

Internet Engineering Task Force (IETF)  
Request for Comments: 8532  
Category: Standards Track  
ISSN: 2070-1721

D. Kumar  
Cisco  
M. Wang  
Q. Wu, Ed.  
Huawei  
R. Rahman  
S. Raghavan  
Cisco  
April 2019

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

Abstract

This document presents a base YANG Data model for the management of Operations, Administration, and Maintenance (OAM) protocols that use connectionless communications. The data model is defined using the YANG data modeling language, as specified in RFC 7950. It provides a technology-independent abstraction of key OAM constructs for OAM protocols that use connectionless communication. The base model presented here can be extended to include technology-specific details.

There are two key benefits of this approach: First, it leads to uniformity between OAM protocols. Second, it supports both nested OAM workflows (i.e., performing OAM functions at the same level or different levels through a unified interface) as well as interactive OAM workflows (i.e., performing OAM functions at the same level through a unified interface).

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <https://www.rfc-editor.org/info/rfc8532>.

## Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Conventions Used in This Document . . . . .	4
2.1. Abbreviations . . . . .	4
2.2. Terminology . . . . .	5
2.3. Tree Diagrams . . . . .	5
3. Overview of the Connectionless OAM Model . . . . .	5
3.1. TP Address . . . . .	6
3.2. Tools . . . . .	7
3.3. OAM Neighboring Test Points . . . . .	7
3.4. Test Point Location Information . . . . .	8
3.5. Test Point Locations . . . . .	8
3.6. Path Discovery Data . . . . .	8
3.7. Continuity Check Data . . . . .	9
3.8. OAM Data Hierarchy . . . . .	9
4. LIME Time Types YANG Module . . . . .	12
5. Connectionless OAM YANG Module . . . . .	15
6. Connectionless Model Applicability . . . . .	44
6.1. BFD Extension . . . . .	45
6.1.1. Augment Method . . . . .	45
6.1.2. Schema Mount . . . . .	47
6.2. LSP Ping Extension . . . . .	49
6.2.1. Augment Method . . . . .	49
6.2.2. Schema Mount . . . . .	50
7. Security Considerations . . . . .	52
8. IANA Considerations . . . . .	54
9. References . . . . .	54
9.1. Normative References . . . . .	54
9.2. Informative References . . . . .	56
Acknowledgments . . . . .	58
Authors' Addresses . . . . .	59

## 1. Introduction

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

1. monitor network communications (i.e., reachability verification and Continuity Check)
2. troubleshoot failures (i.e., fault verification and localization)
3. monitor service-level agreements and performance (i.e., performance management)

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

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

The different sets of OAM tools may support both connection-oriented or connectionless technologies. In connection-oriented technologies, a connection is established prior to the transmission of data. After the connection is established, no additional control information such as signaling or operations and maintenance information is required to transmit the actual user data. In connectionless technologies, data is typically sent between communicating endpoints without prior arrangement, but control information is required to identify the destination (e.g., [G.800] and [RFC7276]). The YANG data model for OAM protocols using connection-oriented communications is specified in [RFC8531].

This document defines a base YANG data model for OAM protocols that use connectionless communications. The data model is defined using the YANG data modeling language [RFC7950]. This generic YANG data model for connectionless OAM includes only configuration and state data. It can be used in conjunction with the data retrieval method model described in [RFC8533], which focuses on the data retrieval procedures such as RPC, or it can be used independently of this data retrieval method model.

## 2. Conventions Used in This Document

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

- o client
- o configuration data
- o server
- o state data

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

- 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 - Remote Procedure Call [RFC1831].

DSCP - Differentiated Services Code Point.

VRF - Virtual Routing and Forwarding [RFC4382].

OWAMP - One-Way Active Measurement Protocol [RFC4656].

TWAMP - Two-Way Active Measurement Protocol [RFC5357].

AS - Autonomous System.

LSP - Label Switched Path.

