Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 18, 2017

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

June 16, 2017

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

Abstract

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

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

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

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

2. Troubleshoot failures (Fault verification and localization).

3. Monitor Performance

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

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

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

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

2. Conventions used in this document

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

o client

o configuration data

o server

o state data

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

o augment

o data model

o data node

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

2.1. Terminology

TP - Test Point

MAC - Media Access Control

BFD - Bidirectional Forwarding Detection

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

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

3. Overview of the Connectionless OAM Model

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

augmented to "/nd:networks/nd:network/nd:node" using 'test-point-locations' defined below.

3.1. TP Address

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

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

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

3.2. Tools

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

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

3.3. OAM-layers

As typical networks have a multi-layer architecture, the set of OAM protocols similarly take a multi-layer structure; each layer may has its own OAM protocol [RFC7276] and is corresponding to specific

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

```
             list oam-layers {
               key "index";
               leaf index {
                  type uint16 {
                     range "0..65535";
                  }
               }
               leaf level {
                   type int32 {
                        range "-1..1";
                   }
                   description
                     "Level";
               }

               description
                  "List of related oam layers.";
             }
```

3.4. Test Point Locations Information

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

3.5. Test Point Locations

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

3.6. Path Discovery Data

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

3.7. Continuity Check Data

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

4. OAM YANG Module

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

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

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

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

    organization
      "IETF LIME Working Group";
    contact
      "Deepak Kumar dekumar@cisco.com
       Qin Wu bill.wu@huawei.com
       S Raghavan srihari@cisco.com
       Zitao Wang wangzitao@huawei.com
       R Rahman rrahman@cisco.com";
    description
      "This YANG module defines the generic configuration,
       data model, statistics for connectionless OAM to be
       used within IETF in a protocol indpendent manner.
       Functional level abstraction is indendent with
       YANG modeling. It is assumed that each protocol maps
       corresponding abstracts to its native format.
       Each protocol may extend the YANG model defined
       here to include protocol specific extensions";

    revision 2017-06-09 {
      description
        " Base model for Connectionless
         Operations, Administration,
         and Maintenance(OAM) ";
      reference
        " RFC XXXX: Connectionless
         Operations, Administration, and
         Maintenance(OAM)YANG Data Model";
    }

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

```
    }

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

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

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

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

    typedef ipv4-multicast-group-address {
      type string {
        pattern "(2((2[4-9])|(3[0-9]))\\.)"+"
  (([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])\\.)"
  +"{2}([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])";
      }
      description
        "The ipv4-multicast-group-address type
         represents an IPv4 multicast address
```

```
         in dotted-quad notation.";
      reference "RFC4607";
    }

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

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

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

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

```
          "Target address attribute type.";
      }

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
            "Average delay value observed.";
        }
      }
    }

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

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

```
        leaf failed-count {
          type uint32;
          description
            "Total number of OAM packets that
             went through an unintended path.";
        }
      }
    }

    grouping session-type {
      description
        "This object indicates the current session
         definition.";
      leaf session-type-enum {
        type enumeration {
          enum "proactive" {
            description
              "The current session is proactive";
          }
          enum "on-demand" {
            description
              "The current session is on-demand.";
          }
        }
        default "on-demand";
        description
          "Session type enum";
      }
    }

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

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

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

    identity ipv6-address-type {
```

```
      base tp-address-technology-type;
      description
        "IPv6 address type";
    }

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

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

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

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

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

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

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

    grouping tp-address {
```

```
      leaf tp-location-type-value {
        type identityref {
          base tp-address-technology-type;
        }

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

```
        case tp-attribute {
          when "'tp-location-type-value' = 'tp-attribute-type'" {
            description
              "Test point attribute type";
          }
          leaf tp-attribute-type {
            type address-attribute-type;
            description
              "Test point type.";
          }
          choice tp-attribute-value {
            description
              "Test point value.";
            case ip-prefix {
              leaf ip-prefix {
                type inet:ip-prefix;
                description
                  "IP prefix.";
              }
            }
            case bgp {
              leaf bgp {
                type inet:ip-prefix;
                description
                  "BGP Labeled Prefix ";
              }
            }
            case tunnel {
              leaf tunnel-interface {
                type uint32;
                description
                  "VPN Prefix ";
              }
            }
            case pw {
              leaf remote-pe-address {
                type inet:ip-address;
                description
                  "Remote pe address.";
              }
              leaf pw-id {
                type uint32;
                description
                  "Pseudowire ID is a non-zero 32-bit ID.";
                reference
                  "RFC 4379 :Detecting Multi-Protocol Label
                  Switched (MPLS) Data Plane Failures";
              }
```

