

Internet Engineering Task Force (IETF)
Request for Comments: 8533
Category: Standards Track
ISSN: 2070-1721

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

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

Abstract

This document presents a retrieval method YANG data model for connectionless Operations, Administration, and Maintenance (OAM) protocols. It provides technology-independent RPC operations for OAM protocols that use connectionless communication. The retrieval methods model herein presented can be extended to include technology-specific details. There are two key benefits of this approach: First, it leads to uniformity between OAM protocols. Second, it supports both nested OAM workflows (i.e., performing OAM functions at different or the same levels through a unified interface) as well as interactive OAM workflows (i.e., performing OAM functions at the same levels through a unified interface).

Status of This Memo

This is an Internet Standards Track document.

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

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <https://www.rfc-editor.org/info/rfc8533>.

Copyright Notice

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

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

Table of Contents

1. Introduction	3
2. Conventions Used in This document	3
2.1. Terminology	4
2.2. Tree Diagrams	4
3. Overview of the Connectionless OAM Retrieval Methods Model .	4
3.1. RPC Operation Definitions	4
3.2. OAM Retrieval Methods Hierarchy	7
4. OAM Retrieval Methods YANG Module	16
5. Security Considerations	26
6. IANA Considerations	26
7. References	27
7.1. Normative References	27
7.2. Informative References	28
Appendix A. Extending Connectionless OAM Method Module Example .	29
A.1. Example of New Retrieval Procedures Model	29
Acknowledgements	40
Authors' Addresses	41

1. Introduction

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

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

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

Ping and Traceroute [RFC4443], as well as Bidirectional Forwarding Detection (BFD) [RFC5880], are well-known fault verification and isolation tools, respectively, for IP networks [RFC792]. Over the years, different technologies have developed similar toolsets for equivalent purposes.

This document presents an on-demand retrieval method YANG data model for OAM protocols that use connectionless communication. This model provides technology-independent RPC operations for OAM protocols that use connectionless communication (i.e., connectionless OAM). It is separated from the generic YANG data model for connectionless OAM [RFC8532] and can avoid mixing the models for the retrieved data from the retrieval procedures. It is expected that retrieval procedures will evolve faster than the data model [RFC8532] and will allow new procedures to be defined for retrieval of the same data defined by the generic YANG data model for connectionless OAM.

2. Conventions Used in This document

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

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

The following terms are defined in [RFC6020] and are used in this document:

- 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

RPC - Remote Procedure Call

RPC Operation - A specific Remote Procedure Call

2.2. Tree Diagrams

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

3. Overview of the Connectionless OAM Retrieval Methods Model

This document describes an on-demand retrieval method YANG data model for OAM protocols that use connectionless communication. This model provides technology-independent retrieval procedures (RPC operations) for connectionless OAM protocols. It provides a flexible way to retrieve the data that is defined by the "ietf-connectionless-oam.yang" module [RFC8532].

3.1. RPC Operation Definitions

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

Under the "connectionless-oam-methods" module, we summarize common OAM functions and define two generic RPC operations: 'continuity-check' and 'path-discovery'. In practice, these RPC operations are activated on demand and 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 operation corresponds to IP Traceroute [RFC792] [RFC4443].

Note that the RPC operation 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] and Label Switched Path (LSP) Ping [RFC8029]). This base building block should be extended with corresponding technology-specific parameters. To facilitate this for future enhancements to data retrieval methods, the RPCs are captured under a separate module.

The generic 'tp-address' grouping is used as data input from different RPCs described in this document. The generic 'path-discovery-data' and 'continuity-check-data' groupings defined by the "ietf-connectionless-oam.yang" module [RFC8532] are used as data outputs from different RPCs described in this document. Similar methods, including other RPCs, can retrieve the data using the same data model (i.e., the "ietf-connectionless-oam.yang" module).

```
rpc continuity-check {
  if-feature cl-oam:continuity-check;
  description
    "Continuity Check RPC operation as per RFC 7276.";
  reference
    "RFC 7276: An Overview of Operations, Administration, and
    Maintenance (OAM) Tools";
  input {
    uses rpc-input-parameters;
    ....
  }
  output {
    container response-info {
      leaf protocol-id {
        type identityref {
          base protocol-id;
        }
        mandatory true;
        description
          "Protocol used in the Continuity Check. ";
      }
      leaf protocol-id-meta-data {
        type identityref {
          base protocol-id-meta-data;
        }
        description
          "An optional metadata related to the protocol ID.";
      }
      leaf status-code {
```

```
    type identityref{
    base status-code;
  }
    mandatory true;
    description
      "Status code for Continuity Check RPC operation.";
  }
  leaf status-sub-code {
    type identityref{
      base status-sub-code;
    }
    mandatory true;
    description
      "Status-sub-code for Continuity Check RPC operation.";
  }
  description
    "Status code and status-sub-code for Continuity Check RPC
    operation.";
}
uses cl-oam:continuity-check-data;
}

rpc path-discovery {
  description
    "Path discovery RPC operation as per RFC 7276.";
  reference
    "RFC 7276: An Overview of Operations, Administration, and
    Maintenance (OAM) Tools";
  input {
    uses rpc-input-parameters;
    .....
  }
  output {
    list response-list {
      key "response-index";
      description
        "Path discovery response list.";
      leaf response-index {
        type uint32;
        mandatory true;
        description
          "Response index.";
      }
      leaf protocol-id {
        type identityref {
          base protocol-id;
        }
      }
    }
  }
}
```

```

    mandatory true;
    description
      "Protocol used in path discovery. ";
  }
  leaf protocol-id-meta-data {
    type identityref {
      base protocol-id-meta-data;
    }
    description
      "An optional metadata related to the protocol ID.";
  }
  leaf status-code {
    type identityref {
      base status-code;
    }
    mandatory true;
    description
      "Status code for path discovery RPC operation. ";
  }
  leaf status-sub-code {
    type identityref {
      base status-sub-code;
    }
    mandatory true;
    description
      "Status-sub-code for path discovery RPC operation. ";
  }
}
uses cl-oam:path-discovery-data;
}
}

