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

Abstract

This document presents a base YANG Data model for connection oriented OAM protocols. It provides a technology-independent abstraction of key OAM constructs for such protocols. The model presented here can be extended to include technology specific details. This guarantees uniformity in the management of OAM protocols and provides support for nested OAM workflows (i.e., performing OAM functions at different 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 January 9, 2017.

Copyright Notice

Copyright (c) 2016 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.
1. Introduction .................................................. 3
2. Conventions used in this document ............................... 4
  2.1. Terminology ............................................. 5
3. Architecture of Generic YANG Model for OAM .................... 6
4. Overview of the OAM Model ....................................... 7
  4.1. Maintenance Domain (MD) configuration ....................... 8
  4.2. Maintenance Association (MA) configuration ................. 9
  4.3. Maintenance Endpoint (MEP) configuration .................... 9
  4.4. rpc definitions ....................................... 10
  4.5. notifications ....................................... 13
  4.6. monitor statistics .................................. 13
  4.7. OAM data hierarchy .................................. 13
5. OAM YANG Module .............................................. 18
6. Base Mode .................................................. 38
  6.1. MEP Address ....................................... 38
  6.2. MEP ID for Base Mode ................................ 39
  6.3. Maintenance Association ................................ 39
7. connection-oriented oam yang model applicability ............. 39
  7.1. Generic YANG Model extension for TRILL OAM ............... 40
  7.1.1. MD Configuration Extension .......................... 40
  7.1.2. MA Configuration Extension .......................... 41
  7.1.3. MEP Configuration Extension .......................... 41
  7.1.4. RPC extension .................................... 42
  7.2. Generic YANG Model extension for MPLS-TP OAM ............. 43
  7.2.1. MD Configuration Extension .......................... 43
  7.2.2. MA Configuration Extension .......................... 44
  7.2.3. MEP Configuration Extension .......................... 45
8. Security Considerations ...................................... 45
9. IANA Considerations .......................................... 46
10. Acknowledgments ............................................ 46
11. References ................................................. 46
  11.1. Normative References ................................ 46
  11.2. Informative References ................................ 47
Appendix A. Contributors’ Addresses ................................ 48
Authors’ Addresses ............................................. 49
1. Introduction

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

1. Monitor networks connections (Connectivity Verification, Continuity Check).
2. Troubleshoot failures (Fault verification and localization).
3. Monitor Performance

An overview of OAM tools is presented in [RFC7276][RFC7276]. Over the years, many technologies have developed similar tools for fault and performance management.

[IEEE802.1Q] Connectivity Fault Management is a well-established OAM standard that is widely adopted for Ethernet networks. ITU-T [G.8013][G.8013], MEF Service OAM, MPLS-TP [RFC6371], TRILL [RFC7455][RFC7455] all define OAM mechanisms based on the manageability framework of [IEEE802.1Q] [IEEE802.1Q]CFM.

Given the wide adoption of the underlying OAM concepts defined in [IEEE802.1Q][IEEE802.1Q] CFM, it is a reasonable choice to develop the unified management framework for connection oriented OAM based on those concepts. In this document, we take the [IEEE802.1Q][IEEE802.1Q] CFM model and extend it to a technology independent framework and define the corresponding YANG model accordingly. The YANG model presented in this document is the base model for connection oriented OAM protocols and supports generic continuity check, connectivity verification (Loopback) and path discovery (traceroute). The generic YANG model for connection oriented OAM is designed to be extensible to other connection oriented technologies. Technology dependent nodes and remote process call (RPC) commands are defined in technology specific YANG models, which use and extend the base model defined here. As an example, VXLAN uses source UDP port number for flow entropy, while TRILL uses either MAC addresses, the VLAN tag or fine grain label, and/or IP addresses for flow entropy in the hashing for multipath selection. To capture this variation, corresponding YANG models would define the applicable structures as augmentation to the generic base model presented here. This accomplishes three goals: First it keeps each YANG model smaller and more manageable. Second, it allows independent development of corresponding YANG models. Third, implementations can limit support to only the applicable set of YANG models. (e.g. TRILL RBridge may only need to implement Generic model and the TRILL YANG model).
All implementations that follow the YANG framework presented in this document MUST implement the generic connection-oriented YANG model presented here.

The YANG data model presented in this document is generated at the management layer. Encapsulations and state machines may differ according to each OAM protocol. A user who wishes to issue a Continuity Check command or a Loopback or initiate a performance monitoring session can do so in the same manner regardless of the underlying protocol or technology or specific vendor implementation.

As an example, consider a scenario where Loopback from device A to Device B fails. Between device A and B there are IEEE 802.1 bridges a, b and c. Let’s assume a, b and c are using [IEEE802.1Q] CFM. Upon detecting the Loopback failures, a user may decide to drill down to the lower level at different segments of the path and issue the corresponding fault verification (LBM) and fault isolation (LTM) tools, using the same API. This ability to drill down to a lower layer of the protocol stack at a specific segment within a path for fault localization and troubleshooting is referred to as "nested OAM workflow". It is a useful concept that leads to efficient network troubleshooting and maintenance workflows. The connection-oriented OAM YANG model presented in this document facilitates that without needing changes to the underlying protocols.

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119]. Lower case uses of these words are not to be interpreted as carrying [RFC2119] significance.

The following notations are used within the data tree and carry the meaning as below.

Each node is printed as:
<status> <flags> <name> <opts> <type>

<status> is one of:
    + for current
    x for deprecated
    o for obsolete

<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
    (choice)/:(case) Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":")

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

In this document, these words will appear with that interpretation only when in ALL CAPS.

2.1. Terminology

CCM   - Continuity Check Message [IEEE802.1Q].
ECMP  - Equal Cost Multipath.
LBM   - Loopback Message [IEEE802.1Q].
MP    - Maintenance Point [IEEE802.1Q].
MEP   - Maintenance End Point [RFC7174] (Maintenance association End Point [IEEE802.1Q], MEG End Points [RFC6371]).
MIP - Maintenance Intermediate Point [RFC7174] (Maintenance domain Intermediate Point [IEEE802.1Q], MEG Intermediate Point [RFC6371]).

MA - Maintenance Association [IEEE802.1Q] [RFC7174].

MD - Maintenance Domain [IEEE802.1Q]

MEG - Maintenance Entity Group [RFC6371]

MTV - Multi-destination Tree Verification Message.

OAM - Operations, Administration, and Maintenance [RFC6291].

TRILL - Transparent Interconnection of Lots of Links [RFC6325].

CFM - Connectivity Fault Management [RFC7174] [IEEE802.1Q].

RPC - Remote Process Call.

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

CV - Connectivity Verification [RFC7276]. Connectivity Verification are used to verify that a destination is connected. It are also referred to as path verification and used to verify not only that the two MPs are connected, but also that they are connected through the expected path, allowing detection of unexpected topology changes.

3. Architecture of Generic YANG Model for OAM

In this document we define a generic YANG model for connection oriented OAM protocols. The YANG model defined here is generic in a sense that other technologies can extend it for technology specific needs. The Generic YANG model acts as the root for other OAM YANG models. This allows users to traverse between different OAM protocols without needing to redefine them within the sub-technology. The Generic YANG model for OAM provides a framework where technology-specific YANG models can inherit constructs from the base YANG models without needing to redefine them within the sub-technology.

Figure 1 depicts relationship of different YANG modules.
4. Overview of the OAM Model

In this document we adopt the concepts of the [IEEE802.1Q] CFM model and structure it such that it can be adapted to different connection oriented OAM protocols.

At the top of the Model is the Maintenance Domain. Each Maintenance Domain is associated with a Maintenance Name and a Domain Level.

Under each Maintenance Domain there is one or more Maintenance Association (MA). In TRILL this can be per Fine-Grained Label or for VPLS this can be per VPLS instance.

Under each MA, there can be two or more MEPs (Maintenance Association End Points). MEPs are addressed by their respective technology specific address identifiers. The YANG model presented here provides flexibility to accommodate different addressing schemes.

In the vertical direction orthogonal to the Maintenance Domain, presented are the commands. Those, in YANG terms, are the rpc
commands. These rpc commands provide uniform APIs for continuity check, connectivity verification (loopback), path discovery (traceroute) and their equivalents as well as other OAM commands.

The generic YANG model defined here does not require explicit configuration of OAM entities prior to using any of the OAM tools. The OAM tools used here are limited to OAM toolset specified in section 5.1 of [RFC7276]. In order to facilitate zero-touch experience, this document defines a default mode of OAM. The default mode of OAM is referred to as the Base Mode and specifies default values for each of model parameters, such as Maintenance Domain Level, Name of the Maintenance Association, Addresses of MEPs and so on. The default values of these depend on the technology. Base Mode for TRILL is defined in [RFC7455]. Base mode for other technologies and future extensions developed in IETF will be defined in their corresponding documents.

It is important to note that, no specific enhancements are needed in the YANG model to support Base Mode. Implementations that comply with this document, by default implement the data nodes of the applicable technology. Data nodes of the Base Mode are read-only nodes.

4.1. Maintenance Domain (MD) configuration

The container "domains" is the top level container within the gen-oam module. Within the container "domains", separate list is maintained per MD. The MD list uses the key MD-name-string for indexing. MD-name-string is a leaf and derived from type string. Additional name formats as defined in [IEEE802.1Q] or other standards can be included by association of the MD-name-format with an identity-ref. MD-name-format indicates the format of the augmented MD-names. MD-name is presented as choice/case construct. Thus, it is easily augmentable by derivative work.

module: ietf-conn-oam
  +++rw domains
    +++rw domain* [technology MD-name-string]
      +++rw technology identityref
      +++rw MD-name-string MD-name-string
      +++rw MD-name-format? identityref
      +++rw (MD-name)?
        |  +++:(MD-name-null)
        |     |  +++rw MD-name-null? empty
        |  +++rw md-level? MD-level

Snippet of data hierarchy related to OAM domains
4.2. Maintenance Association (MA) configuration

Within a given Maintenance Domain there can be one or more Maintenance Associations (MA). MAs are represented as a list and indexed by the MA-name-string. Similar to MD-name defined previously, additional name formats can be added by augmenting the name-format identity-ref and adding applicable case statements to MA-name.

```
module: ietf-conn-oam
  +--rw domains
    +--rw domain* [technology MD-name-string]
      
    +--rw MAs
      +--rw MA* [MA-name-string]
        +--rw MA-name-string MA-name-string
        +--rw MA-name-format? identityref
        +--rw (MA-name)?
          | +--:(MA-name-null)
          |   +--rw MA-name-null? empty
```

Snippet of data hierarchy related to Maintenance Associations (MA)

4.3. Maintenance Endpoint (MEP) configuration

Within a given Maintenance Association (MA), there can be one or more Maintenance End Points (MEP). MEPs are represented as a list within the data hierarchy and indexed by the key MEP-name.
4.4. 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 obtain a response. rpc model defined here abstracts OAM specific commands in a technology independent manner.

There are several rpc commands defined for the purpose of OAM. In this section we present a snippet of the continuity check command for illustration purposes. Please refer to Section 4 for the complete data hierarchy and Section 5 for the YANG model.
rpcs:
  +---x continuity-check {continuity-check}?
    +---w input
      | +---w technology? identityref
      | +---w MD-name-string -> /domains/domain/MD-name-string
      | +---w MA-name-string -> /domains/domain/MAs/MA/MA-name-string
      | +---w cos-id? uint8
      | +---w (ttl)?
      |     +--:(ip-ttl)
      |        | +---w ip-ttl? uint8
      |     +--:(mpls-ttl) +---w mpls-ttl? uint8
      | +---w sub-type? identityref
      | +---w source-mep? -> /domains/domain/MAs/MA/MEP/mep-name
      | +---w destination-mep
      |     +---w (mep-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
      |     +---w (MEP-ID)?
      |        +--:(MEP-ID-int)
      |     +---w MEP-ID-int? int32
    +---w count? uint32
    +---w cc-transmit-interval? Interval
    +---w packet-size? uint32
  +---ro output
    +---ro (monitor-stats)?
    | +---:(monitor-null)
    |   +---ro monitor-null? empty
  +---x continuity-verification {connectivity-verification}?
    +---w input
      | +---w MD-name-string -> /domains/domain/MD-name-string
      | +---w MA-name-string -> /domains/domain/MAs/MA/MA-name-string
      | +---w cos-id? uint8
      | +---w (ttl)?
      |     +--:(ip-ttl)
      |        | +---w ip-ttl? uint8
      |     +--:(mpls-ttl)
      |     +---w mpls-ttl? uint8
      | +---w sub-type? identityref
      | +---w source-mep? -> /domains/domain/MAs/MA/MEP/mep-name
      | +---w destination-mep
      |     +---w (mep-address)?
      |        | +--:(mac-address)
4.5.  notifications

Notification is sent on defect condition and defect clears with Maintenance Domain Name, MA Name, defect-type (The currently active defects), generating-mepid, and defect-message to indicate more details.

4.6.  monitor statistics

Grouping for monitoring statistics is to be used by Yang modules which Augment Yang to provide statistics due to pro-active OAM like CCM Messages. For example CCM Transmit, CCM Receive, CCM Errors, etc.

