Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 10, 2017

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

June 8, 2017

Retrieval Methods YANG Data Model for Connectionless Operations, Administration, and Maintenance(OAM) protocols draft-ietf-lime-yang-connectionless-oam-methods-04

Abstract

This document presents a retrieval method YANG Data model for connectionless OAM protocols. It provides a technology-independent RPC commands for connectionless OAM protocols. The retrieval methods model presented here can be extended to include technology specific details. This is leading to uniformity between OAM protocols and support both nested OAM workflows (i.e., performing OAM functions at different levels through a unified interface) and interactive OAM workflows (i.e., performing OAM functions at same levels through a unified interface).

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 10, 2017.

Copyright Notice

Copyright (c) 2017 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 (http://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 2
2. Conventions used in this document 3
2.1. Terminology 3
2.2. Tree Diagrams 4
3. Overview of the Connectionless OAM retrieval methods Model . 4
3.1. RPC definitions 5
3.2. OAM Retrieval Methods Hierarchy 8
4. OAM Retrieval Methods YANG Module 17
5. Security Considerations 20
6. IANA Considerations 21
7. References 21
7.1. Normative References 21
7.2. Informative References 22
Appendix A. Appendix 23
A.1. OAM Retrieval Persistent Methods YANG Module 23
Authors' Addresses 33

1. Introduction

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

1. Monitor reachability of destinations (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] and BFD [RFC5880] are well-known fault verification and isolation tools, respectively, for IP networks. Over the years, different technologies have developed similar tools for similar purposes.

In this document, we present a retrieval method YANG Data model for connectionless OAM protocols. This module provides technology-independent RPC commands for connectionless OAM protocols. It is separated from the generic YANG model for connectionless OAM [I-D.ietf-lime-yang-connectionless-oam] and can avoid mixing the models for the retrieved-data from the retrieval procedures. It is expected that retrieval procedures would evolve faster than the data model [I-D.ietf-lime-yang-connectionless-oam] and will allow new procedures to be defined for retrieval of the same data defined by the base data model.

2. Conventions used in this document

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

- client
- configuration data
- server
- state data

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

- augment
- data model
- data node

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

2.1. Terminology

TP - Test Point

MAC - Media Access Control

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

2.2. Tree Diagrams

A simplified graphical representation of the data model is used in this document. The meaning of the symbols in these diagrams is as follows:

Each node is printed as:

<status> <flags> <name> <opts> <type>

<status> is one of:

```
     +  for current
```

<flags> is one of:

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

<name> is the name of the node

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

<opts> is one of:

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

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

3. Overview of the Connectionless OAM retrieval methods Model

In this document, we present a retrieval method YANG Data model for connectionless OAM protocols. This module provides technology-independent retrieval procedures (RPC commands) for connectionless OAM protocols. It provides a flexible way to retrieve the retrieved-data which defined by the "ietf-connectionless-oam.yang" [I-D.ietf-lime-yang-connectionless-oam].

3.1. RPC definitions

The RPC model facilitates issuing commands to a NETCONF server (in this case to the device that need to execute the OAM command) and obtaining a response.

Under 'connectionless-oam-methods' module, we summarize the common OAM functions and define the generic RPC commands: 'continuity-check' and 'path-discovery'. In practice, these commands are supported by corresponding technology-specific OAM tools [RFC7276]. For example, for the IP OAM model, the continuity-check RPC corresponds to the IP Ping [RFC792] [RFC4443], while the path-discovery RPC command corresponds to IP Traceroute [RFC792] [RFC4443].

Note that the RPC command presented in this document is the base building block, which is used to derive a model for a technology-specific OAM (i.e., ICMP ping [RFC792] [RFC4443], LSP ping [RFC8029]), the base building block should be extended with corresponding technology specific parameters. To facilitate this and for future enhancements to data retrieval methods, the RPCs are captured under a separate module.