```

Snippet of Data Hierarchy Related to RPC Operations

3.2. OAM Retrieval Methods Hierarchy

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

```

module: ietf-connectionless-oam-methods

rpcs:
  +---x continuity-check {cl-oam:continuity-check}?
  |   +---w input
  |   |   +---w destination-tp
  |   |   |   +---w tp-location-type    identityref
  |   |   |   +---w mac-address
  |   |   |   +---w mac-address    yang:mac-address

```

```

+---w ipv4-address
|   +---w ipv4-address      inet:ipv4-address
+---w ipv6-address
|   +---w ipv6-address      inet:ipv6-address
+---w tp-attribute
|   +---w tp-attribute-type?
|       |   address-attribute-type
+---w (tp-attribute-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?
|       |   rt:route-distinguisher
|   +---w sender-ve-id?           uint16
|   +---w receiver-ve-id?         uint16
+---:(mpls-ldp)
|   +---w (root-address)?
|       |   +---:(ip-address)
|           |   +---w source-address?
|               |   inet:ip-address
|           +---w group-ip-address?
|               |   inet:ip-address
+---:(vpn)
|   +---w as-number?
|       |   inet:as-number
+---:(global-id)
|   +---w lsp-id?                  string
+---w system-info
|   +---w router-id?      rt:router-id
+---w source-interface      if:interface-ref
+---w outbound-interface    if:interface-ref
+---w vrf?
|   cl-oam:routing-instance-ref
+---w session-type?          enumeration
+---w count?                  uint32
+---w ttl?                     uint8
+---w packet-size?            uint32
+--ro output

```



```

+--ro response-info
|   +--ro protocol-id          identityref
|   +--ro protocol-id-meta-data? identityref
|   +--ro status-code          identityref
|   +--ro status-sub-code      identityref
+--ro src-test-point
|   +--ro ni?                  routing-instance-ref
|   +--ro tp-location-type     identityref
|   +--ro mac-address
|   |   +--ro mac-address      yang:mac-address
+--ro ipv4-address
|   +--ro ipv4-address         inet:ipv4-address
+--ro ipv6-address
|   +--ro ipv6-address         inet:ipv6-address
+--ro tp-attribute
|   +--ro tp-attribute-type?
|   |   address-attribute-type
+--ro (tp-attribute-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?
|   |   rt:route-distinguisher
+--ro sender-ve-id?          uint16
+--ro receiver-ve-id?       uint16
+--:(mpls-ldp)
|   +--ro (root-address)?
|   |   +--:(ip-address)
|   |   |   +--ro source-address?
|   |   |   |   inet:ip-address
|   |   |   +--ro group-ip-address?
|   |   |   |   inet:ip-address
|   |   +--:(vpn)
|   |   |   +--ro as-number?
|   |   |   |   inet:as-number
|   |   +--:(global-id)
|   |   |   +--ro lsp-id?      string
+--ro system-info