TE - Traffic Engineering.

MPLS - Multiprotocol Label Switching.

NI - Network Instance.

PTP - Precision Time Protocol [IEEE.1588v2].

NTP - Network Time Protocol [RFC5905].

## 2.2. Terminology

MAC - Media Access Control.

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

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

RPC operation - A specific Remote Procedure Call.

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

## 2.3. Tree Diagrams

Tree diagrams used in this document follow the notation defined in [RFC8340].

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

The "ietf-connectionless-oam" module augments the "/networks/network/node" path defined in the "ietf-network" module [RFC8345] with the 'test-point-locations' grouping defined in Section 3.5. The network nodes in the "/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 the '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, and
- o 'connectionless-oam-tps' grouping.

The groupings of 'tp-address' and 'tp-address-ni' are kept out of the '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'), each container differs 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 indicates 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 describes the OAM tools supported.

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

### 3.1. TP Address

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

- o MAC address [RFC6136] at the data-link layer for TPs
- o IPv4 or IPv6 address at the 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 [RFC8294].

To define a forwarding treatment of a test packet, the 'tp-address' grouping needs to be associated with additional parameters, e.g., DSCP for IP or Traffic Class [RFC5462] for MPLS. In the generic

connectionless OAM YANG data model, these parameters are not explicitly configured. The model user can add corresponding parameters according to their requirements.

### 3.2. Tools

The different OAM tools may be used in one of two basic types of activation: proactive and on-demand. Proactive OAM refers to OAM actions that are carried out continuously to permit proactive reporting of faults. The proactive OAM method requires persistent configuration. On-demand OAM refers to OAM actions that are initiated via manual intervention for a limited time to carry out specific diagnostics. The on-demand OAM method requires only transient configuration (e.g., [RFC7276] and [G.8013]). In connectionless OAM, the '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. In addition, it can serve as a constraint condition when the base model is extended to a specific OAM technology. For example, to fulfill the ICMP PING configuration, the "../coam:continuity-check" leaf should be set to "true", and then the LIME base model should be augmented with details specific to ICMP PING.

### 3.3. OAM Neighboring Test Points

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

Each OAM test point may have an associated list of neighboring test points that are in other layers up and down the protocol stack for the same interface and are therefore related to the current test point. This allows users to easily navigate between related neighboring layers to efficiently troubleshoot a defect. In this model, the 'position' leaf defines the relative position of the neighboring test point corresponding to the current test point, and it 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.

```

+-- oam-neighboring-tps* [index]
  +-- index?                               uint16
  +-- position?                             int8
  +-- (tp-location)?
    +--:(mac-address)
      | +-- mac-address-location?          yang:mac-address
    +--:(ipv4-address)
      | +-- ipv4-address-location?         inet:ipv4-address
    +--:(ipv6-address)
      | +-- ipv6-address-location?         inet:ipv6-address
    +--:(as-number)
      | +-- as-number-location?           inet:as-number
    +--:(router-id)
      | +-- router-id-location?           rt:router-id

```

### 3.4. Test Point Location Information

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

### 3.5. Test Point Locations

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

### 3.6. Path Discovery Data

This is a generic grouping for the path discovery data model that can be retrieved by any data retrieval method, 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, and information related to path verification and path trace. Path discovery includes data to be retrieved on a 'per-hop' basis via a list of 'path-trace-info-list' items which includes information such as 'timestamp' grouping, 'ingress-intf-name', 'egress-intf-name', and 'app-meta-data'. The path discovery data model is made generic enough to allow



different methods of data retrieval. None of the fields are made mandatory for that reason. Note that a set of retrieval methods are defined in [RFC8533].

