Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 3, 2018

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

October 30, 2017

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

draft-ietf-lime-yang-connectionless-oam-16

Abstract

This document presents a base YANG Data model for Operations Administration, and Maintenance(OAM) protocols that use Connectionless Communications. The data model is defined using the YANG in RFC7950 data modeling language. It provides a technology-independent abstraction of key OAM constructs for OAM protocols that use connectionless communication. The base model presented here can be extended to include technology specific details. This is leading to uniformity between OAM protocols and support both nested OAM workflows (i.e., performing OAM functions at different or same levels through a unified interface) and interacting OAM workflows (i.e., performing OAM functions at same levels through a unified interface).

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

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

1. Monitor networks communication (Reachability Verification, Continuity Check).

2. Troubleshoot failures (Fault verification and localization).

3. Monitor Performance

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

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

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

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

2. Conventions used in this document

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

- client

- configuration data

o server

o state data

The following terms are defined in [RFC7950] and are not redefined here:

o augment

o data model

o data node

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

2.1. Abbreviations

BFD - Bidirectional Forwarding Detection [RFC5880].

RPC - A Remote Procedure Call [RFC1831].

DSCP - Differentiated Services Code Point.

VRF - Virtual Routing and Forwarding (VRF) [RFC 4382].

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

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

AS - Autonomous System.

LSP - Label Switched Path.

TE - Traffic Engineering.

MPLS - Multiprotocol Label Switching.

NI - Network Instance.

PTP - Precision Time Protocol [IEEE.1588].

NTP - Network Time Protocol [RFC5905].

2.2. Terminology

MAC address- Address for data link layer interface.

TP - Test Point. Test point is a functional entity that is defined at a node in the network and can initiate and/or react to OAM diagnostic test. This document focuses on the data-plane functionality of TPs, while TPs interact with the control plane and with the management plane as well.

RPC operation - A specific Remote Procedure Call.

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

3. Overview of the Connectionless OAM Model

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

- o The module ietf-lime-common-types.yang provides common definitions such as Time-specific data types and Timestamp specific data types.

- o The module ietf-connectionless-oam.yang defines technology-independent abstraction of key OAM constructs for OAM protocols that use Connectionless communication.

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

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

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

- o 'tp-technology' grouping,

- o 'tp-tools' grouping,

- o and 'connectionless-oam-tps' grouping.

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

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

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

3.1. TP Address

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

- o MAC address [RFC6136] at link layer for TPs

- o IPv4 or IPv6 address at IP layer for TPs

- o TP-attribute identifying a TP associated with an application layer function

- o Router-id to represent the device or node, which is commonly used to identify nodes in routing and other control plane protocols.[I-D.ietf-rtgwg-routing-types]

To define a forwarding treatment of a test packet, the 'tp-address' grouping needs to be associated with additional parameters, e.g., DSCP for IP or EXP (renamed to Traffic Classic in [RFC5462]) for MPLS. In generic connectionless OAM YANG model, these parameters are not explicitly configured. The model user can add corresponding parameters according to their requirements.

3.2. Tools

The different OAM tools may be used in one of two basic types of activation: proactive and on-demand. The proactive OAM refers to OAM actions which are carried out continuously to permit proactive reporting of fault. The proactive OAM method requires persistent

configuration. The on-demand OAM refers to OAM actions which are initiated via manual intervention for a limited time to carry out diagnostics. The on-demand OAM method requires only transient configuration.[RFC7276] [G.8013]. In connectionless OAM, 'session-type' grouping is defined to indicate which kind of activation will be used by the current session.

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

3.3. OAM neighboring test points

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

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

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

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

              }
```

3.4. Test Point Locations Information

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

3.5. Test Point Locations

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

3.6. Path Discovery Data

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

3.7. Continuity Check Data

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

3.8. OAM data hierarchy

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

```
 module: ietf-connectionless-oam
     +--ro cc-session-statistics-data {continuity-check}?
        +--ro cc-session-statistics* [type]
           +--ro type                             identityref
           +--ro cc-ipv4-sessions-statistics
           |  +--ro cc-session-statistics
           |     +--ro session-count?              uint32
           |     +--ro session-up-count?           uint32
           |     +--ro session-down-count?         uint32
           |     +--ro session-admin-down-count?   uint32
           +--ro cc-ipv6-sessions-statistics
              +--ro cc-session-statistics
                 +--ro session-count?              uint32
                 +--ro session-up-count?           uint32
                 +--ro session-down-count?         uint32
```