4.7.  OAM data hierarchy

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

module: ietf-conn-oam
  +--rw domains
    +--rw domain* [technology MD-name-string]
      +--rw technology identityref
      +--rw MD-name-string MD-name-string
      +--rw MD-name-format? identityref
      +--rw (MD-name)?
        +--:(MD-name-null)
          +--rw MD-name-null? empty
+++rw md-level?  MD-level
++-rw MAs
  +++rw MA* [MA-name-string]
    ++-rw MA-name-string  MA-name-string
    +++rw MA-name-format?  identityref
    +++rw (MA-name)?
      ++-: (MA-name-null)
        +++rw MA-name-null?  empty
    +++rw (MA-ID)?
      ++-: (MA-id)
        ++-rw MA-id?  uint32
      ++-: (MEG-ID)
        ++-rw meg-id?  string
    +++rw (connectivity-context)?
      ++-: (context-null)
        +++rw context-null?  empty
    +++rw cos-id?  uint8
    +++rw cc-enable?  boolean
  +++rw MEP* [mep-name]
    ++-rw mep-name  MEP-name
    +++rw (MEP-ID)?
      ++-: (MEP-ID-int)
        ++-rw MEP-ID-int?  int32
      +++rw MEP-ID-format?  identityref
    +++rw (mep-address)?
      ++-: (mac-address)
        ++-rw mac-address?  yang:mac-address
      ++-: (ipv4-address)
        ++-rw ipv4-address?  inet:ipv4-address
      ++-: (ipv6-address)
        +++rw ipv6-address?  inet:ipv6-address
    +++rw cos-id?  uint8
    +++rw cc-enable?  boolean
  +++rw session* [session-cookie]
    ++-rw session-cookie  uint32
    +++rw destination-mep
      ++-rw (MEP-ID)?
        ++-: (MEP-ID-int)
          ++-rw MEP-ID-int?  int32
      +++rw MEP-ID-format?  identityref
      +++rw destination-mep-address
        ++-rw (mep-address)?
          ++-: (mac-address)
            ++-rw mac-address?  yang:mac-address
          ++-: (ipv4-address)
            ++-rw ipv4-address?  inet:ipv4-address
          ++-: (ipv6-address)
            +++rw ipv6-address?  inet:ipv6-address
Internet-Draft     Connection-Oriented OAM YANG model          July 2016

|     +--rw cos-id?                    uint8
+++rw MIP* [interface]
     ++-rw interface     if:interface-ref
rpcs:
    +++x continuity-check {continuity-check}?
        +++w input
            +++-w technology?       identityref
            +++-w MD-name-string     -> /domains/domain/MD-name-string
            +++-w MA-name-string     -> /domains/domain/MAs/MA/MA-name-string
            +++-w cos-id?            uint8
            +++-w (ttl)?
                ++-:(ip-ttl)
                |   +++-w ip-ttl?         uint8
                +-:(mpls-ttl)
                |   +++-w mpls-ttl?       uint8
            +++-w sub-type?          identityref
            +++-w source-mep?        -> /domains/domain/MAs/MA/MEP/mep-name
            +++-w (mep-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
            ++-:(MEP-ID)?
                ++-:(MEP-ID-int)        int32
                |   +++-w MEP-ID-int?     int32
            +++-w MEP-ID-format?      identityref
            +++-w count?              uint32
            +++-w cc-transmit-interval? Interval
            +++-w packet-size?        uint32
        +++ro output
            +++-ro (monitor-stats)?
                ++-:(monitor-null)
                |   +++-ro monitor-null? empty
    +++x connectivity-verification {connectivity-verification}?
        +++w input
            +++-w MD-name-string     -> /domains/domain/MD-name-string
            +++-w MA-name-string     -> /domains/domain/MAs/MA/MA-name-string
            +++-w cos-id?            uint8
            +++-w (ttl)?
                ++-:(ip-ttl)
                |   +++-w ip-ttl?         uint8
                +-:(mpls-ttl)
                |   +++-w mpls-ttl?       uint8
            +++-w sub-type?          identityref
            +++-w source-mep?        -> /domains/domain/MAs/MA/MEP/mep-name

```yang
destination-mep
  (mep-address)?
    (mac-address)
      mac-address? yang:mac-address
    (ipv4-address)
      ipv4-address? inet:ipv4-address
    (ipv6-address)
      ipv6-address? inet:ipv6-address
  (MEP-ID)?
    (MEP-ID-int)
      MEP-ID-int? int32
  MEP-ID-format? identityref
  count? uint32
  interval? Interval
  packet-size? uint32
output
  (monitor-stats)?
    (monitor-null)
      monitor-null? empty
traceroute {traceroute}?
input
  MD-name-string
  MA-name-string
  cos-id? uint8
  (ttl)?
    ip-ttl? uint8
    mpls-ttl?
      mpls-ttl? uint8
  command-sub-type? identityref
  source-mep?
    destination-mep
      (mep-address)?
        (mac-address)
          mac-address? yang:mac-address
        (ipv4-address)
          ipv4-address? inet:ipv4-address
        (ipv6-address)
          ipv6-address? inet:ipv6-address
      (MEP-ID)?
        (MEP-ID-int)
          MEP-ID-int? int32
      MEP-ID-format? identityref
      count? uint32
      interval? Interval
ro output
response* [response-index]
  response-index uint8
```
+++ro (ttl)?
  +--:(ip-ttl)
    |  +++ro ip-ttl?  uint8
  +--:(mpls-ttl)
    +++ro mpls-ttl?  uint8

+++ro destination-mep
  +++ro (mep-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
    +++ro (MEP-ID)?
      |  +--:(MEP-ID-int)
        |  +++ro MEP-ID-int?  int32
    +++ro MEP-ID-format?  identityref

+++ro (monitor-stats)?
  +--:(monitor-null)
    +++ro monitor-null?  empty

notifications:
  +++n defect-condition-notification
    +++ro technology?  identityref
    +++ro MD-name-string  -> /domains/domain/MD-name-string
    +++ro MA-name-string  -> /domains/domain/MAs/MA/MA-name-string
    +++ro mep-name?  -> /domains/domain/MAs/MA/MEP/mep-name
    +++ro defect-type?  identityref
    +++ro generating-mepid
      +++ro (MEP-ID)?
        |  +--:(MEP-ID-int)
        |  +++ro MEP-ID-int?  int32
        +++ro MEP-ID-format?  identityref
      +++ro (defect)?
        |  +--:(defect-null)
        |  +++ro defect-null?  empty
        +--:(defect-code)
          |  +++ro defect-code?  int32

  +++n defect-cleared-notification
    +++ro technology?  identityref
    +++ro MD-name-string  -> /domains/domain/MD-name-string
    +++ro MA-name-string  -> /domains/domain/MAs/MA/MA-name-string
    +++ro mep-name?  -> /domains/domain/MAs/MA/MEP/mep-name
    +++ro defect-type?  identityref
    +++ro generating-mepid
      +++ro (MEP-ID)?
        |  +--:(MEP-ID-int)
        |  +++ro MEP-ID-int?  int32
        +++ro MEP-ID-format?  identityref
5. OAM YANG Module

<CODE BEGINS> file "ietf-conn-oam.yang"

module ietf-conn-oam {
    prefix goam;
    import ietf-interfaces {
        prefix if;
    }
    import ietf-yang-types {
        prefix yang;
    }
    import ietf-inet-types {
        prefix inet;
    }
    organization "IETF LIME Working Group";
    contact
        "WG Web: http://tools.ietf.org/wg/lime"
        "WG List: mailto:lime@ietf.org"
        "WG Chair: Carlos Pignataro cpignata@cisco.com"
        "WG Chair: Ron Bonica rbonica@juniper.net"
        "Editor: Deepak Kumar dekumar@cisco.com"
        "Editor: Qin Wu bill.wu@huawei.com"
        "Editor: Zitao Wang wangzitao@huawei.com"
    description
        "This YANG module defines the generic configuration, statistics and rpc for connection oriented OAM to be used within IETF in a protocol independent manner. Functional level abstraction is independent 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 2016-03-15 {
        description
            "Initial revision. - 05 version";
    }
}
reference "draft-ietf-lime-yang-oam-model";
}

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

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

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

/* Identities */

identity technology-types {
  description
    "this is the base identity of technology types which are TRILL,MPLS-TP,vpls etc";
}

identity command-sub-type {
  description
    "defines different rpc command subtypes, e.g rfc6905 trill OAM, this is optional for most cases";
}

identity name-format {

This defines the name format, IEEE 802.1Q CFM defines varying styles of names. It is expected name format as an identity ref to be extended with new types.

identity name-format-null {
  base name-format;
  description
    "defines name format as null";
}

identity identifier-format {
  description
    "identifier-format identity can be augmented to define other format identifiers used in MEP-ID etc";
}

identity identifier-format-integer {
  base identifier-format;
  description
    "defines identifier-format to be integer";
}

identity defect-types {
  description
    "defines different defect types, e.g. remote rdi, mis-connection defect, loss of continuity";
}

identity rdi {
  base defect-types;
  description
    "Indicates the aggregate health of the remote MEPs. ";
}

identity remote-mep-defect {
  base defect-types;
  description
    "Indicates that one or more of the remote MEPs is reporting a failure ";
}

identity loss-of-continuity {
  base defect-types;
  description
    "If no proactive CC-V OAM packets from the source MEP (and in the case of CV, this includes the
requirement to have the expected globally unique
Source MEP identifier) are received within the interval
equal to 3.5 times the receiving MEP’s
configured CC-V reception period.

identity invalid-oam-defect{
  base defect-types;
  description
  "Indicates that one or more invalid OAM messages has been
  received and that 3.5 times that OAM message transmission
  interval has not yet expired."
}

identity cross-connect-defect{
  base defect-types;
  description
  "Indicates that one or more cross-connect defect
  (for example, a service ID does not match the VLAN.)
  messages has been received and that 3.5 times that OAM message
  transmission interval has not yet expired."
}

/* typedefs */
typedef MEP-name {
  type string;
  description
    "Generic administrative name for a MEP";
}

typedef Interval{
  type decimal64{
    fraction-digits 2;
  }
  units "milliseconds";
  description
    "Interval between packets in milliseconds.
    0 means no packets are sent."
}

typedef MD-name-string {
  type string;
  description
    "Generic administrative name for an MD";
}
typedef MA-name-string {
    type string;
    description
        "Generic administrative name for an MA";
}

typedef oam-counter32 {
    type yang:zero-based-counter32;
    description
        "defines 32 bit counter for OAM";
}

typedef MD-level {
    type uint32 {
        range "0..255";
    }
    description
        "Maintenance Domain level. The level may be restricted in
         certain protocols (eg to 0-7)";
}

/* groupings */
grouping MEG-ID{
    leaf meg-id{
        type string;
        description
            "concatenation of domain and ma, For example a co-routed
             bidirectional LSP, MEG_ID is A1-{Global_ID::Node_ID::
              Tunnel_Num}::Z9-{Global_ID::Node_ID::Tunnel_Num}::LSP_Num.";
    }
    description
        "MEG-ID grouping.";
}

grouping time-to-live {
    choice ttl{
        case ip-ttl{
            leaf ip-ttl{
                type uint8;
                default "255";
                description
                    "time to live";
            }
        }
        case mpls-ttl{
            leaf mpls-ttl{

type uint8;
description
"Time to live. When an IP packet is imposed with a label, the IP TTL value is first decremented then copied into the MPLS TTL. As each LSR the MPLS frame’s TTL is decremented. This behavior can be modified with no mpls ip ttl. When a MPLS label is popped, the MPLS TTL value is decremented then copied in the IP TTL field. If the MPLS TTL value is greater than IP TTL, that value is not copied over. This is to prevent a possible condition of forwarding loop and TTL never reaching 0. When two MPLS labels are swapped, decrement by 1 and copy over the result into the new label. When a new MPLS labels is pushed, decrement by 1 and copy over the result into the new label. When a new MPLS labels is popped, decrement by 1 and copy over the result into the label below.[RFC3443]";

} }
description
"Time to Live."
} }
description
"Time to Live grouping."
}
grouping defect-message {
choice defect {
 case defect-null {
 description
 "this is a placeholder when no defect status is needed";
 leaf defect-null {
 type empty;
 description
 "there is no defect define, it will be defined in technology specific model.";
 }
 }
 case defect-code {
 description
 "this is a placeholder to display defect code."
 leaf defect-code {
 type int32;
 description
 "defect code is integer value specific to technology.";
 }
 }
}
description
  "defect Message choices.";
}

description
  "defect Message.";
}
grouping mep-address {
  choice mep-address {
    case mac-address {
      leaf mac-address {
        type yang:mac-address;
        description
          "MAC Address";
      }
      description
        "MAC Address based MEP Addressing.";
    }
    case ipv4-address {
      leaf ipv4-address {
        type inet:ipv4-address;
        description
          "Ipv4 Address";
      }
      description
        "Ip Address based MEP Addressing.";
    }
    case ipv6-address {
      leaf ipv6-address {
        type inet:ipv6-address;
        description
          "Ipv6 Address";
      }
      description
        "ipv6 Address based MEP Addressing.";
    }
    description
      "MEP Addressing.";
  }
  description
    "MEP Address";
}
grouping maintenance-domain-id {
  description
    "Grouping containing leaves sufficient to identify an MD";
  leaf technology {

type identityref {
  base technology-types;
}
mandatory true;

description
  "Defines the technology";
}
leaf MD-name-string {
  type MD-name-string;
  mandatory true;
  description
  "Defines the generic administrative maintenance domain name";
}

grouping MD-name {
  leaf MD-name-format {
    type identityref {
      base name-format;
    }
    description
      "Name format.";
  }
  choice MD-name {
    case MD-name-null {
      leaf MD-name-null {
        when ".../../MD-name-format = name-format-null" {
          description
            "MD name format is equal to null format.";
          type empty;
          description
            "MD name Null.";
        }
      }
    }
    description
      "MD name.";
  }
  description
    "MD name";
}

grouping ma-identifier {
  description
    "Grouping containing leaves sufficient to identify an MA";
  leaf MA-name-string {
    type MA-name-string;
  }
}

description
     "MA name string."

