int32
+--:(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
|        +--ro (tools)?
|        |  +--:(tools-empty)
```

```
|        |  |  +--ro tools-null?          empty
|        |  +--:(tools-ip)
|        |  |  +--ro rfc792?               boolean
|        |  |  +--ro rfc4443?              boolean
|        |  |  +--ro rfc4884?              boolean
|        |  |  +--ro rfc5837?              boolean
|        |  +--:(tools-bfd)
|        |  |  +--ro rfc5881?              boolean
|        |  |  +--ro rfc5883?              boolean
|        |  |  +--ro rfc5884?              boolean
|        |  |  +--ro rfc5885?              boolean
|        |  +--:(tools-mpls)
|        |  |  +--ro rfc4379?              boolean
|        |  |  +--ro rfc4687?              boolean
|        |  |  +--ro rfc4950?              boolean
|        |  |  +--ro mpls-rfc5884?         boolean
|        |  +--:(tools-pw)
|        |     +--ro rfc5085?              boolean
|        |     +--ro pw_rfc5885?           boolean
|        |     +--ro rfc6423?              boolean
|        |     +--ro rfc6310?              boolean
|        |     +--ro rfc7023?              boolean
|        +--rw oam-layers* [index]
|           +--rw index    uint16
|           +--rw level?   int32
+--:(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
|        +--ro (tools)?
|        |  +--:(tools-empty)
|        |  |  +--ro tools-null?          empty
|        |  +--:(tools-ip)
|        |  |  +--ro rfc792?               boolean
|        |  |  +--ro rfc4443?              boolean
|        |  |  +--ro rfc4884?              boolean
|        |  |  +--ro rfc5837?              boolean
|        |  +--:(tools-bfd)
|        |  |  +--ro rfc5881?              boolean
|        |  |  +--ro rfc5883?              boolean
|        |  |  +--ro rfc5884?              boolean
|        |  |  +--ro rfc5885?              boolean
```

```
|        |  +--:(tools-mpls)
|        |  |  +--ro rfc4379?              boolean
|        |  |  +--ro rfc4687?              boolean
|        |  |  +--ro rfc4950?              boolean
|        |  |  +--ro mpls-rfc5884?         boolean
|        |  +--:(tools-pw)
|        |     +--ro rfc5085?              boolean
|        |     +--ro pw_rfc5885?           boolean
|        |     +--ro rfc6423?              boolean
|        |     +--ro rfc6310?              boolean
|        |     +--ro rfc7023?              boolean
|        +--rw oam-layers* [index]
|           +--rw index    uint16
|           +--rw level?   int32
+--:(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
|        +--ro (tools)?
|        |  +--:(tools-empty)
|        |  |  +--ro tools-null?                 empty
|        |  +--:(tools-ip)
|        |  |  +--ro rfc792?                     boolean
|        |  |  +--ro rfc4443?                    boolean
|        |  |  +--ro rfc4884?                    boolean
|        |  |  +--ro rfc5837?                    boolean
|        |  +--:(tools-bfd)
|        |  |  +--ro rfc5881?                    boolean
|        |  |  +--ro rfc5883?                    boolean
|        |  |  +--ro rfc5884?                    boolean
|        |  |  +--ro rfc5885?                    boolean
|        |  +--:(tools-mpls)
|        |  |  +--ro rfc4379?                    boolean
|        |  |  +--ro rfc4687?                    boolean
|        |  |  +--ro rfc4950?                    boolean
|        |  |  +--ro mpls-rfc5884?               boolean
|        |  +--:(tools-pw)
|        |     +--ro rfc5085?                    boolean
|        |     +--ro pw_rfc5885?                 boolean
|        |     +--ro rfc6423?                    boolean
|        |     +--ro rfc6310?                    boolean
|        |     +--ro rfc7023?                    boolean
```

```
|        +--rw oam-layers* [index]
|           +--rw index    uint16
|           +--rw level?   int32
+--:(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
|        +--ro (tools)?
|        |  +--:(tools-empty)
|        |  |  +--ro tools-null?           empty
|        |  +--:(tools-ip)
|        |  |  +--ro rfc792?               boolean
|        |  |  +--ro rfc4443?              boolean
|        |  |  +--ro rfc4884?              boolean
|        |  |  +--ro rfc5837?              boolean
|        |  +--:(tools-bfd)
|        |  |  +--ro rfc5881?              boolean
|        |  |  +--ro rfc5883?              boolean
|        |  |  +--ro rfc5884?              boolean
|        |  |  +--ro rfc5885?              boolean
|        |  +--:(tools-mpls)
|        |  |  +--ro rfc4379?              boolean
|        |  |  +--ro rfc4687?              boolean
|        |  |  +--ro rfc4950?              boolean
|        |  |  +--ro mpls-rfc5884?         boolean
|        |  +--:(tools-pw)
|        |     +--ro rfc5085?              boolean
|        |     +--ro pw_rfc5885?           boolean
|        |     +--ro rfc6423?              boolean
|        |     +--ro rfc6310?              boolean
|        |     +--ro rfc7023?              boolean
|        +--rw oam-layers* [index]
|           +--rw index    uint16
|           +--rw level?   int32
+--:(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
|        +--ro (tools)?
|        |  +--:(tools-empty)
|        |  |  +--ro tools-null?        empty
|        |  +--:(tools-ip)
|        |  |  +--ro rfc792?            boolean
|        |  |  +--ro rfc4443?           boolean
|        |  |  +--ro rfc4884?           boolean
|        |  |  +--ro rfc5837?           boolean
|        |  +--:(tools-bfd)
|        |  |  +--ro rfc5881?           boolean
|        |  |  +--ro rfc5883?           boolean
|        |  |  +--ro rfc5884?           boolean
|        |  |  +--ro rfc5885?           boolean
|        |  +--:(tools-mpls)
|        |  |  +--ro rfc4379?           boolean
|        |  |  +--ro rfc4687?           boolean
|        |  |  +--ro rfc4950?           boolean
|        |  |  +--ro mpls-rfc5884?      boolean
|        |  +--:(tools-pw)
|        |     +--ro rfc5085?           boolean
|        |     +--ro pw_rfc5885?        boolean
|        |     +--ro rfc6423?           boolean
|        |     +--ro rfc6310?           boolean
|        |     +--ro rfc7023?           boolean
|        +--rw oam-layers* [index]
|           +--rw index    uint16
|           +--rw level?   int32
|
+--:(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
         +--ro (tools)?
         |  +--:(tools-empty)
         |  |  +--ro tools-null?          empty
         |  +--:(tools-ip)
         |  |  +--ro rfc792?              boolean
         |  |  +--ro rfc4443?             boolean
         |  |  +--ro rfc4884?             boolean
         |  |  +--ro rfc5837?             boolean
         |  +--:(tools-bfd)
```

```
               |  |  +--ro rfc5881?              boolean
               |  |  +--ro rfc5883?              boolean
               |  |  +--ro rfc5884?              boolean
               |  |  +--ro rfc5885?              boolean
               |  +--:(tools-mpls)
               |  |  +--ro rfc4379?              boolean
               |  |  +--ro rfc4687?              boolean
               |  |  +--ro rfc4950?              boolean
               |  |  +--ro mpls-rfc5884?         boolean
               |  +--:(tools-pw)
               |     +--ro rfc5085?              boolean
               |     +--ro pw_rfc5885?           boolean
               |     +--ro rfc6423?              boolean
               |     +--ro rfc6310?              boolean
               |     +--ro rfc7023?              boolean
               +--rw oam-layers* [index]
                  +--rw index    uint16
                  +--rw level?   int32
```

data hierarchy of OAM

4. OAM YANG Module

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

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

    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 2016-10-31 {
      description
        "Initial revision. - 07 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 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.";
      } // typedef IP-Multicast-Group-Address

    identity fec-types {

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

    typedef fec-type {
      type identityref {
        base fec-types;
      }
      description "Target FEC type.";
    }

    typedef oam-counter32 {
      type yang:zero-based-counter32;
      description
        "defines 32 bit counter for OAM";
    }

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

    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";
    }

    /* 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 received packet count.";
        }
        leaf tx-packet-count {
          type uint32;
          description "Total transmitted packet count.";
        }
        leaf rx-bad-packet {

          type uint32;
          description "Total received bad packet.";
        }
        leaf tx-packet-failed {
          type uint32;
          description "Total send 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 packets that went through a path as inten
ded.";
        }
        leaf failed-count {
          type uint32;
          description "Total number of packets that went through an unintended p
ath.";
        }
      }
    }

    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-type {
      description
        "Test point address type";
    } //base identity

    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 src-dst-address-type {
      base tp-address-type;
      description
        "Source/Dest address type";
    }

    identity fec-address-type {
```

```
      base tp-address-type;
      description
        "FEC address type";
    }

    identity tlv-address-type {
      base tp-address-type;
      description
        "TLV address 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-address-type-value {
        type identityref {
          base tp-address-type;
        }
        description "Test point address type.";
      }

      choice tp-address {

        case mac-address {
          when "'tp-address-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-address-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-address-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 src-dst-address {
          when "’tp-address-type-value’ = ’src-dst-address-type’" {
            description "Src dest address type for BFD";
          }
          leaf src-ip-address {
            type inet:ip-address;
                    description
                    "source ip address.";
          }
          leaf dst-ip-address {
            type inet:ip-address;
                    description

                    "destination ip address.";
          }
          leaf Interface {
            type if:interface-ref;
                    description
                    "interface.";
          }
        }
        case fec {
          when "’tp-address-type-value’ = ’fec-address-type’" {
            description "FEC address type";
          }
          leaf fec-type {
            type fec-type;
                    description
                    "fec type.";
          }
          choice fec-value {
                   description
                   "fec 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 tlv-address {
             when "’tp-address-type-value’ = ’tlv-address-type’" {
               description "TLV address type";
             }
              leaf tlv-type {
                type int16;
                description
                  "Type of MEP-ID";
              }
              leaf tlv-len {
                type int16;
                description
                  "Length of MEP-ID value";
              }
              leaf tlv-value {
                type binary {
                  length "12..255";
              }
              description
                "Value please refer RFC6428 (Figure 4,5,6).";
              }
```

```
        description
          "MEP-ID";
        }
        case system-info {
          when "’tp-address-type-value’ = ’system-id-address-type’" {
            description "System id address type";
          }
          leaf system-id {
            type inet:uri;
            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.";
         }
         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.";
      choice tools {
           default tools-empty;
                   config false;
           description
           "choice of test point tools.
            Empty tools means based on Test Point it's implicit
            all OAM tools are present and no further configuration
            is supported.";

         case tools-empty {
           description
             "this is a placeholder when oam toolset is not needed.";
           leaf tools-null {
             type empty;
             description
               "there is no oam toolset defined.";
           }
         }
         case tools-ip{
           description
             "Oam Toolset for Ip";
           leaf rfc792 {
             type boolean;
             description
               "rfc792 (icmpv4) supported.";
           }
           leaf rfc4443 {
             type boolean;
             description
               "rfc4443 supported.";
           }
           leaf rfc4884 {
             type boolean;
                     description
                     "rfc4884 supported.";
           }
           leaf rfc5837 {
             type boolean;
                     description
                     "rfc5837 supported.";
           }
```

```
        }
        case tools-bfd {
          leaf rfc5881 {
            type boolean;
                    description
                    "rfc5881 supported.";
          }
          leaf rfc5883 {
            type boolean;
                    description
                    "rfc5883 supported.";
          }
          leaf rfc5884 {
            type boolean;
                    description

                    "rfc5884 supported.";
          }
          leaf rfc5885 {
            type boolean;
                    description
                    "rfc5885 supported.";
          }
        }
        case tools-mpls {
          description
            "Oam Toolset for mpls";
          leaf rfc4379 {
            type boolean;
                    description
                    "rfc4379 supported.";
          }
          leaf rfc4687 {
            type boolean;
                    description
                    "rfc4687 supported.";
          }
          leaf rfc4950 {
            type boolean;
                    description
                    "rfc4950 supported.";
          }
          leaf mpls-rfc5884 {
            type boolean;
                    description
                    "rfc5884 supported.";
          }
        }
```

```
        case tools-pw {
          description
            "Oam Toolset for pw oam.";
          leaf rfc5085 {
            type boolean;
                    description
                    "rfc5085 supported.";
          }
          leaf pw_rfc5885 {
            type boolean;
                    description
                    "rfc5885 supported.";
          }
          leaf rfc6423 {
            type boolean;
                    description
                    "rfc6423 supported.";
          }
          leaf rfc6310 {
            type boolean;
                    description
                    "rfc6310 supported.";
          }
          leaf rfc7023 {
            type boolean;
                    description
                    "rfc7023 supported.";
          }
        }
      }
    }

    grouping test-point-location-info {
      uses tp-technology;
      uses tp-tools;
      uses connectionless-oam-layers;
      description
        "Test point Location";
    }

    grouping test-point-locations {
      description "Group of test point locations.";
      leaf tp-address-type-value {

        type identityref {
          base tp-address-type;
        }
        description "Test point address type.";
```

```
      }
      choice location-type {
        case ipv4-location-type {
          when "'tp-address-type-value' = 'ipv4-address-type'" {
          description
            "when test point address 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-address-type-value' = 'ipv6-address-type'" {
          description
            "when test point address 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-address-type-value' = 'mac-address-type'" {
            description
              "when test point address 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-address-type-value' = 'tunnel-address-type'" {
            description
              "when test point address is equal to tunnel type.";
          }
          container test-point-tunnel-address-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-address-type-value' = 'ip-prefix-address-type'" {
            description
              "when test point address 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-address-type-value' = 'route-distinguisher-address-type'" {
            description "when test point address 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-address-type-value' = 'group-ip-address-type'" {
            description "when test point address 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-address-type-value' = 'as-number-address-type'" {
            description "when test point address 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-address-type-value' = 'lsp-id-address-type'" {
            description "when test point address 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-address-type-value' = 'system-id-address-type'" {
            description "when test point address 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
            "ACL name that refers to the flow, if any.";
        }
        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-val {
            type yang:date-and-time;
            description "Timestamp value";
          }
          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 app-meta-data {
            type uint32;
            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 oper {
      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;
      }
    }
}
```

YANG module of OAM

<CODE ENDS>

5. CL 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

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 both BFD model and "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. 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 LIME BFD model.

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;
  }
}
```

5.1.2. test point attributes extension

To derive a model for bfd technology, the "ietf-connectionless-oam" model can be extended. Some data nodes for bfd specific parameters can be defined and inserted into the proper "test-point-location"

list. Or some new "location-type" cases can be added to support the some bfd technologies such as "bfd over MPLS-TE", etc.

5.1.2.1. 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 model for some bfd technologies(such as ip single-hop, ip multi-hops, etc), data nodes for bfd specific details can be defined and inserted into corresponding "test-point-locations" list. In this section, we reuse some groupings which are defined in [I-D.ietf-bfd-yang] to derive following example.

The snippet below shows how the "ietf-connectionless-oam" model can be extended to "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;
  }
}
}
```

Edit note: To prevent new attribute to be defined in the CL extension model,[I-D.ietf-bfd-yang] should define the attributes that are used by CL model as grouping, then CL extension model can reuse these grouping. We will make accordingly change when these grouping are available in the BFD model.