```
                 +--ro session-admin-down-count?   uint32
     augment /nd:networks/nd:network/nd:node:
       +--rw tp-location-type?                identityref
       +--rw ipv4-location-type
       |  +--rw test-point-ipv4-location-list
       |     +--rw test-point-locations* [ipv4-location ni]
       |        +--rw ipv4-location          inet:ipv4-address
       |        +--rw ni                     routing-instance-ref
       |        +--rw (technology)?
       |        |  +--:(technology-null)
       |        |     +--rw tech-null?             empty
       |        +--rw tp-tools
       |        |  +--rw continuity-check    boolean
       |        |  +--rw path-discovery      boolean
       |        +--rw root?                  <anydata>
       |        +--rw oam-neighboring-tps* [index]
       |           +--rw index                    uint16
       |           +--rw position?                int8
       |           +--rw (tp-location)?
       |              +--:(mac-address)
       |              |  +--rw mac-address-location?    yang:mac-address
       |              +--:(ipv4-address)
       |              |  +--rw ipv4-address-location?   inet:ipv4-address
       |              +--:(ipv6-address)
       |              |  +--rw ipv6-address-location?   inet:ipv6-address
       |              +--:(as-number)
       |              |  +--rw as-number-location?      inet:as-number
       |              +--:(router-id)
       |                 +--rw router-id-location?      rt:router-id
       +--rw ipv6-location-type
       |  +--rw test-point-ipv6-location-list
       |     +--rw test-point-locations* [ipv6-location ni]
       |        +--rw ipv6-location          inet:ipv6-address
       |        +--rw ni                     routing-instance-ref
       |        +--rw (technology)?
       |        |  +--:(technology-null)
       |        |     +--rw tech-null?             empty
       |        +--rw tp-tools
       |        |  +--rw continuity-check    boolean
       |        |  +--rw path-discovery      boolean
       |        +--rw root?                  <anydata>
       |        +--rw oam-neighboring-tps* [index]
       |           +--rw index                    uint16
       |           +--rw position?                int8
       |           +--rw (tp-location)?
       |              +--:(mac-address)
       |              |  +--rw mac-address-location?    yang:mac-address
       |              +--:(ipv4-address)
```

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

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

4. LIME Time Types YANG Module

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

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

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

```
        Working Group";