}

grouping MA-name {
    description
     "MA name";
    leaf MA-name-format {
        type identityref {
            base name-format;
            } description
            "Ma name format";
    }
    choice MA-name {
        case MA-name-null {
            leaf MA-name-null {
                when " ../../../MA-name-format = name-format-null" {
                    description
                    "MA";
                } type empty;
                } description
                "empty";
            }
        description
        "MA name";
    }
}

grouping MEP-ID {
    choice MEP-ID {
        default "MEP-ID-int";
        case MEP-ID-int {
            leaf MEP-ID-int {
                type int32;
                description
                "MEP ID in integer format";
            } description
            "MEP-ID";
        }
    }
    leaf MEP-ID-format {

type identityref {
    base identifier-format;
}
description
"MEP ID format."
}
description
"MEP-ID"
}
grouping MEP {
description
"Defines elements within the MEP"
leaf mep-name {
    type MEP-name;
    mandatory true;
    description
    "Generic administrative name of the MEP"
}
    uses MEP-ID;
    uses mep-address;
}

grouping monitor-stats {
description
"grouping for monitoring statistics, this will be augmented
by others who use this component"
choice monitor-stats {
    default "monitor-null"
    case monitor-null {
        description
        "this is a place holder when
        no monitoring statistics is needed"
        leaf monitor-null {
            type empty;
            description
            "there is no monitoring statistics to be defined"
        }
        description
        "define the monitor stats"
    }
}

grouping MIP {
    description
    "defines MIP";
}
leaf interface {
    type if:interface-ref;
    description
        "Interface";
}

grouping connectivity-context {
    description
        "Grouping defining the connectivity context for an MA; for
         example, a VRF for VPLS, or an LSP for MPLS-TP. This will be
         augmented by each protocol who use this component";
    choice connectivity-context {
        default "context-null";
        case context-null {
            description
                "this is a place holder when no context is needed";
            leaf context-null {
                type empty;
                description
                    "there is no context define";
            }
            description
                "connectivity context";
        }
    }

grouping cos {
    description
        "Priority used in transmitted packets; for example, in the
         EXP field in MPLS-TP.";
    leaf cos-id {
        type uint8;
        description
            "class of service";
    }
}

container domains {
    description
        "Contains configuration related data. Within the container
         is list of fault domains. Within each domain has List of MA.";
    list domain {
        key "technology MD-name-string";
        ordered-by system;
        description
            "...";
    }
}
"Define the list of Domains within the IETF-OAM";
uses maintenance-domain-id;
uses MD-name;
leaf md-level {
    type MD-level;
    description
    "Defines the MD-Level";
}
container MAs {
    description
    "This container defines MA, within that have multiple MA
    and within MA have MEP, MIP";
list MA {
    key "MA-name-string";
    ordered-by system;
    uses ma-identifier;
    uses MA-name;
    choice MA-ID{
        case MA-id{
            leaf MA-id{
                type uint32;
                description
                "MA Identifier";
            }
            description
            "MA ID case";
        }
        case MEG-ID{
            uses MEG-ID;
            description
            "In case MPLS-TP, the MA equivalent to MEG";
        }
        description
        "The MA/MEG identifier";
    }
    uses connectivity-context;
    uses cos {
        description
        "Default class of service for this MA, which may be overridden
        for particular MEPs, sessions or operations.";
    }
    leaf cc-enable{
        type boolean;
        description
        "Indicate whether the CC enable.";
    }
list MEP {
    key "mep-name";
ordered-by system;
description
"contain list of MEPS";
uses MEP;
uses cos;
leaf cc-enable{
  type boolean;
  description
  "Indicate whether the CC enable.";
}
list session {
  key "session-cookie";
  ordered-by user;
  description
  "Monitoring session to/from a particular remote MEP. Depending on the protocol, this could represent CC messages received from a single remote MEP (if the protocol uses multicast CCs) or a target to which unicast echo request CCs are sent and from which responses are received (if the protocol uses a unicast request/response mechanism.";
  leaf session-cookie {
    type uint32;
    description
    "Cookie to identify different sessions, when there are multiple remote MEPs or multiple sessions to the same remote MEP.";
  }
  container destination-mep {
    uses MEP-ID;
    description
    "Destination MEP";
  }
  container destination-mep-address {
    uses mep-address;
    description
    "Destination MEP Address";
  }
  uses cos;
}
list MIP {
  key "interface";
  uses MIP;
  description
  "Maintenance Intermediate Point";
}
description

"Maintenance Association list";
}
}
}

notification defect-condition-notification {

description
 "When defect condition is met this notification is sent";
leaf technology {

type identityref {
  base technology-types;
}

description
 "the technology";
}
leaf MD-name-string {

type leafref{
  path "/domains/domain/MD-name-string";
}
mandatory true;

description
 "Indicate which MD is seeing the defect";
}
leaf MA-name-string{

type leafref{
  path "/domains/domain/MAs/MA/MA-name-string";
}
mandatory true;

description
 "Indicate which MA is seeing the defect";
}
leaf mep-name {

type leafref{
  path "/domains/domain/MAs/MA/MEP/mep-name";
}
description
 "Indicate which MEP is seeing the defect";
}
leaf defect-type {

type identityref {
  base defect-types;
}
description
 "The currently active defects on the specific MEP.";
}
container generating-mepid {

uses MEP-ID;

description
  "Who is generating the defect (if known) if
  unknown make it 0.";
}

uses defect-message {
  description
    "defect message to indicate more details.";
}

notification defect-cleared-notification {
  description
    "When defect cleared is met this notification is sent";
  leaf technology {
    type identityref {
      base technology-types;
    } description
      "the technology";
  }
  leaf MD-name-string {
    type leafref{
      path "/domains/domain/MD-name-string";
    } mandatory true;
    description
      "Indicate which MD is seeing the defect";
  }
  leaf MA-name-string{
    type leafref{
      path "/domains/domain/MAs/MA/MA-name-string";
    } mandatory true;
    description
      "Indicate which MA is seeing the defect";
  }
  leaf mep-name {
    type leafref{
      path "/domains/domain/MAs/MA/MEP/mep-name";
    } description
      "Indicate which MEP is seeing the defect";
  }

  leaf defect-type {
    type identityref {
      base defect-types;
}{

description
"The currently active defects on the specific MEP."
}
}

container generating-mepid {
    uses MEP-ID;
    description
    "Who is generating the defect (if known) if
    unknown make it 0.";
}

uses defect-message {
    description
    "defect message to indicate more details.";
}
}
}

rpc continuity-check {
    if-feature continuity-check;
    description
    "Generates continuity-check as per RFC7276 Table 4.";
    input {
        leaf technology {
            type identityref {
                base technology-types;
            }
            description
            "the technology";
        }
        leaf MD-name-string {
            type leafref{
                path "/domains/domain/MD-name-string";
            }
            mandatory true;
            description
            "Indicate which MD is seeing the defect";
        }
        leaf MA-name-string{
            type leafref{
                path "/domains/domain/MAs/MA/MA-name-string";
            }
            mandatory true;
            description
            "Indicate which MA is seeing the defect";
        }
        uses cos;
        uses time-to-live;
        leaf sub-type {

type identityref {
  base command-sub-type;
}
description
  "defines different command types";
} leaf source-mep {
  type leafref{
    path "/domains/domain/MAs/MA/MEP/mep-name";
  }
description
  "Source MEP";
} container destination-mep {
  uses mep-address;
  uses MEP-ID {
    description "Only applicable if the destination is a MEP";
  }
description
  "Destination MEP";
}
leaf count {
  type uint32;
  default "3";
  description
    "Number of continuity-check message to send";
}
leaf cc-transmit-interval {
  type Interval;
  description
    "Interval between echo requests";
}
leaf packet-size {
  type uint32 {
    range "64..10000";
  }
  default "64";
  description
    "Size of continuity-check packets, in octets";
}
output {
  uses monitor-stats {
    description
      "Stats of continuity check.";
  }
}
rpc continuity-verification {
  if-feature connectivity-verification;
  description
    "Generates continuity-verification as per RFC7276 Table 4.";
  input {
    leaf MD-name-string {
      type leafref{
        path "/domains/domain/MD-name-string";
      }
      mandatory true;
      description
        "Indicate which MD is seeing the defect";
    }
    leaf MA-name-string{
      type leafref{
        path "/domains/domain/MAs/MA/MA-name-string";
      }
      mandatory true;
      description
        "Indicate which MA is seeing the defect";
    }
    uses cos;
    uses time-to-live;
    leaf sub-type {
      type identityref {
        base command-sub-type;
      }
      description
        "defines different command types";
    }
    leaf source-mep {
      type leafref{
        path "/domains/domain/MAs/MA/MEP/mep-name";
      }
      description
        "Source MEP";
    }
    container destination-mep {
      uses mep-address;
      uses MEP-ID {
        description "Only applicable if the destination is a MEP";
      }
      description
        "Destination MEP";
    }
  }
}
leaf count {
    type uint32;
    default "3";
    description
        "Number of continuity-verification message to send";
}
leaf interval {
    type Interval;
    description
        "Interval between echo requests";
}
leaf packet-size {
    type uint32 {
        range "64..10000";
    }
    default "64";
    description
        "Size of continuity-verification packets, in octets";
}
}
output {
    uses monitor-stats {
        description
            "Stats of continuity check.";
    }
}
}
rpc traceroute {
    if-feature traceroute;
    description
        "Generates Traceroute or Path Trace and return response. 
        Referencing RFC7276 for common Toolset name, for 
        MPLS-TP OAM it’s Route Tracing, and for TRILL OAM It’s 
        Path Tracing tool. Starts with TTL of one and increment 
        by one at each hop. Untill destination reached or TTL 
        reach max value";
    input {
        leaf MD-name-string {
            type leafref{
                path "/domains/domain/MD-name-string";
            }
            mandatory true;
            description
                "Indicate which MD is seeing the defect";
        }
        leaf MA-name-string{
            type leafref{
path "/domains/domain/MAs/MA/MA-name-string";
}
mandatory true;
description
"Indicate which MA is seeing the defect";
}
uses cos;
uses time-to-live;
leaf command-sub-type {
  type identityref {
    base command-sub-type;
  }
description
"defines different command types";
}
leaf source-mep {
  type leafref{
    path "/domains/domain/MAs/MA/MEP/mep-name";
  }
description
"Source MEP";
}
container destination-mep {
  uses mep-address;
  uses MEP-ID {
    description "Only applicable if the destination is a MEP";
  }
description
"Destination MEP";
}
leaf count {
  type uint32;
  default "1";
description
"Number of traceroute probes to send. In protocols where a separate message is sent at each TTL, this is the number of packets to send at each TTL.";
}
leaf interval {
  type Interval;
description
"Interval between echo requests";
}
}
output {
list response {
  key "response-index";
leaf response-index {
  type uint8;
  description
    "Arbitrary index for the response. In protocols that
     guarantee there is only a single response at each TTL
     , the TTL can be used as the response
     index."
}

uses time-to-live;

container destination-mep {
  description "MEP from which the response has been received";
  uses mep-address;
  uses MEP-ID {
    description
      "Only applicable if the destination is a MEP"
  }
}

uses monitor-stats {
  description
    "Stats of traceroute."
}

description
  "List of response."
}

YANG module of OAM

<CODE ENDS>

6. Base Mode

The Base Mode defines default configuration that MUST be present in
the devices that comply with this document. Base Mode allows users
to have "zero-touch" experience. Several parameters require
technology specific definition.

6.1. MEP Address

In the Base Mode of operation, the MEP Address is by default the IP
address of the interface on which the MEP is located.
6.2. MEP ID for Base Mode

In the Base Mode of operation, each device creates a single UP MEP associated with a virtual OAM port with no physical layer (NULL PHY). The MEPID associated with this MEP is zero (0). The choice of MEP-ID zero is explained below.

MEPID is 2 octet field by default. It is never used on the wire except when using CCM. It is important to have method that can derive MEP ID of base mode in an automatic manner with no user intervention. IP address cannot be directly used for this purpose as the MEP ID is much smaller field. For Base Mode of operation we propose to use MEP ID zero (0) as the default MEP-ID.

CCM packet use MEP-ID on the payload. CCM MUST NOT be used in the Base Mode. Hence CCM MUST be disabled on the Maintenance Association of the Base Mode.

If CCM is required, users MUST configure a separate Maintenance association and assign unique value for the corresponding MEP IDs.

[IEEE802.1Q] CFM defines MEP ID as an unsigned integer in the range 1 to 8191. In this document we propose to extend the range to 0 to 65535. Value 0 is reserved for MEP ID of Base Mode operation and MUST NOT be used for other purposes.

6.3. Maintenance Association

MAID [IEEE802.1Q] has a flexible format and includes two parts: Maintenance Domain Name and Short MA name. In the Based Mode of operation, the value of the Maintenance Domain Name must be the character string "GenericBaseMode" (excluding the quotes "). In Base Mode operation Short MA Name format is set to 2-octet integer format (value 3 in Short MA Format field [IEEE802.1Q]) and Short MA name set to 65532 (0xFFFC).

7. connection-oriented oam yang model applicability

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

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

7.1. Generic YANG Model extension for TRILL OAM

The TRILL YANG module is augmenting connection oriented OAM module for both configuration and RPC commands.

The TRILL YANG module requires the base TRILL module ([I-D.ietf-trill-yang]) to be supported as there is a strong relationship between those modules.

