Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 20, 2017

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

July 19, 2016

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

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 Base 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).

Table of Contents

1. Introduction 2
2. Conventions used in this document 3
2.1. Terminology 4
2.2. Tree Diagrams 4
3. Overview of the Connectionless OAM Model 5
3.1. TP Address 5
3.2. Tools 6
3.3. OAM-layers 6
3.4. Test Point Locations Information 7
3.5. Test Point Locations 7
3.6. Path discovery data 7
3.7. Continuity check data 8
3.8. RPC definitions 8
3.9. Relation with other OAM YANG Model 11
3.10. OAM data hierarchy 11
4. OAM YANG Module 29
5. Security Considerations 60
6. IANA Considerations 60
7. Normative References 60
Authors' Addresses 61

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 two modules, one to represent the base independent and stand-alone YANG data model for connectionless OAM protocols and the other one focuses on data retrival procedures like RPCs. The split module approach avoids mixing the models for the retrieved-data from the retrieval procedures. It is expected that retrieval procedures would evolve faster than the data model and will allow new procedures to be defined for retrieval of the same data defined by the base data model. This also allows the data model to change at its own pace.

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

TLV - Type Length Value

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 topology nodes. 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 or stiched 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
- TLV address (RFC6428 (Figure 4,5, and 6))
- 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.

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.

3.8. RPC definitions

The rpc model described here as a separate model outside of the data 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

3.9. Relation with other OAM YANG Model

In this document we define a generic YANG data model for connectionless OAM protocols. The other model defined adds generic data-retrieval methods. The YANG data model defined here is generic such that other technologies can extend it for technology specific needs. The Generic YANG model acts as the root for other OAM YANG models. This allows users to traverse between different OAM protocols at ease through a uniform API set. The Generic YANG model for OAM provides a framework where technology- specific YANG models can choose to inherit constructs from the base YANG models without needing to redefine them within the sub-technology.

3.10. 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
      |        +--rw (tools)?
      |        |  +--:(tools-empty)
      |        |  |  +--rw tools-null?      empty
      |        |  +--:(tools-ip)
      |        |  |  +--rw rfc792?          boolean
      |        |  |  +--rw rfc4443?         boolean
      |        |  |  +--rw rfc4884?         boolean
      |        |  |  +--rw rfc5837?         boolean
      |        |  +--:(tools-bfd)
      |        |  |  +--rw rfc5881?         boolean
      |        |  |  +--rw rfc5883?         boolean
      |        |  |  +--rw rfc5884?         boolean
      |        |  |  +--rw rfc5885?         boolean
      |        |  +--:(tools-mpls)
      |        |  |  +--rw rfc4379?         boolean
      |        |  |  +--rw rfc4687?         boolean
      |        |  |  +--rw rfc4950?         boolean
      |        |  |  +--rw mpls-rfc5884?    boolean
      |        |  +--:(tools-mpls-tp)
      |        |  |  +--rw rfc6426?         boolean
      |        |  |  +--rw rfc6435?         boolean
      |        |  |  +--rw rfc6374?         boolean
      |        |  +--:(tools-pw)
```