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

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

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

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

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

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

   <CODE ENDS>
```

5. Connectionless OAM YANG Module

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

module ietf-connectionless-oam {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-connectionless-oam";
  prefix cl-oam;
  import ietf-yang-schema-mount {
```

```
     prefix yangmnt;
    }
    import ietf-network {
      prefix nd;
    }
    import ietf-yang-types {
      prefix yang;
    }
    import ietf-interfaces {
      prefix if;
    }
    import ietf-inet-types {
      prefix inet;
    }
    import ietf-network-instance {
      prefix ni;
    }
    import ietf-routing-types {
     prefix rt;
    }
    import ietf-lime-time-types {
     prefix lime;
    }
    organization
      "IETF LIME Working Group";
    contact
      "Deepak Kumar dekumar@cisco.com
       Qin Wu bill.wu@huawei.com
       S Raghavan srihari@cisco.com
       Zitao Wang wangzitao@huawei.com
       R Rahman rrahman@cisco.com";
    description
      "This YANG module defines the generic configuration,
       data model, and statistics for OAM protocols using
       connectionless communications, described in a
       protocol independent manner.It is assumed that each
       protocol maps corresponding abstracts to its native
       format. Each protocol mayextend the YANG model defined
       here to include protocol specific extensions.";
    revision 2017-09-06 {
      description
        " Base model for Connectionless
         Operations, Administration,
         and Maintenance(OAM) ";
      reference
        " RFC XXXX: Connectionless
         Operations, Administration, and
         Maintenance(OAM)YANG Data Model";
```

```
  }
  feature connectionless {
    description
      "This feature indicates that OAM solution is connectionless.";
  }
  feature continuity-check {
    description
      "This feature indicates that the server supports
       executing continuity check OAM command and
       returning a response. Servers that do not advertise
       this feature will not support executing
       continuity check command or RPC operation model for
       continuity check command.";
  }
  feature path-discovery {
    description
      "This feature indicates that the server supports
       executing path discovery OAM command and
       returning a response. Servers that do not advertise
       this feature will not support executing
       path discovery command or RPC operation model for
       path discovery command.";
  }
  feature ptp-long-format {
    description
      "This feature indicates that timestamp is PTP long format.";
  }
  feature ntp-short-format {
    description
      "This feature indicates that timestamp is NTP short format.";
  }
  feature icmp-timestamp {
    description
      "This feature indicates that timestamp is ICMP timestamp.";
  }
  identity traffic-type {
   description
   "This is base identity of traffic type
   which include IPv4 and IPv6,etc.";
  }
  identity ipv4 {
  base traffic-type;
  description
   "identity for IPv4 traffic type.";
 }
  identity ipv6 {
  base traffic-type;
   description
```

```
     "identity for IPv4 traffic type.";
   }
    identity address-attribute-types {
      description
        "This is base identity of address
         attribute types which are Generic
         IPv4/IPv6 Prefix,BGP Labeled
         IPv4/IPv6 Prefix,Tunnel ID,
         PW ID, vpls VE ID, etc.(See RFC8029
         for details.)";
    }
    typedef address-attribute-type {
      type identityref {
        base address-attribute-types;
      }
      description
        "Target address attribute type.";
    }
    typedef percentage {
        type decimal64 {
           fraction-digits 5;
           range "0..100";
       }
     description "Percentage";
     }
    typedef routing-instance-ref {
      type leafref {
        path "/ni:network-instances/ni:network-instance/ni:name";
      }
      description
        "This type is used for leafs that reference a routing instance
         configuration.";
    }
    grouping cc-session-statistics {
      description
        "Grouping for session statistics.";
      container cc-session-statistics {
        description
          "cc session counters";
        leaf session-count {
          type uint32;
          default "0";
          description
            "Number of Continuity Check sessions.
            A value of zero indicates that no session
            count is sent.";
        }
        leaf session-up-count {
```

```
        type uint32;
        default "0";
        description
          "Number of sessions which are up.
          A value of zero indicates that no up
          session count is sent.";
      }
      leaf session-down-count {
        type uint32;
        default "0";
        description
          "Number of sessions which are down.
         A value of zero indicates that no down
         session count is sent.";
      }
      leaf session-admin-down-count {
        type uint32;
       default "0";
        description
          "Number of sessions which are admin-down.
          A value of zero indicates that no admin
          down session count is sent.";
      }
    }
  }
  grouping session-packet-statistics {
    description
      "Grouping for per session packet statistics";
    container session-packet-statistics {
      description
        "Per session packet statistics.";

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

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

      leaf create-time {
        type yang:date-and-time;
        description
```

```
          "Time and date when session is created.";
      }
      leaf last-down-time {
        type yang:date-and-time;
        description
          "Time and date last time session is down.";
      }
      leaf last-up-time {
        type yang:date-and-time;
        description
          "Time and date last time session is up.";
      }
      leaf down-count {
        type uint32 {
        range "0..4294967295";
        }
        default "0";
        description
          "Total Continuity Check sessions down count.
           The value of count will be set to zero (0)
           on creation and will thereafter increase
           monotonically until it reaches a maximum value
           of 2^32-1 (4294967295 decimal), when it wraps
           around and starts increasing again from zero.";
      }
      leaf admin-down-count {
        type uint32 {
        range "0..4294967295";
        }
        default "0";
        description
          "Total Continuity Check sessions admin down count.
           The value of count will be set to zero (0)
           on creation and will thereafter increase
           monotonically until it reaches a maximum value
           of 2^32-1 (4294967295 decimal), when it wraps
           around and starts increasing again from zero.";
      }
      uses session-packet-statistics;
    }
  }
  grouping session-error-statistics {
    description
      "Grouping for per session error statistics";
    container session-error-statistics {
      description
        "Per session error statistics.";
      leaf packet-loss-count {
```