### 3.7. Continuity Check Data

This is a generic grouping for the Continuity Check data model that can be retrieved by any data retrieval methods including RPC operations. Continuity Check data output from methods, includes 'src-test-point' container, 'dst-test-point' container, 'sequence-number' leaf, 'hop-cnt' leaf, and session statistics of various kinds. The Continuity Check data model is made generic enough to allow different methods of data retrieval. None of the fields are made mandatory for that reason. Noted that a set of retrieval methods are defined in [RFC8533].

### 3.8. OAM Data Hierarchy

The complete data hierarchy related to the OAM YANG data 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-neighbor-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-neighbor-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@2019-04-16.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 LIME Working Group";
  contact
    "WG Web:  <https://datatracker.ietf.org/wg/lime>
     WG List: <mailto:lmmap@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 in the
     Multi-Layer Environment (LIME).  This module defines
     identities but no schema tree elements."

```

Copyright (c) 2019 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC 8532; see the RFC itself for full legal notices.";

```
revision 2019-04-16 {
  description
    "Initial version.";
  reference
    "RFC 8532: Generic YANG Data Model for the Management of
    Operations, Administration, and Maintenance (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 64-bit short-format PTP timestamp.";
}

identity truncated-ntp {
  base timestamp-type;
  description
    "Identity for 32-bit short-format NTP timestamp.";
}

identity ntp64 {
  base timestamp-type;
  description
    "Identity for 64-bit NTP timestamp.";
}

identity icmp {
  base timestamp-type;
  description
    "Identity for 32-bit ICMP timestamp.";
}
```