```
|        |     +--rw rfc5085?         boolean
|        |     +--rw pw_rfc5885?      boolean
|        |     +--rw rfc6423?         boolean
|        |     +--rw rfc6310?         boolean
|        |     +--rw 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
|        +--rw (tools)?
|        |  +--:(tools-empty)
|        |  |  +--rw tools-null?      empty
|        |  +--:(tools-ip)
|        |  |  +--rw rfc792?          boolean
|        |  |  +--rw rfc4443?         boolean
|        |  |  +--rw rfc4884?         boolean
|        |  |  +--rw rfc5837?         boolean
|        |  +--:(tools-bfd)
|        |  |  +--rw rfc5881?         boolean
|        |  |  +--rw rfc5883?         boolean
|        |  |  +--rw rfc5884?         boolean
|        |  |  +--rw rfc5885?         boolean
|        |  +--:(tools-mpls)
|        |  |  +--rw rfc4379?         boolean
|        |  |  +--rw rfc4687?         boolean
|        |  |  +--rw rfc4950?         boolean
|        |  |  +--rw mpls-rfc5884?    boolean
|        |  +--:(tools-mpls-tp)
|        |  |  +--rw rfc6426?         boolean
|        |  |  +--rw rfc6435?         boolean
|        |  |  +--rw rfc6374?         boolean
|        |  +--:(tools-pw)
|        |     +--rw rfc5085?         boolean
|        |     +--rw pw_rfc5885?      boolean
|        |     +--rw rfc6423?         boolean
|        |     +--rw rfc6310?         boolean
|        |     +--rw 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
|        +--rw (tools)?
|        |  +--:(tools-empty)
|        |  |  +--rw tools-null?             empty
|        |  +--:(tools-ip)
|        |  |  +--rw rfc792?                 boolean
|        |  |  +--rw rfc4443?                boolean
|        |  |  +--rw rfc4884?                boolean
|        |  |  +--rw rfc5837?                boolean
|        |  +--:(tools-bfd)
|        |  |  +--rw rfc5881?                boolean
|        |  |  +--rw rfc5883?                boolean
|        |  |  +--rw rfc5884?                boolean
|        |  |  +--rw rfc5885?                boolean
|        |  +--:(tools-mpls)
|        |  |  +--rw rfc4379?                boolean
|        |  |  +--rw rfc4687?                boolean
|        |  |  +--rw rfc4950?                boolean
|        |  |  +--rw mpls-rfc5884?           boolean
|        |  +--:(tools-mpls-tp)
|        |  |  +--rw rfc6426?                boolean
|        |  |  +--rw rfc6435?                boolean
|        |  |  +--rw rfc6374?                boolean
|        |  +--:(tools-pw)
|        |     +--rw rfc5085?                boolean
|        |     +--rw pw_rfc5885?             boolean
|        |     +--rw rfc6423?                boolean
|        |     +--rw rfc6310?                boolean
|        |     +--rw 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
|        +--rw (tools)?
|        |  +--:(tools-empty)
|        |  |  +--rw tools-null?        empty
|        |  +--:(tools-ip)
|        |  |  +--rw rfc792?            boolean
|        |  |  +--rw rfc4443?           boolean
|        |  |  +--rw rfc4884?           boolean
|        |  |  +--rw rfc5837?           boolean
|        |  +--:(tools-bfd)
|        |  |  +--rw rfc5881?           boolean
|        |  |  +--rw rfc5883?           boolean
|        |  |  +--rw rfc5884?           boolean
|        |  |  +--rw rfc5885?           boolean
|        |  +--:(tools-mpls)
|        |  |  +--rw rfc4379?           boolean
|        |  |  +--rw rfc4687?           boolean
|        |  |  +--rw rfc4950?           boolean
|        |  |  +--rw mpls-rfc5884?      boolean
|        |  +--:(tools-mpls-tp)
|        |  |  +--rw rfc6426?           boolean
|        |  |  +--rw rfc6435?           boolean
|        |  |  +--rw rfc6374?           boolean
|        |  +--:(tools-pw)
|        |     +--rw rfc5085?           boolean
|        |     +--rw pw_rfc5885?        boolean
|        |     +--rw rfc6423?           boolean
|        |     +--rw rfc6310?           boolean
|        |     +--rw 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
|        +--rw (tools)?
|        |  +--:(tools-empty)
|        |  |  +--rw tools-null?          empty
|        |  +--:(tools-ip)
```

```
|        |  |  +--rw rfc792?              boolean
|        |  |  +--rw rfc4443?             boolean
|        |  |  +--rw rfc4884?             boolean
|        |  |  +--rw rfc5837?             boolean
|        |  +--:(tools-bfd)
|        |  |  +--rw rfc5881?             boolean
|        |  |  +--rw rfc5883?             boolean
|        |  |  +--rw rfc5884?             boolean
|        |  |  +--rw rfc5885?             boolean
|        |  +--:(tools-mpls)
|        |  |  +--rw rfc4379?             boolean
|        |  |  +--rw rfc4687?             boolean
|        |  |  +--rw rfc4950?             boolean
|        |  |  +--rw mpls-rfc5884?        boolean
|        |  +--:(tools-mpls-tp)
|        |  |  +--rw rfc6426?             boolean
|        |  |  +--rw rfc6435?             boolean
|        |  |  +--rw rfc6374?             boolean
|        |  +--:(tools-pw)
|        |     +--rw rfc5085?             boolean
|        |     +--rw pw_rfc5885?          boolean
|        |     +--rw rfc6423?             boolean
|        |     +--rw rfc6310?             boolean
|        |     +--rw 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
|        +--rw (tools)?
|        |  +--:(tools-empty)
|        |  |  +--rw tools-null?            empty
|        |  +--:(tools-ip)
|        |  |  +--rw rfc792?                boolean
|        |  |  +--rw rfc4443?               boolean
|        |  |  +--rw rfc4884?               boolean
|        |  |  +--rw rfc5837?               boolean
|        |  +--:(tools-bfd)
|        |  |  +--rw rfc5881?               boolean
|        |  |  +--rw rfc5883?               boolean
```