The generic 'path-discovery-data' and 'continuity-check-data' are used as data outputs from the different RPCs described in the document. Similar methods including other RPCs can retrieve the data using the same data model.

```
    rpc continuity-check {
      if-feature coam:continuity-check;
      description
        "Generates continuity-check as per RFC7276.";
      input {
        container destination-tp {
          uses coam:tp-address;
                  description
                  "destination test point.";
        }
        uses coam:session-type;
        leaf source-interface {
          type if:interface-ref;
                  description
                  "source interface.";
        }
        leaf outbound-interface {
          type if:interface-ref;

                  description
                  "outbound interface.";
```

```
      }
      leaf count {
        type uint32;
        default "5";
        description
          "Specifies the number of packets that will be sent.";
      }
      leaf vrf {
        type coam:routing-instance-ref;
        description
          "vrf instance.";
      }
      leaf ttl {
        type uint8;
        default "255";
        description
          "Time to live (TTL).";
      }
      leaf packet-size {
        type uint32 {
          range "64..10000";
        }
        default "64";
        description
          "Size of ping echo request packets, in octets";
      }
    }
    output {
      list error-code-list {
        key "response-index";
        leaf response-index {
          type uint32;
                  description
                  "response index.";
        }
        leaf status-code {
          type int32;
          description
            "error code is ";
        }
        leaf status-sub-code {
          type uint8;
                  description
                  "sub code.";
        }
                description
                "error code list.";
      }
```

```
          uses coam:continuity-check-data;
        }
      }

      rpc path-discovery {
        description
          "Generates path discovery as per RFC7276.";
        input {
          container destination-tp {
            uses coam:tp-address;
                    description
                    "destination test point.";
          }
          uses coam:session-type;
          leaf source-interface {
            type if:interface-ref;
            description
              "source interface.";
          }
          leaf outbound-interface {
            type if:interface-ref;
            description
              "outbound interface.";
          }
          leaf vrf {
            type coam:routing-instance-ref;
            description
              "vrf";
          }
          leaf max-ttl {
            type uint8;
            default "255";
            description
              "max ttl.";
          }
        }
        output {
          list response-list {
            key "response-index";
            description
              "path discovery response list.";
            leaf response-index {
              type uint32;
              description
                "response index.";
            }
            leaf status-code {
              type int32;
```

```
              description
                "error code is ";
            }
            leaf status-sub-code {
              type uint8;

                      description
                      "sub code is ";
            }
          }

          uses coam:path-discovery-data;
        }
      }
```