```

```

| | +--ro router-id?    rt:router-id
| | +--ro egress-intf-name?  if:interface-ref
+--ro dest-test-point
| | +--ro ni?          routing-instance-ref
| | +--ro tp-location-type  identityref
| | +--ro mac-address
| | | +--ro mac-address  yang:mac-address
+--ro ipv4-address
| | +--ro ipv4-address  inet:ipv4-address
+--ro ipv6-address
| | +--ro ipv6-address  inet:ipv6-address
+--ro tp-attribute
| | +--ro tp-attribute-type?
| | | address-attribute-type
+--ro (tp-attribute-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?
| | | | rt:route-distinguisher
| | | +--ro sender-ve-id?         uint16
| | | +--ro receiver-ve-id?      uint16
| | +--:(mpls-mldp)
| | | +--ro (root-address)?
| | | | +--:(ip-address)
| | | | | +--ro source-address?
| | | | | | inet:ip-address
| | | | | +--ro group-ip-address?
| | | | | | inet:ip-address
| | | +--:(vpn)
| | | | +--ro as-number?
| | | | | inet:as-number
| | | +--:(global-id)
| | | | +--ro lsp-id?            string
+--ro system-info
| | +--ro router-id?    rt:router-id
+--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-loss-count?              uint32
|      |      +--ro loss-ratio?                     percentage
|      |      +--ro packet-reorder-count?           uint32
|      |      +--ro packets-out-of-seq-count?       uint32
|      |      +--ro packets-dup-count?              uint32
|      +--ro session-delay-statistics
|      |      +--ro time-unit-value?                identityref
|      |      +--ro min-delay-value?                uint32
|      |      +--ro max-delay-value?                uint32
|      |      +--ro average-delay-value?            uint32
|      +--ro session-jitter-statistics
|      |      +--ro unit-value?                     identityref
|      |      +--ro min-jitter-value?               uint32
|      |      +--ro max-jitter-value?               uint32
|      |      +--ro average-jitter-value?           uint32
+----x path-discovery {cl-oam:path-discovery}?
|      +---w input
|      |      +---w destination-tp
|      |      |      +---w tp-location-type          identityref
|      |      |      +---w mac-address
|      |      |      |      +---w mac-address        yang:mac-address
|      |      |      +---w ipv4-address
|      |      |      |      +---w ipv4-address       inet:ipv4-address
|      |      |      +---w ipv6-address
|      |      |      |      +---w ipv6-address       inet:ipv6-address
|      |      |      +---w tp-attribute
|      |      |      |      +---w tp-attribute-type?
|      |      |      |      |      address-attribute-type
|      |      |      |      +---w (tp-attribute-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?
|   |       rt:route-distinguisher
|   +---w sender-ve-id?          uint16
|   +---w receiver-ve-id?       uint16
+--:(mpls-mldp)
|   +---w (root-address)?
|   |   +--:(ip-address)
|   |   |   +---w source-address?
|   |   |   |       inet:ip-address
|   |   |   +---w group-ip-address?
|   |   |   |       inet:ip-address
|   |   +--:(vpn)
|   |   |   +---w as-number?
|   |   |   |       inet:as-number
|   |   +--:(global-id)
|   |   |   +---w lsp-id?          string
|   +---w system-info
|   |   +---w router-id?          rt:router-id
+---w source-interface            if:interface-ref
+---w outbound-interface          if:interface-ref
+---w vrf?
|   cl-oam:routing-instance-ref
+---w session-type?              enumeration
+---w max-ttl?                   uint8
+--ro output
+--ro response-list* [response-index]
|   +--ro response-index          uint32
|   +--ro protocol-id            identityref
|   +--ro protocol-id-meta-data?  identityref
|   +--ro status-code            identityref
|   +--ro status-sub-code        identityref
+--ro src-test-point
|   +--ro ni?                    routing-instance-ref
|   +--ro tp-location-type        identityref
|   +--ro mac-address
|   |   +--ro mac-address        yang:mac-address
|   +--ro ipv4-address
|   |   +--ro ipv4-address        inet:ipv4-address
|   +--ro ipv6-address
|   |   +--ro ipv6-address        inet:ipv6-address
|   +--ro tp-attribute
|   |   +--ro tp-attribute-type?
|   |   |       address-attribute-type
|   |   +--ro (tp-attribute-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?
    |       |
    |       rt:route-distinguisher
    |   +---ro sender-ve-id?              uint16
    |   +---ro receiver-ve-id?           uint16
    +---:(mpls-mldp)
    |   +---ro (root-address)?
    |       +---:(ip-address)
    |       |   +---ro source-address?
    |       |       |
    |       |       inet:ip-address
    |       |   +---ro group-ip-address?
    |       |       |
    |       |       inet:ip-address
    |       +---:(vpn)
    |       |   +---ro as-number?
    |       |       |
    |       |       inet:as-number
    |       +---:(global-id)
    |       |   +---ro lsp-id?              string
    +---ro system-info
    |   +---ro router-id?      rt:router-id
+---ro dest-test-point
|   +---ro ni?                routing-instance-ref
|   +---ro tp-location-type   identityref
|   +---ro mac-address
|       |
|       +---ro mac-address    yang:mac-address
+---ro ipv4-address
|   +---ro ipv4-address       inet:ipv4-address
+---ro ipv6-address
|   +---ro ipv6-address       inet:ipv6-address
+---ro tp-attribute
|   +---ro tp-attribute-type?
|       |
|       address-attribute-type
+---ro (tp-attribute-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?
|   |   |   |   |       rt:route-distinguisher
|   |   |   |   |--ro sender-ve-id?           uint16
|   |   |   |   |--ro receiver-ve-id?         uint16
|   |   |   +--:(mpls-mldp)
|   |   |   |   |--ro (root-address)?
|   |   |   |   |       +--:(ip-address)
|   |   |   |   |       |       |--ro source-address?
|   |   |   |   |       |       |       inet:ip-address
|   |   |   |   |       |       |--ro group-ip-address?
|   |   |   |   |       |       |       inet:ip-address
|   |   |   |   |       +--:(vpn)
|   |   |   |   |       |       |--ro as-number?
|   |   |   |   |       |       |       inet:as-number
|   |   |   |   |       +--:(global-id)
|   |   |   |   |       |       |--ro lsp-id?          string
|   |   |   +--ro system-info
|   |   |   |       |--ro router-id?    rt:router-id
|   |   |--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-loss-count?    uint32
|   |   |   |--ro loss-ratio?           percentage
|   |   |   |--ro packet-reorder-count? uint32
|   |   |   |--ro packets-out-of-seq-count? uint32
|   |   |   |--ro packets-dup-count?    uint32
|   |   |--ro session-delay-statistics
|   |   |   |--ro time-unit-value?      identityref
|   |   |   |--ro min-delay-value?      uint32
|   |   |   |--ro max-delay-value?      uint32
|   |   |   |--ro average-delay-value?  uint32
|   |   |--ro session-jitter-statistics
|   |   |   |--ro unit-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 ni?
|   |   |   routing-instance-ref
|   |   +--ro tp-location-type      identityref
|   |   +--ro mac-address
|   |   |   +--ro mac-address      yang:mac-address
|   |   +--ro ipv4-address
|   |   |   +--ro ipv4-address     inet:ipv4-address
|   |   +--ro ipv6-address
|   |   |   +--ro ipv6-address     inet:ipv6-address
|   |   +--ro tp-attribute
|   |   |   +--ro tp-attribute-type?
|   |   |   |   address-attribute-type
|   |   |   +--ro (tp-attribute-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?
|   |   |   |   |   |   rt:route-distinguisher
|   |   |   |   |   +--ro sender-ve-id?
|   |   |   |   |   |   uint16
|   |   |   |   |   +--ro receiver-ve-id?
|   |   |   |   |   |   uint16
|   |   |   |   +--:(mpls-ldp)
|   |   |   |   |   +--ro (root-address)?
|   |   |   |   |   |   +--:(ip-address)
|   |   |   |   |   |   |   +--ro source-address?
|   |   |   |   |   |   |   |   inet:ip-address
|   |   |   |   |   |   |   +--ro group-ip-address?