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

5.1.2.2. Add new location-type cases

In the "ietf-connectionless-oam" model, If no a proper "test-point-locations" can be extended, new "location-type" cases can be defined and inserted into the "location-type" choice node.

Therefore, the model user can flexible add "location-type" to support other kinds 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] to derive the following example.

The snippet below shows how the "ietf-connectionless-oam" model can be extended to "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.

5.2. LSP ping extension

The following sections shows how the "ietf-connectionless-oam" model can be extended to cover 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 both LSP Ping model and "ietf-connectioless-oam" as basis and augment the "ietf-connectionless-oam" model with LSP Ping specific details. 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 LIME LSP ping model.

The snippet below depicts an example of augmenting "lsp-ping" 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 lsp-ping{
   type string;
  }
}
```

5.2.2. test point attributes extension

In order to derive a model for lsp-ping, the "ietf-connectionless-oam" model can be extended. Some data nodes for lsp-ping specific parameters can be defined and inserted into the proper "test-point-location" list.

User can reuse the attributes or groupings which are defined in [I-D.draft-zheng-mpls-lsp-ping-yang-cfg] to derive the "lime lsp ping" model.

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

```
          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.";
          ......
        }
```

Edit note: To prevent new attribute to be defined in the CL extension model,[I-D.draft-zheng-mpls-lsp-ping-yang-cfg] should define the attributes that are used by CL model as grouping, then CL extension model can reuse these grouping. We will make accordingly change when these grouping are available in the LSP ping model.

6. Security Considerations

TBD.

7. 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-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: goam reference: RFC XXXX

8. Normative References

[I-D.draft-ietf-i2rs-yang-network-topo]
Clemm, A., Medved, J., Tkacik, T., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A YANG Data Model for Network Topologies", I-D draft-ietf-i2rs-yang-network-topo-05, July 2016.

[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-03, 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-03 (work in progress), July 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>.

[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>.

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

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: May 17, 2018

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

November 13, 2017

Generic YANG Data Model for the Management of Operations, Administration, and Maintenance (OAM) Protocols that use Connectionless Communications

draft-ietf-lime-yang-connectionless-oam-18

Abstract

This document presents a base YANG Data model for the management of Operations Administration, and Maintenance (OAM) protocols that use Connectionless Communications. The data model is defined using the YANG, as specified in RFC7950 data modeling language. It provides a technology-independent abstraction of key OAM constructs for OAM protocols that use connectionless communication. The base model presented here can be extended to include technology-specific details.

There are two key benefits of this approach: First, it leads to uniformity between OAM protocols. And second, it support both nested OAM workflows (i.e., performing OAM functions at different or same levels through a unified interface) as well as 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 https://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 May 17, 2018.

Table of Contents

1. Introduction

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

1. Monitor network communications (i.e., Reachability Verification, Continuity Check)

2. Troubleshoot failures (i.e., Fault verification and Localization)

3. Monitor service-level agreements and performance (i.e., Performance Management)

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

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

The different sets of OAM tools may support both connection-oriented technologies or connectionless technologies. In connection-oriented technologies, a connection is established prior to the transmission of data. After the connection is established, no additional control information such as signaling or operations and maintenance information is required to transmit the actual user data. In connectionless technologies, data is typically sent between communicating end points without prior arrangement, but control information is required to identify the destination (e.g., [G.800] and [RFC7276]). The YANG Data model for OAM protocols using connection-oriented communications is specified in [I-D.ietf-lime-yang-connection-oriented-oam-model].

This document defines a base YANG Data model for OAM protocols that use connectionless communications. The data model is defined using the YANG [RFC7950] data modeling language. This generic YANG model for connectionless OAM includes only configuration and state data. It can be used in conjunction with data retrieval method model described in [I-D.ietf-lime-yang-connectionless-oam-methods], which focuses on the data retrieval procedures such as RPC, or it can be used independently of this data retrieval method model.

2. Conventions used in this document

The following terms are defined in [RFC6241] and are used in this specification:

o client

- configuration data
- server
- state data

The following terms are defined in [RFC7950] and are used in this specification:

- augment
- data model
- data node

The terminology for describing YANG data models is found in [RFC7950].

2.1. Abbreviations

BFD - Bidirectional Forwarding Detection [RFC5880].

RPC - Remote Procedure Call [RFC1831].

DSCP - Differentiated Services Code Point.

VRF - Virtual Routing and Forwarding [RFC 4382].

OWAMP - One-Way Active Measurement Protocol [RFC 4656].

TWAMP - Two-Way Active Measurement Protocol [RFC 5357].

AS - Autonomous System.

LSP - Label Switched Path.

TE - Traffic Engineering.

MPLS - Multiprotocol Label Switching.

NI - Network Instance.

PTP - Precision Time Protocol [IEEE.1588].

NTP - Network Time Protocol [RFC5905].

2.2. Terminology

MAC - Media Access Control.

MAC address - Address for the data-link layer interface.

TP - Test Point. The TP is a functional entity that is defined at a node in the network and can initiate and/or react to OAM diagnostic tests. This document focuses on the data-plane functionality of TPs.

RPC Operation - A specific Remote Procedure Call.

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

3. Overview of the Connectionless OAM Model

The YANG data model for OAM protocols that use Connectionless Communications has been split into two modules:

- The ietf-lime-common-types.yang module provides common definitions such as Time-related data types and Timestamp-related data types.
- The ietf-connectionless-oam.yang module defines technology-independent abstraction of key OAM constructs for OAM protocols that use connectionless communication.

The ietf-connectionless-oam module augments the "/networks/network/node" path defined in the ietf-network module [I-D.ietf-i2rs-yang-network-topo] with 'test-point-locations' grouping defined in Section 3.5. The network node in "/networks/network/node" path are used to describe the network hierarchies and the inventory of nodes contained in a network.

Under the 'test-point-locations' grouping, each test point location is chosen based on 'tp-location-type' leaf which when chosen, leads to a container that includes a list of 'test-poit-locations'.

Each 'test-point-locations' list includes a 'test-point-location-info' grouping. The 'test-point-location-info' grouping includes:

- 'tp-technology' grouping,
- 'tp-tools' grouping, and
- 'connectionless-oam-tps' grouping.

The groupings of 'tp-address' and 'tp-address-ni' are kept out of 'test-point-location-info' grouping to make it addressing agnostic and allow varied composition. Depending upon the choice of the 'tp-location-type' (determined by the 'tp-address-ni'), the containers differ in its composition of 'test-point-locations' while the 'test-point-location-info', is a common aspect of every 'test-point-locations'.

The 'tp-address-ni' grouping is used to describe the corresponding network instance. The 'tp-technology' grouping indicate OAM technology details. The 'connectionless-oam-tps' grouping is used to describe the relationship of one test point with other test points. The 'tp-tools' grouping describe the OAM tools supported.

In addition, at the top of the model, there is an 'cc-oper-data' container for session statistics. A grouping is also defined for common session statistics and these are only applicable for proactive (see Section 3.2) OAM sessions.

3.1. TP Address

With connectionless OAM protocols, the TP address can be one of the following types:

- MAC address [RFC6136] at the data-link layer for TPs
- IPv4 or IPv6 address at IP layer for TPs
- TP-attribute identifying a TP associated with an application layer function
- Router-id to represent the device or node, which is commonly used to identify nodes in routing and other control plane protocols [I-D.ietf-rtgwg-routing-types].

To define a forwarding treatment of a test packet, the 'tp-address' grouping needs to be associated with additional parameters, e.g., DSCP for IP or Traffic Classic [RFC5462] for MPLS. In the generic connectionless OAM YANG model, these parameters are not explicitly 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. Proactive OAM refers to OAM actions which are carried out continuously to permit proactive reporting of faults. The proactive OAM method requires persistent

configuration. On-demand OAM refers to OAM actions which are initiated via manual intervention for a limited time to carry out specific diagnostics. The on-demand OAM method requires only transient configuration (e.g., [RFC7276] and [G.8013]). In connectionless OAM, tbe 'session-type' grouping 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 a specific OAM technology. For example, to fulfill the ICMP PING configuration, the "../coam:continuity-check" leaf should be set to "true", and then the lime base model should be augmented with ICMP PING specific details.

3.3. OAM neighboring test points

Given tbat typical network communication stacks have a multi-layer architecture, the set of associated OAM protocols has also a multi-layer structure; each communication layer in the stack may have its own OAM protocol [RFC7276] that may also be linked to a specific administrative domain. Management of these OAM protocols will necessitate associated test points in the nodes accessible by appropriate management domains. Accordingly, a given network interface may actually present several test points.

Each OAM test point may have an associated list of neighboring test points in other layers up and down the protocol stack for the same interface and are therefore related to the current test point. This allows users to easily navigate between related neighboring layers to efficiently troubleshoot a defect. In this model, the 'position' leaf defines the relative position of the neighboring test point corresponding to the current test point, and is provided to allow correlation of faults at different locations. If there is one neighboring test point placed before the current test point, the 'position' leaf is set to -1. If there is one neighboring test point placed after the current test point, the 'position' leaf is set to 1. If there is no neighboring test point placed before or after the current test point, the 'position' leaf is set to 0.

```
                list oam-neighboring-tps {
                  key "index";
                  leaf index {
                     type uint16 {
                        range "0..65535";
                     }
                    description
                     "Index of a list of neighboring test points
                      in layers up and down the stack for
                      the same interface that are related to the
                      current test point.";
                  }
                  leaf position {
                      type int8 {
                           range "-1..1";
                      }
                      description
                        "The relative position
                        of neighboring test point
                        corresponding to the current
                        test point";
                  }

                  description
                     "List of related neighboring test points in adjacent
                     layers up and down the stack for the same interface
                     that are related to the current test point.";

              }
```

3.4. Test Point Locations Information

This is a generic grouping for Test Point Locations Information (i.e., test-point-location-info grouping). It Provide details of Test Point Location using 'tp-technology','tp-tools' grouping, 'oam-neighboring-tps' grouping, all of which are defined above.

3.5. Test Point Locations

This is a generic grouping for Test Point Locations. 'tp-location-type 'leaf 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 network instance name (e.g., VRF instance name) if required.

3.6. Path Discovery Data

This is a generic grouping for the path discovery data model that can be retrieved by any data retrieval methods including RPC operations. Path discovery data output from methods, includes 'src-test-point' container, 'dst-test-point' container, 'sequence-number'leaf, 'hop-cnt' leaf, 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' items which includes information such as 'timestamp' grouping, 'ingress-intf-name', 'egress-intf-name' 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. Note that a set of retrieval methods are defined in [I-D.ietf-lime-yang-connectionless-oam-methods].

3.7. Continuity Check Data

This is a generic grouping for the continuity check data model that can be retrieved by any data retrieval methods including RPC operations. Continuity check data output from methods, includes 'src-test- point' container, 'dst-test-point' container, 'sequence-number'leaf, 'hop-cnt' leaf 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 a set of 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-session-statistics-data {continuity-check}?
       +--ro cc-session-statistics* [type]
          +--ro type                          identityref
          +--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?                identityref
      +--rw ipv4-location-type
      |  +--rw test-point-ipv4-location-list
      |     +--rw test-point-locations* [ipv4-location ni]
      |        +--rw ipv4-location          inet:ipv4-address
      |        +--rw ni                     routing-instance-ref
      |        +--rw (technology)?
      |        |  +--:(technology-null)
      |        |     +--rw tech-null?             empty
      |        +--rw tp-tools
      |        |  +--rw continuity-check    boolean
      |        |  +--rw path-discovery      boolean
      |        +--rw root?                  <anydata>
      |        +--rw oam-neighboring-tps* [index]
      |           +--rw index                     uint16
      |           +--rw position?                 int8
      |           +--rw (tp-location)?
      |              +--:(mac-address)
      |              |  +--rw mac-address-location?    yang:mac-address
      |              +--:(ipv4-address)
      |              |  +--rw ipv4-address-location?   inet:ipv4-address
      |              +--:(ipv6-address)
      |              |  +--rw ipv6-address-location?   inet:ipv6-address
      |              +--:(as-number)
      |              |  +--rw as-number-location?      inet:as-number
      |              +--:(router-id)
      |                 +--rw router-id-location?      rt:router-id
      +--rw ipv6-location-type
      |  +--rw test-point-ipv6-location-list
      |     +--rw test-point-locations* [ipv6-location ni]
      |        +--rw ipv6-location          inet:ipv6-address
      |        +--rw ni                     routing-instance-ref
      |        +--rw (technology)?
      |        |  +--:(technology-null)
      |        |     +--rw tech-null?             empty
      |        +--rw tp-tools
      |        |  +--rw continuity-check    boolean
      |        |  +--rw path-discovery      boolean
      |        +--rw root?                  <anydata>
      |        +--rw oam-neighboring-tps* [index]
      |           +--rw index                     uint16
      |           +--rw position?                 int8
      |           +--rw (tp-location)?
      |              +--:(mac-address)
      |              |  +--rw mac-address-location?    yang:mac-address
      |              +--:(ipv4-address)
```

```
|              |  +--rw ipv4-address-location?   inet:ipv4-address
|              +--:(ipv6-address)
|              |  +--rw ipv6-address-location?   inet:ipv6-address
|              +--:(as-number)
|              |  +--rw as-number-location?      inet:as-number
|              +--:(router-id)
|                 +--rw router-id-location?      rt:router-id
+--rw 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
|        +--rw tp-tools
|        |  +--rw continuity-check    boolean
|        |  +--rw path-discovery      boolean
|        +--rw root?                  <anydata>
|        +--rw oam-neighboring-tps* [index]
|           +--rw index                    uint16
|           +--rw position?                int8
|           +--rw (tp-location)?
|              +--:(mac-address)
|              |  +--rw mac-address-location?    yang:mac-address
|              +--:(ipv4-address)
|              |  +--rw ipv4-address-location?   inet:ipv4-address
|              +--:(ipv6-address)
|              |  +--rw ipv6-address-location?   inet:ipv6-address
|              +--:(as-number)
|              |  +--rw as-number-location?      inet:as-number
|              +--:(router-id)
|                 +--rw router-id-location?      rt:router-id
+--rw 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 ni?                    routing-instance-ref
|        +--rw (technology)?
|        |  +--:(technology-null)
|        |     +--rw tech-null?             empty
|        +--rw tp-tools
|        |  +--rw continuity-check    boolean
|        |  +--rw path-discovery      boolean
|        +--rw root?                  <anydata>
|        +--rw oam-neighboring-tps* [index]
|           +--rw index                    uint16
|           +--rw position?                int8
|           +--rw (tp-location)?
```

```
      |              +--:(mac-address)
      |              |  +--rw mac-address-location?    yang:mac-address
      |              +--:(ipv4-address)
      |              |  +--rw ipv4-address-location?   inet:ipv4-address
      |              +--:(ipv6-address)
      |              |  +--rw ipv6-address-location?   inet:ipv6-address
      |              +--:(as-number)
      |              |  +--rw as-number-location?      inet:as-number
      |              +--:(router-id)
      |                 +--rw router-id-location?      rt:router-id
      +--rw group-router-id-location-type
         +--rw test-point-system-info-location-list
            +--rw test-point-locations* [router-id-location]
               +--rw router-id-location     rt:router-id
               +--rw ni?                    routing-instance-ref
               +--rw (technology)?
               |  +--:(technology-null)
               |     +--rw tech-null?             empty
               +--rw tp-tools
               |  +--rw continuity-check    boolean
               |  +--rw path-discovery      boolean
               +--rw root?                  <anydata>
               +--rw oam-neighboring-tps* [index]
                  +--rw index                     uint16
                  +--rw position?                 int8
                  +--rw (tp-location)?
                     +--:(mac-address)
                     |  +--rw mac-address-location?    yang:mac-address
                     +--:(ipv4-address)
                     |  +--rw ipv4-address-location?   inet:ipv4-address
                     +--:(ipv6-address)
                     |  +--rw ipv6-address-location?   inet:ipv6-address
                     +--:(as-number)
                     |  +--rw as-number-location?      inet:as-number
                     +--:(router-id)
                        +--rw router-id-location?      rt:router-id
```