The configuration extensions for connection oriented OAM include MD configuration extension, Technology type extension, MA configuration extension, Connectivity-Context Extension, MEP Configuration Extension, ECMP extension. In the RPC extension, the continuity-check and path-discovery RPC are extended with TRILL specific.

7.1.1. MD Configuration Extension

MD level configuration parameters are management information which can be inherited in the TRILL OAM model and set by connection oriented base model as default values. For example domain name can be set to area-ID in the TRILL OAM case. In addition, at the Maintenance Domain level, domain data node at root level can be augmented with technology type.

Note that MD level configuration parameters provides context information for management system to correlate faults, defects, network failures with location information, which helps quickly identify root causes of network failures.

7.1.1.1. Technology Type Extension

No TRILL technology type has been defined in the connection oriented base model. Therefore a technology type extension is required in the TRILL OAM model. The technology type "trill" is defined as an identity that augments the base "technology-types" defined in the connection oriented base model:

```yang
identity trill{
  base goam:technology-types;
  description
    "trill type";
}
```

7.1.2. MA Configuration Extension

MA level configuration parameters are management information which can be inherited in the TRILL OAM model and set by connection oriented base model as default values. In addition, at the Maintenance Association (MA) level, MA data node at the second level can be augmented with connectivity-context extension.

Note that MA level configuration parameters provides context information for management system to correlate faults, defects, network failures with location information, which helps quickly identify root causes of network failures.

7.1.2.1. Connectivity-Context Extension

In TRILL OAM, one example of connectivity-context is either a 12 bit VLAN ID or a 24 bit Fine Grain Label. The connection oriented base model defines a placeholder for context-id. This allows other technologies to easily augment that to include technology specific extensions. The snippet below depicts an example of augmenting connectivity-context to include either VLAN ID or Fine Grain Label.

```
augment /goam:domains/goam:domain/goam:MAs
  /goam:MA /goam:connectivity-context:
    +--:(connectivity-context-vlan)
      |  +--rw connectivity-context-vlan?   vlan
    +--:(connectivity-context-fgl)
      +--rw connectivity-context-fgl?    fgl
```

7.1.3. MEP Configuration Extension

The MEP configuration definition in the connection oriented base model already supports configuring the interface of MEP with either MAC address or IP address. In addition, the MEP address can be represented using a 2 octet RBridge Nickname in TRILL OAM. Hence, the TRILL OAM model augments the MEP configuration in base model to add a nickname case into the MEP address choice node as follows:

```
augment /goam:domains/goam:domain/goam:MAs
  /goam:MA/ goam:MEP/goam:mep-address:
    +--:( mep-address-trill)
      |  +--rw mep-address-trill?  tril-rb-nickname
```

In addition, at the Maintenance Association Endpoint (MEP) level, MEP data node at the third level can be augmented with ECMP extension.
7.1.3.1. ECMP Extension

Since TRILL supports ECMP path selection, flow-entropy in TRILL is defined as a 96 octet field in the LIME model extension for TRILL OAM. The snippet below illustrates its extension.

```
augment /goam:domains/goam:domain/goam:MAs/goam:MA/goam:MEP:
    +--rw flow-entropy-trill?   flow-entropy-trill
augment /goam:domains/goam:domain/goam:MAs/goam:MA/goam:MEP:
    +--rw flow-entropy-trill?   flow-entropy-trill
```

7.1.4. RPC extension

In the TRILL OAM YANG model, the continuity-check and path-discovery RPC commands are extended with TRILL specific requirements. The snippet below depicts an example of illustrates the TRILL OAM RPC extension.

```
augment /goam:continuity-check/goam:input:
    +--ro (out-of-band)?
    |  +--:(ipv4-address)
    |     |  |  +--ro ipv4-address? inet:ipv4-address
    |  +--:(ipv6-address)
    |     |  +--ro ipv6-address? inet:ipv6-address
    |  +--:(trill-nickname)
    |     |     +--ro trill-nickname?    tril-rb-nickname
    +--ro diagnostic-vlan?   boolean
augment /goam:continuity-check/goam:input:
    +--ro flow-entropy-trill?   flow-entropy-trill
augment /goam:continuity-check/goam:output:
    +--ro upstream-rbridge?   tril-rb-nickname
    +--ro next-hop-rbridge*   tril-rb-nickname
augment /goam:path-discovery/goam:input:
    +--ro (out-of-band)?
    |  +--:(ipv4-address)
    |     |  |  +--ro ipv4-address? inet:ipv4-address
    |  +--:(ipv6-address)
    |     |  +--ro ipv6-address? inet:ipv6-address
    |  +--:(trill-nickname)
    |     |     +--ro trill-nickname?    tril-rb-nickname
    +--ro diagnostic-vlan?   boolean
augment /goam:path-discovery/goam:input:
    +--ro flow-entropy-trill?   flow-entropy-trill
augment /goam:path-discovery/goam:output/goam:response:
    +--ro upstream-rbridge?   tril-rb-nickname
    +--ro next-hop-rbridge*   tril-rb-nickname
```
7.2. Generic YANG Model extension for MPLS-TP OAM

The MPLS-TP OAM YANG module can augment connection oriented OAM Module with some technology-specific details. And the [mpls-tp-oam-yang] presents the YANG Data model for MPLS-TP OAM.

The configuration extensions for connection oriented OAM include MD configuration extension, Technology type extension, Sub Technology Type Extension, MA configuration extension, MEP Configuration Extension.

7.2.1. MD Configuration Extension

MD level configuration parameters are management information which can be inherited in the MPLS-TP OAM model and set by LIME base model as default values. For example domain name can be set to area-ID or the provider’s Autonomous System Number (ASN) [RFC6370] in the MPLS-TP OAM case. In addition, at the Maintenance Domain level, domain data node at root level can be augmented with technology type and sub-technology type.

Note that MD level configuration parameters provides context information for management system to correlate faults, defects, network failures with location information, which helps quickly identify root causes of network failures.

7.2.1.1. Technology Type Extension

No MPLS-TP technology type has been defined in the connection oriented base model, hence it is required in the MPLS OAM model. The technology type "mpls-tp" is defined as an identity that augments the base "technology-types" defined in the connection oriented base model:

```yml
identity mpls-tp{
    base goam:technology-types;
    description
    "mpls-tp type";
}
```

7.2.1.2. Sub Technology Type Extension

In MPLS-TP, since different encapsulation types such as IP/UDP Encapsulation, PW-ACH encapsulation can be employed, the "technology-sub-type" data node is defined and added into the MPLS OAM model to further identify the encapsulation types within the MPLS-TP OAM model. Based on it, we also define a technology sub-type for IP/UDP encapsulation and PW-ACH encapsulation. Other Encapsulation types...
can be defined in the same way. The snippet below depicts an example of several encapsulation types.

```yang
identity technology-sub-type {
    description
    "certain implementations can have different encapsulation types such as ip/udp, pw-ach and so on. Instead of defining separate models for each encapsulation, we define a technology sub-type to further identify different encapsulations. Technology sub-type is associated at the MA level";
}

identity technology-sub-type-udp {
    base technology-sub-type;
    description
    "technology sub-type is IP/UDP encapsulation";
}

identity technology-sub-type-ach {
    base technology-sub-type;
    description
    "technology sub-type is PW-ACH encapsulation";
}

augment "/goam:domains/goam:domain/goam:MAs/goam:MA" {
    leaf technology-sub-type {
        type identityref {
            base technology-sub-type;
        }
    }
}

7.2.2. MA Configuration Extension

MA level configuration parameters are management information which can be inherited in the MPLS-TP OAM model and set by Connection Oriented base model as default values. Meg-Id parameter under MA data node will be selected for MPLS-TP OAM model. Therefore one example of MA Name could be MEG LSP ID or MEG Section ID or MEG PW ID[RFC6370].

Note that MA level configuration parameters provides context information for management system to correlate faults, defects, network failures with location information, which helps quickly identify root causes of network failures.
7.2.3. MEP Configuration Extension

In MPLS-TP, MEP-ID is either a variable length label value in case of G-ACH encapsulation or a 2 octet unsigned integer value in case of IP/UDP encapsulation. One example of MEP-ID is MPLS-TP LSP_MEP_ID [RFC6370]. In the connection-oriented base model, MEP-ID is defined as a choice/case node which can supports an int32 value, and the same definition can be used for MPLS-TP with no further modification. In addition, at the Maintenance Association Endpoint(MEP) level, MEP data node at the third level can be augmented with Session extension and interface extension.

8. Security Considerations

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

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

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

```
/goam:domains/goam:domain/
```

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

This document registers a URI in the IETF XML registry [RFC3688] [RFC3688]. Following the format in RFC 3688, the following registration is requested to be made:


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].

prefix: goam reference: RFC XXXX

10. Acknowledgments

Giles Heron came up with the idea of developing a YANG model as a way of creating a unified OAM API set (interface), work in this document is largely an inspiration of that. Alexander Clemm provided many valuable tips, comments and remarks that helped to refine the YANG model presented in this document.

Carlos Pignataro, David Ball, Mahesh Jethanandani, Benoit Claise, Ladislav Lhotka, GUBALLA JENS, Yuji Tochio, Gregory Mirsky, Huub van Helvoort, Tom Taylor, Dapeng Liu, Mishael Wexler, Adi Molkho participated and contributed to this document.

11. References

11.1. Normative References


11.2. Informative References


[IEEE802.1Q] "Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks", IEEE Std 802.1Q-2011, August 2011.


Appendix A. Contributors’ Addresses

Tissa Seneviratne
Consultant

Email: tsenevir@gmail.com

Norman Finn
CISCO Systems
510 McCarthy Blvd
Milpitas, CA 95035
USA

Email: nfinn@cisco.com

Samer Salam
CISCO Systems
595 Burrard St. Suite 2123
Vancouver, BC V7X 1J1
Canada

Email: ssalam@cisco.com
Authors’ Addresses

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

Email: dekumar@cisco.com

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

Email: bill.wu@huawei.com

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

Email: wangzitao@huawei.com
Abstract

This document presents a base YANG Data model for connection oriented OAM protocols. It provides a technology-independent abstraction of key OAM constructs for such protocols. The model presented here can be extended to include technology specific details. This guarantees uniformity in the management of OAM protocols and provides support for nested OAM workflows (i.e., performing OAM functions at different 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 October 11, 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.
1.  Introduction .................................................. 3
2.  Conventions used in this document .......................... 4
  2.1.  Terminology ................................................ 5
3.  Architecture of Generic YANG Model for OAM ............... 6
4.  Overview of the OAM Model .................................... 7
  4.1.  Maintenance Domain (MD) configuration .................. 8
  4.2.  Maintenance Association (MA) configuration .............. 9
  4.3.  Maintenance Endpoint (MEP) configuration ............... 9
  4.4.  RPC definitions ........................................... 10
  4.5.  Notifications .............................................. 13
  4.6.  Monitor statistics ......................................... 13
  4.7.  OAM data hierarchy ........................................ 13
5.  OAM YANG Module .............................................. 18
6.  Base Mode ..................................................... 40
  6.1.  MEP Address ................................................. 40
  6.2.  MEP ID for Base Mode ...................................... 41
  6.3.  Maintenance Association ................................... 41
7.  Connection-oriented OAM YANG model applicability .......... 41
  7.1.  Generic YANG Model extension for TRILL OAM ............ 42
    7.1.1.  MD Configuration Extension .......................... 42
    7.1.2.  MA Configuration Extension .......................... 43
    7.1.3.  MEP Configuration Extension .......................... 43
    7.1.4.  RPC extension ......................................... 44
  7.2.  Generic YANG Model extension for MPLS-TP OAM ........ 45
    7.2.1.  MD Configuration Extension .......................... 45
    7.2.2.  MA Configuration Extension .......................... 47
    7.2.3.  MEP Configuration Extension .......................... 47
8.  Security Considerations ...................................... 48
9.  IANA Considerations ......................................... 48
10. Acknowledgments .............................................. 49
11. References ................................................... 49
  11.1.  Normative References .................................... 49
  11.2.  Informative References ................................... 50
Appendix A.  Contributors’ Addresses ........................... 51
Authors’ Addresses .............................................. 52
1. Introduction

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

1. Monitor networks connections (Connectivity Verification, Continuity Check).
2. Troubleshoot failures (Fault verification and localization).
3. Monitor Performance

An overview of OAM tools is presented in [RFC7276]. Over the years, many technologies have developed similar tools for fault and performance management.

[IEEE802.1ag] Connectivity Fault Management is a well-established OAM standard that is widely adopted for Ethernet networks. ITU-T [G.8013], MEF Service OAM, MPLS-TP [RFC6371], TRILL [RFC7455] all define OAM mechanisms based on the manageability framework of CFM [IEEE802.1ag].

Given the wide adoption of the underlying OAM concepts defined in CFM [IEEE802.1ag], it is a reasonable choice to develop the unified management framework for connection oriented OAM based on those concepts. In this document, we take the CFM [IEEE802.1ag] model and extend it to a technology independent framework and define the corresponding YANG model accordingly. The YANG model presented in this document is the base model for connection oriented OAM protocols and supports generic continuity check, connectivity verification and path discovery (traceroute). The generic YANG model for connection oriented OAM is designed to be extensible to other connection oriented technologies. Technology dependent nodes and remote process call (RPC) commands are defined in technology specific YANG models, which use and extend the base model defined here. As an example, VXLAN uses source UDP port number for flow entropy, while TRILL uses either MAC addresses, the VLAN tag or fine grain label, and/or IP addresses for flow entropy in the hashing for multipath selection. To capture this variation, corresponding YANG models would define the applicable structures as augmentation to the generic base model presented here. This accomplishes three goals: First it keeps each YANG model smaller and more manageable. Second, it allows independent development of corresponding YANG models. Third, implementations can limit support to only the applicable set of YANG models. (e.g. TRILL RBridge may only need to implement Generic model and the TRILL YANG model).
All implementations that follow the YANG framework presented in this document MUST implement the generic connection oriented YANG model presented here.

The YANG data model presented in this document is generated at the management layer. Encapsulations and state machines may differ according to each OAM protocol. A user who wishes to issues a Continuity Check command or a Loopback or initiate a performance monitoring session can do so in the same manner regardless of the underlying protocol or technology or specific vendor implementation.

As an example, consider a scenario where Loopback from device A to Device B fails. Between device A and B there are IEEE 802.1 bridges a, b and c. Let’s assume a,b and c are using CFM [IEEE802.1ag]. Upon detecting the Loopback failures, a user may decide to drill down to the lower level at different segments of the path and issue the corresponding fault verification (LBM) and fault isolation (LTM) tools, using the same API. This ability to drill down to a lower layer of the protocol stack at a specific segment within a path for fault localization and troubleshooting is referred to as "nested OAM workflow". It is a useful concept that leads to efficient network troubleshooting and maintenance workflows. The connection oriented OAM YANG model presented in this document facilitates that without needing changes to the underlying protocols.

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119]. In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying [RFC2119] significance.

The following notations are used within the data tree and carry the meaning as below.

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
   -w for writable

<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
   (choice)/:(case) Parentheses enclose choice and case nodes,
   and case nodes are also marked with a colon (":")

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

2.1. Terminology

  CCM  - Continuity Check Message [IEEE802.1ag].
  ECMP - Equal Cost Multipath.
  LBM  - Loopback Message [IEEE802.1ag].
  MP   - Maintenance Point [IEEE802.1ag].
  MEP  - Maintenance End Point [RFC7174] (Maintenance association End
         Point [IEEE802.1ag], MEG End Points [RFC6371]).
  MIP  - Maintenance Intermediate Point [RFC7174] (Maintenance domain
         Intermediate Point [IEEE802.1ag], MEG Intermediate Point
         [RFC6371]).
  MA   - Maintenance Association [IEEE802.1ag] [RFC7174].
MD  - Maintenance Domain [IEEE802.1ag]
MEG - Maintenance Entity Group [RFC6371]
MTV - Multi-destination Tree Verification Message.
OAM - Operations, Administration, and Maintenance [RFC6291].
TRILL - Transparent Interconnection of Lots of Links [RFC6325].
CFM - Connectivity Fault Management [RFC7174] [IEEE802.1ag].
RPC - Remote Process Call.
CC  - Continuity Check [RFC7276]. Continuity Checks are used to verify that a destination is reachable and therefore also referred to as reachability verification.
CV  - Connectivity Verification [RFC7276]. Connectivity Verification are used to verify that a destination is connected. It are also referred to as path verification and used to verify not only that the two MPs are connected, but also that they are connected through the expected path, allowing detection of unexpected topology changes.

Proactive OAM - The proactive OAM refers to OAM actions which are carried out continuously to permit proactive reporting of fault. Proactive OAM method requires persistent configuration.

On-demand OAM - The on-demand OAM refers to OAM actions which are initiated via manual intervention for a limited time to carry out diagnostics. On-demand OAM method requires only transient configuration.

3. Architecture of Generic YANG Model for OAM

In this document we define a generic YANG model for connection oriented OAM protocols. The YANG model defined here is generic in a sense that other technologies can extend it for technology specific needs. The Generic YANG model acts as the root for other OAM YANG models. This allows users to traverse between different OAM protocols with ease through a uniform API set. This also enables a nested OAM workflow. Figure 1 depicts the relationship of different OAM YANG models to the Generic YANG Model for connection oriented OAM. The Generic YANG model for OAM provides a framework where technology-specific YANG models can inherit constructs from the base YANG models without needing to redefine them within the sub-technology.
Figure 1 depicts relationship of different YANG modules.