```

```

|               |               inet:ip-address
|               +---:(vpn)
|               |   +---ro as-number?
|               |               inet:as-number
|               +---:(global-id)
|               |   +---ro lsp-id?
|               |               string
+---ro system-info
|   +---ro router-id?   rt:router-id
+---ro timestamp-type?   identityref
+---ro timestamp-64bit
|   +---ro timestamp-sec?   uint32
|   +---ro timestamp-nanosec?   uint32
+---ro timestamp-80bit {ptp-long-format}?
|   +---ro timestamp-sec?   uint64
|   +---ro timestamp-nanosec?   uint32
+---ro ntp-timestamp-32bit
|   {ntp-short-format}?
|   +---ro timestamp-sec?   uint16
|   +---ro timestamp-nanosec?   uint16
+---ro icmp-timestamp-32bit {icmp-timestamp}?
|   +---ro timestamp-millsec?   uint32
+---ro ingress-intf-name?
|   if:interface-ref
+---ro egress-intf-name?
|   if:interface-ref
+---ro queue-depth?   uint32
+---ro transit-delay?   uint32
+---ro app-meta-data?   uint64

```

Data Hierarchy of OAM Retrieval Methods

4. OAM Retrieval Methods YANG Module

```

<CODE BEGINS> file
    "ietf-connectionless-oam-methods@2019-04-16.yang"

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

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