4. LIME Time Types YANG Module

<CODE BEGINS> file "ietf-lime-time-types@2017-09-06.yang"

```
 module ietf-lime-time-types {
     yang-version 1.1;
     namespace "urn:ietf:params:xml:ns:yang:ietf-lime-time-types";
     prefix "lime";

     organization
       "IETF Layer Independent OAM Management (LIME)
```

```
       Working Group";

    contact
      "WG Web:   <https://datatracker.ietf.org/wg/lime>
       WG List:  <mailto:lmap@ietf.org>

       Editor:   Qin Wu
                 <bill.wu@huawei.com>";

    description
      "This module provides time related definitions used by the data
       models written for Layer Independent OAM Management (LIME).
       This module defines identities but no schema tree elements.";

    revision "2017-09-06" {
      description
        "Initial version";
      reference
        "RFC xxxx: A YANG Data Model for OAM Protocols that use Connectionless
          Communications";
    }

  /*** Collection of common types related to time ***/
    /*** Time unit identity ***/
  identity time-unit-type {
    description
      "Time unit type";
  }
  identity hours {
    base time-unit-type;
    description
      "Time unit in Hours";
  }
  identity minutes {
    base time-unit-type;
    description
      "Time unit in Minutes";
  }
  identity seconds {
    base time-unit-type;
    description
      "Time unit in Seconds";
  }
  identity milliseconds {
    base time-unit-type;
    description
      "Time unit in Milliseconds";
  }
```

```
  identity microseconds {
    base time-unit-type;
    description
      "Time unit in Microseconds";
  }
  identity nanoseconds {
    base time-unit-type;
    description
      "Time unit in Nanoseconds";
  }
   /*** Timestamp format Identity ***/
 identity timestamp-type {
  description
   "Base identity for Timestamp Type.";
 }
 identity truncated-ptp {
  base timestamp-type;
  description
   "Identity for 64bit short format PTP timestamp.";
 }
 identity truncated-ntp {
  base timestamp-type;
  description
   "Identity for 32bit short format NTP timestamp.";
 }
 identity ntp64 {
  base timestamp-type;
  description
   "Identity for 64bit NTP timestamp.";
 }
 identity icmp {
  base timestamp-type;
  description
   "Identity for 32bit ICMP timestamp.";
 }
}
```

<CODE ENDS>

5. Connectionless OAM YANG Module

This module imports Core YANG Derived Types definition (i.e.,ietf-yang-types.yang module) and Internet-Specific Derived Types definitions (ietf-inet-types.yang module) from [RFC6991], ietf-routing-types.yang module from [I-D.ietf-rtgwg-routing-types], ietf-interfaces.yang module from [RFC7223],ietf-network.yang module from [I-D.ietf-i2rs-yang-network-topo],ietf-network-instance.yang module

from [I-D.ietf-rtgwg-ni-model] and the ietf-lime-common-types.yang module in Section 4.