```
+----------+
|Connection|
|Oriented   |
gen       |
|OAM YANG  |
+----------+

+------------------------------------------+
|                       |                  |
|                       |                  |
+-+-+-+-+-+          +-+-+-+-+-+          +-+-+-+-+-+ |
| TRILL   |          | MPLS-TP |     . . .|  foo    |
|OAM YANG |          |OAM YANG |          |OAM YANG |
+-+-+-+-+-+          +-+-+-+-+-+          +-+-+-+-+-+ |
|                    |                  |                  |
|                    |              +-+-+-+-+-+ |
|                    |         . . .|  foo    |
|                    |              |sub tech |
|                    |              +-+-+-+-+-+ |
|                    |                  |
|                    |                  |
+-------------------------------------------------------+

Relationship of OAM YANG model to generic (base) YANG model

4. Overview of the OAM Model

In this document we adopt the concepts of the CFM [IEEE802.1ag] model and structure it such that it can be adapted to different connection oriented OAM protocols.

At the top of the Model is the Maintenance Domain. Each Maintenance Domain is associated with a Maintenance Name and a Domain Level.

Under each Maintenance Domain there is one or more Maintenance Association (MA). In TRILL this can be per Fine-Grained Label or for VPLS this can be per VPLS instance [RFC6136].

Under each MA, there can be two or more MEPs (Maintenance End Points). MEPs are addressed by their respective technology specific address identifiers. The YANG model presented here provides flexibility to accommodate different addressing schemes.
In the vertical direction orthogonal to the Maintenance Domain, presented are the commands. Those, in YANG terms, are the RPC commands. These RPC commands provide uniform APIs for continuity check, connectivity verification, path discovery (traceroute) and their equivalents as well as other OAM commands.

The OAM entities in the generic YANG model defined here will be either explicitly or implicitly configured using any of the OAM tools. The OAM tools used here are limited to OAM toolset specified in section 5.1 of [RFC7276]. In order to facilitate zero-touch experience, this document defines a default mode of OAM. The default mode of OAM is referred to as the Base Mode and specifies default values for each of model parameters, such as Maintenance Domain Level, Name of the Maintenance Association, Addresses of MEPs and so on. The default values of these depend on the technology. Base Mode for TRILL is defined in [RFC7455]. Base mode for other technologies and future extensions developed in IETF will be defined in their corresponding documents.

It is important to note that, no specific enhancements are needed in the YANG model to support Base Mode. Implementations that comply with this document, by default implement the data nodes of the applicable technology. Data nodes of the Base Mode are read-only nodes.

4.1. Maintenance Domain (MD) configuration

The container "domains" is the top level container within the gen-oam module. Within the container "domains", separate list is maintained per MD. The MD list uses the key MD-name-string for indexing. MD-name-string is a leaf and derived from type string. Additional name formats as defined in [IEEE802.1ag] or other standards can be included by association of the MD-name-format with an identity-ref. MD-name-format indicates the format of the augmented MD-names. MD-name is presented as choice/case construct. Thus, it is easily augmentable by derivative work.
4.2. Maintenance Association (MA) configuration

Within a given Maintenance Domain there can be one or more Maintenance Associations (MA). MAs are represented as a list and indexed by the MA-name-string. Similar to MD-name defined previously, additional name formats can be added by augmenting the name-format identity-ref and adding applicable case statements to MA-name.

4.3. Maintenance Endpoint (MEP) configuration

Within a given Maintenance Association (MA), there can be one or more Maintenance End Points (MEP). MEPs are represented as a list within the data hierarchy and indexed by the key MEP-name.
module: ietf-conn-oam
  ++-rw domains
    ++-rw domain* [technology MD-name-string]
      ++-rw technology identityref
        .
        .
      ++-rw MAs
        ++-rw MA* [MA-name-string]
          ++-rw MA-name-string MA-name-string
            .
            .
        ++-rw MEP* [mep-name]
          ++-rw mep-name MEP-name
            ++-rw (MEP-ID)?
              |    ++-:(MEP-ID-int) int32
              |    ++-rw MEP-ID-format? identityref
            ++-rw (mep-address)?
              |    ++-:(mac-address)
              |    |    ++-rw mac-address? yang:mac-address
              |    ++-:(ipv4-address)
              |    |    ++-rw ipv4-address? inet:ipv4-address
              |    ++-:(ipv6-address)
              |    |    ++-rw ipv6-address? inet:ipv6-address
            .
            .
            .

Snippet of data hierarchy related to Maintenance Endpoint (MEP)

4.4. 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
obtain a response. RPC model defined here abstracts OAM specific
commands in a technology independent manner.

There are several RPC commands defined for the purpose of OAM. In
this section we present a snippet of the continuity check command for
illustration purposes. Please refer to Section 4.5 for the complete
data hierarchy and Section 5 for the YANG model.

module: ietf-conn-oam
  ++-rw domains
    ++-rw domain* [technology MD-name-string]
      ++-rw technology identityref
        .
        .

rpcs:
++++-x continuity-check {continuity-check}?
++++-w input
|++++-w technology? identityref
|++++-w MD-name-string -> /domains/domain/MD-name-string
|++++-w md-level? -> /domains/domain/md-level
|++++-w MA-name-string -> /domains/domain/MA/MA-name-string
|++++-w cos-id? uint8
|++++-w ttl? uint8
|++++-w sub-type? identityref
|++++-w source-mep? -> /domains/domain/MAs/MA/MEP/mep-name
|++++-w destination-mep
|++++-w (mep-address)?
| |++++-(mac-address) yang:mac-address
| |++++-(ipv4-address) inet:ipv4-address
| |++++-(ipv6-address)
| |++++-w ipv6-address? inet:ipv6-address
|++++-w (MEP-ID)?
| |++++-(MEP-ID-int)
| | |++++-w MEP-ID-int? int32
| |++++-w MEP-ID-format? identityref
|++++-w count? uint32
|++++-w cc-transmit-interval? Interval
|++++-w packet-size? uint32
++++-ro output
++++-ro (monitor-stats)?
|++++-ro monitor-null? empty
++++-x connectivity-verification {connectivity-verification}?
++++-w input
|++++-w MD-name-string -> /domains/domain/MD-name-string
|++++-w md-level? -> /domains/domain/md-level
|++++-w MA-name-string -> /domains/domain/MA/MA-name-string
|++++-w cos-id? uint8
|++++-w ttl? uint8
|++++-w sub-type? identityref
|++++-w source-mep? -> /domains/domain/MAs/MA/MEP/mep-name
|++++-w destination-mep
|++++-w (mep-address)?
| |++++-(mac-address) yang:mac-address
| |++++-(ipv4-address) inet:ipv4-address
| |++++-(ipv6-address)
| |++++-w ipv6-address? inet:ipv6-address
|++++-w (MEP-ID)?
4.5. Notifications

Notification is sent on defect condition and defect clears with Maintenance Domain Name, MA Name, defect-type (The currently active defects), generating-mepid, and defect-message to indicate more details.

4.6. Monitor statistics

Grouping for monitoring statistics is to be used by Yang modules which Augment Yang to provide statistics due to pro-active OAM like CCM Messages. For example CCM Transmit, CCM Receive, CCM Errors, etc.

4.7. OAM data hierarchy

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

module: ietf-conn-oam

---rw domains
    ---rw domain* [technology MD-name-string]
        ---rw technology identityref
        ---rw MD-name-string MD-name-string
        ---rw MD-name-format? identityref
        ---rw (MD-name)?
            ---:(MD-name-null)
                ---rw MD-name-null? empty
        ---rw md-level? MD-level
    ---rw MAs
        ---rw MA* [MA-name-string]
            ---rw MA-name-string MA-name-string
---rw MA-name-format? identityref
+---rw (MA-name)?
    +--:(MA-name-null)
      +---rw MA-name-null? empty
+---rw (connectivity-context)?
    +--:(context-null)
      +---rw context-null? empty
+---rw cos-id? uint8
+---rw cc-enable? boolean
+---rw MEP* [mep-name]
    +---rw mep-name MEP-name
    +---rw (MEP-ID)?
        +--:(MEP-ID-int)
          +---rw MEP-ID-int? int32
        +---rw MEP-ID-format? identityref
        +---rw (mep-address)?
            +--:(mac-address)
            +--:ipv4-address
            +--:ipv6-address
            +---rw mac-address? yang:mac-address
            +---rw ipv4-address? inet:ipv4-address
            +---rw ipv6-address? inet:ipv6-address
        +---rw cos-id? uint8
        +---rw cc-enable? boolean
+---rw session* [session-cookie]
    +---rw session-cookie uint32
    +---rw destination-mep
        +---rw (MEP-ID)?
            +--:(MEP-ID-int)
            +---rw MEP-ID-int? int32
        +---rw destination-mep-address
            +---rw (mep-address)?
                +--:(mac-address)
                +--:ipv4-address
                +--:ipv6-address
                +---rw mac-address? yang:mac-address
                +---rw ipv4-address? inet:ipv4-address
                +---rw ipv6-address? inet:ipv6-address
        +---rw cos-id? uint8
+---rw MIP* [interface] {mip}?
    +---rw interface if:interface-ref
    +---rw (mip-address)?
        +--:(mac-address)
        +--:ipv4-address
        +--:ipv6-address
        +---rw mac-address? yang:mac-address
        +---rw ipv4-address? inet:ipv4-address
        +---rw ipv6-address?
---rw ipv6-address?  inet:ipv6-address