Snippet of data hierarchy related to RPC calls

3.2. OAM Retrieval Methods Hierarchy

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

```
module: ietf-connectionless-oam-methods

  rpcs:
    +---x continuity-check {coam:continuity-check}?
    |  +---w input
    |  |  +---w destination-tp
    |  |  |  +---w tp-address-type-value?   identityref
    |  |  |  +---w (tp-address)?
    |  |  |     +--:(mac-address)
    |  |  |     |  +---w mac-address?            yang:mac-address
    |  |  |     +--:(ipv4-address)
    |  |  |     |  +---w ipv4-address?           inet:ipv4-address
    |  |  |     +--:(ipv6-address)
    |  |  |     |  +---w ipv6-address?           inet:ipv6-address
    |  |  |     +--:(src-dst-address)
    |  |  |     |  +---w src-ip-address?         inet:ip-address
    |  |  |     |  +---w dst-ip-address?         inet:ip-address
    |  |  |     |  +---w Interface?              if:interface-ref
    |  |  |     +--:(fec)
    |  |  |     |  +---w fec-type?               fec-type
    |  |  |     |  +---w (fec-value)?
    |  |  |     |     +--:(ip-prefix)
    |  |  |     |     |  +---w ip-prefix?          inet:ip-prefix
    |  |  |     |     +--:(bgp)
    |  |  |     |     |  +---w bgp?                inet:ip-prefix
    |  |  |     |     +--:(tunnel)
```

```
      |  |  |     |     |  +---w tunnel-interface?        uint32
      |  |  |     |     +--:(pw)
      |  |  |     |     |  +---w remote-pe-address?   inet:ip-address
      |  |  |     |     |  +---w pw-id?                   uint32
      |  |  |     |     +--:(vpls)
      |  |  |     |     |  +---w route-distinguisher?     uint32
      |  |  |     |     |  +---w sender-ve-id?            uint32
      |  |  |     |     |  +---w receiver-ve-id?          uint32
      |  |  |     |     +--:(mpls-mldp)
      |  |  |     |        +---w (root-address)?
      |  |  |     |           +--:(ip-address)
      |  |  |     |           |  +---w source-address? inet:ip-address
      |  |  |     |           |  +---w group-ip-address? IP-Multicast-
          Group-Address
      |  |  |     |           +--:(vpn)
      |  |  |     |           |  +---w as-number?        inet:as-number
      |  |  |     |           +--:(global-id)
      |  |  |     |              +---w lsp-id?           string
      |  |  |     +--:(tlv-address)
      |  |  |     |  +---w tlv-type?               int16
      |  |  |     |  +---w tlv-len?                int16
      |  |  |     |  +---w tlv-value?              binary
      |  |  |     +--:(system-info)
      |  |  |        +---w system-id?              inet:uri
      |  |  +---w session-type-enum?    enumeration
      |  |  +---w source-interface?     if:interface-ref
      |  |  +---w outbound-interface?   if:interface-ref
      |  |  +---w count?                uint32
      |  |  +---w vrf?                  coam:routing-instance-ref
      |  |  +---w ttl?                  uint8
      |  |  +---w packet-size?          uint32
      |  +--ro output
      |     +--ro error-code-list* [response-index]
      |     |  +--ro response-index     uint32
      |     |  +--ro status-code?       int32
      |     |  +--ro status-sub-code?   uint8
      |     +--ro src-test-point
      |     |  +--ro vrf?                    routing-instance-ref
      |     |  +--ro tp-address-type-value?  identityref
      |     |  +--ro (tp-address)?
      |     |  |  +--:(mac-address)
      |     |  |  |  +--ro mac-address?            yang:mac-address
      |     |  |  +--:(ipv4-address)
      |     |  |  |  +--ro ipv4-address?          inet:ipv4-address
      |     |  |  +--:(ipv6-address)
      |     |  |  |  +--ro ipv6-address?          inet:ipv6-address
      |     |  |  +--:(src-dst-address)
      |     |  |  |  +--ro src-ip-address?        inet:ip-address
```

```
      |     |  |  |  +--ro dst-ip-address?         inet:ip-address
      |     |  |  |  +--ro Interface?              if:interface-ref
      |     |  |  +--:(fec)
      |     |  |  |  +--ro fec-type?               fec-type
      |     |  |  |  +--ro (fec-value)?
      |     |  |  |     +--:(ip-prefix)
      |     |  |  |     |  +--ro ip-prefix?           inet:ip-prefix
      |     |  |  |     +--:(bgp)
      |     |  |  |     |  +--ro bgp?                 inet:ip-prefix
      |     |  |  |     +--:(tunnel)
      |     |  |  |     |  +--ro tunnel-interface?       uint32
      |     |  |  |     +--:(pw)
      |     |  |  |     |  +--ro remote-pe-address?    inet:ip-address
      |     |  |  |     |  +--ro pw-id?                  uint32
      |     |  |  |     +--:(vpls)
      |     |  |  |     |  +--ro route-distinguisher?    uint32
      |     |  |  |     |  +--ro sender-ve-id?           uint32
      |     |  |  |     |  +--ro receiver-ve-id?         uint32
      |     |  |  |     +--:(mpls-mldp)
      |     |  |  |        +--ro (root-address)?
      |     |  |  |           +--:(ip-address)
      |     |  |  |           |  +--ro source-address? inet:ip-address
      |     |  |  |           |  +--ro group-ip-address? IP-Multicast-
          Group-Address
      |     |  |  |           +--:(vpn)
      |     |  |  |           |  +--ro as-number?       inet:as-number
      |     |  |  |           +--:(global-id)
      |     |  |  |              +--ro lsp-id?          string
      |     |  |  +--:(tlv-address)
      |     |  |  |  +--ro tlv-type?               int16
      |     |  |  |  +--ro tlv-len?                int16
      |     |  |  |  +--ro tlv-value?              binary
      |     |  |  +--:(system-info)
      |     |  |     +--ro system-id?              inet:uri
      |     |  +--ro egress-intf-name?        if:interface-ref
      |     +--ro dest-test-point
      |     |  +--ro vrf?                     routing-instance-ref
      |     |  +--ro tp-address-type-value?   identityref
      |     |  +--ro (tp-address)?
      |     |  |  +--:(mac-address)
      |     |  |  |  +--ro mac-address?            yang:mac-address
      |     |  |  +--:(ipv4-address)
      |     |  |  |  +--ro ipv4-address?           inet:ipv4-address
      |     |  |  +--:(ipv6-address)
      |     |  |  |  +--ro ipv6-address?           inet:ipv6-address
      |     |  |  +--:(src-dst-address)
      |     |  |  |  +--ro src-ip-address?         inet:ip-address