```
identity ptp80 {
  base timestamp-type;
  description
    "Identity for 80-bit PTP timestamp.";
}
```

<CODE ENDS>

## 5. Connectionless OAM YANG Module

This module imports the Core YANG Derived Types definition ("ietf-yang-types" module) and Internet-Specific Derived Types definitions ("ietf-inet-types" module) from [RFC6991], the "ietf-routing-types" module from [RFC8294], the "ietf-interfaces" module from [RFC8343], the "ietf-network" module from [RFC8345], the "ietf-network-instance" module from [RFC8529], and the "ietf-lime-time-types" module in Section 4. This module references [IEEE.1588v1], [IEEE.1588v2], [RFC8029], and additional RFCs cited elsewhere in this document.

<CODE BEGINS> file "ietf-connectionless-oam@2019-04-16.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
  "WG Web:    <https://datatracker.ietf.org/wg/lime>
  WG List:   <mailto:lmap@ietf.org>

  Deepak Kumar <dekumar@cisco.com>
  Qin Wu <bill.wu@huawei.com>
  Srihari Raghavan <srihari@cisco.com>
  Michael Wang <>wangzitao@huawei.com>
  Reshad 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 may extend the YANG data model defined
  here to include protocol specific extensions.

  Copyright (c) 2019 IETF Trust and the persons identified as
  authors of the code. All rights reserved.

  Redistribution and use in source and binary forms, with or
  without modification, is permitted pursuant to, and subject
  to the license terms contained in, the Simplified BSD License
  set forth in Section 4.c of the IETF Trust's Legal Provisions
  Relating to IETF Documents
  (http://trustee.ietf.org/license-info).
```



```
    description
      "This feature indicates that the OAM solution is connectionless.";
  }

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

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

  feature ptp-long-format {
    description
      "This feature indicates that the timestamp is PTP long format.";
  }

  feature ntp-short-format {
    description
      "This feature indicates that the timestamp is NTP short format.";
  }

  feature icmp-timestamp {
    description
      "This feature indicates that the timestamp is ICMP timestamp.";
  }

  identity traffic-type {
    description
      "This is the base identity of the traffic type,
      which includes IPv4, IPv6, etc.";
  }

  identity ipv4 {
    base traffic-type;
    description
```

```
    "identity for IPv4 traffic type.";
}

identity ipv6 {
  base traffic-type;
  description
    "identity for IPv6 traffic type.";
}

identity address-attribute-types {
  description
    "This is the base identity of the address attribute types, which
    are Generic IPv4/IPv6 Prefix, BGP Labeled IPv4/IPv6 Prefix,
    Tunnel ID, PW ID, VPLS VE ID, etc. (See RFC 8029 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 that 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 that 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 that are admin-down.
     A value of zero indicates that no admin-
     down session count is sent.";
}
}
}

grouping session-packet-statistics {
  description
    "Grouping for statistics per session packet.";
  container session-packet-statistics {
    description
      "Statistics per session packet.";
    leaf rx-packet-count {
      type uint32 {
        range "0..4294967295";
      }
      default "0";
      description
        "Total count of received OAM packets."
    }
  }
}
```

```

    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 count of transmitted OAM packets.
    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 packets.
    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 OAM packets that failed when sent.
    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 of the last time session was down.";
    }
    leaf last-up-time {
      type yang:date-and-time;
      description
        "Time and date of the last time session was up.";
    }
    leaf down-count {
      type uint32 {
        range "0..4294967295";
      }
      default "0";
      description
        "Total count of Continuity Check sessions down.
        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 count of Continuity Check sessions admin down.
        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 count of received packet drops.
        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. Expressed as percentage
        of packets lost with respect to packets sent.";
    }
    leaf packet-reorder-count {
      type uint32 {
        range "0..4294967295";
      }
      default "0";
      description
        "Total count of received packets that were reordered.
        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 count of packets received out of sequence.
        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 count of received packet duplicates.
            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 delay statistics per session.";
    container session-delay-statistics {
        description
            "Session delay summarized information. By default, a
            one-way measurement protocol (e.g., OWAMP) is used
            to measure delay. When a two-way measurement protocol
            (e.g., TWAMP) is used instead, it can be indicated
            using the protocol-id defined in RPC operation of
            retrieval methods for connectionless OAM (RFC 8533),
            i.e., set protocol-id as OWAMP. Note that only one
            measurement protocol for delay is specified for
            interoperability reasons.";
        leaf time-unit-value {
            type identityref {
                base lime:time-unit-type;
            }
            default "lime:milliseconds";
            description
                "Time units, where the options are 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
      "Summarized information about session jitter. By default,
      jitter is measured using IP Packet Delay Variation
      (IPDV) as defined in RFC 3393. When the other measurement
      method is used instead (e.g., Packet Delay Variation used
      in ITU-T Recommendation Y.1540, it can be indicated using
      protocol-id-meta-data defined in RPC operation of
      retrieval methods for connectionless OAM (RFC 8533).
      Note that only one measurement method for jitter is
      specified for interoperability reasons.";
    leaf unit-value {
      type identityref {
        base lime:time-unit-type;
      }
      default "lime:milliseconds";
      description
        "Time units, where the options are 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 path verification statistics per session.";
  container session-path-verification-statistics {
    description
      "OAM path verification statistics per session.";
    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 a proactive session.";
      }
    }
  }
}

```

```
    }
    enum on-demand {
      description
        "The current session is an 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 Sections 3.2.13 and
            3.2.14 of RFC 8029.";
        reference
          "RFC 8029: Detecting Multiprotocol Label
            Switched (MPLS) Data-Plane Failures";
      }
    }
    case bgp {
      leaf bgp {
        type inet:ip-prefix;
        description
          "BGP Labeled IPv4/IPv6 Prefix.  See Sections