```



```
organization
  "IETF LIME Working Group";
contact
  "WG Web:  <https://datatracker.ietf.org/wg/lime>
  WG List:  <mailto:imap@ietf.org>

  Deepak Kumar <dekumar@cisco.com>
  Qin Wu <bill.wu@huawei.com>
  Srihari Raghavan <rihari@cisco.com>
  Michael Wang <wangzitao@huawei.com>
  Reshad Rahman <rrahman@cisco.com>";
description
  "This YANG module defines the RPC operations for
  connectionless OAM to be used within the IETF
  in a protocol-independent manner.  It is
  assumed that each protocol maps corresponding
  abstracts to its native format.  Each protocol
  may extend the YANG data model defined here to
  include protocol-specific extensions.

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

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

  This version of this YANG module is part of RFC 8533; see
  the RFC itself for full legal notices.";

revision 2019-04-16 {
  description
    "Initial revision.";
  reference
    "RFC 8533: Retrieval Methods YANG Data Model for the Management
    of Operations, Administration, and Maintenance (OAM)
    Protocols That Use Connectionless Communications";
}

identity protocol-id {
  description
    "This is the base identity for a generic protocol
    ID.  The protocol registry can be found at
    https://www.iana.org/protocols.";
}
```

```
identity protocol-id-internet {
  base protocol-id;
  description
    "Identity for Internet Protocols.";
}

identity protocol-id-proprietary {
  base protocol-id;
  description
    "Identity for proprietary protocols (e.g.,
    IP SLA).";
}

identity protocol-id-sfc {
  base protocol-id;
  description
    "Identity for Service Function Chaining.";
}

identity protocol-id-mpls {
  base protocol-id;
  description
    "The MPLS protocol.";
}

identity protocol-id-mpls-tp {
  base protocol-id;
  description
    "The MPLS-TP protocol.";
}

identity protocol-id-twamp {
  base protocol-id;
  description
    "The Two-Way Active Measurement Protocol (TWAMP)
    protocol.";
}

identity protocol-id-bier {
  base protocol-id;
  description
    "The Bit Index Explicit Replication (BIER)
    protocol.";
}

identity status-code {
  description
    "This is base identity for a status code.";
```

```
}

identity success-reach {
  base status-code;
  description
    "Indicates that the destination being verified
     is reachable (see RFC 7276).";
  reference
    "RFC 7276: An Overview of Operations, Administration, and
     Maintenance (OAM) Tools";
}

identity fail-reach {
  base status-code;
  description
    "Indicates that the destination being verified
     is not reachable (see RFC 7276).";
  reference
    "RFC 7276: An Overview of Operations, Administration, and
     Maintenance (OAM) Tools";
}

identity success-path-verification {
  base status-code;
  description
    "Indicates that the path verification is performed
     successfully (see RFC 7276).";
  reference
    "RFC 7276: An Overview of Operations, Administration, and
     Maintenance (OAM) Tools";
}

identity fail-path-verification {
  base status-code;
  description
    "Indicates that the path verification fails
     (see RFC 7276).";
  reference
    "RFC 7276: An Overview of Operations, Administration, and
     Maintenance (OAM) Tools";
}

identity status-sub-code {
  description
    "IdentityBase status-sub-code.";
}

identity invalid-cc {
```

```
    base status-sub-code;
    description
        "Indicates that the Continuity Check message is invalid
        (see RFC 7276).";
    reference
        "RFC 7276: An Overview of Operations, Administration, and
        Maintenance (OAM) Tools";
}

identity invalid-pd {
    base status-sub-code;
    description
        "Indicates that the path discovery message is invalid
        (see RFC 7276).";
    reference
        "RFC 7276: An Overview of Operations, Administration, and
        Maintenance (OAM) Tools";
}

identity protocol-id-meta-data {
    description
        "This is the base identity for metadata that corresponds
        to the protocol ID.";
}

identity protocol-internet-number {
    base protocol-id-meta-data;
    description
        "Internet Protocol number for standard
        Internet Protocols (IANA-assigned Internet
        Protocol numbers) to help in protocol processing.
        The Protocol Numbers registry can be found at
        https://www.iana.org/assignments/protocol-numbers.";
}

grouping rpc-input-parameters {
    container destination-tp {
        uses cl-oam:tp-address;
        description
            "Destination test point.";
    }
    leaf source-interface {
        type if:interface-ref;
        mandatory true;
        description
            "Source interface.";
    }
    leaf outbound-interface {
```

```
    type if:interface-ref;
    mandatory true;
    description
        "Outbound interface.";
}
leaf vrf {
    type cl-oam:routing-instance-ref;
    description
        "Virtual Routing and Forwarding (VRF) instance.";
}
description
    "Grouping for RPC input parameters";
}

rpc continuity-check {
    if-feature "cl-oam:continuity-check";
    description
        "Continuity Check RPC operation as per RFC 7276.";
    reference
        "RFC 7276: An Overview of Operations, Administration, and
        Maintenance (OAM) Tools";
    input {
        uses rpc-input-parameters;
        uses cl-oam:session-type {
            description
                "If session-type is specified, then session-type
                must be set to on demand";
        }
        leaf count {
            type uint32 {
                range "0..4294967295" {
                    description
                        "The overall number of packets to be transmitted
                        by the sender. The value of the count will be set
                        to zero (0) on creation and will thereafter
                        increase monotonically until it reaches a maximum
                        value of 2^32-1 (4294967295 decimal), when it wraps
                        around and starts increasing again from zero.";
                }
            }
            default "5";
            description