<CODE BEGINS> file "ietf-connectionless-oam@2017-09-06.yang"

```
module ietf-connectionless-oam {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-connectionless-oam";
  prefix cl-oam;
  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;
  }
  import ietf-routing-types {
   prefix rt;
  }
  import ietf-lime-time-types {
   prefix lime;
  }
  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, and statistics for OAM protocols using
     connectionless communications, described in a
     protocol independent manner. It is assumed that each
     protocol maps corresponding abstracts to its native
     format. Each protocol mayextend the YANG model defined
```

```
     here to include protocol specific extensions.";
  revision 2017-09-06 {
    description
      "Base model for Connectionless
       Operations, Administration,
       and Maintenance (OAM)";
    reference
      "RFC XXXX: Connectionless
       Operations, Administration, and
       Maintenance (OAM) YANG Data Model";
  }
  feature connectionless {
    description
      "This feature indicates that OAM solution is connectionless.";
  }
  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 operation 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 operation model for
       path discovery command.";
  }
  feature ptp-long-format {
    description
      "This feature indicates that timestamp is PTP long format.";
  }
  feature ntp-short-format {
    description
      "This feature indicates that timestamp is NTP short format.";
  }
  feature icmp-timestamp {
    description
      "This feature indicates that timestamp is ICMP timestamp.";
  }
  identity traffic-type {
   description
   "This is base identity of traffic type
```

```
   which include IPv4 and IPv6, etc.";
  }
  identity ipv4 {
  base traffic-type;
  description
   "identity for IPv4 traffic type.";
 }
  identity ipv6 {
  base traffic-type;
   description
   "identity for IPv4 traffic type.";
 }
  identity address-attribute-types {
    description
      "This is base identity of address
       attribute types which are Generic
       IPv4/IPv6 Prefix, BGP Labeled
       IPv4/IPv6 Prefix, Tunnel ID,
       PW ID, VPLS VE ID, etc. (see RFC8029
       for details.)";
  }
  typedef address-attribute-type {
    type identityref {
      base address-attribute-types;
    }
    description
      "Target address attribute type.";
  }
  typedef percentage {
      type decimal64 {
         fraction-digits 5;
         range "0..100";
     }
   description "Percentage.";
   }
  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.";
  }
  grouping cc-session-statistics {
    description
      "Grouping for session statistics.";
    container cc-session-statistics {
      description
```

```
        "cc session counters";
      leaf session-count {
        type uint32;
        default "0";
        description
          "Number of Continuity Check sessions.
          A value of zero indicates that no session
          count is sent.";
      }
      leaf session-up-count {
        type uint32;
        default "0";
        description
          "Number of sessions which are up.
          A value of zero indicates that no up
          session count is sent.";
      }
      leaf session-down-count {
        type uint32;
        default "0";
        description
          "Number of sessions which are down.
         A value of zero indicates that no down
         session count is sent.";
      }
      leaf session-admin-down-count {
        type uint32;
       default "0";
        description
          "Number of sessions which are admin-down.
          A value of zero indicates that no admin
          down session count is sent.";
      }
    }
  }
  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{
            range "0..4294967295";
        }
        default "0";
        description
```

```
          "Total number of received OAM packet count.
           The value of count will be set to zero (0)
           on creation and will thereafter increase
           monotonically until it reaches a maximum value
           of 2^32-1 (4294967295 decimal), when it wraps
           around and starts increasing again from zero.";
      }
      leaf tx-packet-count {
        type uint32{
        range "0..4294967295";
        }
       default "0";
        description
          "Total number of transmitted OAM packet count.
           The value of count will be set to zero (0)
           on creation and will thereafter increase
           monotonically until it reaches a maximum value
           of 2^32-1 (4294967295 decimal), when it wraps
           around and starts increasing again from zero.";
      }
      leaf rx-bad-packet {
        type uint32 {
        range "0..4294967295";
        }
        default "0";
        description
          "Total number of received bad OAM packet.
           The value of count will be set to zero (0)
           on creation and will thereafter increase
           monotonically until it reaches a maximum value
           of 2^32-1 (4294967295 decimal), when it wraps
           around and starts increasing again from zero.";
      }
      leaf tx-packet-failed {
        type uint32 {
        range "0..4294967295";
        }
        default "0";
        description
          "Total number of failed sending OAM packet.
          The value of count will be set to zero (0)
           on creation and will thereafter increase
           monotonically until it reaches a maximum value
           of 2^32-1 (4294967295 decimal), when it wraps
           around and starts increasing again from zero.";
      }
    }
  }
```

```
  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 {
        range "0..4294967295";
        }
        default "0";
        description
          "Total Continuity Check sessions down count.
           The value of count will be set to zero (0)
           on creation and will thereafter increase
           monotonically until it reaches a maximum value
           of 2^32-1 (4294967295 decimal), when it wraps
           around and starts increasing again from zero.";
      }
      leaf admin-down-count {
        type uint32 {
        range "0..4294967295";
        }
        default "0";
        description
          "Total Continuity Check sessions admin down count.
           The value of count will be set to zero (0)
           on creation and will thereafter increase
           monotonically until it reaches a maximum value
           of 2^32-1 (4294967295 decimal), when it wraps
           around and starts increasing again from zero.";
      }
```

```
      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-loss-count {
        type uint32 {
        range "0..4294967295";
        }
       default "0";
        description
          "Total received packet drops count.
           The value of count will be set to zero (0)
           on creation and will thereafter increase
           monotonically until it reaches a maximum value
           of 2^32-1 (4294967295 decimal), when it wraps
           around and starts increasing again from zero.";
      }
          leaf loss-ratio{
           type percentage;
                description
                 "Loss ratio of the packets. Express as percentage
                 of packets lost with respect to packets sent.";
                }
      leaf packet-reorder-count {
        type uint32 {
        range "0..4294967295";
        }
        default "0";
        description
          "Total received packet reordered count.
           The value of count will be set to zero (0)
           on creation and will thereafter increase
           monotonically until it reaches a maximum value
           of 2^32-1 (4294967295 decimal), when it wraps
           around and starts increasing again from zero.";
      }
      leaf packets-out-of-seq-count {
        type uint32 {
        range "0..4294967295";
        }
        description
          "Total received out of sequence count.
           The value of count will be set to zero (0)
```

```
            on creation and will thereafter increase
            monotonically until it reaches a maximum value
            of 2^32-1 (4294967295 decimal), when it wraps
            around and starts increasing again from zero..";
        }
        leaf packets-dup-count {
          type uint32 {
          range "0..4294967295";
          }
          description
            "Total received packet duplicates count.
            The value of count will be set to zero (0)
             on creation and will thereafter increase
             monotonically until it reaches a maximum value
             of 2^32-1 (4294967295 decimal), when it wraps
             around and starts increasing again from zero.";
        }
      }
    }
    grouping session-delay-statistics {
      description
        "Grouping for per session delay statistics";
      container session-delay-statistics {
        description
          "Session delay summarised information. By default,
           one way measurement protocol (e.g., OWAMP) is used
           to measure delay. When two way measurement protocol
           (e.g., TWAMP) is used instead, it can be indicated
           using and protocol-id defined in RPC operation of
           draft-ietf-lime-yang-connectionless-oam-methods, i.e.,
           set protocol-id as OWAMP. Note that only one measurement
           protocol for delay is specified for interoperability reason.";
        leaf time-unit-value {
          type identityref {
            base lime:time-unit-type;
          }
          default lime:milliseconds;
          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. By default,
         jitter is measured using IP Packet Delay Variation
         (IPDV) as defined in RFC3393. When the other measurement
         method is used instead (e.g., Packet Delay Variation used
         in Y.1540, it can be indicated using protocol-id-meta-data
         defined in RPC operation of
         draft-ietf-lime-yang-connectionless-oam-methods. Note that
         only one measurement method for jitter is specified
         for interoperability reason.";
      leaf unit-value {
        type identityref {
          base lime:time-unit-type;
        }
       default lime:milliseconds;
        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 {
        range "0..4294967295";
        }
        description
          "Total number of OAM packets that
           went through a path as intended.
           The value of count will be set to zero (0)
           on creation and will thereafter increase
           monotonically until it reaches a maximum value
           of 2^32-1 (4294967295 decimal), when it wraps
           around and starts increasing again from zero.";
      }
      leaf failed-count {
        type uint32 {
        range "0..4294967295";
        }
        description
          "Total number of OAM packets that
           went through an unintended path.
          The value of count will be set to zero (0)
           on creation and will thereafter increase
           monotonically until it reaches a maximum value
           of 2^32-1 (4294967295 decimal), when it wraps
           around and starts increasing again from zero.";
      }
    }
  }
  grouping session-type {
    description
      "This object indicates which kind
      of activation will be used by the current
      session.";
    leaf session-type {
      type enumeration {
        enum "proactive" {
          description
            "The current session is proactive session.";
        }
        enum "on-demand" {
          description
            "The current session is on-demand session.";
```

```
        }
      }
      default "on-demand";
      description
        "Indicate which kind of activation will be used
         by the current session";
    }
  }
  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 router-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 route-distinguisher-address-type {
    base tp-address-technology-type;
    description
      "Route Distinguisher address type";
  }
```

```
  grouping tp-address {
    leaf tp-location-type {
      type identityref {
        base tp-address-technology-type;
      }
      mandatory true;
      description
        "Test point address type.";
    }
      container mac-address {
        when "derived-from-or-self(../tp-location-type,"+
       "'cl-oam:mac-address-type')" {
          description
            "MAC address type";
        }
        leaf mac-address {
          type yang:mac-address;
          mandatory true;
          description
            "MAC Address";
        }
        description
          "MAC Address based TP Addressing.";
      }
      container ipv4-address {
        when "derived-from-or-self(../tp-location-type,"+
        "'cl-oam:ipv4-address-type')" {
          description
            "IPv4 address type";
        }
        leaf ipv4-address {
          type inet:ipv4-address;
          mandatory true;

          description
            "IPv4 Address";
        }
        description
          "IP Address based TP Addressing.";
      }
      container ipv6-address {
        when "derived-from-or-self(../tp-location-type,"+
        "'cl-oam:ipv6-address-type')" {
          description
            "IPv6 address type";
        }
        leaf ipv6-address {
```

```
          type inet:ipv6-address;
          mandatory true;
          description
            "IPv6 Address";
        }
        description
          "ipv6 Address based TP Addressing.";
      }
      container tp-attribute {
        when "derived-from-or-self(../tp-location-type,"+
        "'cl-oam: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
                "Generic IPv4/IPv6 prefix. See Section 3.2.13 and
                Section 3.2.14 of RFC8029.";
              reference
                "RFC 8029 :Detecting Multi-Protocol Label
                Switched (MPLS) Data Plane Failures";
            }
          }
          case bgp {
            leaf bgp {
              type inet:ip-prefix;
              description
               "BGP Labeled IPv4/IPv6 Prefix. See section
               3.2.11 and section 3.2.12 of RFC8029 for details. ";
              reference
                "RFC 8029 :Detecting Multi-Protocol Label
                Switched (MPLS) Data Plane Failures";
            }
          }
          case tunnel {
            leaf tunnel-interface {
              type uint32;
              description
```

```
              "Basic IPv4/IPv6 Tunnel ID. See section 3.2.3
               and Section 3.2.4 of RFC8029 for details.";
            reference
              "RFC 8029 :Detecting Multi-Protocol Label
              Switched (MPLS) Data Plane Failures.";
          }
        }
        case pw {
          leaf remote-pe-address {
            type inet:ip-address;
            description
              "Remote PE address. See section 3.2.8
              of RFC8029 for details.";
              reference
              "RFC 8029 :Detecting Multi-Protocol Label
              Switched (MPLS) Data Plane Failures";
          }
          leaf pw-id {
            type uint32;
            description
              "Pseudowire ID is a non-zero 32-bit ID. See section
               3.2.8 and Section 3.2.9 for details.";
            reference
              "RFC 8029 :Detecting Multi-Protocol Label
              Switched (MPLS) Data Plane Failures";
          }
        }
        case vpls {
          leaf route-distinguisher {
            type rt:route-distinguisher;
            description
              "Route Distinguisher is an 8 octets identifier
              used to distinguish information about various
              L2VPN advertised by a node.";
            reference
              "RFC 8029 :Detecting Multi-Protocol Label
              Switched (MPLS) Data Plane Failures";
          }
          leaf sender-ve-id {
            type uint16;
            description
              "Sender's VE ID. The VE ID (VPLS Edge Identifier)
               is a 2-octet identifier.";
            reference
              "RFC 8029 :Detecting Multi-Protocol Label
              Switched (MPLS) Data Plane Failures";
          }
          leaf receiver-ve-id {
```

```
              type uint16;
              description
                "Receiver's VE ID. The VE ID (VPLS Edge Identifier)
                 is a 2-octet identifier.";
              reference
                "RFC 8029 :Detecting Multi-Protocol Label
                Switched (MPLS) Data Plane Failures";
            }
          }
          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 inet:ip-address;
                  description
                    "Group ip address.";
                }
              }
              case vpn {
                leaf as-number {
                  type inet:as-number;
                  description
                  "The AS number represents autonomous system
                  numbers which identify an Autonomous System.";
                }
              }
              case global-id {
                leaf lsp-id {
                  type string;
                  description
                    "LSP ID is an identifier of a LSP
                     within a MPLS network.";
                  reference
                  "RFC 8029 :Detecting Multi-Protocol Label
                   Switched (MPLS) Data Plane Failures";
                }
              }
            }
          }
        }
        description
```

```
          "Test Point Attribute Container";
        }
        container system-info {
          when "derived-from-or-self(../tp-location-type,"+
         "'cl-oam:router-id-address-type')" {
            description
              "System id address type";
          }
          leaf router-id {
            type rt:router-id;
            description
              "Router ID assigned to this node.";
          }
          description
             "Router ID container.";
        }
      description
        "TP Address";
    }
    grouping tp-address-ni {
      description
        "Test point address with VRF.";
      leaf ni {
        type routing-instance-ref;
        description
          "The ni is used to describe virtual resource partitioning
          that may be present on a network device. Example of common
          industry terms for virtual resource partitioning is VRF
          instance.";
      }
      uses tp-address;
    }
    grouping connectionless-oam-tps {
      list oam-neighboring-tps {
        key "index";
        leaf index {
          type uint16{
          range "0..65535";
         }
          description
            "Index of a list of neighboring test points
             in layers up and down the stack for
             the same interface that are related to the
             current test point.";
        }
        leaf position {
          type int8 {
            range "-1..1";
```

```
        }
        default "0";
        description
          " The relative position of neighboring test point corresponding
            to the current test point. Level 0 indicates test point correspondin
g
            to a specific index is in the same layer as the current test point.-
1
            means there is test point corresponding to a specific index is the t
est
            point down the stack and +1 means there is a test point correspondin
g to
             a specific index is the test point up the stack.";
      }
      choice tp-location {
        case mac-address {
          leaf mac-address-location {
            type yang:mac-address;
            description
              "MAC Address";
          }
          description
            "MAC Address based TP Addressing.";
        }
        case ipv4-address {
          leaf ipv4-address-location {
            type inet:ipv4-address;
            description
              "Ipv4 Address";
          }
          description
            "IP Address based TP Addressing.";
        }
        case ipv6-address {
          leaf ipv6-address-location {
            type inet:ipv6-address;
            description
              "IPv6 Address";
          }
          description
            "IPv6 Address based TP Addressing.";
        }
        case as-number {
          leaf as-number-location {
            type inet:as-number;
            description
              "AS number location";
          }
          description
            "AS number for point to multipoint OAM";
        }
        case router-id {
```

```
          leaf router-id-location {
            type rt:router-id;
            description
              "System id location";
          }

          description
            "System ID";
        }
        description
          "TP location.";
      }
      description
        "List of neighboring test points in the same layer that are related to c
urrent test
         point. If the neighboring test-point is placed after the current test p
oint, the
         position is specified as +1. If neighboring test-point
         is placed before the current test point, the position is specified
         as -1, if no neighboring test points placed before or after the current
         test point in the same layer, the position is specified as 0.";
    }
    description
      "Connectionless OAM related neighboring test points list.";
  }
  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 to be defined.";
        }
      }
      description
        "Technology choice.";
    }
    description
      "OAM Technology";
  }
  grouping tp-tools {
    description
      "Test Point OAM Toolset.";
    container tp-tools {
      leaf continuity-check {
        type boolean;
        mandatory true;
```

```
        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.
           RFC 8029: Detecting Multiprotocol Label Switched
          (MPLS) Data-Plane Failures.";
      }
      leaf path-discovery {
        type boolean;
        mandatory true;
        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 8029: Detecting Multiprotocol Label Switched (MPLS)
           Data-Plane Failures.";
      }
      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-tps;
```

```
    description
      "Test point Location";
  }
  grouping test-point-locations {
    description
      "Group of test point locations.";
        leaf tp-location-type {
      type identityref {
        base tp-address-technology-type;
      }
      description
        "Test point location type.";
    }
  container ipv4-location-type {
  when "derived-from-or-self(../tp-location-type,"+
  "'cl-oam: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 ni";
            leaf ipv4-location {
              type inet:ipv4-address;
              description
                "IPv4 Address.";
            }
            leaf ni {
              type routing-instance-ref;
              description
                "The ni is used to describe the
                 corresponding network instance";
            }
            uses test-point-location-info;
            description
              "List of test point locations.";
          }
          description
            "Serves as top-level container
             for test point location list.";
        }
        description
           "ipv4 location type container.";
      }
 container ipv6-location-type {
   when "derived-from-or-self(../tp-location-type,"+
    "'cl-oam: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 ni";
            leaf ipv6-location {
              type inet:ipv6-address;
              description
                "IPv6 Address.";
            }
            leaf ni {
              type routing-instance-ref;
              description
                "The ni is used to describe the
                 corresponding network instance";
            }
            uses test-point-location-info;
            description
              "List of test point locations.";
          }
          description
            "Serves as top-level container
             for test point location list.";
        }
        description
           "ipv6 location type container.";
      }
      container mac-location-type {
       when "derived-from-or-self(../tp-location-type,"+
       "'cl-oam: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;
            description
              "List of test point locations.";
          }
          description
            "Serves as top-level container
             for test point location list.";
```

```
        }
        description
           "mac address location type container.";
      }
container group-as-number-location-type {
   when "derived-from-or-self(../tp-location-type,"+
   "'cl-oam: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 ni {
              type routing-instance-ref;
              description
                "The ni is used to describe the
                 corresponding network instance";
            }
            uses test-point-location-info;
            description
              "List of test point locations.";
          }
          description
            "Serves as top-level container
             for test point location list.";
        }
        description
           "as number location type container.";
      }
container group-router-id-location-type {
when "derived-from-or-self(../tp-location-type,"+
  "'cl-oam:router-id-address-type')" {
  description
  "when test point location type is equal to system-info.";
   }
        container test-point-system-info-location-list {
          list test-point-locations {
            key "router-id-location";
            leaf router-id-location {
              type rt:router-id;
              description
                "System Id.";
```

```
            }
            leaf ni {
              type routing-instance-ref;
              description
                "The ni is used to describe the
                 corresponding network instance";
            }
            uses test-point-location-info;
            description
              "List of test point locations.";
          }
          description
            "Serves as top-level container for
             test point location list.";
        }
        description
           "system ID location type container.";
      }
  }
  augment "/nd:networks/nd:network/nd:node" {
    description
      "augments the /networks/network/node path defined in the
      ietf-network module (I-D.ietf-i2rs-yang-network-topo) with
      test-point-locations grouping.";
      uses test-point-locations;
  }
  grouping timestamp {
    description
      "Grouping for timestamp.";
    leaf timestamp-type {
      type identityref {
      base lime:timestamp-type;
      }
      description
        "Type of Timestamp, such as Truncated PTP, NTP.";
    }
 container timestamp-64bit {
  when "derived-from-or-self(../timestamp-type, 'cl-oam:truncated-ptp')"+
  "or derived-from-or-self(../timestamp-type,'cl-oam:ntp64')" {
   description
   "Only applies when Truncated PTP or 64bit NTP 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.";
      }
      description
      "Container for 64bit timestamp.See section 4.2.1 of
      draft-ietf-ntp-packet-timestamps for NTP 64-bit Timestamp
      Format and section 4.3 of draft-ietf-ntp-packet-timestamps
      for The PTP Truncated Timestamp Format.";
    }
 container timestamp-80bit {
 when "derived-from-or-self(../timestamp-type, 'cl-oam:ptp80')"{
         description
          "Only applies when 80bit PTP Timestamp.";
        }
  if-feature ptp-long-format;
      leaf timestamp-sec {
      type uint64 {
      range "0..281474976710655";
      }
      description
        "48bit Timestamp in seconds as per IEEE1588v2.";
       }
      leaf timestamp-nanosec {
      type uint32;
      description
        "Fractional part in nanoseconds as per IEEE1588v2.";
      }
      description
      "Container for 80bit timestamp.";
    }
 container ntp-timestamp-32bit {
 when "derived-from-or-self(../timestamp-type, 'cl-oam:truncated-ntp')"{
  description
   "Only applies when 32 bit NTP Short format Timestamp.";
   }
  if-feature ntp-short-format;
      leaf timestamp-sec {
      type uint16;
      description
        "Timestamp in seconds as per short format NTP.";
       }
      leaf timestamp-nanosec {
      type uint16;
      description
        "Truncated Fractional part in 16-bit NTP timestamp.";
```

```
      }
      description
      "Container for 32bit timestamp.See section 4.2.2 of
     draft-ietf-ntp-packet-timestamps for NTP 32-bit Timestamp
      Format.";
    }
 container icmp-timestamp-32bit {
 when "derived-from-or-self(../timestamp-type, 'cl-oam:icmp-ntp')"{
  description
  "Only applies when Truncated NTP or 64bit NTP Timestamp.";
  }
  if-feature icmp-timestamp;
      leaf timestamp-millisec {
      type uint32;

      description
        "timestamp in milliseconds for ICMP timestamp.";
       }
      description
      "Container for 32bit timestamp.See RFC792 for ICMP
       timestamp format.";
    }
  }
  grouping path-discovery-data {
    description
      "Path discovery related data output from nodes.";
    container src-test-point {
      description
        "Source test point.";
      uses tp-address-ni;
    }
    container dest-test-point {
      description
        "Destination test point.";
      uses tp-address-ni;
    }
    leaf sequence-number {
      type uint64;
      default "0";
      description
        "Sequence number in data packets. A value of
        zero indicates that no sequence number is sent.";
    }
    leaf hop-cnt {
      type uint8;
      default "0";
      description
        "Hop count. A value of zero indicates
```

```
        that no hop count is sent";
    }
    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 path trace information list has typically a single
         element for per-hop cases such as path-discovery RPC operation
         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-ni;
        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 queue of the interface from where
```

```
              the packet is forwarded out.  The queue depth could
              be the current number of memory buffers used by the
              queue and a packet can consume one or more memory buffers
              thus constituting device-level information.";
          }
          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-ni;
        leaf egress-intf-name {
          type if:interface-ref;
          description
            "Egress interface name.";
        }
      }
      container dest-test-point {
        description
        "Destination test point.";
        uses tp-address-ni;
        leaf ingress-intf-name {
          type if:interface-ref;
          description
            "Ingress interface name.";
        }
      }
      leaf sequence-number {
        type uint64;
        default "0";
        description
```

```
       "Sequence number in data packets. A value of
       zero indicates that no sequence number is sent.";
    }
    leaf hop-cnt {
      type uint8;
      default "0";
      description
        "Hop count. A value of zero indicates
         that no hop count is sent";
    }
    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;
    list cc-session-statistics {
       key type;
       leaf type {
        type identityref {
         base traffic-type;
        }
        description
         "Type of traffic.";
       }
        container cc-ipv4-sessions-statistics {
        when "../type = 'ipv4'" {
         description
          "Only applies when traffic type is Ipv4.";
        }
      description
       "CC ipv4 sessions";
      uses cc-session-statistics;
      }
       container cc-ipv6-sessions-statistics {
        when "../type = 'ipv6'" {
         description
          "Only applies when traffic type is Ipv6.";
        }
      description
        "CC ipv6 sessions";
      uses cc-session-statistics;
      }
       description
      "List of CC session statistics data.";
      }
```

```
    description
    "CC operational information.";
  }
}
```

<CODE ENDS>

6. Connectionless model applicability

The "ietf-connectionless-oam" model defined in this document provides a technology-independent abstraction of key OAM constructs for OAM protocols that use connectionless communication. 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, several snippets of technology-specific model extensions are presented for illustrative purposes. The complete model extensions should be worked on in respective protocol working groups.

6.1. BFD Extension

RFC 7276 defines BFD as a connection-oriented protocol. It is used to monitor a connectionless protocol in the case of basic BFD for IP.

6.1.1. 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 a dedicated BFD YANG data model [I-D.ietf-bfd-yang] is also standardized. Augmentation of the "ietf-connectionless-oam" model with BFD specific details provides an alternative approach that provides a unified view of management information across various OAM protocols. The BFD specific details can be the grouping defined in the BFD model avoiding duplication of effort.