```

```
        3.2.11 and 3.2.12 of RFC 8029 for details.";
    reference
        "RFC 8029: Detecting Multiprotocol Label
        Switched (MPLS) Data-Plane Failures";
    }
}
case tunnel {
    leaf tunnel-interface {
        type uint32;
        description
            "Basic IPv4/IPv6 Tunnel ID. See Sections 3.2.3
            and 3.2.4 of RFC 8029 for details.";
        reference
            "RFC 8029: Detecting Multiprotocol 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 RFC 8029 for details.";
        reference
            "RFC 8029: Detecting Multiprotocol Label
            Switched (MPLS) Data-Plane Failures";
    }
    leaf pw-id {
        type uint32;
        description
            "Pseudowire ID is a non-zero 32-bit ID. See Sections
            3.2.8 and 3.2.9 of RFC 8029 for details.";
        reference
            "RFC 8029: Detecting Multiprotocol Label
            Switched (MPLS) Data-Plane Failures";
    }
}
case vpls {
    leaf route-distinguisher {
        type rt:route-distinguisher;
        description
            "Route Distinguisher is an 8-octet identifier
            used to distinguish information about various
            L2VPNs advertised by a node.";
        reference
            "RFC 8029: Detecting Multiprotocol 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 Multiprotocol 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 Multiprotocol 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 that identifies 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 Multiprotocol 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. An example of a
            common industry term for virtual resource partitioning is
            'VRF instance'.";
    }
    uses tp-address;
}

grouping connectionless-oam-tps {
    list oam-neighboring-tps {
        key "index";
        leaf index {

```

```
type uint16 {
  range "0..65535";
}
description
  "Index of a list of neighboring test points
  in layers up and down the stack for
  the same interface that are related to the
  current test point.";
}
leaf position {
  type int8 {
    range "-1..1";
  }
  default "0";
  description
    "The position of the neighboring test point relative to
    the current test point. Level 0 indicates a test point
    corresponding to a specific index in the same layer as
    the current test point. -1 means there is a test point
    corresponding to a specific index in the test point down
    the stack, and +1 means there is a test point corresponding
    to a specific index in the test point up the stack.";
}
choice tp-location {
  case mac-address {
    leaf mac-address-location {
      type yang:mac-address;
      description
        "MAC address.";
    }
    description
      "MAC address based TP addressing.";
  }
  case ipv4-address {
    leaf ipv4-address-location {
      type inet:ipv4-address;
      description
        "IPv4 address.";
    }
    description
      "IP address based TP addressing.";
  }
  case ipv6-address {
    leaf ipv6-address-location {
      type inet:ipv6-address;
      description
        "IPv6 address.";
    }
  }
}
```



```
    description
      "IPv6 address based TP addressing.";
  }
  case as-number {
    leaf as-number-location {
      type inet:as-number;
      description
        "AS number location.";
    }
    description
      "AS number for point-to-multipoint OAM.";
  }
  case router-id {
    leaf router-id-location {
      type rt:router-id;
      description
        "System ID location.";
    }
    description
      "System ID.";
  }
  description
    "TP location.";
}
description
  "List of neighboring test points in the same layer that are
  related to current test point.  If the neighboring test point
  is placed after the current test point, the position is
  specified as +1.  If the neighboring test point is placed
  before the current test point, the position is specified
  as -1; if no neighboring test points are placed before or
  after the current test point in the same layer, the
  position is specified as 0.";
}
description
  "List of neighboring test points related to connectionless OAM.";
}

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 Messages
        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 toolset.";
  }
}

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
  "Container for IPv4 location types.";
}
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
      "Container for MAC address location types.";
  }
  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-multipoint 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