```
|        |  |  +--rw rfc5884?              boolean
|        |  |  +--rw rfc5885?              boolean
|        |  +--:(tools-mpls)
|        |  |  +--rw rfc4379?              boolean
|        |  |  +--rw rfc4687?              boolean
|        |  |  +--rw rfc4950?              boolean
|        |  |  +--rw mpls-rfc5884?         boolean
|        |  +--:(tools-mpls-tp)
|        |  |  +--rw rfc6426?              boolean
|        |  |  +--rw rfc6435?              boolean
|        |  |  +--rw rfc6374?              boolean
|        |  +--:(tools-pw)
|        |     +--rw rfc5085?              boolean
|        |     +--rw pw_rfc5885?           boolean
|        |     +--rw rfc6423?              boolean
|        |     +--rw rfc6310?              boolean
|        |     +--rw 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
|        +--rw (tools)?
|        |  +--:(tools-empty)
|        |  |  +--rw tools-null?                 empty
|        |  +--:(tools-ip)
|        |  |  +--rw rfc792?                     boolean
|        |  |  +--rw rfc4443?                    boolean
|        |  |  +--rw rfc4884?                    boolean
|        |  |  +--rw rfc5837?                    boolean
|        |  +--:(tools-bfd)
|        |  |  +--rw rfc5881?                    boolean
|        |  |  +--rw rfc5883?                    boolean
|        |  |  +--rw rfc5884?                    boolean
|        |  |  +--rw rfc5885?                    boolean
|        |  +--:(tools-mpls)
|        |  |  +--rw rfc4379?                    boolean
|        |  |  +--rw rfc4687?                    boolean
|        |  |  +--rw rfc4950?                    boolean
|        |  |  +--rw mpls-rfc5884?               boolean
```

```
|        |  +--:(tools-mpls-tp)
|        |  |  +--rw rfc6426?                    boolean
|        |  |  +--rw rfc6435?                    boolean
|        |  |  +--rw rfc6374?                    boolean
|        |  +--:(tools-pw)
|        |     +--rw rfc5085?                    boolean
|        |     +--rw pw_rfc5885?                 boolean
|        |     +--rw rfc6423?                    boolean
|        |     +--rw rfc6310?                    boolean
|        |     +--rw 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
|        +--rw (tools)?
|        |  +--:(tools-empty)
|        |  |  +--rw tools-null?           empty
|        |  +--:(tools-ip)
|        |  |  +--rw rfc792?               boolean
|        |  |  +--rw rfc4443?              boolean
|        |  |  +--rw rfc4884?              boolean
|        |  |  +--rw rfc5837?              boolean
|        |  +--:(tools-bfd)
|        |  |  +--rw rfc5881?              boolean
|        |  |  +--rw rfc5883?              boolean
|        |  |  +--rw rfc5884?              boolean
|        |  |  +--rw rfc5885?              boolean
|        |  +--:(tools-mpls)
|        |  |  +--rw rfc4379?              boolean
|        |  |  +--rw rfc4687?              boolean
|        |  |  +--rw rfc4950?              boolean
|        |  |  +--rw mpls-rfc5884?         boolean
|        |  +--:(tools-mpls-tp)
|        |  |  +--rw rfc6426?              boolean
|        |  |  +--rw rfc6435?              boolean
|        |  |  +--rw rfc6374?              boolean
|        |  +--:(tools-pw)
|        |     +--rw rfc5085?              boolean
|        |     +--rw pw_rfc5885?           boolean
```