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

```
          "Total received packet duplicates count.
          The value of count will be set to zero (0)
           on creation and will thereafter increase
           monotonically until it reaches a maximum value
           of 2^32-1 (4294967295 decimal), when it wraps
           around and starts increasing again from zero.";
      }
    }
  }
  grouping session-delay-statistics {
    description
      "Grouping for per session delay statistics";
    container session-delay-statistics {
      description
        "Session delay summarised information.By default,
         one way measurement protocol (e.g., OWAMP)is used
         to measure delay. When two way measurement protocol
         (e.g., TWAMP) is used instead, it can be indicated
         using and protocol-id defined in RPC operation of
         draft-ietf-lime-yang-connectionless-oam-methods,i.e.,
         set protocol-id as OWAMP. Note that only one measurement
         protocol for delay is specified for interoperability reason.";
      leaf time-unit-value {
        type identityref {
          base lime:time-unit-type;
        }
        default lime:milliseconds;
        description
          "Time units among choice of s,ms,ns etc.";
      }
      leaf min-delay-value {
        type uint32;
        description
          "Minimum delay value observed.";
      }
      leaf max-delay-value {
        type uint32;
        description
          "Maximum delay value observed.";
      }
      leaf average-delay-value {
        type uint32;
        description
          "Average delay value observed.";
      }
    }
  }
  grouping session-jitter-statistics {
```

```
    description
      "Grouping for per session jitter statistics";
    container session-jitter-statistics {
      description
        "Session jitter summarised information. By default,
         jitter is measured using IP Packet Delay Variation
         (IPDV) as defined in RFC3393. When the other measurement
         method is used instead(e.g., Packet Delay Variation used in
         Y.1540, it can be indicated using protocol-id-meta-data
         defined in RPC operation of
         draft-ietf-lime-yang-connectionless-oam-methods. Note that
         only one measurement method for jitter is specified
         for interoperability reason.";
      leaf unit-value {
        type identityref {
          base lime:time-unit-type;
        }
       default lime:milliseconds;
        description
          "Time units among choice of s,ms,ns etc.";
      }
      leaf min-jitter-value {
        type uint32;
        description
          "Minimum jitter value observed.";
      }
      leaf max-jitter-value {
        type uint32;
        description
          "Maximum jitter value observed.";
      }
      leaf average-jitter-value {
        type uint32;
        description
          "Average jitter value observed.";
      }
    }
  }
  grouping session-path-verification-statistics {
    description
      "Grouping for per session path verification statistics";
    container session-path-verification-statistics {
      description
        "OAM per session path verification statistics.";
      leaf verified-count {
        type uint32 {
        range "0..4294967295";
        }
```

```
        description
          "Total number of OAM packets that
           went through a path as intended.
           The value of count will be set to zero (0)
           on creation and will thereafter increase
           monotonically until it reaches a maximum value
           of 2^32-1 (4294967295 decimal), when it wraps
           around and starts increasing again from zero.";
      }
      leaf failed-count {
        type uint32 {
        range "0..4294967295";
        }
        description
          "Total number of OAM packets that
           went through an unintended path.
          The value of count will be set to zero (0)
           on creation and will thereafter increase
           monotonically until it reaches a maximum value
           of 2^32-1 (4294967295 decimal), when it wraps
           around and starts increasing again from zero.";
      }
    }
  }
  grouping session-type {
    description
      "This object indicates which kind
      of activation will be used by the current
      session.";
    leaf session-type {
      type enumeration {
        enum "proactive" {
          description
            "The current session is proactive session.";
        }
        enum "on-demand" {
          description
            "The current session is on-demand session.";
        }
      }
      default "on-demand";
      description
        "Indicate which kind of activation will be used
         by the current session";
    }
  }
  identity tp-address-technology-type {
    description
```

```
      "Test point address type";
  }
  identity mac-address-type {
    base tp-address-technology-type;
    description
      "MAC address type";
  }
  identity ipv4-address-type {
    base tp-address-technology-type;
    description
      "IPv4 address type";
  }
  identity ipv6-address-type {
    base tp-address-technology-type;
    description
      "IPv6 address type";
  }
  identity tp-attribute-type {
    base tp-address-technology-type;
    description

      "Test point attribute type";
  }
  identity router-id-address-type {
    base tp-address-technology-type;
    description
      "System id address type";
  }
  identity as-number-address-type {
    base tp-address-technology-type;
    description
      "AS number address type";
  }
  identity route-distinguisher-address-type {
    base tp-address-technology-type;
    description
      "Route Distinguisher address type";
  }
  grouping tp-address {
    leaf tp-location-type {
      type identityref {
        base tp-address-technology-type;
      }
      mandatory true;
      description
        "Test point address type.";
    }
      container mac-address {
```

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

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

          type inet:ipv6-address;
          mandatory true;
          description
            "IPv6 Address";
        }
        description
          "ipv6 Address based TP Addressing.";
      }
      container tp-attribute {
        when "derived-from-or-self(../tp-location-type,"+
```

```
        "'cl-oam:tp-attribute-type')" {
          description
            "Test point attribute type";
        }
        leaf tp-attribute-type {
          type address-attribute-type;
          description
            "Test point type.";
        }
        choice tp-attribute-value {
          description
            "Test point value.";
          case ip-prefix {
            leaf ip-prefix {
              type inet:ip-prefix;
              description
                "Generic IPv4/IPv6 prefix.See Section 3.2.13 and
                Section 3.2.14 of RFC8029.";
              reference
                "RFC 8029 :Detecting Multi-Protocol Label
                Switched (MPLS) Data Plane Failures";
            }
          }
          case bgp {
            leaf bgp {
              type inet:ip-prefix;
              description
               "BGP Labeled IPv4/IPv6 Prefix.See section
               3.2.11 and section 3.2.12 of RFC8029 for details. ";
              reference
                "RFC 8029 :Detecting Multi-Protocol Label
                Switched (MPLS) Data Plane Failures";
            }
          }
          case tunnel {
            leaf tunnel-interface {
              type uint32;
              description
               "Basic IPv4/IPv6 Tunnel ID. See section 3.2.3
                and Section 3.2.4 of RFC8029 for details.";
              reference
                "RFC 8029 :Detecting Multi-Protocol Label
                Switched (MPLS) Data Plane Failures.";
            }
          }
          case pw {
            leaf remote-pe-address {
              type inet:ip-address;
```

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