                "Specifies the number of
                packets that will be sent. By
                default, the packet number is
                set to 5.";
        }
    }
    leaf ttl {
```

```
    type uint8;
    default "255";
    description
        "Time to live (TTL) used to limit the lifetime
        of data packets transmitted in the network
        to prevent looping. The TTL value is decremented
        for every hop that the packet traverses. If the
        TTL is zero, the data packet will be discarded.";
}
leaf packet-size {
    type uint32 {
        range "64..10000";
    }
    default "64";
    description
        "Packet size of the Continuity Check message, in octets.
        By default, the packet size is set to 64 octets.";
}
}
output {
    container response-info {
        leaf protocol-id {
            type identityref {
                base protocol-id;
            }
            mandatory true;
            description
                "Protocol used in the Continuity Check message.
                This could be a standard protocol (e.g.,
                TCP/IP protocols, MPLS, etc.) or a proprietary
                protocol as identified by this field.";
        }
        leaf protocol-id-meta-data {
            type identityref {
                base protocol-id-meta-data;
            }
            description
                "An optional metadata related to the protocol ID.
                For example, this could be the Internet Protocol
                number for standard Internet Protocols used for
                help with protocol processing.";
        }
        leaf status-code {
            type identityref {
                base status-code;
            }
            mandatory true;
            description
```

```
    "Status code for Continuity Check RPC operation.
    This could be a basic status code (e.g., destination
    is reachable or destination is not reachable; see RFC 7276)
    or some customized status code as identified by this
    field.";
  reference
    "RFC 7276: An Overview of Operations, Administration, and
    Maintenance (OAM) Tools";
}
leaf status-sub-code {
  type identityref {
    base status-sub-code;
  }
  mandatory true;
  description
    "An optional status-sub-code for Continuity Check
    RPC operation. If the basic status code is destination
    reachable, this status-sub-code doesn't need to be
    specified. If the basic status code is destination
    unreachable, the status-sub-code can be used to specify
    the detailed reasons. This could be a basic
    sub-status-code (such as an invalid Continuity Check) or
    other error codes specific to the protocol under use for
    the Continuity Checks. For example, if ICMP is the
    protocol under use, the error codes defined in RFC 4443
    can be used to specify the reasons specific to ICMP.
    This technology-specific status-sub-code can be
    defined in technology-specific models.";
  reference
    "RFC 4443: The IETF Administrative Oversight Committee
    (IAOC) Member Selection Guidelines and Process.";
}
description
  "Status code and status-sub-code for Continuity Check RPC
  operation.";
}
uses cl-oam:continuity-check-data;
}

rpc path-discovery {
  if-feature "cl-oam:path-discovery";
  description
    "Path discovery RPC operation as per RFC 7276.";
  reference
    "RFC 7276: An Overview of Operations, Administration, and
    Maintenance (OAM) Tools";
  input {
```

```
uses rpc-input-parameters;
uses cl-oam:session-type {
  description
    "If session-type is specified, then session-type
     must be set to on demand";
}
leaf max-ttl {
  type uint8;
  default "255";
  description
    "Maximum TTL indicates the maximum number of hops that
     a packet is permitted to travel before being discarded
     by a router.  By default, the maximum TTL is set to
     255.";
}
}
output {
  list response-list {
    key "response-index";
    description
      "Path discovery response list.";
    leaf response-index {
      type uint32;
      mandatory true;
      description
        "Response index.";
    }
    leaf protocol-id {
      type identityref {
        base protocol-id;
      }
      mandatory true;
      description
        "Protocol used in path discovery.  This could be a
         standard protocol (e.g., TCP/IP protocols, MPLS, etc.)
         or a proprietary protocol as identified by
         this field.";
    }
    leaf protocol-id-meta-data {
      type identityref {
        base protocol-id-meta-data;
      }
      description
        "An optional metadata related to the protocol ID.
         For example, this could be the Internet Protocol
         number for standard Internet Protocols used for
         help with protocol processing.";
    }
  }
}
```



```
leaf status-code {
  type identityref {
    base status-code;
  }
  mandatory true;
  description
    "Status code for Continuity Check RPC operation.
    This could be a basic status code (e.g., destination
    is reachable or destination is not reachable) or some
    customized status code as identified by this field.";
}
leaf status-sub-code {
  type identityref {
    base status-sub-code;
  }
  mandatory true;
  description
    "An optional status-sub-code for Continuity Check
    RPC operation. If the basic status code is destination
    reachable, this status-sub-code doesn't need to be
    specified. If the basic status code is destination
    unreachable, the status-sub-code can be used to specify
    the detailed reasons. This could be a basic
    sub-status-code (such as an invalid Continuity Check) or
    other error codes specific to the protocol under use for
    Continuity Checks. For example, if ICMP is the protocol
    under use, the error codes defined in RFC 4443
    can be used to specify the reasons specific to ICMP.
    This technology-specific status-sub-code can be defined
    in technology-specific models.";
  reference
    "RFC 4443: The IETF Administrative Oversight Committee
    (IAOC) Member Selection Guidelines and Process.";
}
}
uses cl-oam:path-discovery-data;
}
}
```