```
|        |     +--rw rfc6423?              boolean
|        |     +--rw rfc6310?              boolean
|        |     +--rw 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
|        +--rw (tools)?
|        |  +--:(tools-empty)
|        |  |  +--rw tools-null?        empty
|        |  +--:(tools-ip)
|        |  |  +--rw rfc792?            boolean
|        |  |  +--rw rfc4443?           boolean
|        |  |  +--rw rfc4884?           boolean
|        |  |  +--rw rfc5837?           boolean
|        |  +--:(tools-bfd)
|        |  |  +--rw rfc5881?           boolean
|        |  |  +--rw rfc5883?           boolean
|        |  |  +--rw rfc5884?           boolean
|        |  |  +--rw rfc5885?           boolean
|        |  +--:(tools-mpls)
|        |  |  +--rw rfc4379?           boolean
|        |  |  +--rw rfc4687?           boolean
|        |  |  +--rw rfc4950?           boolean
|        |  |  +--rw mpls-rfc5884?      boolean
|        |  +--:(tools-mpls-tp)
|        |  |  +--rw rfc6426?           boolean
|        |  |  +--rw rfc6435?           boolean
|        |  |  +--rw rfc6374?           boolean
|        |  +--:(tools-pw)
|        |     +--rw rfc5085?           boolean
|        |     +--rw pw_rfc5885?        boolean
|        |     +--rw rfc6423?           boolean
|        |     +--rw rfc6310?           boolean
|        |     +--rw 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
               +--rw (tools)?
               |  +--:(tools-empty)
               |  |  +--rw tools-null?          empty
               |  +--:(tools-ip)
               |  |  +--rw rfc792?              boolean
               |  |  +--rw rfc4443?             boolean
               |  |  +--rw rfc4884?             boolean
               |  |  +--rw rfc5837?             boolean
               |  +--:(tools-bfd)
               |  |  +--rw rfc5881?             boolean
               |  |  +--rw rfc5883?             boolean
               |  |  +--rw rfc5884?             boolean
               |  |  +--rw rfc5885?             boolean
               |  +--:(tools-mpls)
               |  |  +--rw rfc4379?             boolean
               |  |  +--rw rfc4687?             boolean
               |  |  +--rw rfc4950?             boolean
               |  |  +--rw mpls-rfc5884?        boolean
               |  +--:(tools-mpls-tp)
               |  |  +--rw rfc6426?             boolean
               |  |  +--rw rfc6435?             boolean
               |  |  +--rw rfc6374?             boolean
               |  +--:(tools-pw)
               |     +--rw rfc5085?             boolean
               |     +--rw pw_rfc5885?          boolean
               |     +--rw rfc6423?             boolean
               |     +--rw rfc6310?             boolean
               |     +--rw rfc7023?             boolean
               +--rw oam-layers* [index]
                  +--rw index    uint16
                  +--rw level?   int32


module: ietf-connectionless-oam-methods
rpcs:
   +---x continuity-check {coam:continuity-check}?
   |  +---w input
   |  |  +---w destination-tp
   |  |  |  +---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
   |  |  |     |     +--:(l3vpn)
   |  |  |     |     |  +---w l3vpn-id?              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-Ad
dress
   |  |  |     |           +--:(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)?
   |     |  |  +--:(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
   |     |  |  |     +--:(l3vpn)
   |     |  |  |     |  +--ro l3vpn-id?             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-Ad
dress
   |     |  |  |           +--:(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)?
   |     |  |  +--:(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
   |     |  |  |     +--:(l3vpn)
   |     |  |  |     |  +--ro l3vpn-id?              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-Ad
dress
   |     |  |  |           +--:(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)?
   |  |     +--:(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
      |  |     |     +--:(l3vpn)
      |  |     |     |  +---w l3vpn-id?              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-Ad
dress
      |  |     |           +--:(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)?
         |     +--:(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
          |     |     +--:(l3vpn)
          |     |     |  +--ro l3vpn-id?              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-Ad
dress
          |     |           +--:(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)?
          |     +--:(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
          |     |     +--:(l3vpn)
          |     |     |  +--ro l3vpn-id?              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-Ad
dress
          |     |           +--:(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)?
      |  +--:(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
      |  |     +--:(l3vpn)
      |  |     |  +--ro l3vpn-id?              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 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

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";
    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 2015-12-22 {
      description
        "Initial revision. - 01 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, l3vpn, 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 l3vpn {
              leaf l3vpn-id {
                type uint32;
                            description
              "FEC layer 3 vpn.";
              }
            }
```

```
            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;
           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-mpls-tp {
          description
            "Oam Toolset for mpls TP.";
          leaf rfc6426 {
            type boolean;
                description
                "rfc6426 supported.";
          }
          leaf rfc6435 {
            type boolean;
                description
                "rfc6435 supported.";
          }
          leaf rfc6374 {
            type boolean;
                description
```

```
                    "rfc6374 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;
      }
    }
}

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";
```

```
    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 2015-12-22 {
      description
        "Initial revision. - 01 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