```
            choice root-address {
              description
                "Root address choice.";
              case ip-address {
                leaf source-address {
                  type inet:ip-address;
                  description
                    "IP address.";
                }
                leaf group-ip-address {
                  type inet:ip-address;
                  description
                    "Group ip address.";
                }
              }
              case vpn {
                leaf as-number {
                  type inet:as-number;
                  description
                  "The AS number represents autonomous system
                  numbers which identify an Autonomous System.";
                }
              }
              case global-id {
                leaf lsp-id {
                  type string;
                  description
                    "LSP ID is an identifier of a LSP
                     within a MPLS network.";
                  reference
                  "RFC 8029 :Detecting Multi-Protocol Label
                   Switched (MPLS) Data Plane Failures";
                }
              }
            }
          }
        }
        description
         "Test Point Attribute Container";
      }
      container system-info {
        when "derived-from-or-self(../tp-location-type,"+
       "'cl-oam:router-id-address-type')" {
          description
            "System id address type";
        }
        leaf router-id {
          type rt:router-id;
```

```
          description
            "Router ID assigned to this node.";
        }
        description
           "Router ID container.";
      }
    description
      "TP Address";
  }
  grouping tp-address-ni {
    description
      "Test point address with VRF.";
    leaf ni {
      type routing-instance-ref;
      description
        "The ni is used to describe virtual resource partitioning
        that may be present on a network device.Example of common
        industry terms for virtual resource partitioning is VRF
        instance.";
    }
    uses tp-address;
  }
  grouping connectionless-oam-tps {
    list oam-neighboring-tps {
      key "index";
      leaf index {
        type uint16{
        range "0..65535";
       }
        description
          "List of related neighboring test points in adjacent
           layers up and down the stack for the same interface
           that are related to the current test point";
      }
      leaf position {
        type int8 {
          range "-1..1";
        }
        default "0";
        description
          "The relative position
           of neighboring test point
           corresponding to the current
           test point.Level 0 indicates no neighboring
           test points placed before or after the current
           test point in the same layer.-1 means there is
           a neighboring test point placed before the current
           test point in the same layer and +1 means there is
```

```
           a neighboring test point placed after the current
           test point in same layer.";
      }
      choice tp-location {
        case mac-address {
          leaf mac-address-location {
            type yang:mac-address;
            description
              "MAC Address";
          }
          description
            "MAC Address based TP Addressing.";
        }
        case ipv4-address {
          leaf ipv4-address-location {
            type inet:ipv4-address;
            description
              "Ipv4 Address";
          }
          description
            "IP Address based TP Addressing.";
        }
        case ipv6-address {
          leaf ipv6-address-location {
            type inet:ipv6-address;
            description
              "IPv6 Address";
          }
          description
            "IPv6 Address based TP Addressing.";
        }
        case as-number {
          leaf as-number-location {
            type inet:as-number;
            description
              "AS number location";
          }
          description
            "AS number for point to multipoint OAM";
        }
        case router-id {
          leaf router-id-location {
            type rt:router-id;
            description
              "System id location";
          }

          description
```

```
            "System ID";
        }
        description
          "TP location.";
      }
      description
        "List of neighboring test points in the same layer that are related to
current test
         point. If the neighboring test-point is placed after the current test
point, the
         position is specified as +1. If neighboring test-point
         is placed before the current test point, the position is specified
         as -1, if no neighboring test points placed before or after the
current
         test point in the same layer, the position is specified as 0.";
    }
    description
      "Connectionless OAM related neighboring test points list.";
  }
  grouping tp-technology {
    choice technology {
      default "technology-null";
      case technology-null {
        description
          "This is a placeholder when no technology is needed.";
        leaf tech-null {
          type empty;
          description
            "There is no technology to be defined.";
        }
      }
      description
        "Technology choice.";
    }
    description
      "OAM Technology";
  }
  grouping tp-tools {
    description
      "Test Point OAM Toolset.";
    container tp-tools {
      leaf continuity-check {
        type boolean;
        mandatory true;
        description
          "A flag indicating whether or not the
           continuity check function is supported.";
        reference
```

"RFC 792: INTERNET CONTROL MESSAGE PROTOCOL.
RFC 4443: Internet Control Message Protocol (ICMPv6)

```
          for the Internet Protocol Version 6 (IPv6) Specification.
          RFC 5880: Bidirectional Forwarding Detection.
          RFC 5881: BFD for IPv4 and IPv6.
          RFC 5883: BFD for Multihop Paths.