      |     |  |  |  +--ro dst-ip-address?         inet:ip-address
```

```
|     |  |  |  +--ro Interface?              if:interface-ref
|     |  |  +--:(fec)
|     |  |  |  +--ro fec-type?              fec-type
|     |  |  |  +--ro (fec-value)?
|     |  |  |     +--:(ip-prefix)
|     |  |  |     |  +--ro ip-prefix?         inet:ip-prefix
|     |  |  |     +--:(bgp)
|     |  |  |     |  +--ro bgp?               inet:ip-prefix
|     |  |  |     +--:(tunnel)
|     |  |  |     |  +--ro tunnel-interface?       uint32
|     |  |  |     +--:(pw)
|     |  |  |     |  +--ro remote-pe-address?  inet:ip-address
|     |  |  |     |  +--ro pw-id?                  uint32
|     |  |  |     +--:(vpls)
|     |  |  |     |  +--ro route-distinguisher?    uint32
|     |  |  |     |  +--ro sender-ve-id?           uint32
|     |  |  |     |  +--ro receiver-ve-id?         uint32
|     |  |  |     +--:(mpls-mldp)
|     |  |  |        +--ro (root-address)?
|     |  |  |           +--:(ip-address)
|     |  |  |           |  +--ro source-address? inet:ip-address
|     |  |  |           |  +--ro group-ip-address? IP-Multicast-
    Group-Address
|     |  |  |           +--:(vpn)
|     |  |  |           |  +--ro as-number?       inet:as-number
|     |  |  |           +--:(global-id)
|     |  |  |              +--ro lsp-id?          string
|     |  |  +--:(tlv-address)
|     |  |  |  +--ro tlv-type?              int16
|     |  |  |  +--ro tlv-len?               int16
|     |  |  |  +--ro tlv-value?             binary
|     |  |  +--:(system-info)
|     |  |     +--ro system-id?             inet:uri
|     |  +--ro ingress-intf-name?       if:interface-ref
|     +--ro sequence-number?             uint64
|     +--ro hop-cnt?                     uint8
|     +--ro session-packet-statistics
|     |  +--ro rx-packet-count?    uint32
|     |  +--ro tx-packet-count?    uint32
|     |  +--ro rx-bad-packet?      uint32
|     |  +--ro tx-packet-failed?   uint32
|     +--ro session-error-statistics
|     |  +--ro packet-drops-count?         uint32
|     |  +--ro packet-reorder-count?       uint32
|     |  +--ro packets-out-of-seq-count?   uint32
|     |  +--ro packets-dup-count?          uint32
|     +--ro session-delay-statistics
|     |  +--ro time-resolution-value?   identityref
```

```
|     |  +--ro min-delay-value?         uint32
|     |  +--ro max-delay-value?         uint32
|     |  +--ro average-delay-value?     uint32
|     +--ro session-jitter-statistics
|        +--ro time-resolution-value?   identityref
|        +--ro min-jitter-value?        uint32
|        +--ro max-jitter-value?        uint32
|        +--ro average-jitter-value?    uint32
+---x path-discovery {coam:path-discovery}?
   +---w input
   |  +---w destination-tp
   |  |  +---w tp-address-type-value?   identityref
   |  |  +---w (tp-address)?
   |  |     +--:(mac-address)
   |  |     |  +---w mac-address?         yang:mac-address
   |  |     +--:(ipv4-address)
   |  |     |  +---w ipv4-address?        inet:ipv4-address
   |  |     +--:(ipv6-address)
   |  |     |  +---w ipv6-address?        inet:ipv6-address
   |  |     +--:(src-dst-address)
   |  |     |  +---w src-ip-address?      inet:ip-address
   |  |     |  +---w dst-ip-address?      inet:ip-address
   |  |     |  +---w Interface?           if:interface-ref
   |  |     +--:(fec)
   |  |     |  +---w fec-type?                fec-type
   |  |     |  +---w (fec-value)?
   |  |     |     +--:(ip-prefix)
   |  |     |     |  +---w ip-prefix?           inet:ip-prefix
   |  |     |     +--:(bgp)
   |  |     |     |  +---w bgp?                 inet:ip-prefix
   |  |     |     +--:(tunnel)
   |  |     |     |  +---w tunnel-interface?        uint32
   |  |     |     +--:(pw)
   |  |     |     |  +---w remote-pe-address?   inet:ip-address
   |  |     |     |  +---w pw-id?                   uint32
   |  |     |     +--:(vpls)
   |  |     |     |  +---w route-distinguisher?     uint32
   |  |     |     |  +---w sender-ve-id?            uint32
   |  |     |     |  +---w receiver-ve-id?          uint32
   |  |     |     +--:(mpls-mldp)
   |  |     |        +---w (root-address)?
   |  |     |           +--:(ip-address)
   |  |     |           |  +---w source-address? inet:ip-address
   |  |     |           |  +---w group-ip-address? IP-Multicast-
       Group-Address
   |  |     |           +--:(vpn)
   |  |     |           |  +---w as-number?   inet:as-number
   |  |     |           +--:(global-id)
```

```
|  |     |              +---w lsp-id?      string
|  |     +--:(tlv-address)
|  |     |  +---w tlv-type?               int16
|  |     |  +---w tlv-len?                int16
|  |     |  +---w tlv-value?              binary
|  |     +--:(system-info)
|  |        +---w system-id?              inet:uri
|  +---w session-type-enum?    enumeration
|  +---w source-interface?     if:interface-ref
|  +---w outbound-interface?   if:interface-ref
|  +---w vrf?                  coam:routing-instance-ref
|  +---w max-ttl?              uint8
+--ro output
   +--ro response-list* [response-index]
   |  +--ro response-index     uint32
   |  +--ro status-code?       int32
   |  +--ro status-sub-code?   uint8
   +--ro src-test-point
   |  +--ro vrf?                     routing-instance-ref
   |  +--ro tp-address-type-value?   identityref
   |  +--ro (tp-address)?
   |     +--:(mac-address)
   |     |  +--ro mac-address?            yang:mac-address
   |     +--:(ipv4-address)
   |     |  +--ro ipv4-address?           inet:ipv4-address
   |     +--:(ipv6-address)
   |     |  +--ro ipv6-address?           inet:ipv6-address
   |     +--:(src-dst-address)
   |     |  +--ro src-ip-address?         inet:ip-address
   |     |  +--ro dst-ip-address?         inet:ip-address
   |     |  +--ro Interface?              if:interface-ref
   |     +--:(fec)
   |     |  +--ro fec-type?                fec-type
   |     |  +--ro (fec-value)?
   |     |     +--:(ip-prefix)
   |     |     |  +--ro ip-prefix?           inet:ip-prefix
   |     |     +--:(bgp)
   |     |     |  +--ro bgp?                 inet:ip-prefix
   |     |     +--:(tunnel)
   |     |     |  +--ro tunnel-interface?       uint32
   |     |     +--:(pw)
   |     |     |  +--ro remote-pe-address?    inet:ip-address
   |     |     |  +--ro pw-id?                  uint32