```
            }
            case vpls {
              leaf route-distinguisher {
                type uint32;
                description
                  "Route Distinguisher is an 8 octets identifier
                  used to distinguish information about various
                  L2VPN advertised by a node.";
                reference
                  "RFC 4379 :Detecting Multi-Protocol Label
                  Switched (MPLS) Data Plane Failures";
              }
              leaf sender-ve-id {
                type uint32;
                description
                  "Sender's VE ID. The VE ID (VPLS Edge Identifier)
                   is a 2-octet identifier.";
                reference
                  "RFC 4379 :Detecting Multi-Protocol Label
                  Switched (MPLS) Data Plane Failures";
              }
              leaf receiver-ve-id {
                type uint32;
                description
                  "Receiver's VE ID.The VE ID (VPLS Edge Identifier)
                   is a 2-octet identifier.";
                reference
                  "RFC 4379 :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 ip-multicast-group-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 4379 :Detecting Multi-Protocol Label
                     Switched (MPLS) Data Plane Failures";
                  }
                }
              }
            }
          }
        }
        case system-info {
          when "'tp-location-type-value' = 'system-id-address-type'" {
            description
              "System id address type";
          }
          leaf system-id {
            type router-id;
            description
              "System ID assigned to this node.";
          }
        }
        description
          "TP Addressing.";
      }
      description
        "TP Address";
    }

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

```
      }
      uses tp-address;
    }

    grouping connectionless-oam-layers {
      list oam-layers {
        key "index";
        leaf index {
          type uint16 {
            range "0..65535";
          }
          description
            "Index";
        }
        leaf level {
          type int32 {
            range "-1..1";
          }
          default "0";
          description
            "Level 0 indicates default level,
             -1 means server and +1 means client layer.
             In relationship 0 means same layer.";
        }
        choice tp-location {
          case mac-address {
            leaf mac-address-location {
              type yang:mac-address;
              description
                "MAC Address";
            }
            description
              "MAC Address based MP Addressing.";
          }
          case ipv4-address {
            leaf ipv4-location {
              type inet:ipv4-address;
              description
                "Ipv4 Address";
            }
            description
              "IP Address based MP Addressing.";
          }
          case ipv6-location {
            leaf ipv6-address {
              type inet:ipv6-address;
              description
                "IPv6 Address";
```

```
            }
            description
              "IPv6 Address based MP Addressing.";
          }
          case group-ip-address-location {
            leaf group-ip-address-location {
              type ip-multicast-group-address;
              description
                "Group IP address location";
            }
            description
              "Group IP address";
          }
          case as-number-location {
            leaf as-number-location {
              type inet:as-number;
              description
                "AS number location";
            }
            description
              "AS number for point to multipoint OAM";
          }
          case system-id-location {
            leaf system-id-location {
              type router-id;
              description
                "System id location";
            }
            description
              "System ID";
          }
          description
            "TP location.";
        }

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

```
    }

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

    grouping tp-tools {
      description
        "Test Point OAM Toolset.";
      container tp-tools {
        leaf continuity-check {
          type boolean;
          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.";
        }
        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 4379: LSP-PING.";
        }
        description
          "Container for test point OAM tools set.";
      }
    }

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

    grouping test-point-locations {
      description
        "Group of test point locations.";
          leaf tp-location-type-value {
        type identityref {
          base tp-address-technology-type;
        }
        description
          "Test point location type.";
      }
      choice location-type {
        case ipv4-location-type {
          when "'tp-location-type-value' = 'ipv4-address-type'" {
            description
              "When test point location type is equal to ipv4 address.";
          }
          container test-point-ipv4-location-list {
            list test-point-locations {
              key "ipv4-location";
              leaf ipv4-location {
```

```
                type inet:ipv4-address;
                description
                  "IPv4 Address.";
              }
              leaf vrf {
                type routing-instance-ref;
                description
                  "The vrf is used to describe the
                   corresponding network instance";
              }
              uses test-point-location-info;

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

              description
                "List of test point locations.";
            }
            description
              "Serves as top-level container
               for test point location list.";
          }
```

```
        }
        case mac-location-type {
          when "'tp-location-type-value' = 'mac-address-type'" {
            description
              "when test point location type is equal to mac address.";
          }
          container test-point-mac-address-location-list {
            list test-point-locations {
              key "mac-address-location";
              leaf mac-address-location {
                type yang:mac-address;
                description
                  "MAC Address";
              }
              uses test-point-location-info;

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

              description
                "List of test point locations.";
```

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

              description
                "List of test point locations.";
            }
            description
              "Serves as top-level container
               for test point location list.";
          }
        }
        case group-system-id-location-type {
          when "'tp-location-type-value' = 'system-id-address-type'" {
            description
              "When test point location is equal to
               system info.";
          }
          container test-point-system-info-location-list {
            list test-point-locations {
              key "system-id-location";
              leaf system-id-location {
                type inet:uri;
                description
```

```
                 "System Id.";
             }
             leaf vrf {
               type routing-instance-ref;
               description
                 "The vrf is used to describe the
                  corresponding network instance";
             }
             uses test-point-location-info;

             description
               "List of test point locations.";
           }
           description
             "Serves as top-level container for
              test point location list.";
         }
       }
       description
         "Choice of address types.";
     }
   }

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

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

   grouping timestamp {
     description
```