          RFC 5884: BFD for MPLS Label Switched Paths.
          RFC 5885: BFD for PW VCCV.
          RFC 6450: Multicast Ping Protocol.
          RFC 8029: Detecting Multiprotocol Label Switched
         (MPLS) Data-Plane Failures.";
      }
      leaf path-discovery {
        type boolean;
        mandatory true;
        description
         "A flag indicating whether or not the
          path discovery function is supported.";
        reference
         "RFC 792: INTERNET CONTROL MESSAGE PROTOCOL.
          RFC 4443: Internet Control Message Protocol (ICMPv6)
          for the Internet Protocol Version 6 (IPv6) Specification.
          RFC 4884: Extended ICMP to Support Multi-part Message.
          RFC 5837:Extending ICMP for Interface.
          and Next-Hop Identification.
          RFC 8029: Detecting Multiprotocol Label Switched (MPLS)
          Data-Plane Failures.";
      }
      description
        "Container for test point OAM tools set.";
    }
  }
  grouping test-point-location-info {
    uses tp-technology;
    uses tp-tools;
    anydata root {
      yangmnt:mount-point "root";
      description
        "Root for models supported per
         test point";
    }
    uses connectionless-oam-tps;
    description
      "Test point Location";
  }
  grouping test-point-locations {
    description
      "Group of test point locations.";
        leaf tp-location-type {
```

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

```
              description
                "IPv6 Address.";
            }
            leaf ni {
              type routing-instance-ref;
              description
                "The ni is used to describe the
                 corresponding network instance";
            }
            uses test-point-location-info;
            description
              "List of test point locations.";
          }
          description
            "Serves as top-level container
             for test point location list.";
        }
        description
           "ipv6 location type container.";
      }
      container mac-location-type {
       when "derived-from-or-self(../tp-location-type,"+
       "'cl-oam:mac-address-type')" {
       description
       "when test point location type is equal to mac address.";
       }
      container test-point-mac-address-location-list {
          list test-point-locations {
            key "mac-address-location";
            leaf mac-address-location {
              type yang:mac-address;
              description
                "MAC Address";
            }
            uses test-point-location-info;
            description
              "List of test point locations.";
          }
          description
            "Serves as top-level container
             for test point location list.";
        }
        description
           "mac address location type container.";
      }
container group-as-number-location-type {
   when "derived-from-or-self(../tp-location-type,"+
   "'cl-oam:as-number-address-type')" {
```

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

```
            uses test-point-location-info;
            description
              "List of test point locations.";
          }
          description
            "Serves as top-level container for
             test point location list.";
        }
        description
           "system ID location type container.";
      }
  }
  augment "/nd:networks/nd:network/nd:node" {
    description
      "augments the /networks/network/node path defined in the ietf-
      network module (I-D.ietf-i2rs-yang-network-topo) with test-point-
      locations grouping.";
      uses test-point-locations;
  }
  grouping timestamp {
    description
      "Grouping for timestamp.";
    leaf timestamp-type {
      type identityref {
      base lime:timestamp-type;
      }
      description
        "Type of Timestamp, such as Truncated PTP, NTP.";
    }
 container timestamp-64bit {
  when "derived-from-or-self(../timestamp-type, 'cl-oam:truncated-ptp')"+
  "or derived-from-or-self(../timestamp-type,'cl-oam:ntp64')" {
   description
   "Only applies when Truncated NTP or 64bit NTP Timestamp.";
  }
      leaf timestamp-sec {
      type uint32;
      description
        "Absolute timestamp in seconds as per IEEE1588v2
         or seconds part in 64-bit NTP timestamp.";
       }
      leaf timestamp-nanosec {
      type uint32;
      description
        "Fractional part in nanoseconds as per IEEE1588v2
         or Fractional part in 64-bit NTP timestamp.";
      }
      description
```

```
      "Container for 64bit timestamp.";
    }
  container timestamp-80bit {
  when "derived-from-or-self(../timestamp-type, 'cl-oam:ptp80')"{
         description
          "Only applies when 80bit PTP Timestamp.";
        }
   if-feature ptp-long-format;
       leaf timestamp-sec {
       type uint64 {
       range "0..281474976710655";
       }
       description
         "48bit Timestamp in seconds as per IEEE1588v2.";
        }
       leaf timestamp-nanosec {
       type uint32;
       description
         "Fractional part in nanoseconds as per IEEE1588v2
          or Fractional part in 64-bit NTP timestamp.";
       }
       description
       "Container for 80bit timestamp.";
    }
  container ntp-timestamp-32bit {
  when "derived-from-or-self(../timestamp-type, 'cl-oam:truncated-ntp')"{
   description
    "Only applies when 32 bit NTP Short format Timestamp.";
    }
   if-feature ntp-short-format;
       leaf timestamp-sec {
       type uint16;
       description
         "Timestamp in seconds as per short format NTP.";
        }
       leaf timestamp-nanosec {
       type uint16;
       description
         "Truncated Fractional part in 16-bit NTP timestamp.";
       }
       description
       "Container for 32bit timestamp.";
    }
  container icmp-timestamp-32bit {
  when "derived-from-or-self(../timestamp-type, 'cl-oam:icmp-ntp')"{
   description
   "Only applies when Truncated NTP or 64bit NTP Timestamp.";
   }
```