   |     |     +--:(vpls)
   |     |     |  +--ro route-distinguisher?    uint32
   |     |     |  +--ro sender-ve-id?           uint32
   |     |     |  +--ro receiver-ve-id?         uint32
   |     |     +--:(mpls-mldp)
```

```
          |     |       +--ro (root-address)?
          |     |          +--:(ip-address)
          |     |          |  +--ro source-address? inet:ip-address
          |     |          |  +--ro group-ip-address?  IP-Multicast
            -Group-Address
          |     |          +--:(vpn)
          |     |          |  +--ro as-number?        inet:as-number
          |     |          +--:(global-id)
          |     |             +--ro lsp-id            string
          |     +--:(tlv-address)
          |     |  +--ro tlv-type?               int16
          |     |  +--ro tlv-len?                int16
          |     |  +--ro tlv-value?              binary
          |     +--:(system-info)
          |        +--ro system-id?              inet:uri
          +--ro dest-test-point
          |  +--ro vrf?                   routing-instance-ref
          |  +--ro tp-address-type-value? identityref
          |  +--ro (tp-address)?
          |     +--:(mac-address)
          |     |  +--ro mac-address?           yang:mac-address
          |     +--:(ipv4-address)
          |     |  +--ro ipv4-address?          inet:ipv4-address
          |     +--:(ipv6-address)
          |     |  +--ro ipv6-address?          inet:ipv6-address
          |     +--:(src-dst-address)
          |     |  +--ro src-ip-address?        inet:ip-address
          |     |  +--ro dst-ip-address?        inet:ip-address
          |     |  +--ro Interface?             if:interface-ref
          |     +--:(fec)
          |     |  +--ro fec-type?                   fec-type
          |     |  +--ro (fec-value)?
          |     |     +--:(ip-prefix)
          |     |     |  +--ro ip-prefix?            inet:ip-prefix
          |     |     +--:(bgp)
          |     |     |  +--ro bgp?                  inet:ip-prefix
          |     |     +--:(tunnel)
          |     |     |  +--ro tunnel-interface?     uint32
          |     |     +--:(pw)
          |     |     |  +--ro remote-pe-address?    inet:ip-address
          |     |     |  +--ro pw-id?                   uint32
          |     |     +--:(vpls)
          |     |     |  +--ro route-distinguisher?     uint32
          |     |     |  +--ro sender-ve-id?            uint32
          |     |     |  +--ro receiver-ve-id?          uint32
          |     |     +--:(mpls-mldp)
          |     |        +--ro (root-address)?
          |     |           +--:(ip-address)
```

```
|     |           |  +--ro source-address? inet:ip-address
|     |           |  +--ro group-ip-address? IP-Multicast-
        Group-Address
|     |           +--:(vpn)
|     |           |  +--ro as-number?        inet:as-number
|     |           +--:(global-id)
|     |              +--ro lsp-id?           string
|     +--:(tlv-address)
|     |  +--ro tlv-type?               int16
|     |  +--ro tlv-len?                int16
|     |  +--ro tlv-value?              binary
|     +--:(system-info)
|        +--ro system-id?              inet:uri
+--ro sequence-number?             uint64
+--ro hop-cnt?                     uint8
+--ro session-packet-statistics
|  +--ro rx-packet-count?    uint32
|  +--ro tx-packet-count?    uint32
|  +--ro rx-bad-packet?      uint32
|  +--ro tx-packet-failed?   uint32
+--ro session-error-statistics
|  +--ro packet-drops-count?         uint32
|  +--ro packet-reorder-count?       uint32
|  +--ro packets-out-of-seq-count?   uint32
|  +--ro packets-dup-count?          uint32
+--ro session-delay-statistics
|  +--ro time-resolution-value?   identityref
|  +--ro min-delay-value?         uint32
|  +--ro max-delay-value?         uint32
|  +--ro average-delay-value?     uint32
+--ro session-jitter-statistics
|  +--ro time-resolution-value?   identityref
|  +--ro min-jitter-value?        uint32
|  +--ro max-jitter-value?        uint32
|  +--ro average-jitter-value?    uint32
+--ro path-verification
|  +--ro flow-info?               string
|  +--ro session-path-verification-statistics
|     +--ro verified-count?   uint32
|     +--ro failed-count?     uint32
+--ro path-trace-info
   +--ro path-trace-info-list* [index]
      +--ro index                  uint32
      +--ro vrf?                   routing-instance-ref
      +--ro tp-address-type-value? identityref
      +--ro (tp-address)?
      |  +--:(mac-address)
      |  |  +--ro mac-address?           yang:mac-address
```

```
|  +--:(ipv4-address)
|  |  +--ro ipv4-address?          inet:ipv4-address
|  +--:(ipv6-address)
|  |  +--ro ipv6-address?          inet:ipv6-address
|  +--:(src-dst-address)
|  |  +--ro src-ip-address?        inet:ip-address
|  |  +--ro dst-ip-address?        inet:ip-address
|  |  +--ro Interface?             if:interface-ref
|  +--:(fec)
|  |  +--ro fec-type?              fec-type
|  |  +--ro (fec-value)?
|  |     +--:(ip-prefix)
|  |     |  +--ro ip-prefix?       inet:ip-prefix
|  |     +--:(bgp)
|  |     |  +--ro bgp?             inet:ip-prefix
|  |     +--:(tunnel)
|  |     |  +--ro tunnel-interface?        uint32
|  |     +--:(pw)
|  |     |  +--ro remote-pe-address?  inet:ip-address
|  |     |  +--ro pw-id?                   uint32
|  |     +--:(vpls)
|  |     |  +--ro route-distinguisher?     uint32
|  |     |  +--ro sender-ve-id?            uint32
|  |     |  +--ro receiver-ve-id?          uint32
|  |     +--:(mpls-mldp)
|  |        +--ro (root-address)?
|  |           +--:(ip-address)
|  |           |  +--ro source-address inet:ip-address
|  |           |  +--ro group-ip-address? IP-Multicast-
|  |            Group-Address
|  |           +--:(vpn)
|  |           |  +--ro as-number?       inet:as-number
|  |           +--:(global-id)
|  |              +--ro lsp-id?          string
|  +--:(tlv-address)
|  |  +--ro tlv-type?               int16
|  |  +--ro tlv-len?                int16
|  |  +--ro tlv-value?              binary
|  +--:(system-info)
|     +--ro system-id?              inet:uri
+--ro timestamp-val?          yang:date-and-time
+--ro ingress-intf-name?      if:interface-ref
+--ro egress-intf-name?       if:interface-ref
+--ro app-meta-data?          uint32
```