6.1.1.1. 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 adding the "bfd" type as an augment to the ietf-connectionless-oam" model:

```
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"
{
    leaf bfd{
   type string;
  }
}
```

6.1.1.2. Test point attributes extension

To support BFD, the "ietf-connectionless-oam" model can be extended by adding specific parameters into the "test-point-locations" list and/or adding a new location type such as "BFD over MPLS TE" under "location-type".

6.1.1.2.1. 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, some groupings which are defined in [I-D.ietf-bfd-yang] are reused as follows:

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.

6.1.1.2.2. 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, a new "location-type" case is added and some groupings that are defined in [I-D.ietf-bfd-yang] are reused 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.

6.1.2. Schema Mount

An alternative method is using the schema mount mechanism [I-D.ietf-netmod-schema-mount] in the "ietf-connectionless-oam" model. Within the "test-point-locations" list, a "root" attribute is defined to provide a mount point for models mounted per "test-point-locations". 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 the "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.

6.1.2.1. 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>192.0.2.1</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>
```

6.2. LSP Ping extension

6.2.1. 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 extensions are introduced such as the "technology-type" extension and the test-point "attributes" extension.

Note that an LSP Ping YANG data model is being specified [I-D.zheng-mpls-lsp-ping-yang-cfg]. As with BFD, users can choose to use the "ietf-connectioless-oam" as basis and augment the "ietf-connectionless-oam" model with LSP Ping specific details in the model extension to provide a unified view across different technologies. The LSP Ping specific details can be the grouping defined in the LSP ping model to avoid duplication of effort.

6.2.1.1. 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"
{
   leaf lsp-ping{
   type string;
  }
}
```

6.2.1.2. 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-locations" list.

Users 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.";
        ......
       }
```

6.2.2. Schema Mount

An alternative method is using schema mount mechanism [I-D.ietf-netmod-schema-mount] in the "ietf-connectionless-oam". Within the "test-point-locations" list, a "root" attribute is defined to provide a mounted point for models mounted per "test-point-locations". 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.

6.2.2.1. 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> 192.0.2.1</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>
```

7. Security Considerations

The YANG module defined in this document is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC5246].

The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in this YANG module that 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/cl-oam:location-type/cl-oam:ipv4-location-type/cl-oam:test-point-ipv4-location-list/cl-oam:test-point-locations/

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

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

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

/nd:networks/nd:network/nd:node/cl-oam:location-type/cl-oam:group-router-id-location-type/cl-oam:test-point-system-info-location-list/cl-oam: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.

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

/coam:cc-session-statistics-data/cl-oam:cc-ipv4-sessions-statistics/cl-oam:cc-session-statistics/cl-oam:session-count/

/coam:cc-session-statistics-data/cl-oam:cc-ipv4-sessions-statistics/cl-oam:cc-session-statistics/cl-oam:session-up-count/

/coam:cc-session-statistics-data/cl-oam:cc-ipv4-sessions-statistics/cl-oam:cc-session-statistics/cl-oam: session-down-count/

/coam:cc-session-statistics-data/cl-oam:cc-ipv4-sessions-statistics/cl-oam:cc-session-statistics/cl-oam:session-admin-down-count/

/coam:cc-session-statistics-data/cl-oam:cc-ipv6-sessions-statistics/cl-oam:cc-session-statistics/cl-oam:session-count/

/coam:cc-session-statistics-data/cl-oam:cc-ipv6-sessions-statistics/cl-oam:cc-session-statistics/cl-oam:session-up-count//

/coam:cc-session-statistics-data/cl-oam:cc-ipv6-sessions-statistics/cl-oam:cc-session-statistics/cl-oam:session-down-count/

```
/coam:cc-session-statistics-data/cl-oam:cc-ipv6-sessions-
statistics/cl-oam:cc-session-statistics/cl-oam:session-admin-down-
count/
```

8. 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-lime-time-types
Registrant Contact: The IESG.
XML: N/A; the requested URI is an XML namespace.

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 [RFC7950].

```
Name: ietf-lime-common-types
Namespace: urn:ietf:params:xml:ns:yang:ietf-lime-time-types
Prefix: lime
Reference: RFC XXXX

Name: ietf-connectionless-oam
Namespace: urn:ietf:params:xml:ns:yang:ietf-connectionless-oam
Prefix: cl-oam
Reference: RFC XXXX
```

9. Acknowlegements

The authors of this document would like to thank Elwyn Davies, Alia Atlas, Brian E Carpenter, Greg Mirsky, Adam Roach, Alissa Cooper, Eric Rescorla, Ben Campbell, Benoit Claise, Kathleen Moriarty, Carlos Pignataro, and others for their substantive review and comments, and proposals to stabilize and improve the document.

10. References

10.1. Normative References

[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-17 (work in progress), October 2017.

[I-D.ietf-rtgwg-ni-model]
Berger, L., Hopps, C., Lindem, A., Bogdanovic, D., and X. Liu, "YANG Network Instances", draft-ietf-rtgwg-ni-model-04 (work in progress), September 2017.

[I-D.ietf-rtgwg-routing-types]
Liu, X., Qu, Y., Lindem, A., Hopps, C., and L. Berger, "Routing Area Common YANG Data Types", draft-ietf-rtgwg-routing-types-17 (work in progress), October 2017.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <https://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", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <https://www.rfc-editor.org/info/rfc4443>.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <https://www.rfc-editor.org/info/rfc5246>.

[RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010, <https://www.rfc-editor.org/info/rfc5905>.

[RFC6021] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6021, DOI 10.17487/RFC6021, October 2010, <https://www.rfc-editor.org/info/rfc6021>.

[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, <https://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, <https://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, <https://www.rfc-editor.org/info/rfc6536>.

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

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

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

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/info/rfc7950>.

[RFC8029] Kompella, K., Swallow, G., Pignataro, C., Ed., Kumar, N., Aldrin, S., and M. Chen, "Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures", RFC 8029, DOI 10.17487/RFC8029, March 2017, <https://www.rfc-editor.org/info/rfc8029>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

10.2. Informative References

[G.800] "Unified functional architecture of transport networks", ITU-T Recommendation G.800, 2016.

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

[I-D.ietf-bfd-yang] Rahman, R., Zheng, L., Jethanandani, M., Networks, J., and G. Mirsky, "YANG Data Model for Bidirectional Forwarding Detection (BFD)", draft-ietf-bfd-yang-07 (work in progress), October 2017.

[I-D.ietf-lime-yang-connection-oriented-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-connection-oriented-oam-model-00 (work in progress), June 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 the Management of Operations, Administration, and Maintenance (OAM) Protocols that use Connectionless Communications", draft-ietf-lime-yang-connectionless-oam-methods-13 (work in progress), November 2017.

[I-D.ietf-netmod-schema-mount]
Bjorklund, M. and L. Lhotka, "YANG Schema Mount", draft-ietf-netmod-schema-mount-08 (work in progress), October 2017.

[I-D.ietf-ntp-packet-timestamps]
Mizrahi, T., Fabini, J., and A. Morton, "Guidelines for Defining Packet Timestamps", draft-ietf-ntp-packet-timestamps-00 (work in progress), October 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-06 (work in progress), October 2017.

[IEEE.1588]
"IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", IEEE IEEE Std 1588-2008, 2008.

[RFC5462] Andersson, L. and R. Asati, "Multiprotocol Label Switching (MPLS) Label Stack Entry: "EXP" Field Renamed to "Traffic Class" Field", RFC 5462, DOI 10.17487/RFC5462, February 2009, <https://www.rfc-editor.org/info/rfc5462>.

[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, <https://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, <https://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 (editor)
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: May 4, 2017

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

October 31, 2016

Retrieval Methods YANG Data Model for Connectionless Operations, Administration, and Maintenance(OAM) protocols
draft-ietf-lime-yang-connectionless-oam-methods-00

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 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 May 4, 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.

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 [lime

base model]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[lime base model] 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
x for deprecated
o for obsolete

<flags> is one of:

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

<name> is the name of the node

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

<opts> is one of:

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

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

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 will allow the user to flexibility to retrieve the retrieved-data which defined by the base data model[lime base model].

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, while the path-discovery rpc command corresponds to IP Traceroute.

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, lsp ping), 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 tp-address-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
   |  |  |     +--:(src-dst-address)
   |  |  |     |  +---w src-ip-address?          inet:ip-address
   |  |  |     |  +---w dst-ip-address?          inet:ip-address
   |  |  |     |  +---w Interface?               if:interface-ref
   |  |  |     +--:(fec)
   |  |  |     |  +---w fec-type?                fec-type
   |  |  |     |  +---w (fec-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
   |  |  |     +--:(tlv-address)
   |  |  |     |  +---w tlv-type?               int16
   |  |  |     |  +---w tlv-len?                int16
   |  |  |     |  +---w tlv-value?              binary
   |  |  |     +--:(system-info)
   |  |  |        +---w system-id?              inet:uri
   |  |  +---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-address-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
   |     |  |  +--:(src-dst-address)
   |     |  |  |  +--ro src-ip-address?          inet:ip-address
   |     |  |  |  +--ro dst-ip-address?          inet:ip-address
   |     |  |  |  +--ro Interface?               if:interface-ref
   |     |  |  +--:(fec)
   |     |  |  |  +--ro fec-type?                fec-type
   |     |  |  |  +--ro (fec-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
   |     |  |  +--:(tlv-address)
   |     |  |  |  +--ro tlv-type?               int16
   |     |  |  |  +--ro tlv-len?                int16
   |     |  |  |  +--ro tlv-value?              binary
   |     |  |  +--:(system-info)
   |     |  |     +--ro system-id?              inet:uri
   |     |  +--ro egress-intf-name?        if:interface-ref
   |     +--ro dest-test-point
   |     |  +--ro vrf?                     routing-instance-ref
   |     |  +--ro tp-address-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
   |     |  |  +--:(src-dst-address)
   |     |  |  |  +--ro src-ip-address?          inet:ip-address
   |     |  |  |  +--ro dst-ip-address?          inet:ip-address
   |     |  |  |  +--ro Interface?               if:interface-ref
   |     |  |  +--:(fec)
   |     |  |  |  +--ro fec-type?                fec-type
   |     |  |  |  +--ro (fec-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
   |     |  |  +--:(tlv-address)
   |     |  |  |  +--ro tlv-type?               int16
   |     |  |  |  +--ro tlv-len?                int16
   |     |  |  |  +--ro tlv-value?              binary
   |     |  |  +--:(system-info)
   |     |  |     +--ro system-id?              inet:uri
   |     |  +--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 path-discovery
     +---w input
     |  +---w destination-tp
```

```
      |  |  +---w tp-address-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
      |  |     +--:(src-dst-address)
      |  |     |  +---w src-ip-address?          inet:ip-address
      |  |     |  +---w dst-ip-address?          inet:ip-address
      |  |     |  +---w Interface?               if:interface-ref
      |  |     +--:(fec)
      |  |     |  +---w fec-type?                fec-type
      |  |     |  +---w (fec-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
      |  |     +--:(tlv-address)
      |  |     |  +---w tlv-type?                int16
      |  |     |  +---w tlv-len?                 int16
      |  |     |  +---w tlv-value?               binary
      |  |     +--:(system-info)
      |  |        +---w system-id?               inet:uri
      |  +---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-address-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
         |     +--:(src-dst-address)
         |     |  +--ro src-ip-address?          inet:ip-address
         |     |  +--ro dst-ip-address?          inet:ip-address
         |     |  +--ro Interface?               if:interface-ref
         |     +--:(fec)
         |     |  +--ro fec-type?                fec-type
         |     |  +--ro (fec-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
         |     +--:(tlv-address)
         |     |  +--ro tlv-type?                int16
         |     |  +--ro tlv-len?                 int16
         |     |  +--ro tlv-value?               binary
```

```
         |     +--:(system-info)
         |        +--ro system-id?              inet:uri
         +--ro dest-test-point
         |  +--ro vrf?                      routing-instance-ref
         |  +--ro tp-address-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
         |     +--:(src-dst-address)
         |     |  +--ro src-ip-address?         inet:ip-address
         |     |  +--ro dst-ip-address?         inet:ip-address
         |     |  +--ro Interface?              if:interface-ref
         |     +--:(fec)
         |     |  +--ro fec-type?               fec-type
         |     |  +--ro (fec-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
         |     +--:(tlv-address)
         |     |  +--ro tlv-type?               int16
         |     |  +--ro tlv-len?                int16
         |     |  +--ro tlv-value?              binary
         |     +--:(system-info)
         |        +--ro system-id?              inet:uri
         +--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-address-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
               |  +--:(src-dst-address)
               |  |  +--ro src-ip-address?         inet:ip-address
               |  |  +--ro dst-ip-address?         inet:ip-address
               |  |  +--ro Interface?              if:interface-ref
               |  +--:(fec)
               |  |  +--ro fec-type?               fec-type
               |  |  +--ro (fec-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-Gro
up-Address
                |  |           +--:(vpn)
                |  |           |  +--ro as-number?               inet:as-number
                |  |           +--:(global-id)
                |  |              +--ro lsp-id?                  string
                |  +--:(tlv-address)
                |  |  +--ro tlv-type?               int16
                |  |  +--ro tlv-len?                int16
                |  |  +--ro tlv-value?              binary
                |  +--:(system-info)
                |     +--ro system-id?              inet:uri
               +--ro timestamp-val?          yang:date-and-time
               +--ro ingress-intf-name?      if:interface-ref
               +--ro egress-intf-name?       if:interface-ref
               +--ro app-meta-data?          uint32
```

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;
    }

    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 2016-06-23 {
      description
        "06 version";
      reference "";
    }

    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;

       }
    }
}
```

YANG module of OAM

<CODE ENDS>

5. Security Considerations

TBD.

6. 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-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: goam reference: RFC XXXX

7. 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>.

[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>.

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

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: May 16, 2018

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

November 12, 2017

Retrieval Methods YANG Data Model for the Management of Operations, Administration, and Maintenance (OAM) Protocols that use Connectionless Communications

draft-ietf-lime-yang-connectionless-oam-methods-13

Abstract

This document presents a retrieval method YANG Data model for connectionless OAM protocols. It provides technology-independent RPC operations for OAM protocols that use connectionless communication. The retrieval methods model herein presented can be extended to include technology specific details. There are two key benefits of this approach: First, it leads to uniformity between OAM protocols. And second, it support both nested OAM workflows (i.e., performing OAM functions at different or same levels through a unified interface) as well as 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 https://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 May 16, 2018.

Table of Contents

1. Introduction

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

1. Monitor network communications (i.e., Reachability Verification, Continuity Check)

2. Troubleshoot failures (i.e., Fault verification and Localization)

3. Monitor service-level agreements and performance (i.e., Performance Management)

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

Ping and Traceroute [RFC792] [RFC4443], as well as BFD [RFC5880] are well-known fault verification and isolation tools, respectively, for IP networks. Over the years, different technologies have developed similar toolsets for equivalent purposes.