```
  if-feature icmp-timestamp;
      leaf timestamp-millisec {
      type uint32;

      description
        "timestamp in milliseconds for ICMP timestamp.";
       }
      description
      "Container for 32bit timestamp.";
    }
  }
  grouping path-discovery-data {
    description
      "Path discovery related data output from nodes.";
    container src-test-point {
      description
        "Source test point.";
      uses tp-address-ni;
    }
    container dest-test-point {
      description
        "Destination test point.";
      uses tp-address-ni;
    }
    leaf sequence-number {
      type uint64;
      default "0";
      description
        "Sequence number in data packets.A value of
        zero indicates that no sequence number is sent.";
    }
    leaf hop-cnt {
      type uint8;
      default "0";
      description
        "Hop count.A value of zero indicates
        that no hop count is sent";
    }
    uses session-packet-statistics;
    uses session-error-statistics;
    uses session-delay-statistics;
    uses session-jitter-statistics;
    container path-verification {
      description
        "Optional path verification related information.";
      leaf flow-info {
        type string;
        description
```

```
            "Informations that refers to the flow.";
        }
        uses session-path-verification-statistics;
      }
      container path-trace-info {
        description
          "Optional path trace per-hop test point information.
           The path trace information list has typically a single
           element for per-hop cases like path-discovery RPC operation
           but allows a list of hop related information for other types of
           data retrieval methods.";
        list path-trace-info-list {
          key "index";
          description
            "Path trace information list.";
          leaf index {
            type uint32;
            description
              "Trace information index.";
          }
          uses tp-address-ni;
          uses timestamp;
          leaf ingress-intf-name {
            type if:interface-ref;
            description
              "Ingress interface name";
          }
          leaf egress-intf-name {
            type if:interface-ref;
            description
              "Egress interface name";
          }
          leaf queue-depth {
            type uint32;
            description
              "Length of the queue of the interface from where
               the packet is forwarded out.  The queue depth could
               be the current number of memory buffers used by the
               queue and a packet can consume one or more memory buffers
               thus constituting device-level information.";
          }
          leaf transit-delay {
            type uint32;
            description
              "Time in nano seconds
               packet spent transiting a node.";
          }
          leaf app-meta-data {
```

```
          type uint64;
          description

            "Application specific
             data added by node.";
        }
      }
    }
  }
  grouping continuity-check-data {
    description
      "Continuity check data output from nodes.";
    container src-test-point {
      description
        "Source test point.";
      uses tp-address-ni;
      leaf egress-intf-name {
        type if:interface-ref;
        description
          "Egress interface name.";
      }
    }
    container dest-test-point {
      description
      "Destination test point.";
      uses tp-address-ni;
      leaf ingress-intf-name {
        type if:interface-ref;
        description
          "Ingress interface name.";
      }
    }
    leaf sequence-number {
      type uint64;
      default "0";
      description
       "Sequence number in data packets.A value of
       zero indicates that no sequence number is sent.";
    }
    leaf hop-cnt {
      type uint8;
      default "0";
      description
        "Hop count.A value of zero indicates
         that no hop count is sent";
    }
    uses session-packet-statistics;
    uses session-error-statistics;
```

```
    uses session-delay-statistics;
    uses session-jitter-statistics;
  }
  container cc-session-statistics-data {
    if-feature "continuity-check";
    config false;
    list cc-session-statistics {
       key type;
       leaf type {
        type identityref {
         base traffic-type;
        }
        description
         "Type of traffic.";
       }
        container cc-ipv4-sessions-statistics {
        when "../type = 'ipv4'" {
         description
          "Only applies when traffic type is Ipv4.";
        }
      description
       "CC ipv4 sessions";
      uses cc-session-statistics;
      }
       container cc-ipv6-sessions-statistics {
        when "../type = 'ipv6'" {
         description
          "Only applies when traffic type is Ipv6.";
        }
      description
        "CC ipv6 sessions";
      uses cc-session-statistics;
      }
       description
      "List of CC session statistics data.";
      }
    description
    "CC operational information.";
  }
}