data hierarchy of OAM Retrieval Methods

4. OAM Retrieval Methods YANG Module

<CODE BEGINS> file "ietf-connectionless-oam-methods@2017-05-18.yang"

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

 import ietf-interfaces {
  prefix if;
 }
 import ietf-connectionless-oam {
  prefix coam;
 }


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

  revision 2017-05-18 {
    description
      "02 version";
    reference "draft-ietf-lime-yang-connectionless-oam-methods";
  }

  rpc continuity-check {
    if-feature "coam:continuity-check";
    description
      "Generates continuity-check as per RFC7276.";
    input {
      container destination-tp {
        uses coam:tp-address;
        description
```

```
          "Destination test point.";
      }
      uses coam:session-type;
      leaf source-interface {
        type if:interface-ref;
        description
          "Source interface.";
      }
      leaf outbound-interface {
        type if:interface-ref;
        description
          "Outbound interface.";
      }
      leaf count {
        type uint32;
        default "5";

        description
          "Specifies the number of
           packets that will be sent.";
      }
      leaf vrf {
        type coam:routing-instance-ref;
        description
          "VRF instance.";
      }
      leaf ttl {
        type uint8;
        default "255";
        description
          "Time to live (TTL).";
      }
      leaf packet-size {
        type uint32 {
          range "64..10000";
        }
        default "64";
        description
          "Size of ping echo request
           packets, in octets";
      }
    }
    output {
      list error-code-list {
        key "response-index";
        leaf response-index {
          type uint32;
          description
```

```
            "Response index.";
        }
        leaf status-code {
          type int32;
          description
            "Error code is ";
        }
        leaf status-sub-code {
          type uint8;
          description
            "Sub code.";
        }
        description
          "Error code list.";
      }
      uses coam:continuity-check-data;
    }

  }

  rpc path-discovery {
    if-feature "coam:path-discovery";
    description
      "Generates path discovery as per RFC7276.";
    input {
      container destination-tp {
        uses coam:tp-address;
        description
          "Destination test point.";
      }
      uses coam:session-type;
      leaf source-interface {
        type if:interface-ref;
        description
          "Source interface.";
      }
      leaf outbound-interface {
        type if:interface-ref;
        description
          "Outbound interface.";
      }
      leaf vrf {
        type coam:routing-instance-ref;
        description
          "VRF";
      }
      leaf max-ttl {
        type uint8;
```

```
        default "255";
        description
          "Max ttl.";
      }
    }
    output {
      list response-list {
        key "response-index";
        description
          "Path discovery response list.";
        leaf response-index {
          type uint32;
          description
            "Response index.";
        }
        leaf status-code {
          type int32;
          description