```

```

    "Container for AS number location types.";
}
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
    "Container for system ID location types.";
}
}

augment "/nd:networks/nd:network/nd:node" {
  description
    "Augments the /networks/network/node path defined in the
      ietf-network module (RFC 8345) 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 or NTP.";
}
container timestamp-64bit {
  when "derived-from-or-self(..../timestamp-type, "
    + "'lime:truncated-ntp') "
    + "or derived-from-or-self(..../timestamp-type, "
    + "'lime:ntp64') " {
    description
      "Only applies when PTP truncated or 64-bit NTP timestamp.";
  }
  leaf timestamp-sec {
    type uint32;
    description
      "Absolute timestamp in seconds as per IEEE 1588v2
      or seconds part in 64-bit NTP timestamp.";
  }
  leaf timestamp-nanosec {
    type uint32;
    description
      "Fractional part in nanoseconds as per IEEE 1588v2
      or fractional part in 64-bit NTP timestamp.";
  }
}
description
  "Container for 64-bit timestamp. The Network Time Protocol
  (NTP) 64-bit timestamp format is defined in RFC 5905. The
  PTP truncated timestamp format is defined in IEEE 1588v1.";
reference
  "RFC 5905: Network Time Protocol Version 4: Protocol and
  Algorithms Specification
  IEEE 1588v1: IEEE Standard for a Precision Clock
  Synchronization Protocol for Networked Measurement and
  Control Systems Version 1";
}
container timestamp-80bit {
  when "derived-from-or-self(..../timestamp-type, 'lime:ptp80') " {
    description
      "Only applies when 80-bit PTP timestamp.";
  }
  if-feature "ptp-long-format";
  leaf timestamp-sec {
    type uint64 {
      range "0..281474976710655";
    }
    description
      "48-bit timestamp in seconds as per IEEE 1588v2.";
  }
}

```

```

    }
    leaf timestamp-nanosec {
      type uint32;
      description
        "Fractional part in nanoseconds as per IEEE 1588v2.";
    }
    description
      "Container for 80-bit timestamp.";
  }
  container ntp-timestamp-32bit {
    when "derived-from-or-self(..../timestamp-type, "
      + "'lime: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 32-bit timestamp RFC5905.";
    reference
      "RFC 5905: Network Time Protocol Version 4: Protocol and
        Algorithms Specification.";
  }
  container icmp-timestamp-32bit {
    when "derived-from-or-self(..../timestamp-type, 'lime:icmp')" {
      description
        "Only applies when ICMP timestamp.";
    }
    if-feature "icmp-timestamp";
    leaf timestamp-millisecc {
      type uint32;
      description
        "Timestamp in milliseconds for ICMP timestamp.";
    }
    description
      "Container for 32-bit timestamp. See RFC 792 for ICMP
        timestamp format.";
  }
}

```



```
grouping path-discovery-data {
  description
    "Data output from nodes related to path discovery.";
  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 information related to path verification.";
    leaf flow-info {
      type string;
      description
        "Information that refers to the flow.";
    }
    uses session-path-verification-statistics;
  }
  container path-trace-info {
    description
      "Optional per-hop path trace information about test points.
      The path trace information list typically has a single
      element for per-hop cases such as path-discovery RPC operation
      but allows a list of hop-related information for other types of
      data retrieval methods.";
```

```

list path-trace-info-list {
  key "index";
  description
    "Path trace information list.";
  leaf index {
    type uint32;
    description
      "Trace information index.";
  }
  uses tp-address-ni;
  uses timestamp;
  leaf ingress-intf-name {
    type if:interface-ref;
    description
      "Ingress interface name.";
  }
  leaf egress-intf-name {
    type if:interface-ref;
    description
      "Egress interface name.";
  }
  leaf queue-depth {
    type uint32;
    description
      "Length of the queue of the interface from where
       the packet is forwarded out. The queue depth could
       be the current number of memory buffers used by the
       queue, and a packet can consume one or more memory buffers,
       thus constituting device-level information.";
  }
  leaf transit-delay {
    type uint32;
    description
      "Time in nanoseconds that the 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" module defined in this document provides a technology-independent abstraction of key OAM constructs for OAM protocols that use connectionless communication. This module 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 and 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 the working groups of the respective protocols.

## 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 show how the "ietf-connectionless-oam" module can be extended to cover BFD technology. For this purpose, a set of extensions are introduced such as the technology-type extension and test-point attributes extension.

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

#### 6.1.1.1. Technology-Type Extension

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

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

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

#### 6.1.1.2. Test Point Attributes Extension

To support BFD, the "ietf-connectionless-oam" module 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" module, 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 multihop, etc.), data nodes for BFD-specific details need to be added to the corresponding "test-point-locations" list. In this section, some groupings that are defined in [BFD-YANG] are reused as follows.

The snippet below shows how the "ietf-connectionless-oam" module 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 Multihop, BFD over MPLS, etc.

#### 6.1.1.2.2. Add New location-type Cases

In the "ietf-connectionless-oam" module, 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, there is flexibility -- the module user can add "location-type" to support other types of test point that are not defined in the "ietf-connectionless-oam" module. In this section, a new "location-type" case is added, and some groupings that are defined in [BFD-YANG] are reused as follows.

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

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

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

#### 6.1.2. Schema Mount

An alternative method is using the schema mount mechanism [RFC8528] in the "ietf-connectionless-oam" module. Within the "test-point-locations" list, a "root" attribute is defined to provide a mount point for models that will be added onto per "test-point-locations". Therefore, the "ietf-connectionless-oam" module 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 module [RFC8528]. Note that the limitation of the schema mount method is that it's 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 that are supported per test point";
}
```