   <CODE ENDS>
```

6. Connectionless model applicability

The "ietf-connectionless-oam" model defined in this document provides a technology-independent abstraction of key OAM constructs for connectionless protocols. This model can be further extended to

include technology specific details, e.g., adding new data nodes with technology specific functions and parameters into proper anchor points of the base model, so as to develop a technology-specific connectionless OAM model.

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

6.1. BFD Extension

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

6.1.1. Augment Method

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

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

6.1.1.1. Technology type extension

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

The snippet below depicts an example of adding the "bfd" type as an augment to the ietf-connectionless-oam" model:

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

6.1.1.2. Test point attributes extension

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

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

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

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

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

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

6.1.1.2.2. Add new location-type cases

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

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

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

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

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

6.1.2. Schema Mount

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

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

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

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

6.1.2.1. BFD Modules be populated in schema-mount

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

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

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

```
<ietf-connectionless-oam
uri="urn:ietf:params:xml:ns:yang:ietf-connectionless-oam">
   ......
 <test-point-locations>
  <ipv4-location>192.0.2.1</ipv4-location>
   ......
  <root>
   <ietf-bfd-ip-sh uri="urn:ietf:params:xml:ns:yang:ietf-bfd-ip-sh">
    <ip-sh>
     foo
     ......
    </ip-sh>
   </ietf-bfd-ip-sh>
   <ietf-bfd-ip-mh uri="urn:ietf:params:xml:ns:yang:ietf-bfd-ip-mh">
    <ip-mh>
     foo
     ......
    </ip-mh>
   </ietf-bfd-ip-mh>
  </root>
 </test-point-locations>
</ietf-connectionless-oam>
```

6.2. LSP ping extension

6.2.1. Augment Method

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

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

6.2.1.1. Technology type extension

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

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

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

6.2.1.2. Test point attributes extension

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

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

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

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

6.2.2. Schema Mount

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

provide a place in the node hierarchy where other OAM YANG data models can be attached, without any special extension in the "ietf-connectionless-oam" YANG data models [I-D.ietf-netmod-schema-mount]. Note that the limitation of the Schema Mount method is it is not allowed to specify certain modules that are required to be mounted under a mount point.

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

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

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

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

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

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

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

```
<ietf-connectionless-oam
uri="urn:ietf:params:xml:ns:yang:ietf-connectionless-oam">
   ......
 <test-point-locations>
  <ipv4-location> 192.0.2.1</ipv4-location>
   ......
  <root>
   <ietf-lspping uri="urn:ietf:params:xml:ns:yang:ietf-lspping">
    <lsp-pings>
     foo
     ......
    </lsp-pings>
   </ietf-lspping>
  </root>
 </test-point-locations>
</ietf-connectionless-oam>
```

7. Security Considerations

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

8. IANA Considerations

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

URI: urn:ietf:params:xml:ns:yang:ietf-lime-time-types
Registrant Contact: The IESG.
XML: N/A; the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-connectionless-oam
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.

This document registers a YANG module in the YANG Module Names registry [RFC7950].

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

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

9. Acknowlegements

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

10. References

10.1. Normative References

[I-D.ietf-i2rs-yang-network-topo]
Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A Data Model for Network Topologies", draft-ietf-i2rs-yang-network-topo-17 (work in progress), October 2017.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

10.2. Informative References

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

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

[I-D.ietf-bfd-yang] Rahman, R., Zheng, L., Jethanandani, M., Networks, J., and G. Mirsky, "YANG Data Model for Bidirectional Forwarding Detection (BFD)", draft-ietf-bfd-yang-06 (work in progress), June 2017.

[I-D.ietf-lime-yang-connection-oriented-oam-model] Kumar, D., Wu, Q., and Z. Wang, "Generic YANG Data Model for Connection Oriented Operations, Administration, and Maintenance(OAM) protocols", draft-ietf-lime-yang-connection-oriented-oam-model-00 (work in progress), June 2017.

[I-D.ietf-lime-yang-connectionless-oam-methods]
Kumar, D., Wang, Z., Wu, Q., Rahman, R., and S. Raghavan, "Retrieval Methods YANG Data Model for Connectionless Operations, Administration, and Maintenance(OAM) protocols", draft-ietf-lime-yang-connectionless-oam-methods-12 (work in progress), October 2017.

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

[I-D.zheng-mpls-lsp-ping-yang-cfg]
Zheng, L., Aldrin, S., Zheng, G., Mirsky, G., and R. Rahman, "YANG Data Model for LSP-Ping", draft-zheng-mpls-lsp-ping-yang-cfg-06 (work in progress), October 2017.

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

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

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

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

Authors' Addresses

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

Email: dekumar@cisco.com

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

Email: wangzitao@huawei.com

Qin Wu
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