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

    grouping path-discovery-data {
      description
        "Path discovery related data output from nodes.";
      container src-test-point {
        description
          "Source test point.";
        uses tp-address-vrf;
      }
      container dest-test-point {
        description
          "Destination test point.";
        uses tp-address-vrf;
      }
      leaf sequence-number {
        type uint64;
        description
          "Sequence number in data packets.";
      }
      leaf hop-cnt {
        type uint8;
        description
          "Hop count.";
      }
      uses session-packet-statistics;
      uses session-error-statistics;
      uses session-delay-statistics;
      uses session-jitter-statistics;
      container path-verification {
        description
          "Optional path verification related information.";
        leaf flow-info {
          type string;
          description
            "Informations that refers to the flow.";
        }
        uses session-path-verification-statistics;
      }
      container path-trace-info {
        description
```

```
          "Optional path trace per-hop test point information.
           The list has typically a single element for per-hop
           cases like path-discovery RPC but allows a list of
           hop related information for other types of
           data retrieval methods.";
        list path-trace-info-list {
          key "index";
          description
            "Path trace information list.";
          leaf index {
            type uint32;
            description
              "Trace information index.";
          }
          uses tp-address-vrf;
          uses timestamp;
          leaf ingress-intf-name {
            type if:interface-ref;
            description
              "Ingress interface name";
          }
          leaf egress-intf-name {
            type if:interface-ref;
            description
              "Egress interface name";
          }
          leaf queue-depth {
            type uint32;
            description
              "Length of the egress interface
               queue of the interface.";
          }
          leaf transit-delay {
            type uint32;
            description
              "Time in nano seconds
               packet spent transiting a node.";
          }
          leaf app-meta-data {
            type uint64;
            description
              "Application specific
               data added by node.";
          }
        }
      }
    }
```

```
    grouping continuity-check-data {
      description
        "Continuity check data output from nodes.";
      container src-test-point {
        description
          "Source test point.";
        uses tp-address-vrf;
        leaf egress-intf-name {
          type if:interface-ref;
          description
            "Egress interface name";
        }
      }
      container dest-test-point {
        description
          "Destination test point.";
        uses tp-address-vrf;
        leaf ingress-intf-name {
          type if:interface-ref;
          description
            "Ingress interface name";
        }
      }
      leaf sequence-number {
        type uint64;
        description
          "Sequence number.";
      }
      leaf hop-cnt {
        type uint8;
        description
          "Hop count.";
      }
      uses session-packet-statistics;
      uses session-error-statistics;
      uses session-delay-statistics;
      uses session-jitter-statistics;
    }

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

```
      }
      container cc-ipv6-sessions-statistics {
        description
          "CC ipv6 sessions";
        uses cc-session-statistics;
      }
    }
  }

   <CODE ENDS>
```

5. Connectionless model applicability

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

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

5.1. BFD Extension

5.1.1. Augment Method

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

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

5.1.1.1. Technology type extension

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

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

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

5.1.1.2. Test point attributes extension

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

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

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

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

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

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

5.1.1.2.2. Add new location-type cases

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

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

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

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

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

5.1.2. Schema Mount

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

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

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

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

5.1.2.1. BFD Modules be populated in schema-mount

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

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

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

```
   <ietf-connectionless-oam
   uri="urn:ietf:params:xml:ns:yang:ietf-connectionless-oam">
      ......
    <test-point-locations>
     <ipv4-location>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>
```

5.2. LSP ping extension

5.2.1. Augment Method

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

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

5.2.1.1. Technology type extension

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

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

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

5.2.1.2. Test point attributes extension

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

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

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

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

5.2.2. Schema Mount

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

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

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

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

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

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

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

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

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

```
<ietf-connectionless-oam
uri="urn:ietf:params:xml:ns:yang:ietf-connectionless-oam">
   ......
 <test-point-locations>
  <ipv4-location> 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>
```

6. Security Considerations

The YANG module defined in this memo is designed to be accessed via the NETCONF protocol [RFC6241]. The lowest NETCONF layer is the secure transport layer and the mandatory-to-implement secure transport is SSH [RFC6242]. The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF users to a pre-configured subset of all available NETCONF protocol operations and content.

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

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

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

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

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

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

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

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

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

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

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

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

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

7. IANA Considerations

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

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

Registrant Contact: The IESG.

XML: N/A, the requested URI is an XML namespace.

This document registers a YANG module in the YANG Module Names registry [RFC6020].

name: ietf-connectionless-oam

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

prefix: coam

reference: RFC XXXX

8. Acknowlegements

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

9. References

9.1. Normative References

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

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

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <http://www.rfc-editor.org/info/rfc6020>.

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

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

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

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

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

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

9.2. Informative References

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

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

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

[I-D.ietf-lime-yang-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-04 (work in progress), June 2017.

[I-D.ietf-netmod-schema-mount]
Bjorklund, M. and L. Lhotka, "YANG Schema Mount", draft-ietf-netmod-schema-mount-05 (work in progress), May 2017.

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

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

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

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

Authors' Addresses

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

Email: dekumar@cisco.com

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

Email: wangzitao@huawei.com

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

Email: bill.wu@huawei.com

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

Email: rrahman@cisco.com

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

Email: srihari@cisco.com