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

#### 6.1.2.1. BFD Modules Might Be Populated in schema-mounts

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

Note that an LSP Ping YANG data model is being specified [LSP-PING-YANG]. As with BFD, users can choose to use the "ietf-connectionless-oam" as the basis and augment the "ietf-connectionless-oam" model with details specific to LSP Ping in the model extension to provide a unified view across different technologies. The details that are specific to LSP Ping 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" module. Therefore, a technology-type extension is required in the module extension.

The snippet below depicts an example of augmenting "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" module can be extended and parameters specific to LSP Ping can be defined and put on the "test-point-locations" list.

Users can reuse the attributes or groupings that are defined in [LSP-PING-YANG] as follows:

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

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

#### 6.2.2. Schema Mount

An alternative method is using the schema mount mechanism [RFC8528] in the "ietf-connectionless-oam" module. 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 module [RFC8528]. Note that the limitation of the schema mount method is that it's 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" module can use schema mount to support LSP Ping technology.

#### 6.2.2.1. LSP Ping Modules Might Be Populated in schema-mounts

To support LSP Ping technology, the "ietf-lsp-ping" YANG module [LSP-PING-YANG] 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-lsp-ping </name>
      <revision>2016-03-18</revision>
      <namespace>
        urn:ietf:params:xml:ns:yang: ietf-lsp-ping
      </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-lsp-ping uri="urn:ietf:params:xml:ns:yang:ietf-lsp-ping">
      <lsp-pings>
        foo
        .....
      </lsp-pings>
    </ietf-lsp-ping>
  </root>
</test-point-locations>
</ietf-connectionless-oam>
```

## 7. Security Considerations

The YANG module specified in this document defines a schema for data that 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 [RFC8446].

The NETCONF Configuration Access Control Model (NACM) [RFC8341] 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 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. These are the subtrees and data nodes and their sensitivity/vulnerability:

```
/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 URIs in the "IETF XML Registry" [RFC3688]. Following the format in [RFC3688], the following registrations have been 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 two YANG modules in the "YANG Module Names" registry [RFC6020].

```
Name: ietf-lime-time-types  
Namespace: urn:ietf:params:xml:ns:yang:ietf-lime-time-types  
Prefix: lime  
Reference: RFC 8532
```

```
Name: ietf-connectionless-oam  
Namespace: urn:ietf:params:xml:ns:yang:ietf-connectionless-oam  
Prefix: cl-oam  
Reference: RFC 8532
```

## 9. References

### 9.1. Normative References

- [RFC792] Postel, J., "Internet Control Message Protocol", RFC 792, September 1981.
- [RFC1831] Srinivasan, R., "RPC: Remote Procedure Call Protocol Specification Version 2", RFC 1831, DOI 10.17487/RFC1831, August 1995, <<https://www.rfc-editor.org/info/rfc1831>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.