This document presents an on-demand retrieval method YANG Data model for OAM protocols that use connectionless communication. This model provides technology-independent RPC operations for OAM protocols that use connectionless communication (i.e., connectionless oam). 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 generic YANG data model for connectionless OAM.

2. Conventions used in this document

The following terms are defined in [RFC6241] and are used in this document:

- client
- configuration data
- server
- state data

The following terms are defined in [RFC6020] and are used in this document:

- 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 - Remote Procedure Call

RPC Operation - A specific Remote Procedure Call

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

This document describes an On-demand retrieval method YANG Data model for OAM protocols that use connectionless communication. This model provides technology-independent retrieval procedures (RPC operations) for connectionless OAM protocols. It provides a flexible way to retrieve the data which defined by the "ietf-connectionless-oam.yang" module [I-D.ietf-lime-yang-connectionless-oam].

3.1. RPC operation 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 common OAM functions and define two generic RPC operations: ’continuity-check’ and ’path-discovery’. In practice, these RPC operations are activated on-demand and 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 operation corresponds to IP Traceroute [RFC792] [RFC4443].

Note that the RPC operation 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], and LSP Ping [RFC8029]). This base building block should be extended with corresponding technology specific parameters. To facilitate this for future enhancements to data retrieval methods, the RPCs are captured under a separate module.

The generic ’tp-address’ grouping is used as data input from different RPCs described in this document. The generic ’path-discovery-data’ and ’continuity-check-data’ groupings defined by the "ietf-connectionless-oam.yang" module [I-D.ietf-lime-yang-connectionless-oam]are used as data outputs from different RPCs described in this document. Similar methods including other RPCs can retrieve the data using the same data model (i.e., the "ietf-connectionless-oam.yang" module).

```
  rpc continuity-check {
   if-feature cl-oam:continuity-check;
   description
     "Continuity-check RPC operation as per RFC7276.";
   input {
   uses rpc-input-parameters;
   ....
```

```
      }
    output {
      container response-info {
        leaf protocol-id {
          type identityref {
            base protocol-id;
          }
          mandatory true;
          description
            "Protocol used in CC. ";
        }
        leaf protocol-id-meta-data {
           type identityref {
            base protocol-id-meta-data;
            }
             description
             "An optional meta-data related to the protocol ID.";
        }
        leaf status-code {
          type identityref{
         base status-code;
       }
           mandatory true;
           description
             "Status code for Continuity Check RPC operation.";
        }
        leaf status-sub-code {
          type identityref{
          base status-sub-code;
       }
           mandatory true;
           description
             "Status Sub code for Continuity Check RPC operation.";
        }
        description
          "Status Code and Status Sub Code for continuity check RPC operation.";
      }
      uses cl-oam:continuity-check-data;
    }
  }

    rpc path-discovery {
      description
        "path discovery RPC operation as per RFC7276.";
      input {
      uses rpc-input-parameters;
      .....
      }
```

```
    output {
      list response-list {
        key "response-index";
        description
          "Path discovery response list.";
        leaf response-index {
          type uint32;
          mandatory true;
          description
            "Response index.";
        }
        leaf protocol-id {
          type identityref {
            base protocol-id;
          }
          mandatory true;
          description
            "Protocol used in PD. ";
        }
        leaf protocol-id-meta-data {
           type identityref {
            base protocol-id-meta-data;
            }
             description
             "An optional meta-data related to the protocol ID.";
        }
        leaf status-code {
          type identityref{
          base status-code;
        }
          mandatory true;
          description
            "Status code for Path Discovery RPC operation. ";
        }
        leaf status-sub-code {
          type identityref{
          base status-sub-code;
       }
          mandatory true;
          description
            "Status Sub code for Path Discovery RPC operation. ";
        }
      }
      uses cl-oam:path-discovery-data;
    }
  }
```

Snippet of data hierarchy related to RPC operations

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 {cl-oam:continuity-check}?
   |  +---w input
   |  |  +---w destination-tp
   |  |  |  +---w tp-location-type    identityref
   |  |  |  +---w mac-address
   |  |  |  |  +---w mac-address    yang:mac-address
   |  |  |  +---w ipv4-address
   |  |  |  |  +---w ipv4-address    inet:ipv4-address
   |  |  |  +---w ipv6-address
   |  |  |  |  +---w ipv6-address    inet:ipv6-address
   |  |  |  +---w tp-attribute
   |  |  |  |  +---w tp-attribute-type?     address-attribute-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?   rt:route-distinguisher
   |  |  |  |     |  +---w sender-ve-id?          uint16
   |  |  |  |     |  +---w receiver-ve-id?        uint16
   |  |  |  |     +--:(mpls-mldp)
   |  |  |  |        +---w (root-address)?
   |  |  |  |           +--:(ip-address)
   |  |  |  |           |  +---w source-address?        inet:ip-address
   |  |  |  |           |  +---w group-ip-address?      inet:ip-address
   |  |  |  |           +--:(vpn)
   |  |  |  |           |  +---w as-number?             inet:as-number
   |  |  |  |           +--:(global-id)
   |  |  |  |              +---w lsp-id?                string
   |  |  |  +---w system-info
   |  |  |     +---w router-id?   rt:router-id
   |  |  +---w source-interface      if:interface-ref
   |  |  +---w outbound-interface    if:interface-ref
   |  |  +---w vrf?                  cl-oam:routing-instance-ref
   |  |  +---w session-type?         enumeration
```

```
|  |  +---w count?               uint32
|  |  +---w ttl?                 uint8
|  |  +---w packet-size?         uint32
|  +--ro output
|     +--ro response-info
|     |  +--ro protocol-id              identityref
|     |  +--ro protocol-id-meta-data?   identityref
|     |  +--ro status-code              identityref
|     |  +--ro status-sub-code          identityref
|     +--ro src-test-point
|     |  +--ro ni?                 routing-instance-ref
|     |  +--ro tp-location-type    identityref
|     |  +--ro mac-address
|     |  |  +--ro mac-address    yang:mac-address
|     |  +--ro ipv4-address
|     |  |  +--ro ipv4-address    inet:ipv4-address
|     |  +--ro ipv6-address
|     |  |  +--ro ipv6-address    inet:ipv6-address
|     |  +--ro tp-attribute
|     |  |  +--ro tp-attribute-type?     address-attribute-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?   rt:route-distinguisher
|     |  |     |  +--ro sender-ve-id?          uint16
|     |  |     |  +--ro receiver-ve-id?        uint16
|     |  |     +--:(mpls-mldp)
|     |  |        +--ro (root-address)?
|     |  |           +--:(ip-address)
|     |  |           |  +--ro source-address?        inet:ip-address
|     |  |           |  +--ro group-ip-address?      inet:ip-address
|     |  |           +--:(vpn)
|     |  |           |  +--ro as-number?             inet:as-number
|     |  |           +--:(global-id)
|     |  |              +--ro lsp-id?                string
|     |  +--ro system-info
|     |  |  +--ro router-id?   rt:router-id
|     |  +--ro egress-intf-name?   if:interface-ref
|     +--ro dest-test-point
|     |  +--ro ni?                 routing-instance-ref
```

```
|     |  +--ro tp-location-type     identityref
|     |  +--ro mac-address
|     |  |  +--ro mac-address    yang:mac-address
|     |  +--ro ipv4-address
|     |  |  +--ro ipv4-address    inet:ipv4-address
|     |  +--ro ipv6-address
|     |  |  +--ro ipv6-address    inet:ipv6-address
|     |  +--ro tp-attribute
|     |  |  +--ro tp-attribute-type?     address-attribute-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?  rt:route-distinguisher
|     |  |     |  +--ro sender-ve-id?         uint16
|     |  |     |  +--ro receiver-ve-id?       uint16
|     |  |     +--:(mpls-mldp)
|     |  |        +--ro (root-address)?
|     |  |           +--:(ip-address)
|     |  |           |  +--ro source-address?       inet:ip-address
|     |  |           |  +--ro group-ip-address?     inet:ip-address
|     |  |           +--:(vpn)
|     |  |           |  +--ro as-number?            inet:as-number
|     |  |           +--:(global-id)
|     |  |              +--ro lsp-id?               string
|     |  +--ro system-info
|     |  |  +--ro router-id?   rt: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-loss-count?          uint32
|     |  +--ro loss-ratio?                 percentage
|     |  +--ro packet-reorder-count?       uint32
|     |  +--ro packets-out-of-seq-count?   uint32
|     |  +--ro packets-dup-count?          uint32
```

```
   |     +--ro session-delay-statistics
   |     |  +--ro time-unit-value?       identityref
   |     |  +--ro min-delay-value?       uint32
   |     |  +--ro max-delay-value?       uint32
   |     |  +--ro average-delay-value?   uint32
   |     +--ro session-jitter-statistics
   |        +--ro unit-value?             identityref
   |        +--ro min-jitter-value?       uint32
   |        +--ro max-jitter-value?       uint32
   |        +--ro average-jitter-value?   uint32
   +---x path-discovery {cl-oam:path-discovery}?
      +---w input
      |  +---w destination-tp
      |  |  +---w tp-location-type    identityref
      |  |  +---w mac-address
      |  |  |  +---w mac-address    yang:mac-address
      |  |  +---w ipv4-address
      |  |  |  +---w ipv4-address    inet:ipv4-address
      |  |  +---w ipv6-address
      |  |  |  +---w ipv6-address    inet:ipv6-address
      |  |  +---w tp-attribute
      |  |  |  +---w tp-attribute-type?     address-attribute-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?   rt:route-distinguisher
      |  |  |     |  +---w sender-ve-id?          uint16
      |  |  |     |  +---w receiver-ve-id?        uint16
      |  |  |     +--:(mpls-mldp)
      |  |  |        +---w (root-address)?
      |  |  |           +--:(ip-address)
      |  |  |           |  +---w source-address?        inet:ip-address
      |  |  |           |  +---w group-ip-address?      inet:ip-address
      |  |  |           +--:(vpn)
      |  |  |           |  +---w as-number?             inet:as-number
      |  |  |           +--:(global-id)
      |  |  |              +---w lsp-id?                string
      |  |  +---w system-info
      |  |     +---w router-id?   rt:router-id
      |  +---w source-interface      if:interface-ref
```

```
      |  +---w outbound-interface    if:interface-ref
      |  +---w vrf?                  cl-oam:routing-instance-ref
      |  +---w session-type?         enumeration
      |  +---w max-ttl?              uint8
      +--ro output
         +--ro response-list* [response-index]
         |  +--ro response-index           uint32
         |  +--ro protocol-id              identityref
         |  +--ro protocol-id-meta-data?   identityref
         |  +--ro status-code              identityref
         |  +--ro status-sub-code          identityref
         +--ro src-test-point
         |  +--ro ni?                 routing-instance-ref
         |  +--ro tp-location-type    identityref
         |  +--ro mac-address
         |  |  +--ro mac-address    yang:mac-address
         |  +--ro ipv4-address
         |  |  +--ro ipv4-address    inet:ipv4-address
         |  +--ro ipv6-address
         |  |  +--ro ipv6-address    inet:ipv6-address
         |  +--ro tp-attribute
         |  |  +--ro tp-attribute-type?     address-attribute-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?   rt:route-distinguisher
         |  |     |  +--ro sender-ve-id?          uint16
         |  |     |  +--ro receiver-ve-id?        uint16
         |  |     +--:(mpls-mldp)
         |  |        +--ro (root-address)?
         |  |           +--:(ip-address)
         |  |           |  +--ro source-address?        inet:ip-address
         |  |           |  +--ro group-ip-address?      inet:ip-address
         |  |           +--:(vpn)
         |  |           |  +--ro as-number?             inet:as-number
         |  |           +--:(global-id)
         |  |              +--ro lsp-id?                string
         |  +--ro system-info
         |     +--ro router-id?   rt:router-id
         +--ro dest-test-point
```

```
|  +--ro ni?                routing-instance-ref
|  +--ro tp-location-type   identityref
|  +--ro mac-address
|  |  +--ro mac-address    yang:mac-address
|  +--ro ipv4-address
|  |  +--ro ipv4-address    inet:ipv4-address
|  +--ro ipv6-address
|  |  +--ro ipv6-address    inet:ipv6-address
|  +--ro tp-attribute
|  |  +--ro tp-attribute-type?     address-attribute-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?  rt:route-distinguisher
|  |     |  +--ro sender-ve-id?         uint16
|  |     |  +--ro receiver-ve-id?       uint16
|  |     +--:(mpls-mldp)
|  |        +--ro (root-address)?
|  |           +--:(ip-address)
|  |           |  +--ro source-address?       inet:ip-address
|  |           |  +--ro group-ip-address?     inet:ip-address
|  |           +--:(vpn)
|  |           |  +--ro as-number?            inet:as-number
|  |           +--:(global-id)
|  |              +--ro lsp-id?               string
|  +--ro system-info
|     +--ro router-id?   rt: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-loss-count?          uint32
|  +--ro loss-ratio?                 percentage
|  +--ro packet-reorder-count?       uint32
|  +--ro packets-out-of-seq-count?   uint32
|  +--ro packets-dup-count?          uint32
```

```
         +--ro session-delay-statistics
         |  +--ro time-unit-value?       identityref
         |  +--ro min-delay-value?       uint32
         |  +--ro max-delay-value?       uint32
         |  +--ro average-delay-value?   uint32
         +--ro session-jitter-statistics
         |  +--ro unit-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 ni?                      routing-instance-ref
               +--ro tp-location-type         identityref
               +--ro mac-address
               |  +--ro mac-address    yang:mac-address
               +--ro ipv4-address
               |  +--ro ipv4-address    inet:ipv4-address
               +--ro ipv6-address
               |  +--ro ipv6-address    inet:ipv6-address
               +--ro tp-attribute
               |  +--ro tp-attribute-type?     address-attribute-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?   rt:route-distinguisher
               |     |  +--ro sender-ve-id?          uint16
               |     |  +--ro receiver-ve-id?        uint16
               |     +--:(mpls-mldp)
               |        +--ro (root-address)?
               |           +--:(ip-address)
               |           |  +--ro source-address?        inet:ip-address
               |           |  +--ro group-ip-address?      inet:ip-address
               |           +--:(vpn)
```

```
               |           |  +--ro as-number?            inet:as-number
               |           +--:(global-id)
               |              +--ro lsp-id?               string
               +--ro system-info
               |  +--ro router-id?   rt:router-id
               +--ro timestamp-type?         identityref
               +--ro timestamp-64bit
               |  +--ro timestamp-sec?       uint32
               |  +--ro timestamp-nanosec?   uint32
               +--ro timestamp-80bit {ptp-long-format}?
               |  +--ro timestamp-sec?       uint64
               |  +--ro timestamp-nanosec?   uint32
               +--ro ntp-timestamp-32bit {ntp-short-format}?
               |  +--ro timestamp-sec?       uint16
               |  +--ro timestamp-nanosec?   uint16
               +--ro icmp-timestamp-32bit {icmp-timestamp}?
               |  +--ro timestamp-millisec?   uint32
               +--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
```

data hierarchy of OAM Retrieval Methods

4. OAM Retrieval Methods YANG Module

<CODE BEGINS> file "ietf-connectionless-oam-methods@2017-09-06.yang"