            "Error code is ";
        }
        leaf status-sub-code {
          type uint8;
          description
            "Sub code is ";
        }
      }
      uses coam:path-discovery-data;
    }
  }
}
```

<CODE ENDS>

5. Security Considerations

The YANG modules defined in this memo are 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. Some of the RPC operations in the "ietf-connectionless-oam-methods" YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control access to these operations.

These are the operations and their sensitivity/vulnerability:

o continuity-check: Generates continuity check.

o path-discovery: Generates path discovery.

6. IANA Considerations

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

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

Registrant Contact: The IESG.

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

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

name: ietf-connectionless-oam-methods

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

prefix: coam-methods

reference: RFC XXXX

7. References

7.1. Normative References

[I-D.ietf-lime-yang-connectionless-oam]
Kumar, D., Wang, Z., Wu, Q., Rahman, R., and S. Raghavan, "Generic YANG Data Model for Connectionless Operations, Administration, and Maintenance(OAM) protocols", draft-ietf-lime-yang-connectionless-oam-05 (work in progress), May 2017.

[I-D.ietf-netconf-yang-push]
Clemm, A., Voit, E., Prieto, A., Tripathy, A., Nilsen-Nygaard, E., Bierman, A., and B. Lengyel, "Subscribing to YANG datastore push updates", draft-ietf-netconf-yang-push-06 (work in progress), April 2017.

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

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

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

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

[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>.

[RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, DOI 10.17487/RFC7011, September 2013, <http://www.rfc-editor.org/info/rfc7011>.

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

7.2. Informative References

[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>.

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

[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>.

[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, <http://www.rfc-editor.org/info/rfc8029>.

Appendix A. Appendix

The following are some examples of extensions possible to the yang model. The example discusses persistent methods.

The persistent method RPCs are commands to trigger persistent continuity check or path discovery OAM while specifying the options for data-export from the device. Internet Protocol Flow Information Export (IPFIX) [RFC7011] or yang-push [I-D.ietf-netconf-yang-push]. are currently outlined here as data export options and more can be added in future. It should be noted that the persistent methods are used to trigger create, modify and delete the persistent state associated with the data export options. The data export specific configurations are beyond the scope of this document.

A.1. OAM Retrieval Persistent Methods YANG Module

```
module example-cl-oam-persistent-methods {
namespace "http://example.com/cl-oam-persistent-methods";
 prefix pcoam-methods;

 import ietf-interfaces {
  prefix if;
 }
 import ietf-connectionless-oam {
  prefix coam;
 }
 import ietf-yang-types {
  prefix yang;
 }

  identity export-method {
    description
      "Base identity to represent a conceptual export-method.";
  }

  identity ipfix-export {
    base export-method;
    description
      "IPFIX based export. Configuration provided separately.";
  }
```

```
identity yang-push-export {
  base export-method;
  description
    "Yang-push from draft-ietf-netconf-yang-push";
}

typedef export-method {
  type identityref {
    base export-method;
  }
  description
    "Export method type.";
}

typedef change-type {
  type enumeration {
    enum "create" {
      description
        "Change due to a create.";
    }
    enum "delete" {
      description
        "Change due to a delete.";
    }
    enum "modify" {
      description
        "Change due to an update.";
    }
  }
  description
    "Different types of changes that may occur.";
}

rpc cc-persistent-create {
  if-feature "coam:continuity-check";
  description
    "Generates persistent continuity-check.";
  input {
    container destination-tp {
      uses coam:tp-address;
      description
        "Destination test point.";
    }
    uses coam:session-type;
    leaf source-interface {
      type if:interface-ref;
      description
        "Source interface.";
```

```
        }
        leaf outbound-interface {
          type if:interface-ref;
          description
            "Outbound interface.";
        }
        leaf vrf {
          type coam:routing-instance-ref;
          description
            "VRF instance.";
        }
        leaf ttl {
          type uint8;
          default "255";
          description
            "Time to live (TTL).";
        }
        leaf data-export-method {
          type export-method;
          description
            "Type of export.";
        }
        choice cc-trigger {
          description
            "Defines necessary conditions for
             periodic or on-change trigger.";
          case periodic {
            description
              "Periodic reports.";
            leaf period {
              type yang:timeticks;
              description
                "Time interval between reports.";
            }
            leaf start-time {
              type yang:date-and-time;
              description
                "Timestamp from which reports are started.";
            }
          }
          case on-change {
            description
              "On-change trigger and not periodic.";
            leaf all-data-on-start {
              type boolean;
              description
                "Full update needed on start or not.";
            }
```

```
            leaf-list excluded-change {
              type change-type;
              description
                "Restrict which changes trigger an update.";
            }
          }
        }
      }
      output {
        list error-code-list {
          key "response-index";
          leaf response-index {
            type uint32;
            description
              "Response index.";
          }
          leaf status-code {
            type int32;
            description
              "Error code.";
          }
          leaf status-sub-code {
            type uint8;
            description
              "Sub code.";
          }
          description
            "Error code list.";
        }
        leaf cc-persistent-id {
          type string;
          description
            "Id to act as a cookie.";
        }
      }
    }
    rpc cc-persistent-modify {
      if-feature "coam:continuity-check";
      description
        "Modifies persistent continuity-check
         as per RFC7276.";
      input {
        leaf cc-persistent-id {
          type string;
          description
            "Cookie Id to be used for modifications.";
        }
        leaf data-export-method {
```

```
            type export-method;
            description
              "Type of export to use.";
          }
          choice cc-trigger {
            description
              "Defines necessary conditions for
               periodic or on-change trigger.";
            case periodic {
              description
                "Periodic reports.";
              leaf period {
                type yang:timeticks;
                description
                  "Time interval between reports.";
              }
              leaf start-time {
                type yang:date-and-time;
                description
                  "Timestamp from which reports are started.";
              }
            }
            case on-change {
              description
                "On-change trigger and not periodic.";
              leaf all-data-on-start {
                type boolean;
                description
                  "Full update needed on start or not.";
              }
              leaf-list excluded-change {
                type change-type;
                description
                  "Restrict which changes trigger an update.";
              }
            }
          }
        }
        output {
          list error-code-list {
            key "response-index";
            leaf response-index {
              type uint32;
              description
                "Response index.";
            }
            leaf status-code {
              type int32;
```