- [RFC4382] Nadeau, T., Ed. and H. van der Linde, Ed., "MPLS/BGP Layer 3 Virtual Private Network (VPN) Management Information Base", RFC 4382, DOI 10.17487/RFC4382, February 2006, <<https://www.rfc-editor.org/info/rfc4382>>.
- [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>>.
- [RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, DOI 10.17487/RFC4656, September 2006, <<https://www.rfc-editor.org/info/rfc4656>>.
- [RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, DOI 10.17487/RFC5357, October 2008, <<https://www.rfc-editor.org/info/rfc5357>>.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <<https://www.rfc-editor.org/info/rfc5880>>.
- [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>>.
- [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>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [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>>.
- [RFC8294] Liu, X., Qu, Y., Lindem, A., Hopps, C., and L. Berger, "Common YANG Data Types for the Routing Area", RFC 8294, DOI 10.17487/RFC8294, December 2017, <<https://www.rfc-editor.org/info/rfc8294>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 8343, DOI 10.17487/RFC8343, March 2018, <<https://www.rfc-editor.org/info/rfc8343>>.
- [RFC8345] Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A YANG Data Model for Network Topologies", RFC 8345, DOI 10.17487/RFC8345, March 2018, <<https://www.rfc-editor.org/info/rfc8345>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.
- [RFC8529] Berger, L., Hopps, C., Lindem, A., Bogdanovic, D., and X. Liu, "YANG Model for Network Instances", RFC 8529, DOI 10.17487/RFC8529, March 2019, <<https://www.rfc-editor.org/info/rfc8529>>.

## 9.2. Informative References

- [BFD-YANG] Rahman, R., Zheng, L., Jethanandani, M., Networks, J., and G. Mirsky, "YANG Data Model for Bidirectional Forwarding Detection (BFD)", Work in Progress, draft-ietf-bfd-yang-17, August 2018.
- [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.
- [IEEE.1588v1] "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Version 1", IEEE Std 1588, 2002.
- [IEEE.1588v2] "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Version 2", IEEE Std 1588, 2008.
- [LSP-PING-YANG] Zheng, L., Zheng, G., Mirsky, G., Rahman, R., and F. Iqbal, "YANG Data Model for LSP-Ping", Work in Progress, draft-zheng-mpls-lsp-ping-yang-cfg-10, January 2019.
- [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>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [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>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8528] Bjorklund, M. and L. Lhotka, "YANG Schema Mount", RFC 8528, DOI 10.17487/RFC8528, March 2019, <<https://www.rfc-editor.org/info/rfc8528>>.

- [RFC8531] Kumar, D., Wu, Q., and M. Wang, "Generic YANG Data Model for Connection-Oriented Operations, Administration, and Maintenance (OAM) Protocols", RFC 8531, DOI 10.17487/RFC8531, April 2019, <<https://www.rfc-editor.org/info/rfc8531>>.
- [RFC8533] Kumar, D., Wang, M., Wu, Q., Ed., Rahman, R., and S. Raghavan, "A YANG Data Model for Retrieval Methods for the Management of Operations, Administration, and Maintenance (OAM) Protocols That Use Connectionless Communications", RFC 8533, DOI 10.17487/RFC8533, April 2019.

#### Acknowledgments

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

## Authors' Addresses

Deepak Kumar  
CISCO Systems  
510 McCarthy Blvd  
Milpitas, CA 95035  
United States of America

Email: dekumar@cisco.com

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

Email: wangzitao@huawei.com

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

Email: bill.wu@huawei.com

Reshad Rahman  
Cisco Systems  
2000 Innovation Drive  
Kanata, Ontario K2K 3E8  
Canada

Email: rrahman@cisco.com

Srihari Raghavan  
Cisco Systems  
Tril Infopark Sez, Ramanujan IT City  
Neville Block, 2nd floor, Old Mahabalipuram Road  
Chennai, Tamil Nadu 600113  
India

Email: srihari@cisco.com