rpcs:
  +---x continuity-check {continuity-check}?
    +---w input
      |  +---w technology?  identityref
      |  +---w MD-name-string -> /domains/domain/MD-name-string
      |  +---w md-level?  -> /domains/domain/md-level
      |  +---w MA-name-string -> /domains/domain/MAs/MA/MA-name-string
      |  +---w cos-id?  uint8
      |  +---w ttl?  uint8
      |  +---w sub-type?  identityref
      |  +---w source-mep?  -> /domains/domain/MAs/MA/MEP/mep-name
      |  +---w destination-mep
      |  |  +---w (mep-address)?
      |  |  |  +---w mac-address?  yang:mac-address
      |  |  |  +---w ipv4-address?  inet:ipv4-address
      |  |  |  +---w ipv6-address?  inet:ipv6-address
      |  |  +---w (MEP-ID)?
      |  |  |  +---w MEP-ID-int?  int32
      |  |  |  +---w MEP-ID-format?  identityref
      |  +---w count?  uint32
      |  +---w cc-transmit-interval? Interval
      |  +---w packet-size?  uint32
    +---ro output
    +---ro (monitor-stats)?
      +---: (monitor-null)
    +---ro monitor-null?  empty
  +---x continuity-verification {connectivity-verification}?
    +---w input
      |  +---w MD-name-string -> /domains/domain/MD-name-string
      |  +---w md-level?  -> /domains/domain/md-level
      |  +---w MA-name-string -> /domains/domain/MAs/MA/MA-name-string
      |  +---w cos-id?  uint8
      |  +---w ttl?  uint8
      |  +---w sub-type?  identityref
      |  +---w source-mep?  -> /domains/domain/MAs/MA/MEP/mep-name
      |  +---w destination-mep
      |  |  +---w (mep-address)?
      |  |  |  +---w mac-address?  yang:mac-address
      |  |  |  +---w ipv4-address?  inet:ipv4-address
      |  |  |  +---w ipv6-address?  inet:ipv6-address
---w ipv6-address?   inet:ipv6-address
  +---w (MEP-ID)?
    +---:(MEP-ID-int)
      +---w MEP-ID-int?   int32
    +---w MEP-ID-format?   identityref
  +---w count?         uint32
  +---w interval?      Interval
  +---w packet-size?   uint32
+---x traceroute {traceroute}?
  +---ro output
    +---ro (monitor-stats)?
      +---:(monitor-null)
        +---ro monitor-null?   empty
  +---x traceroute {traceroute}?
    +---ro output
      +---ro (monitor-stats)?
        +---:(monitor-null)
          +---ro monitor-null?   empty
  +---w MD-name-string -> /domains/domain/MD-name-string
  +---w md-level?      -> /domains/domain/md-level
  +---w MA-name-string -> /domains/domain/MAs/MA/MA-name-string
  +---w cos-id?         uint8
  +---w ttl?            uint8
  +---w command-sub-type?   identityref
  +---w source-mep?    -> /domains/domain/MAs/MA/MEP/mep-name
  +---w destination-mep
    +---w (mep-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
    +---w (MEP-ID)?
      +---:(MEP-ID-int)
        +---w MEP-ID-int?   int32
    +---w MEP-ID-format?   identityref
  +---w count?         uint32
  +---w interval?      Interval
  +---ro output
    +---ro response* [response-index]
      +---ro response-index   uint8
      +---ro ttl?             uint8
    +---ro destination-mep
      +---ro (mep-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
      +---ro (MEP-ID)?
notifications:
  +--n defect-condition-notification
   +--ro technology? identityref
   +--ro MD-name-string -> /domains/domain/MD-name-string
   +--ro MA-name-string -> /domains/domain/MAs/MA/MA-name-string
   +--ro mep-name? -> /domains/domain/MAs/MA/MEP/mep-name
   +--ro defect-type? identityref
   +--ro generating-mepid
    |  +--ro (MEP-ID)?
    |     +--ro MEP-ID-int? int32
    |     +--ro MEP-ID-format? identityref
    +--ro (defect)?
     |  +--ro defect-null? empty
     +--ro defect-code? int32
  +--n defect-cleared-notification
   +--ro technology? identityref
   +--ro MD-name-string -> /domains/domain/MD-name-string
   +--ro MA-name-string -> /domains/domain/MAs/MA/MA-name-string
   +--ro mep-name? -> /domains/domain/MAs/MA/MEP/mep-name
   +--ro defect-type? identityref
   +--ro generating-mepid
    |  +--ro (MEP-ID)?
    |     +--ro MEP-ID-int? int32
    |     +--ro MEP-ID-format? identityref
    +--ro (defect)?
     |  +--ro defect-null? empty
     +--ro defect-code?
5. OAM YANG Module

<CODE BEGINS> file "ietf-conn-oam.yang"

module ietf-conn-oam {
   prefix goam;

   import ietf-yang-types {
      prefix yang;
   }
   import ietf-inet-types {
      prefix inet;
   }
   import ietf-interfaces {
      prefix if;
   }

   organization "IETF LIME Working Group";
   contact
      "WG Web: http://tools.ietf.org/wg/lime
      WG List: mailto:lime@ietf.org
      WG Chair: Carlos Pignataro cpignata@cisco.com
      WG Chair: Ron Bonica rbonica@juniper.net
      Editor: Deepak Kumar dekumar@cisco.com
      Editor: Qin Wu bill.wu@huawei.com
      Editor: Zitao Wang wangzitao@huawei.com";

   description
      "This YANG module defines the generic configuration,
      statistics and rpc for connection oriented 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 2017-04-10 {
      description
         "Initial revision. - 08 version";

      reference "draft-ietf-lime-yang-oam-model";
   }
}
/* features */
feature connectivity-verification {
  description
    "This feature indicates that the server supports executing connectivity verification OAM command and returning a response. Servers that do not advertise this feature will not support executing connectivity verification command or rpc model for connectivity verification command.";
}

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

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

feature mip {
  description
    "This feature indicates that the MIP (Maintenance Intermediate Point) need to be explicit configured";
}

/* Identities */

identity technology-types {
  description
    "This is the base identity of technology types which are TRILL, MPLS-TP, vplsp etc";
}

identity command-sub-type {
  description
    "Defines different rpc command subtypes, e.g rfc6905 trill OAM, this is optional for most cases";
}
identity on-demand {
  base command-sub-type;
  description
  "On demand activation - indicates that the tool is activated manually to detect a specific anomaly. On-demand OAM method requires only transient configuration.";
}

identity proactive {
  base command-sub-type;
  description
  "Proactive activation - indicates that the tool is activated on a continual basis, where messages are sent periodically, and errors are detected when a certain number of expected messages are not received. Proactive OAM method requires persistent configuration.";
}

identity name-format {
  description
  "This defines the name format, IEEE 8021ag CFM defines varying styles of names. It is expected name format as an identity ref to be extended with new types.";
}

identity name-format-null {
  base name-format;
  description
  "Defines name format as null";
}

identity identifier-format {
  description
  "Identifier-format identity can be augmented to define other format identifiers used in MEP-ID etc";
}

identity identifier-format-integer {
  base identifier-format;
  description
  "Defines identifier-format to be integer";
}

identity defect-types {
  description
  "Defines different defect types, e.g. rdi (Remote Defect Indication), loss of continuity";
}
identity rdi {
    base defect-types;
    description
        "Indicates the aggregate health of the remote MEPs. ";
}

identity remote-mep-defect{
    base defect-types;
    description
        "Indicates that one or more of the remote MEPs is reporting a failure ";
}

identity loss-of-continuity{
    base defect-types;
    description
        "If no proactive CC OAM packets from the source MEP (and in the case of CV, this includes the requirement to have the expected unique, technology dependent source MEP identifier) are received within the interval. ";
}

identity cv-defect {
    base defect-types;
    description
        "This function should support monitoring between the MEPs and, in addition, between a MEP and MIP.[RFC6371] highlights, when performing Connectivity Verification, the need for the Continuity Check and Connectivity Verification (CC-V) messages to include unique identification of the MEG that is being monitored and the MEP that originated the message."
}

identity invalid-oam-defect{
    base defect-types;
    description
        "Indicates that one or more invalid OAM messages has been received and that 3.5 times that OAM message transmission interval has not yet expired.";
}

identity cross-connect-defect{
    base defect-types;
    description
        "Indicates that one or more cross-connect defect
(for example, a service ID does not match the VLAN.)
messages has been received and that 3.5 times that OAM message
transmission interval has not yet expired.
*/

typedef MEP-name {
  type string;
  description
    "Generic administrative name for a MEP";
}

typedef Interval{
  type decimal64{
    fraction-digits 2;
  }
  units "milliseconds";
  description
    "Interval between packets in milliseconds. 0 means no packets are sent.";
}

typedef MD-name-string {
  type string;
  description
    "Generic administrative name for an MD";
}

typedef MA-name-string {
  type string;
  description
    "Generic administrative name for an MA";
}

typedef oam-counter32 {
  type yang:zero-based-counter32;
  description
    "Defines 32 bit counter for OAM";
}

typedef MD-level {
  type uint32 {
    range "0..255";
  }
  description
    "Maintenance Domain level. The level may be restricted in
certain protocols (eg to 0-7)";
}
grouping maintenance-domain-reference {
  description "This grouping uniquely identifies a maintenance domain.";
  leaf maintenance-domain {
    type leafref {
      path "/goam:domains/goam:domain/goam:MD-name-string";
    }
    description "A reference to a specific Maintenance Domain.";
  }
}

grouping maintenance-association-reference {
  description "This grouping uniquely identifies a maintenance association. It consists of a maintenance-domain-reference and a maintenance-association leafref";
  uses maintenance-domain-reference;
  leaf maintenance-association {
    type leafref {
      path "/goam:domains/goam:domain"
        +"[goam:MD-name-string = current()]/""n
        +"../maintenance-domain]/goam:MAs"
        +"/goam:MA/goam:MA-name-string";
    }
    description "A reference to a specific Maintenance Association.";
  }
}

grouping maintenance-association-end-point-reference {
  description "This grouping uniquely identifies a maintenance association. It consists of a maintenance-association-reference and a maintenance-association-end-point leafref";
  uses maintenance-association-reference;
  leaf maintenance-association-end-point {
    type leafref {
      path "/goam:domains/goam:domain"
        +"[goam:MD-name-string = current()]/""n
        +"../maintenance-domain]/goam:MAs"
        +"/goam:MA[goam:MA-name-string = "
        }
    }
}
grouping time-to-live {
  leaf ttl {
    type uint8;
    description
    "Time to Live.";
  }
  description
  "Time to Live grouping.";
}

grouping defect-message {
  choice defect {
    case defect-null {
      description
      "This is a placeholder when no defect status is needed";
      leaf defect-null {
        type empty;
        description
        "there is no defect define, it will be defined in technology specific model.";
      }
    }
    case defect-code {
      description
      "This is a placeholder to display defect code.";
      leaf defect-code {
        type int32;
        description
        "Defect code is integer value specific to technology.";
      }
    }
  }
  description
  "Defect Message choices.";
}

description
"Defect Message.";
grouping mep-address {
  choice mep-address {
    case mac-address {
      leaf mac-address {
        type yang:mac-address;
        description "MAC Address";
      }
      description "MAC Address based MEP Addressing.";
    }
    case ipv4-address {
      leaf ipv4-address {
        type inet:ipv4-address;
        description "IPv4 Address";
      }
      description "IP Address based MEP Addressing.";
    }
    case ipv6-address {
      leaf ipv6-address {
        type inet:ipv6-address;
        description "IPv6 Address";
      }
      description "IPv6 Address based MEP Addressing.";
    }
    description "MEP Addressing.";
  }
  description "MEP Address";
}

grouping mip-address {
  choice mip-address {
    case mac-address {
      leaf mac-address {
        type yang:mac-address;
        description "MAC Address";
      }
      description "MAC Address based MIP Addressing.";
    }
    case ipv4-address {
      leaf ipv4-address {
        type inet:ipv4-address;
        description "IPv4 Address";
      }
      description "IP Address based MIP Addressing.";
    }
    description "MEP Addressing.";
  }
  description "MEP Address";
}
type inet:ipv4-address;
description
 "IPv4 Address";
 }

description
 "IP Address based MIP Addressing.";
}
case ipv6-address {
 leaf ipv6-address {
  type inet:ipv6-address;
  description
  "IPv6 Address";
  }
  description
  "IPv6 Address based MIP Addressing.";
  }
  description
  "MIP Addressing.";
  }
  description
  "MIP Address";
  }
grouping maintenance-domain-id {
  description
  "Grouping containing leaves sufficient to identify an MD";
  leaf technology {
    type identityref {
      base technology-types;
    }
    mandatory true;
    description
    "Defines the technology";
  }
  leaf MD-name-string {
    type MD-name-string;
    mandatory true;
    description
    "Defines the generic administrative maintenance domain name";
  }
  }

  grouping MD-name {
    leaf MD-name-format {
      type identityref {
        base name-format;
      }
      description
      "";
    }
  }

"Name format.";
}
choice MD-name {
  case MD-name-null {
    leaf MD-name-null {
      when "../../../MD-name-format' = 'name-format-null'" {
        description
        "MD name format is equal to null format.";
      }
      type empty;
      description
      "MD name Null.";
    }
    description
    "MD name.";
  }
  description
  "MD name";
}

grouping ma-identifier {
  description
  "Grouping containing leaves sufficient to identify an MA";
  leaf MA-name-string {
    type MA-name-string;
    description
    "MA name string.";
  }
}

grouping MA-name {
  description
  "MA name";
  leaf MA-name-format {
    type identityref {
      base name-format;
    }
    description
    "Ma name format";
  }
  choice MA-name {
    case MA-name-null {
      leaf MA-name-null {
        when "../../../MA-name-format' = 'name-format-null'" {
          description
          "MA name is equal to null format.";
        }
        type empty;
        description
        "MA name Null.";
      }
      description
      "MA name.";
    }
  }
}