```
module ietf-connectionless-oam-methods {
namespace "urn:ietf:params:xml:ns:yang:ietf-connectionless-oam-methods";
 prefix cloam-methods;
 import ietf-interfaces {
  prefix if;
 }
 import ietf-connectionless-oam {
  prefix cl-oam;
 }
  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 RPC operations for
```

```
     connectionless OAM to be used within IETF
     in a protocol Independent manner.
     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 2017-09-06{
    description
      "08 version";
    reference "draft-ietf-lime-yang-connectionless-oam-methods";
  }
  identity protocol-id {
    description
    "This is base identity for a generic protocol ID. The protocol
     registry can be found in https://www.iana.org/protocols.";
  }
  identity protocol-id-internet {
    base protocol-id;
    description
      "Identity for Internet Protocols.";
  }
  identity protocol-id-proprietary {
    base protocol-id;
    description
      "Identity for proprietary protocol (e.g.,IP SLA).";
  }
  identity protocol-id-sfc {
    base protocol-id;
    description
      "Identity for Service Function Chaining.";
  }
  identity protocol-id-mpls {
    base protocol-id;
    description
      "MPLS protocol.";
  }
  identity protocol-id-mpls-tp {
    base protocol-id;
    description
      "MPLS-TP protocol.";
  }
  identity protocol-id-twamp {
    base protocol-id;
    description
      "TWAMP protocol.";
  }
  identity protocol-id-bier {
```

```
    base protocol-id;
    description
      "BIER protocol.";
  }
  identity status-code {
   description
   "This is Base Identity for status code.";
  }
  identity success-reach {
    base status-code;
   description
   "Indicate that the destination being verified
    is reachable (See RFC7276).";
  }
  identity fail-reach {
    base status-code;
   description
   "Indicate that the destination being verified
    is not reachable (See RFC7276).";
  }
  identity success-path-verification {
    base status-code;
   description
   "Indicate that the path verification is performed
   successfully (See RFC7276).";
  }
  identity fail-path-verification {
    base status-code;
   description
   "Indicate that the path verification fails (See RFC7276).";
  }
  identity status-sub-code {
   description
   "IdentityBase status sub code.";
  }
    identity invalid-cc {
   base status-sub-code;
   description
   "Indicates that the Continuity check message is invalid (See RFC7276).";
  }
  identity invalid-pd {
   base status-sub-code;
   description
   "Indicates that the path discovery message is invalid (See RFC7276).";
  }
  identity protocol-id-meta-data {
    description
    "This is base identity for meta-data corresponding
```

```
     to protocol ID.";
  }
  identity protocol-internet-number {
    base protocol-id-meta-data;
    description
      "Internet Protocol Number for standard
     Internet Protocols (IANA assigned Internet
     Protocol numbers) to help in protocol processing.
     The protocol IDs registry can be found in
     https://www.iana.org/assignments/protocol-numbers
     /protocol-numbers.xhtml.";
  }

  grouping rpc-input-parameters {
     container destination-tp {
       uses cl-oam:tp-address;
         description
         "Destination test point.";
       }
       leaf source-interface {
         type if:interface-ref;
         mandatory true;
         description
          "Source interface.";
       }
      leaf outbound-interface {
         type if:interface-ref;
         mandatory true;
         description
          "Outbound interface.";
      }
      leaf vrf {
         type cl-oam:routing-instance-ref;
         description
          "VRF instance.";
      }
    description
   "Grouping for RPC input parameters";
  }
  rpc continuity-check {
     if-feature "cl-oam:continuity-check";
     description
     "Continuity-check RPC operation as per RFC7276.";
    input {
        uses rpc-input-parameters;
        uses cl-oam:session-type {
         description
      "If session-type is specified, then session-type
```

```
          must be set to on-demand";
        }
         leaf count {
         type uint32 {
          range 0..4294967295 {
              description
              "The overall number of packet to be transmitted
              by the sender. The value of count will be set
             to zero (0) on creation and will thereafter increase
             monotonically until it reaches a maximum value of 2^32-1
             (4294967295 decimal), when it wraps around and starts
             increasing again from zero.";
             }
         }
         default "5";
         description
           "Specifies the number of
           packets that will be sent. By
           default, the packet number is
           set to 5.";
        }
         leaf ttl {
         type uint8;
         default "255";
         description
          "Time to live (TTL) used to limit lifetime
          of data packet transmitted in the network
          and prevent looping. The TTL value is decremented
          for every hop which the packet traverses. If the
          TTL is zero, the data packet will be discarded.";
       }
       leaf packet-size {
          type uint32 {
          range "64..10000";
         }
         default "64";
         description
          "Packet size of continuity-check message, in octets.
          By default, the packet size is set to 64 octets.";
       }
     }
     output {
     container response-info {
         leaf protocol-id {
           type identityref {
             base protocol-id;
           }
           mandatory true;
```

```
            description
            "Protocol used in continuity check message.
            This could be a standard protocol (e.g.,
            TCP/IP protocols, MPLS etc.,) or a proprietary
            protocol as identified by this field.";
        }
        leaf protocol-id-meta-data {
         type identityref {
            base protocol-id-meta-data;
            }
            description
            "An optional meta-data related to the protocol ID.
            For e.g., this could be the Internet Protocol number
            for standard Internet Protocols for help in protocol
            processing.";
        }
        leaf status-code {
          type identityref{
          base status-code;
          }
          mandatory true;
          description
           "Status code for continuity check RPC operation.
           This could be a basic status code (e.g., destination
           is reachable or destination is not reachable (See RFC7276))
           or some customized status code as identified by this field.";
        }
        leaf status-sub-code {
            type identityref{
            base status-sub-code;
          }
          mandatory true;
          description
          "An optional status sub code for continuity check
          RPC operation. If the basic status code is destination
          reachable, this status-sub-code doesn't need to be specified.
          If the basic status code is destination unreachable, the
          status-sub-code can be used to specify the detailed reasons.
          This could be a basic sub-status-code (such as invalid cc) or
          other error codes specific to the protocol under use for CC
          For example if ICMP is the protocol under use, the error codes
          defined in [RFC4443] can be used to specify the reasons specific to
          ICMP. These technology specific status-sub-code can be defined
          in technology specific models.";
        }
        description
         "Status Code and Status sub code for continuity check RPC operation.";
      }
```

```
      uses cl-oam:continuity-check-data;
    }
  }

  rpc path-discovery {
    if-feature "cl-oam:path-discovery";
    description
      "Path discovery RPC operation as per RFC7276.";
    input {
      uses rpc-input-parameters;
      uses cl-oam:session-type {
         description
       "If session-type is specified, then session-type
         must be set to on-demand";
    }
      leaf max-ttl {
        type uint8;
        default "255";
         description
          "Maximum TTL indicates the maixmum number of hops that
          a packet is permitted to travel before being discarded
          by a router. By default, the maximimum TTL is set to 255.";
      }
    }
    output {
      list response-list {
        key "response-index";
        description
          "Path discovery response list.";
        leaf response-index {
          type uint32;
          mandatory true;
          description
           "Response index.";
        }
        leaf protocol-id {
          type identityref {
            base protocol-id;
          }
          mandatory true;
             description
             "Protocol used in PD. This could be a standard
             protocol (e.g., TCP/IP protocols, MPLS etc.,)
             or a proprietary protocol as identified by
             this field.";
        }
        leaf protocol-id-meta-data {
          type identityref {
```

```
            base protocol-id-meta-data;
            }
             description
             "An optional meta-data related to the protocol ID.
             For e.g., this could be the Internet Protocol number
             for standard Internet Protocols for help in protocol
             processing.";
        }
        leaf status-code {
          type identityref{
            base status-code;
          }
          mandatory true;
          description
           "Status code for continuity check RPC operation.
           This could be a basic status code (e.g., destination
           is reachable or destination is not reachable) or some
           customized status code as identified by this field.";
        }
        leaf status-sub-code {
          type identityref{
            base status-sub-code;
          }
          mandatory true;
          description
          "An optional status sub code for continuity check
          RPC operation. If the basic status code is destination
          reachable, this status-sub-code doesn't need to be specified.
          If the basic status code is destination unreachable, the
          status-sub-code can be used to specify the detailed reasons.
          This could be a basic sub-status-code (such as invalid cc) or
          other error codes specific to the protocol under use for CC
          For example if ICMP is the protocol under use, the error codes
          defined in [RFC4443] can be used to specify the reasons specific to
          ICMP. These technology specific status-sub-code can be defined
          in technology specific models.";
        }
      }
      uses cl-oam:path-discovery-data;
    }
  }
}
```

<CODE ENDS>

5. Security Considerations

The YANG module defined in this document is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC5246].

The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

Some of the RPC operations in this 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:

- continuity-check: Generates continuity check.

- path-discovery: Generates path discovery.

These operations are used to retrieve the data from the device that need to execute the OAM command. Unauthorized source access to some sensitive information in the above data may be used for network reconnaissance or lead to Denial-of-Service attack on both the local device and the network.

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: cloam-methods

reference: RFC XXXX

7. Acknowlegements

The authors of this document would like to thank Elwyn Davies, Alia Atlas, Brian E Carpenter, Greg Mirsky, Adam Roach, Alissa Cooper, Eric Rescorla, Ben Campbell, Benoit Claise, Kathleen Moriarty, Carlos Pignataro, Benjamin Kaduk, and others for their substantive review, comments, an d proposals to improve the document.

8. References

8.1. 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 the Management of Operations, Administration, and Maintenance (OAM) Protocols that use Connectionless Communications", draft-ietf-lime-yang-connectionless-oam-16 (work in progress), October 2017.

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

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <https://www.rfc-editor.org/info/rfc5246>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <https://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, <https://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, <https://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, <https://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, <https://www.rfc-editor.org/info/rfc7011>.

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

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

8.2. Informative References

[I-D.ietf-netconf-yang-push] Clemm, A., Voit, E., Prieto, A., Tripathy, A., Nilsen-Nygaard, E., Bierman, A., and B. Lengyel, "YANG Datastore Subscription", draft-ietf-netconf-yang-push-11 (work in progress), October 2017.

[RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <https://www.rfc-editor.org/info/rfc4443>.

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

[RFC6087] Bierman, A., "Guidelines for Authors and Reviewers of YANG Data Model Documents", RFC 6087, DOI 10.17487/RFC6087, January 2011, <https://www.rfc-editor.org/info/rfc6087>.

[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, <https://www.rfc-editor.org/info/rfc7276>.

[RFC8029] Kompella, K., Swallow, G., Pignataro, C., Ed., Kumar, N., Aldrin, S., and M. Chen, "Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures", RFC 8029, DOI 10.17487/RFC8029, March 2017, <https://www.rfc-editor.org/info/rfc8029>.

Appendix A. Appdendix A.1 Extending Connectionless OAM Method Module Example

The following is an example of extensions possible to "ietf-connectionless-oam-methods" YANG model defined in this document.

The snippet below depicts an example of augmenting the "ietf-connectionless-oam-methods" YANG model with ICMP ping attributes:

```
augment "/cloam-methods:continutiy-check"
+"/cloam-methods:output"{
  container session-rtt-statistics{
   leaf min-rtt{
    type uint32;
 description
 "This minimum ping round-trip-time (RTT) received.";
   }
   leaf max-rtt{
    type uint32;
 description
 "This maximum ping round-trip-time (RTT) received.";
   }
   leaf avg-rtt{
    type uint32;
 description
  "The current average ping round-trip-time (RTT)";
   }
   description
   "This container presents the ping round-trip-time statistics.";
  }
}
```

Appendix B. Appendix A.2 Example of new retrieval procedures Model

As discussed in the Introduction section of this document, the new retrieval procedures can be defined for retrieval of the same data defined by base YANG Data model for connectionless OAM protocols. This appendix demonstrates how the base connectionless OAM data model can be extended to support persistent data retrieval besides on demand retrieval procedures defined in section 3, i.e., first retrieve a persistent-id based on the destination test point location information and then retrieve the export details based on persistent-

id. Internet Protocol Flow Information Export (IPFIX) [RFC7011] or YANG-push [I-D.ietf-netconf-yang-push] are currently outlined here as data export options. Additional export options can be added in future.

The YANG module "example-cl-oam-persistent-methods" shown below is intended as an illustration rather than a real definition of a RPC operation model for persistent data retrieval. For the sake of brevity, this module does not obey all the guidelines specified in [RFC6087].