<CODE ENDS>

5. Security Considerations

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

The Network Configuration Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

Some of the RPC operations in this 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.

These operations are used to retrieve the data from the device that needs to execute the OAM command. Unauthorized source access to some sensitive information in the above data may be used for network reconnaissance or lead to denial-of-service attacks on both the local device and the network.

6. IANA Considerations

This document registers a URI in the "IETF XML Registry" [RFC3688]. The following registration has been 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: cloam-methods
reference: RFC 8533

7. References

7.1. Normative References

- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://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, <<https://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, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [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, <<https://www.rfc-editor.org/info/rfc7011>>.
- [RFC792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, DOI 10.17487/RFC0792, September 1981.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.

- [RFC8532] Kumar, D., Wang, M., Wu, Q., Ed., Rahman, R., and S. Raghavan, "Generic YANG Data Model for the Management of Operations, Administration, and Maintenance (OAM) Protocols That Use Connectionless Communications", RFC 8532, DOI 10.17487/RFC8532, April 2019, <<https://www.rfc-editor.org/info/rfc8532>>.

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", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <<https://www.rfc-editor.org/info/rfc4443>>.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <<https://www.rfc-editor.org/info/rfc5880>>.
- [RFC7276] Mizrahi, T., Sprecher, N., Bellagamba, E., and Y. Weingarten, "An Overview of Operations, Administration, and Maintenance (OAM) Tools", RFC 7276, DOI 10.17487/RFC7276, June 2014, <<https://www.rfc-editor.org/info/rfc7276>>.
- [RFC8029] Kompella, K., Swallow, G., Pignataro, C., Ed., Kumar, N., Aldrin, S., and M. Chen, "Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures", RFC 8029, DOI 10.17487/RFC8029, March 2017, <<https://www.rfc-editor.org/info/rfc8029>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8407] Bierman, A., "Guidelines for Authors and Reviewers of Documents Containing YANG Data Models", BCP 216, RFC 8407, DOI 10.17487/RFC8407, October 2018, <<https://www.rfc-editor.org/info/rfc8407>>.
- [YANG-Push] Clemm, A., Voit, E., Prieto, A., Tripathy, A., Nilsen-Nygaard, E., Bierman, A., and B. Lengyel, "Subscription to YANG Datastores", Work in Progress, draft-ietf-netconf-yang-push-22, February 2019.

Appendix A. Extending Connectionless OAM Method Module Example

The following is an example of extensions possible to the "ietf-connectionless-oam-methods" YANG data model defined in this document.

The snippet below depicts an example of augmenting the "ietf-connectionless-oam-methods" YANG data model with ICMP ping attributes:

```
augment "/cloam-methods:continuity-check"
+ "/cloam-methods:output" {
  container session-rtt-statistics {
    leaf min-rtt {
      type uint32;
    }
    description
      "This minimum ping round-trip-time (RTT) received.";
  }
  leaf max-rtt {
    type uint32;
  }
  description
    "This maximum ping RTT received.";
  }
  leaf avg-rtt {
    type uint32;
  }
  description
    "The current average ping RTT.";
  }
  description
    "This container presents the ping RTT statistics.";
  }
}
```

A.1. Example of New Retrieval Procedures Model

As discussed in the Introduction section of this document, the new retrieval procedures can be defined for retrieval of the same data defined by the base YANG data model for connectionless OAM protocols. This appendix demonstrates how the base connectionless OAM data model can be extended to support persistent data retrieval besides on-demand retrieval procedures defined in Section 3, i.e., first retrieve a persistent-id based on the destination test point location information, and then retrieve the export details based on persistent-id. Internet Protocol Flow Information Export (IPFIX) [RFC7011] or YANG-Push [YANG-Push] are currently outlined here as data export options. Additional export options can be added in the future.

The YANG module "example-cl-oam-persistent-methods" shown below is intended as an illustration rather than a real definition of an RPC operation model for persistent data retrieval. For the sake of brevity, this module does not obey all the guidelines specified in [RFC8407].

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

  import ietf-interfaces {
    prefix if;
  }
  import ietf-connectionless-oam {
    prefix cl-oam;
  }
  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.";
  }

  identity protocol-id {
    description
      "A generic protocol identifier.";
  }

  identity status-code {
    description
      "Base status code.";
  }
}
```

```
identity success-reach {
  base status-code;
  description
    "Indicates that the destination being verified
     is reachable.";
}

identity fail-reach {
  base status-code;
  description
    "Indicates that the destination being verified
     is not reachable";
}

identity success-path-verification {
  base status-code;
  description
    "Indicates that the path verification is performed
     successfully.";
}

identity fail-path-verification {
  base status-code;
  description
    "Indicates that the path verification fails.";
}

identity status-sub-code {
  description
    "Base status-sub-code.";
}

identity invalid-cc {
  base status-sub-code;
  description
    "Indicates that the Continuity Check message is
     invalid.";
}

identity invalid-pd {
  base status-sub-code;
  description
    "Indicates that the path discovery message is invalid.";
}

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-get-persistent-id {
    if-feature "cl-oam:continuity-check";
    description
      "Obtains Continuity Check persistent identification
      given mapping parameters as input.";
    input {
      container destination-tp {
        uses cl-oam:tp-address;
        description
          "Destination test point.";
      }
      uses cl-oam: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 cl-oam:routing-instance-ref;
      }
    }
  }
}
```



```
    description
      "VRF instance.";
  }
}
output {
  container error-code {
    leaf protocol-id {
      type identityref {
        base protocol-id;
      }
      mandatory true;
      description
        "Protocol used. This could be a standard
        protocol (e.g., TCP/IP protocols, MPLS, etc.)
        or a proprietary protocol as identified by
        this field.";
    }
    leaf protocol-id-meta-data {
      type uint64;
      description
        "An optional metadata related to the protocol ID.
        For example, this could be the Internet Protocol
        number for standard Internet Protocols used for
        help with protocol processing.";
    }
    leaf status-code {
      type identityref {
        base status-code;
      }
      mandatory true;
      description
        "Status code.";
    }
    leaf status-sub-code {
      type identityref {
        base status-sub-code;
      }
      mandatory true;
      description
        "Sub code for the Continuity Check.";
    }
    description
      "Status code and sub code.";
  }
  leaf cc-persistent-id {
    type string;
    description
      "Id to act as a cookie.";
```