"MA";
}
type empty;

description "Empty";
}

description "MA name";
}

grouping MEP-ID {
  choice MEP-ID {
    default "MEP-ID-int";
    case MEP-ID-int {
      leaf MEP-ID-int {
        type int32;
        description "MEP ID in integer format";
      }
    }
    description "MEP-ID";
  }

  leaf MEP-ID-format {
    type identityref {
      base identifier-format;
    }
    description "MEP ID format.";
  }
  description "MEP-ID";
}

grouping MEP {
  description "Defines elements within the MEP";
  leaf mep-name {
    type MEP-name;
    mandatory true;
    description "Generic administrative name of the MEP";
  }
}
uses MEP-ID;
uses mep-address;
}

grouping monitor-stats {
    description "grouping for monitoring statistics, this will be augmented by others who use this component";
    choice monitor-stats {
        default "monitor-null";
        case monitor-null {
            description "This is a place holder when no monitoring statistics is needed";
            leaf monitor-null {
                type empty;
                description "There is no monitoring statistics to be defined";
            }
        }
    }
    description "Define the monitor stats";
}

grouping connectivity-context {
    description "Grouping defining the connectivity context for an MA; for example, a VRF for VPLS, or an LSP for MPLS-TP. This will be augmented by each protocol who use this component";
    choice connectivity-context {
        default "context-null";
        case context-null {
            description "This is a place holder when no context is needed";
            leaf context-null {
                type empty;
                description "There is no context define";
            }
        }
    }
    description "Connectivity context";
}

grouping cos {
    description
"Priority used in transmitted packets; for example, in the EXP field in MPLS-TP."

leaf cos-id {
  type uint8;
  description
  "Class of service";
}

grouping MIP-grouping {
  uses mip-address;
  description
  "Grouping for MIP configuration";
}

container domains {
  description
  "Contains configuration related data. Within the container is list of fault domains. Within each domain has List of MA.";
  list domain {
    key "technology MD-name-string";
    ordered-by system;
    description
    "Define the list of Domains within the IETF-OAM";
    uses maintenance-domain-id;
    uses MD-name;
    leaf md-level {
      type MD-level;
      description
      "Defines the MD-Level";
    }
  }
  container MAs {
    description
    "This container defines MA, within that have multiple MA and within MA have MEP";
    list MA {
      key "MA-name-string";
      ordered-by system;
      uses ma-identifier;
      uses MA-name;
      uses connectivity-context;
      uses cos {
        description
        "Default class of service for this MA, which may be overridden for particular MEPs, sessions or operations.";
      }
    }
  }
}
leaf cc-enable{
  type boolean;
  description
  "Indicate whether the CC enable.";
}

list MEP {
  key "mep-name";
  ordered-by system;
  description
  "Contain list of MEPS";
  uses MEP;
  uses cos;
  leaf cc-enable{
    type boolean;
    description
    "Indicate whether the CC enable.";
  }
}

list session {
  key "session-cookie";
  ordered-by user;
  description
  "Monitoring session to/from a particular remote MEP. Depending on the protocol, this could represent CC messages received from a single remote MEP (if the protocol uses multicast CCs) or a target to which unicast echo request CCs are sent and from which responses are received (if the protocol uses a unicast request/response mechanism).";
  leaf session-cookie {
    type uint32;
    description
    "Cookie to identify different sessions, when there are multiple remote MEPS or multiple sessions to the same remote MEP.";
  }
  container destination-mep {
    uses MEP-ID;
    description
    "Destination MEP";
  }
  container destination-mep-address {
    uses mep-address;
    description
    "Destination MEP Address";
  }
  uses cos;
}


list MIP {
  if-feature mip;
  key "interface";
  leaf interface {
    type if:interface-ref;
    description "Interface";
  }
  uses MIP-grouping;
  description "List for MIP";
}

notification defect-condition-notification {
  description "When defect condition is met this notification is sent";
  leaf technology {
    type identityref {
      base technology-types;
    }
    description "The technology";
  }
  leaf MD-name-string {
    type leafref {
      path "domains/domain/MD-name-string";
    }
    mandatory true;
    description "Indicate which MD is seeing the defect";
  }
  leaf MA-name-string {
    type leafref {
      path "domains/domain/MAs/MA/MA-name-string";
    }
    mandatory true;
    description "Indicate which MA is seeing the defect";
  }
  leaf mep-name {

type leafref{
  path "/domains/domain/MAs/MA/MEP/mep-name";
} description "Indicate which MEP is seeing the defect";

leaf defect-type {
  type identityref {
    base defect-types;
  } description "The currently active defects on the specific MEP.";
}

container generating-mepid {
  uses MEP-ID;
  description "Who is generating the defect (if known) if unknown make it 0.";
}
uses defect-message {
  description "Defect message to indicate more details.";
}
}

notification defect-cleared-notification {
  description "When defect cleared is met this notification is sent";
leaf technology {
  type identityref {
    base technology-types;
  } description "The technology";
}
leaf MD-name-string {
  type leafref{
    path "/domains/domain/MD-name-string";
  } mandatory true;
  description "Indicate which MD is seeing the defect";
}
leaf MA-name-string{
  type leafref{
    path "/domains/domain/MAs/MA/MA-name-string";
  }
}
mandatory true;
description
"Indicate which MA is seeing the defect";
}
leaf mep-name {
  type leafref {
    path "/domains/domain/MA/MEP/mep-name";
  } 
  description
  "Indicate which MEP is seeing the defect";
}

leaf defect-type {
  type identityref {
    base defect-types;
  } 
  description
  "The currently active defects on the specific MEP.";
}

container generating-mepid {
  uses MEP-ID;
  description
  "Who is generating the defect (if known) if unknown make it 0.";
}
uses defect-message {
  description
  "Defect message to indicate more details.";
}
}

rpc continuity-check {
  if-feature "continuity-check";
  description
  "Generates continuity-check as per RFC7276 Table 4.";
  input {
    leaf technology {
      type identityref {
        base technology-types;
      } 
      description
        "The technology";
    }
    leaf MD-name-string {
      type leafref {
        path "/domains/domain/MD-name-string";
      } 
      mandatory true;
    }
}
description "Indicate which MD is seeing the defect";
}
leaf md-level {
    type leafref {
        path "./domains/domain/md-level";
    }
    description "The maintenance domain level."
}
leaf MA-name-string{
    type leafref{
        path "./domains/domain/MA/MA-name-string";
    }
    mandatory true;
    description "Indicate which MA is seeing the defect"
}
uses cos;
uses time-to-live;
leaf sub-type {
    type identityref {
        base command-sub-type;
    }
    description "Defines different command types"
}
leaf source-mep {
    type leafref{
        path "./domains/domain/MA/MEP/mep-name";
    }
    description "Source MEP"
}
container destination-mep {
    uses mep-address;
    uses MEP-ID {
        description "Only applicable if the destination is a MEP"
    }
    description "Destination MEP"
}
leaf count {
    type uint32;
    default "3";
    description
"Number of continuity-check message to send";
}  
leaf cc-transmit-interval {
  type Interval;
  description  
    "Interval between echo requests";
}
leaf packet-size {
  type uint32 {
    range "0..10000";
  
    default "64";
    description  
      "Size of continuity-check packets, in octets";
  }
}
output {
  uses monitor-stats {
    description  
      "Stats of continuity check.";
  }
}
}

rpc continuity-verification {
  if-feature connectivity-verification;
  description  
    "Generates continuity-verification as per RFC7276 Table 4.";
  input {
    leaf MD-name-string {
      type leafref{
        path "/domains/domain/MD-name-string";
      
        mandatory true;
        description  
          "Indicate which MD is seeing the defect";
      }
    
    leaf md-level {
      type leafref {
        path "/domains/domain/md-level";
      
        description  
          "The maintenance domain level.";
      }
    
    leaf MA-name-string{
type leafref 
  path "/domains/domain/MA/MA/MA-name-string";
} 
mandatory true; 
description
"Indicate which MA is seeing the defect";
} 
uses cos; 
uses time-to-live; 
leaf sub-type { 
  type identityref { 
    base command-sub-type; 
  } 
  description
"Defines different command types";
} 
leaf source-mep { 
  type leafref{ 
    path "/domains/domain/MA/MEP/mep-name";
  } 
  description
"Source MEP";
} 
container destination-mep { 
  uses mep-address; 
  uses MEP-ID { 
    description "Only applicable if the destination is a MEP";
  } 
  description
"Destination MEP";
} 
leaf count { 
  type uint32; 
  default "3"; 
  description
"Number of continuity-verification message to send";
} 
leaf interval { 
  type Interval; 
  description
"Interval between echo requests";
} 
leaf packet-size { 
  type uint32 { 
    range "64..10000"; 
  } 
  default "64";
description
"Size of continuity-verification packets, in octets";
}
}

output {
  uses monitor-stats {
    description
    "Stats of continuity check.";
  }
}
}

call traceroute {
  if-feature traceroute;
  description
  "Generates Traceroute or Path Trace and return response.
  Referencing RFC7276 for common Toolset name, for
  MPLS-TP OAM it’s Route Tracing, and for TRILL OAM It’s
  Path Tracing tool. Starts with TTL of one and increment
  by one at each hop. Untill destination reached or TTL
  reach max value";
  input {
    leaf MD-name-string {
      type leafref{
        path "/domains/domain/MD-name-string";
      } mandatory true;
      description
      "Indicate which MD is seeing the defect";
    }
    leaf md-level {
      type leafref {
        path "/domains/domain/md-level";
      } description
      "The maintenance domain level.";
    }
    leaf MA-name-string{
      type leafref{
        path "/domains/domain/MAs/MA/MA-name-string";
      } mandatory true;
      description
      "Indicate which MA is seeing the defect";
    }
  }
  uses cos;
}
uses time-to-live;
leaf command-sub-type {
  type identityref {
    base command-sub-type;
  }
  description
    "Defines different command types";
}
leaf source-mep {
  type leafref{
    path "/domains/domain/MA/MA/MEP/mep-name";
  }
  description
    "Source MEP";
}
container destination-mep {
  uses mep-address;
  uses MEP-ID {
    description
      "Only applicable if the destination is a MEP";
  }
  description
    "Destination MEP";
}
leaf count {
  type uint32;
  default "1";
  description
    "Number of traceroute probes to send. In protocols where a separate message is sent at each TTL, this is the number of packets to send at each TTL.";
}
leaf interval {
  type Interval;
  description
    "Interval between echo requests";
}
output {
  list response {
    key "response-index";
    leaf response-index {
      type uint8;
      description
        "Arbitrary index for the response. In protocols that guarantee there is only a single response at each TTL, the TTL can be used as the response index.";
  }
6. Base Mode

The Base Mode ('default mode' described in section 4) defines default configuration that MUST be present in the devices that comply with this document. Base Mode allows users to have "zero-touch" experience. Several parameters require technology specific definition.

6.1. MEP Address

In the Base Mode of operation, the MEP Address is by default the IP address of the interface on which the MEP is located.
6.2. MEP ID for Base Mode

In the Base Mode of operation, each device creates a single MEP associated with a virtual OAM port with no physical layer (NULL PHY). The MEP-ID associated with this MEP is zero (0). The choice of MEP-ID zero is explained below.

MEP-ID is 2 octet field by default. It is never used on the wire except when using CCM. It is important to have method that can derive MEP-ID of base mode in an automatic manner with no user intervention. IP address cannot be directly used for this purpose as the MEP-ID is much smaller field. For Base Mode of operation we propose to use MEP-ID zero (0) as the default MEP-ID.

CCM packet use MEP-ID on the payload. CCM MUST NOT be used in the Base Mode. Hence CCM MUST be disabled on the Maintenance Association of the Base Mode.

If CCM is required, users MUST configure a separate Maintenance association and assign unique value for the corresponding MEP IDs.

CFM [IEEE802.1ag] defines MEP ID as an unsigned integer in the range 1 to 8191. In this document we propose extend the range to 0 to 65535. Value 0 is reserved for MEP-ID of Base Mode operation and MUST NOT be used for other purposes.

6.3. Maintenance Association

The ID of the Maintenance Association (MA-ID) [IEEE802.1ag] has a flexible format and includes two parts: Maintenance Domain Name and Short MA name. In the Based Mode of operation, the value of the Maintenance Domain Name must be the character string "GenericBaseMode" (excluding the quotes "). In Base Mode operation Short MA Name format is set to 2-octet integer format (value 3 in Short MA Format field [IEEE802.1ag]) and Short MA name set to 65532 (0xFFFFC).

7. Connection-oriented OAM YANG model applicability

"ietf-conn-oam" model defined in this document provides technology-independent abstraction of key OAM constructs for connection oriented protocols. This model can be further extended to include technology specific details, e.g., adding new data nodes with technology specific functions and parameters into proper anchor points of the base model, so as to develop a technology-specific connection-oriented OAM model.
This section demonstrates the usability of the connection-oriented YANG OAM data model to various connection-oriented OAM technologies, e.g., TRILL and MPLS-TP. Note that, in this section, we only present several snippets of technology-specific model extensions for illustrative purposes. The complete model extensions should be worked on in respective protocol working groups.

7.1. Generic YANG Model extension for TRILL OAM

The TRILL YANG module is augmenting connection oriented OAM module for both configuration and RPC commands.

The TRILL YANG module requires the base TRILL module ([I-D.ietf-trill-yang]) to be supported as there is a strong relationship between those modules.

The configuration extensions for connection oriented OAM include MD configuration extension, Technology type extension, MA configuration extension, Connectivity-Context Extension, MEP Configuration Extension, ECMP extension. In the RPC extension, the continuity-check and path-discovery RPC are extended with TRILL specific.