```
            description
              "Error code";
          }
          leaf status-sub-code {
            type uint8;
            description
              "Sub code.";
          }
          description
            "Error code list.";
        }
        leaf cc-persistent-id {
          type string;
          description
            "Id to represent a cookie.";
        }
      }
    }
    rpc cc-persistent-delete {
      if-feature "coam:continuity-check";
      description
        "Deletes persistent continuity-check as per RFC7276.";
      input {
        leaf cc-persistent-id {
          type string;
          description
            "Cookie Id to be used in deletion.";
        }
      }
      output {
        list error-code-list {
          key "response-index";
          leaf response-index {
            type uint32;
            description
              "Response index.";
          }
          leaf status-code {
            type int32;
            description
              "Error code.";
          }
          leaf status-sub-code {
            type uint8;
            description
              "Sub code.";
          }
          description
```

```
            "Error code list.";
        }
      }
    }

    rpc pd-persistent-create {
       description
         "Generates persistent path discovery.";
       input {
         container destination-tp {
           uses coam:tp-address;
           description
             "Destination test point.";
         }
         uses coam:session-type;
         leaf source-interface {
           type if:interface-ref;
           description
             "Source interface.";
         }
         leaf outbound-interface {
           type if:interface-ref;
           description
             "Outbound interface.";
         }
         leaf vrf {
           type coam:routing-instance-ref;
           description
             "VRF";
         }
         leaf max-ttl {
           type uint8;
           default "255";
           description
             "Max ttl.";
         }
         leaf data-export-method {
           type export-method;
           description
             "Type of export.";
         }
         choice pd-trigger {
           description
             "Defines necessary conditions
              for periodic or on-change
              trigger.";
           case periodic {
             description
```

```
            "Periodic reports.";
          leaf period {
            type yang:timeticks;
            description
              "Time interval between reports.";
          }
          leaf start-time {
            type yang:date-and-time;
            description
              "Timestamp from which reports are started.";
          }
        }
        case on-change {
          description
            "On-change trigger and not periodic.";
          leaf all-data-on-start {
            type boolean;
            description
              "Full update needed on start or not.";
          }
          leaf-list excluded-change {
            type change-type;
            description
              "Restrict which changes trigger an update.";
          }
        }
      }
    }
    output {
      list response-list {
        key "response-index";
        description
          "Path discovery response list.";
        leaf response-index {
          type uint32;
          description
            "Response index.";
        }
        leaf status-code {
          type int32;
          description
            "Error code is ";
        }
        leaf status-sub-code {
          type uint8;
          description
            "Sub code is ";
        }
```

```
          leaf pd-persistent-id {
            type string;
            description
              "Id to act as a cookie.";
          }
        }
      }
    }
    rpc pd-persistent-modify {
      description
        "Modifies persistent path discovery.";
      input {
        leaf pd-persistent-id {
          type string;
          description
            "Cookie Id to be used for modifications.";
        }
        leaf data-export-method {
          type export-method;
          description
            "Type of export.";
        }
        choice pd-trigger {
          description
            "Defines necessary conditions for periodic or on-change
             trigger.";
          case periodic {
            description
              "Periodic reports.";
            leaf period {
              type yang:timeticks;
              description
                "Time interval between reports.";
            }
            leaf start-time {
              type yang:date-and-time;
              description
                "Timestamp from which reports are started.";
            }
          }
          case on-change {
            description
              "On-change trigger and not periodic.";
            leaf all-data-on-start {
              type boolean;
              description
                "Full update needed on start or not.";
            }
```

```
            leaf-list excluded-change {
              type change-type;
              description
                "Restrict which changes trigger an update.";
            }
          }
        }
      }
      output {
        list response-list {
          key "response-index";
          description
            "path discovery response list.";
          leaf response-index {
            type uint32;
            description
              "response index.";
          }
          leaf status-code {
            type int32;
            description
              "error code is ";
          }
          leaf status-sub-code {
            type uint8;
            description
              "sub code is ";
          }
          leaf pd-persistent-id {
            type string;
            description
              "Id to act as a cookie.";
          }
        }
      }
    }
    rpc pd-persistent-delete {
      description
        "Deletes persistent path discovery.";
      input {
        leaf pd-persistent-id {
          type string;
          description
            "Cookie Id to be used in deletion.";
        }
      }
      output {
        list response-list {
```

```
          key "response-index";
          description
            "path discovery response list.";
          leaf response-index {
            type uint32;
            description
              "response index.";
          }
          leaf status-code {
            type int32;
            description
              "error code is ";
          }
          leaf status-sub-code {
            type uint8;
            description
              "sub code is ";
          }
        }
      }
    }
   }
```

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