```
    }  
  }  
}  
  
rpc cc-persistent-get-export-details {  
  if-feature "cl-oam:continuity-check";  
  description  
    "Given the persistent ID, gets the configuration  
    options and details related to the configured data  
    export.";  
  input {  
    leaf cc-persistent-id {  
      type string;  
      description  
        "Persistent ID for use as a key in search.";  
    }  
  }  
  output {  
    container error-code {  
      leaf protocol-id {  
        type identityref {  
          base protocol-id;  
        }  
        mandatory true;  
        description  
          "Protocol used. This could be a standard  
          protocol (e.g., TCP/IP protocols, MPLS, etc.)  
          or a proprietary protocol as identified by  
          this field.";  
      }  
      leaf protocol-id-meta-data {  
        type uint64;  
        description  
          "An optional metadata related to the protocol ID.  
          For example, this could be the Internet Protocol  
          number for standard Internet Protocols used for  
          help with protocol processing.";  
      }  
      leaf status-code {  
        type identityref {  
          base status-code;  
        }  
        mandatory true;  
        description  
          "Status code.";  
      }  
      leaf status-sub-code {  
        type identityref {
```

```
        base status-sub-code;
    }
    mandatory true;
    description
        "Sub code for the Continuity Check.";
}
description
    "Status code and sub code.";
}
leaf data-export-method {
    type export-method;
    description
        "Type of export in use.";
}
choice cc-trigger {
    description
        "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 were started.";
        }
    }
    case on-change {
        description
            "On-change trigger and not periodic.";
        leaf all-data-on-start {
            type boolean;
            description
                "Full update done on start or not.";
        }
        leaf-list excluded-change {
            type change-type;
            description
                "Changes that will not trigger an update.";
        }
    }
}
}
```

```
}

rpc pd-get-persistent-id {
  if-feature "cl-oam:path-discovery";
  description
    "Obtains persistent path discovery identification.";
  input {
    container destination-tp {
      uses cl-oam:tp-address;
      description
        "Destination test point.";
    }
    uses cl-oam: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 cl-oam:routing-instance-ref;
      description
        "VRF";
    }
  }
}

output {
  list response-list {
    key "response-index";
    description
      "Path discovery response list.";
    leaf response-index {
      type uint32;
      mandatory true;
      description
        "Response index.";
    }
    leaf protocol-id {
      type identityref {
        base protocol-id;
      }
      mandatory true;
      description
        "Protocol used. This could be a standard
        protocol (e.g., TCP/IP protocols, MPLS, etc.)";
    }
  }
}
```

```

        or a proprietary protocol as identified by
        this field.";
    }
    leaf protocol-id-meta-data {
        type uint64;
        description
            "An optional metadata related to the protocol ID.
            For example, this could be the Internet Protocol
            number for standard Internet Protocols used for
            help with protocol processing.";
    }
    leaf status-code {
        type identityref {
            base status-code;
        }
        mandatory true;
        description
            "Status code for persistent path discovery
            information.";
    }
    leaf status-sub-code {
        type identityref {
            base status-sub-code;
        }
        mandatory true;
        description
            "Sub code for persistent path discovery
            information.";
    }
    leaf pd-persistent-id {
        type string;
        description
            "Id to act as a cookie.";
    }
}
}
}

rpc pd-persistent-get-export-details {
    if-feature "cl-oam:path-discovery";
    description
        "Given the persistent ID, gets the configuration
        options and details related to the configured data
        export.";
    input {
        leaf cc-persistent-id {
            type string;
            description

```

```
        "Persistent ID for use as a key in search.";
    }
}
output {
  list response-list {
    key "response-index";
    description
      "Path discovery response list.";
    leaf response-index {
      type uint32;
      mandatory true;
      description
        "Response index.";
    }
    leaf protocol-id {
      type identityref {
        base protocol-id;
      }
      mandatory true;
      description
        "Protocol used. This could be a standard
        protocol (e.g., TCP/IP protocols, MPLS, etc.)
        or a proprietary protocol as identified by
        this field.";
    }
    leaf protocol-id-meta-data {
      type uint64;
      description
        "An optional metadata related to the protocol ID.
        For example, this could be the Internet Protocol
        number for standard Internet Protocols used for
        help with protocol processing.";
    }
    leaf status-code {
      type identityref {
        base status-code;
      }
      mandatory true;
      description
        "Status code for persistent path discovery
        creation.";
    }
    leaf status-sub-code {
      type identityref {
        base status-sub-code;
      }
      mandatory true;
      description
```


Acknowledgements

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

Authors' Addresses

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

Email: dekumar@cisco.com

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

Email: wangzitao@huawei.com

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

Email: bill.wu@huawei.com

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

Email: rrahman@cisco.com

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

Email: srihari@cisco.com