```
module example-cl-oam-persistent-methods {
namespace "http://example.com/cl-oam-persistent-methods";
 prefix pcloam-methods;
 import ietf-interfaces {
  prefix if;
 }
 import ietf-connectionless-oam {
  prefix cl-oam;
 }
 import ietf-yang-types {
  prefix 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";
  }
  identity protocol-id {
    description
    "A generic protocol identifier.";
  }
identity status-code {
 description
 "Base status code";
}
identity success-reach {
  base status-code;
 description
```

```
   "Indicate that the destination being verified
    is reachable";
  }
  identity fail-reach {
    base status-code;
   description
   "Indicate that the destination being verified
    is not reachable";
  }
  identity success-path-verification {
    base status-code;
   description
   "Indicate that the path verification is performed
   successfully.";
  }
  identity fail-path-verification {
    base status-code;
   description
   "Indicate that the path verification fails.";
  }
  identity status-sub-code {
   description
   "Base status sub code";
  }
    identity invalid-cc {
   base status-sub-code;
   description
   "Indicates that the Continuity check message is invalid";
  }
  identity invalid-pd {
   base status-sub-code;
   description
   "Indicates that the path discovery message is invalid";
  }
    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-get-persistent-id {
      if-feature "cl-oam:continuity-check";
      description
        "Obtains continuity-check persistent identification given mapping
         parameters as input.";
      input {
        container destination-tp {
          uses cl-oam:tp-address;
          description
            "Destination test point.";
        }
        uses cl-oam: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 cl-oam:routing-instance-ref;
          description
            "VRF instance.";
        }
      }
      output {
        container error-code {
          leaf protocol-id {
            type identityref {
              base protocol-id;
            }
            mandatory true;
            description
```

```
              "Protocol used. This could be a standard
               protocol (e.g., TCP/IP protocols, MPLS etc.,)
               or a proprietary protocol as identified by
               this field.";
          }
          leaf protocol-id-meta-data {
            type uint64;
            description
              "An optional meta-data related to the protocol ID.
               For e.g., this could be the Internet Protocol number
               for standard Internet Protocols for help in protocol
               processing.";
          }
          leaf status-code {
            type identityref{
              base status-code;
            }
            mandatory true;
            description
              "Status code.";
          }
          leaf status-sub-code {
            type identityref{
              base status-sub-code;
            }
            mandatory true;
            description
              "Sub code for CC.";
          }
          description
            "Status code and Sub Code.";
        }

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

    rpc cc-persistent-get-export-details {
      if-feature "cl-oam:continuity-check";
      description
        "Given the persistent id, gets the configuration
         options, details related to the configured data
         export.";
      input {
```

```
        leaf cc-persistent-id {
          type string;
          description
            "Persistent Id for use as a key in search.";
        }
      }
      output {
        container error-code {
          leaf protocol-id {
            type identityref {
              base protocol-id;
            }
            mandatory true;
            description
              "Protocol used. This could be a standard
               protocol (e.g., TCP/IP protocols, MPLS etc.,)
               or a proprietary protocol as identified by
               this field.";
          }
          leaf protocol-id-meta-data {
            type uint64;
            description
              "An optional meta-data related to the protocol ID.
               For e.g., this could be the Internet Protocol number
               for standard Internet Protocols for help in protocol
               processing.";
          }
          leaf status-code {
            type identityref{
              base status-code;
            }
            mandatory true;
            description
              "Status code.";
          }
          leaf status-sub-code {
            type identityref{
              base status-sub-code;
            }
            mandatory true;
            description
              "Sub code for CC.";
          }
          description
            "Status code and Sub Code.";
        }

        leaf data-export-method {
```

```
          type export-method;
          description
            "Type of export in use.";
        }

        choice cc-trigger {
          description
            "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 were started.";
            }
          }
          case on-change {
            description
              "On-change trigger and not periodic.";
            leaf all-data-on-start {
              type boolean;
              description
                "Full update done on start or not.";
            }
            leaf-list excluded-change {
              type change-type;
              description
                "Changes that will not trigger an update.";
            }
          }
        }
      }
    }

   rpc pd-get-persistent-id {
      if-feature "cl-oam:path-discovery";
      description
        "Obtains persistent path discovery identification.";

      input {
        container destination-tp {
```

```
          uses cl-oam:tp-address;
          description
            "Destination test point.";
        }
        uses cl-oam: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 cl-oam:routing-instance-ref;
          description
            "VRF";
        }
      }
      output {
        list response-list {
          key "response-index";
          description
            "Path discovery response list.";
          leaf response-index {
            type uint32;
            mandatory true;
            description
              "Response index.";
          }
          leaf protocol-id {
            type identityref {
              base protocol-id;
            }
            mandatory true;
            description
              "Protocol used. This could be a standard
               protocol (e.g., TCP/IP protocols, MPLS etc.,)
               or a proprietary protocol as identified by
               this field.";
          }
          leaf protocol-id-meta-data {
            type uint64;
            description
              "An optional meta-data related to the protocol ID.
               For e.g., this could be the Internet Protocol number
```

```
              for standard Internet Protocols for help in protocol
              processing.";
          }
          leaf status-code {
            type identityref {
              base status-code;
            }
            mandatory true;
            description
              "Status code for Persistent Path Discovery Information. ";
          }
          leaf status-sub-code {
            type identityref{
              base status-sub-code;
            }
            mandatory true;
            description
              "Sub code for Persistent Path Discovery Information. ";
          }
          leaf pd-persistent-id {
            type string;
            description
              "Id to act as a cookie.";
          }
        }
      }
    }

  rpc pd-persistent-get-export-details {
     if-feature "cl-oam:path-discovery";
     description
       "Given the persistent id, gets the configuration
        options, details related to the configured data
        export.";
     input {
       leaf cc-persistent-id {
         type string;
         description
           "Persistent ID for use as a key in search.";
       }
     }

     output {
       list response-list {
         key "response-index";
         description
           "Path discovery response list.";
         leaf response-index {
```

```
            type uint32;
            mandatory true;
            description
              "Response index.";
          }
          leaf protocol-id {
            type identityref {
              base protocol-id;
            }
            mandatory true;
            description
              "Protocol used. This could be a standard
               protocol (e.g., TCP/IP protocols, MPLS etc.,)
               or a proprietary protocol as identified by
               this field.";
          }
          leaf protocol-id-meta-data {
             type uint64;
             description
               "An optional meta-data related to the protocol ID.
                For e.g., this could be the Internet Protocol number
                for standard Internet Protocols for help in protocol
                processing.";
          }
          leaf status-code {
            type identityref{
              base status-code;
            }
            mandatory true;
            description
              "Status code for Persistent Path Discovery Creation. ";
          }
          leaf status-sub-code {
            type identityref{
              base status-sub-code;
            }
            mandatory true;
            description
              "Sub code for Persistent Path Discovery Creation. ";
          }
          leaf data-export-method {
            type export-method;
            description
              "Type of export.";
          }
          choice pd-trigger {
            description
              "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 done on start or not.";
              }
              leaf-list excluded-change {
                type change-type;
                description
                  "Changes that will not trigger an update.";
              }
            }
          }
        }
      }
    }
  }
```

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 (editor)
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: January 9, 2017

D. Kumar
Cisco
Q. Wu
M. Wang
Huawei

July 8, 2016

Generic YANG Data Model for Connection Oriented Operations, Administration, and Maintenance(OAM) protocols
draft-ietf-lime-yang-oam-model-07

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 January 9, 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][RFC7276]. Over the years, many technologies have developed similar tools for fault and performance management.

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

Given the wide adoption of the underlying OAM concepts defined in [IEEE802.1Q][IEEE802.1Q] CFM, it is a reasonable choice to develop the unified management framework for connection oriented OAM based on those concepts. In this document, we take the [IEEE802.1Q][IEEE802.1Q] CFM model and extend it to a technology independent framework and 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 (Loopback) 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 [IEEE802.1Q] CFM. 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]. 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
     x  for deprecated
     o  for obsolete


<flags> is one of:

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


<name> is the name of the node

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

<opts> is one of:

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

In this document, these words will appear with that interpretation only when in ALL CAPS.

2.1. Terminology

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

ECMP - Equal Cost Multipath.

LBM - Loopback Message [IEEE802.1Q].

MP - Maintenance Point [IEEE802.1Q].

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

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

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

MD - Maintenance Domain [IEEE802.1Q]

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] [IEEE802.1Q].

RPC - Remote Process Call.

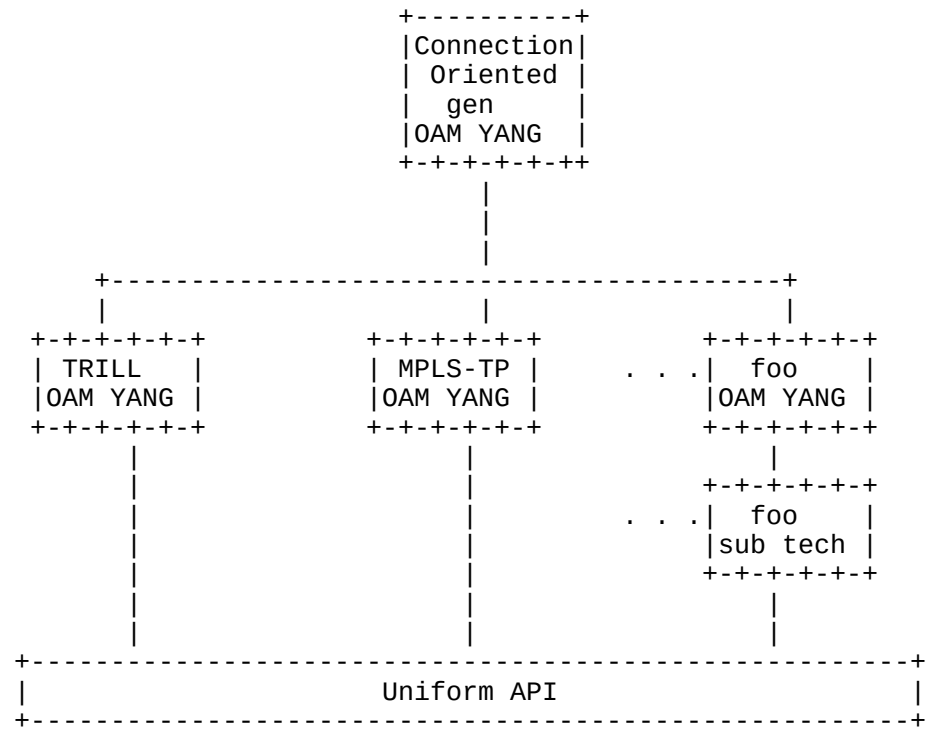
CC - Continuity Check [RFC7276]. Continuity Checks are used to verify that a destination is reachable and therefore also referred to as reachability verification.

CV - Connectivity Verification [RFC7276].Connectivity 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 [IEEE802.1Q] CFM 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.

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

In the vertical direction orthogonal to the Maintenance Domain, presented are the commands. Those, in YANG terms, are the rpc

commands. These rpc commands provide uniform APIs for continuity check, connectivity verification(loopback), path discovery(traceroute) and their equivalents as well as other OAM commands.

The generic YANG model defined here does not require explicit configuration of OAM entities prior to using any of the OAM tools.The OAM tools used here are limited to OAM toolset specified in section 5.1 of [RFC7276]. 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 [IEEE802.1Q] or other standards can be included by association of the MD-name-format with an identity-ref. MD-name-format indicates the format of the augmented MD-names. MD-name is presented as choice/case construct. Thus, it is easily augmentable by derivative work.

```
 module: ietf-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 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 (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 (MA-ID)?
               |  +--:(MA-id)
               |  |  +--rw MA-id?            uint32
               |  +--:(MEG-ID)
               |     +--rw meg-id?           string
               +--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]
                  +--rw interface    if:interface-ref
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 (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-interfaces {
    prefix if;
  }
  import ietf-yang-types {
    prefix yang;
  }
  import ietf-inet-types {
    prefix inet;
  }

  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 2016-03-15 {
    description
      "Initial revision. - 05 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.";
   }

  /* 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 name-format {
```

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

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

  identity identifier-format {
    description
      "identifier-format identity can be augmented to define other
     format identifiers used in 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-V 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
     equal to 3.5 times the receiving MEP's
     configured CC-V reception period. ";
   }

  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 MEG-ID{
    leaf meg-id{
      type string;
      description
       "concatenation of domain and ma, For example a co-routed
        bidirectional LSP, MEG_ID is A1-{Global_ID::Node_ID::
        Tunnel_Num}::Z9-{Global_ID::Node_ID::Tunnel_Num}::LSP_Num.";
    }
    description
      "MEG-ID grouping.";
  }

  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 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 MIP {
    description
      "defines MIP";
```

```
    leaf interface {
      type if:interface-ref;
      description
         "Interface";
    }
  }

  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";
    }
  }

  container domains {
    description
      "Contains configuration related data. Within the container
       is list of fault domains. Wihin each domian has List of MA.";
    list domain {
      key "technology MD-name-string";

      ordered-by system;
      description
```

```
        "Define the list of Domains within the IETF-OAM";
      uses maintenance-domain-id;
      uses MD-name;
      leaf md-level {
        type MD-level;
        description
          "Defines the MD-Level";
      }
      container MAs {
        description
          "This container defines MA, within that have multiple MA
           and within MA have MEP, MIP";
        list MA {
          key "MA-name-string";
          ordered-by system;
          uses ma-identifier;
          uses MA-name;
          choice MA-ID{
               case MA-id{
                leaf MA-id{
                type uint32;
                description
                "MA Identifier";
                }
               description
               "MA ID case";
      }
               case MEG-ID{
                uses MEG-ID;
                description
        "In case MPLS-TP, the MA equivalent to MEG";
                }
               description
      "The MA/MEG identifier";
              }
          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 {
            key "interface";
            uses MIP;
            description
            "Maintenance Intermediate Point";
          }
          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 "64..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";
          }
        }
        uses monitor-stats {
          description
            "Stats of traceroute.";
        }
      description
        "List of response.";
      }
    }
  }
}
```

YANG module of OAM

<CODE ENDS>

6. Base Mode

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

6.1. MEP Address

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

6.2. MEP ID for Base Mode

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

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

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

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

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

6.3. Maintenance Association

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

7. 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. Meg-Id parameter under MA data node will be selected for MPLT-TP OAM model. Therefore 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.

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

[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.

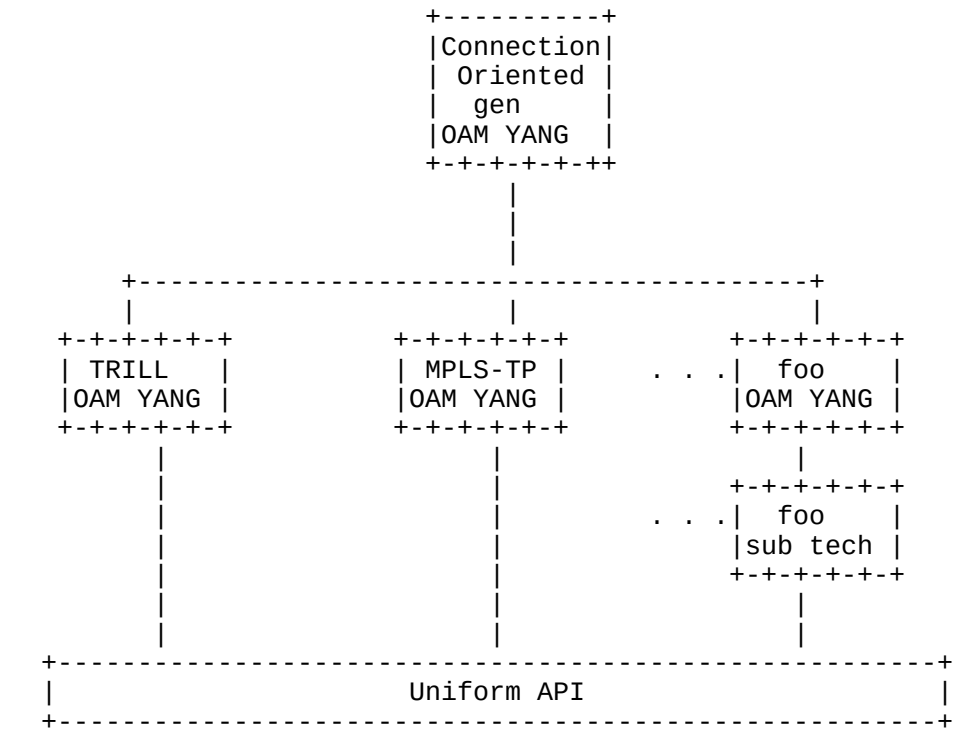
Proactive OAM - The proactive OAM refers to OAM actions which are carried out continuously to permit proactive reporting of fault. Proactive OAM method requires persistent configuration.

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