7.1.1. MD Configuration Extension

MD level configuration parameters are management information which can be inherited in the TRILL OAM model and set by connection oriented base model as default values. For example domain name can be set to area-ID in the TRILL OAM case. In addition, at the Maintenance Domain level, domain data node at root level can be augmented with technology type.

Note that MD level configuration parameters provides context information for management system to correlate faults, defects, network failures with location information, which helps quickly identify root causes of network failures.

7.1.1.1. Technology Type Extension

No TRILL technology type has been defined in the connection oriented base model. Therefore a technology type extension is required in the TRILL OAM model. The technology type "trill" is defined as an identity that augments the base "technology-types" defined in the connection oriented base model:
7.1.2. MA Configuration Extension

MA level configuration parameters are management information which can be inherited in the TRILL OAM model and set by connection oriented base model as default values. In addition, at the Maintenance Association (MA) level, MA data node at the second level can be augmented with connectivity-context extension.

Note that MA level configuration parameters provides context information for management system to correlate faults, defects, network failures with location information, which helps quickly identify root causes of network failures.

7.1.2.1. Connectivity-Context Extension

In TRILL OAM, one example of connectivity-context is either a 12 bit VLAN ID or a 24 bit Fine Grain Label. The connection oriented base model defines a placeholder for context-id. This allows other technologies to easily augment that to include technology specific extensions. The snippet below depicts an example of augmenting connectivity-context to include either VLAN ID or Fine Grain Label.

```
augment /goam:domains/goam:domain/goam:MAs
   /goam:MA /goam:connectivity-context:
      +--:(connectivity-context-vlan)
         |  +--rw connectivity-context-vlan? vlan
      +--:(connectivity-context-fgl)
         +--rw connectivity-context-fgl? fgl
```

7.1.3. MEP Configuration Extension

The MEP configuration definition in the connection oriented base model already supports configuring the interface of MEP with either MAC address or IP address. In addition, the MEP address can be represented using a 2 octet RBridge Nickname in TRILL OAM. Hence, the TRILL OAM model augments the MEP configuration in base model to add a nickname case into the MEP address choice node as follows:

```
augment /goam:domains/goam:domain/goam:MAs
   /goam:MA/ goam:MEP/goam:mep-address:
      +--:( mep-address-trill)
         |  +--rw mep-address-trill? tril-rb-nickname
```
In addition, at the Maintenance Association Endpoint (MEP) level, MEP data node at the third level can be augmented with ECMP extension.

7.1.3.1. ECMP Extension

Since TRILL supports ECMP path selection, flow-entropy in TRILL is defined as a 96 octet field in the LIME model extension for TRILL OAM. The snippet below illustrates its extension.

```yang
augment /goam:domains/goam:domain/goam:MAs/goam:MA/goam:MEP:
    +--rw flow-entropy-trill?   flow-entropy-trill
augment /goam:domains/goam:domain/goam:MAs/goam:MA/goam:MEP/goam:session:
    +--rw flow-entropy-trill?   flow-entropy-trill
```

7.1.4. RPC extension

In the TRILL OAM YANG model, the continuity-check and path-discovery RPC commands are extended with TRILL specific requirements. The snippet below depicts an example of illustrates the TRILL OAM RPC extension.
augment /goam:continuity-check/goam:input:
  +++-ro (out-of-band)?
  |  +++-:(ipv4-address)
  |  |  +++-ro ipv4-address? inet:ipv4-address
  |  |  +++-:(ipv6-address)
  |  |  |  +++-ro ipv6-address? inet:ipv6-address
  |  |  +++-:(trill-nickname)
  |  |  |  +++-ro trill-nickname? tril-rb-nickname
  |  +--ro diagnostic-vlan? boolean
augment /goam:continuity-check/goam:output:
  +++-ro upstream-rbridge? tril-rb-nickname
  +++-ro next-hop-rbridge* tril-rb-nickname
augment /goam:path-discovery/goam:input:
  +++-ro (out-of-band)?
  |  +++-:(ipv4-address)
  |  |  +++-ro ipv4-address? inet:ipv4-address
  |  |  +++-:(ipv6-address)
  |  |  |  +++-ro ipv6-address? inet:ipv6-address
  |  |  +++-:(trill-nickname)
  |  |  |  +++-ro trill-nickname? tril-rb-nickname
  |  +--ro diagnostic-vlan? boolean
augment /goam:path-discovery/goam:output/goam:response:
  +++-ro upstream-rbridge? tril-rb-nickname
  +++-ro next-hop-rbridge* tril-rb-nickname

7.2. Generic YANG Model extension for MPLS-TP OAM

The MPLS-TP OAM YANG module can augment connection oriented OAM Module with some technology-specific details. And the [mpls-tp-oam-yang] presents the YANG Data model for MPLS-TP OAM.

The configuration extensions for connection oriented OAM include MD configuration extension, Technology type extension, Sub Technology Type Extension, MA configuration extension, MEP Configuration Extension.

7.2.1. MD Configuration Extension

MD level configuration parameters are management information which can be inherited in the MPLS-TP OAM model and set by LIME base model as default values. For example domain name can be set to area-ID or the provider’s Autonomous System Number (ASN) [RFC6370] in the MPLS-TP OAM case. In addition, at the Maintenance Domain level, domain data...
node at root level can be augmented with technology type and sub-technology type.

Note that MD level configuration parameters provides context information for management system to correlate faults, defects, network failures with location information, which helps quickly identify root causes of network failures.

7.2.1.1. Technology Type Extension

No MPLS-TP technology type has been defined in the connection oriented base model, hence it is required in the MPLS OAM model. The technology type "mpls-tp" is defined as an identity that augments the base "technology-types" defined in the connection oriented base model:

```yaml
identity mpls-tp{
    base goam:technology-types;
    description
    "mpls-tp type";
}
```

7.2.1.2. Sub Technology Type Extension

In MPLS-TP, since different encapsulation types such as IP/UDP Encapsulation, PW-ACH encapsulation can be employed, the "technology-sub-type" data node is defined and added into the MPLS OAM model to further identify the encapsulation types within the MPLS-TP OAM model. Based on it, we also define a technology sub-type for IP/UDP encapsulation and PW-ACH encapsulation. Other Encapsulation types can be defined in the same way. The snippet below depicts an example of several encapsulation types.
identity technology-sub-type {
    description
    "certain implementations can have different
    encapsulation types such as ip/udp, pw-ach and so on.
    Instead of defining separate models for each
    encapsulation, we define a technology sub-type to
    further identify different encapsulations.
    Technology sub-type is associated at the MA level";
}

    identity technology-sub-type-udp {
        base technology-sub-type;
        description
        "technology sub-type is IP/UDP encapsulation";
    }

    identity technology-sub-type-ach {
        base technology-sub-type;
        description
        "technology sub-type is PW-ACH encapsulation";
    }

    augment "/goam:domains/goam:domain/goam:MAAs/goam:MA" {
        leaf technology-sub-type {
            type identityref {
        base technology-sub-type;
            }
        }
    }

7.2.2. MA Configuration Extension

MA level configuration parameters are management information which
can be inherited in the MPLS-TP OAM model and set by Connection
Oriented base model as default values. One example of MA Name could
be MEG LSP ID or MEG Section ID or MEG PW ID[RFC6370].

Note that MA level configuration parameters provides context
information for management system to correlate faults, defects,
network failures with location information, which helps quickly
identify root causes of network failures.

7.2.3. MEP Configuration Extension

In MPLS-TP, MEP-ID is either a variable length label value in case of
G-ACH encapsulation or a 2 octet unsigned integer value in case of
IP/UDP encapsulation. One example of MEP-ID is MPLS-TP LSP_MEP_ID
[RFC6370]. In the connection-oriented base model, MEP-ID is defined
as a choice/case node which can supports an int32 value, and the same
definition can be used for MPLS-TP with no further modification. In
addition, at the Maintenance Association Endpoint (MEP) level, MEP
data node at the third level can be augmented with Session extension
and interface extension.

8. Security Considerations

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

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

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

/goam:domains/goam:domain/

/goam:domains/goam:domain/goam:MAs/goam:MA/


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

9. IANA Considerations

This document registers a URI in the IETF XML registry [RFC3688]
[RFC3688]. Following the format in RFC 3688, the following
registration is requested to be made:
This document registers a YANG module in the YANG Module Names registry [RFC6020].

name: ietf-gen-oam
prefix: goam
reference: RFC XXXX

10. Acknowledgments

Giles Heron came up with the idea of developing a YANG model as a way of creating a unified OAM API set (interface), work in this document is largely an inspiration of that. Alexander Clemm provided many valuable tips, comments and remarks that helped to refine the YANG model presented in this document.

Carlos Pignataro, David Ball, Mahesh Jethanandani, Benoit Claise, Ladislav Lhotka, GUBALLA JENS, Yuji Tochio, Gregory Mirsky, Huub van Helvoort, Tom Taylor, Dapeng Liu, Mishael Wexler, Adi Molkho participated and contributed to this document.

11. References

11.1. Normative References


11.2. Informative References


Appendix A. Contributors’ Addresses

Tissa Senevirathne
Consultant

Email: tsenevir@gmail.com

Norman Finn
CISCO Systems
510 McCarthy Blvd
Milpitas, CA 95035
USA

Email: nfinn@cisco.com

Samer Salam
CISCO Systems
595 Burrard St. Suite 2123
Vancouver, BC V7X 1J1
Canada

Email: ssalam@cisco.com
Authors’ Addresses

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

Email: dekumar@cisco.com

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

Email: bill.wu@huawei.com

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

Email: wangzitao@huawei.com
Abstract

This document presents a base YANG Data model for connectionless OAM protocols. It provides a technology-independent abstraction of key OAM constructs for connectionless protocols. Based model presented here can be extended to include technology specific details. This is leading to uniformity between OAM protocols and support nested OAM workflows (i.e., performing OAM functions at different or 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 November 27, 2016.

Copyright Notice

Copyright (c) 2016 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
1. Introduction

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

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

2. Troubleshoot failures (Fault verification and localization).

3. Monitor Performance

An overview of OAM tools is presented at [RFC7276].
Ping and Traceroute [RFC792], [RFC4443] are well-known fault verification and isolation tools, respectively, for IP networks. Over the years, different technologies have developed similar tools for similar purposes.

In this document, we present a base YANG Data model for connectionless OAM protocols which supports generic continuity or reachability check, and path discovery. The generic YANG model for connectionless OAM is designed such that it can be extended to cover various connectionless technologies. Technology dependent nodes and RPC (remote process call) commands are defined in technology specific YANG models, which use and extend the base model defined here.

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

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
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
  x for deprecated
  o for obsolete

<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 Model

At the top of the Model there is oper container for ipv4 and ipv6 session statistics. Grouping is also defined for common per session continuity-check statistics. This statistics is applicable for long-lived OAM sessions. Multiple test-point-location-list keyed using test point location type is augmented to network topology. Under test-point-location-list, there is one or more test-point-locations. Each test-point-location is associated with vrf, oam-layers, tools, and technology. The vrf is used to describe the corresponding network instance. The technology indicate oam technology details. The tools is used to describe oam tools supported. The oam-layers is used to indicate relation of test point with other test points. The level in oam-layers indicate whether related oam test point is client layer, server layer or same or stitched layer. The Model is augmented to /nd:networks/nd:network/nd:node. rpc for continuity-check and path discovery.

Under test-point-locations, there are tp-addresses, technology type, tools, and oam layer which describe the attributes associated with test-point.

3.1. TP address

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

- MAC address
- IPv4 or IPv6 address
- a pair of source, destination addresses, and interface (Useful for BFD)
- FEC
- TLV address (RFC6428 (Figure 4, 5, and 6))

3.2. Tools

In connectionless OAM, the tools attribute is used to describe a toolset for fault detection and isolation, and for performance measurement. And it can serve as a constraint condition when the base model be extended to specific OAM technology. For example, to fulfill the icmp ping configuration, the "./coam:tools-ip/coam:rfc792" should be set to "true", and then the base model should be augmented with icmp ping specific details.
3.3. Passive OAM

Passive OAM [1] defines mechanisms to record OAM information in the data packet itself while the packet traverses a particular network domain. Passive OAM defines a set of meta-data that is carried as part of the live traffic. The draft identifies several transport options for carrying passive OAM meta-data in packets and include IPv4, IPv6 as well as VXLAN-GPE, SR for IPv6 and NSH. Passive OAM is meant to complement active OAM mechanisms such as ping or trace route. Passive OAM mechanisms can be leveraged where current active OAM methods do not apply or do not offer the desired results, such as proving that a certain set of traffic takes a pre-defined path, SLA verification for the live data traffic, detailed statistics on traffic distribution paths in ECMP networks or scenarios where active OAM traffic is potentially handled differently from regular data traffic by the network devices. Passive OAM encompasses path trace, path packet counters and proof of transit sub-features and are further described below.

3.3.1. Path Trace

Path-trace feature is part of the passive OAM feature set, where meta-data is expected to be collected at every hop that a packet traverses, i.e. in a typical deployment all nodes in an passive OAM domain would participate in the domain and thus be transit nodes, encapsulating or decapsulating nodes. The network diameter of the domain is assumed to be known. The passive OAM path-trace decapsulating node removes the meta-data and process and/or export the meta-data. Some examples of the meta-data include node identification, incoming interface identification, outgoing interface identification, timestamp or generic application specific meta-data.

3.3.2. Path Packet Counters

Path-packet-counters feature is part of the passive OAM feature set, where the meta-data carried is to be interpreted by the passive OAM domain’s encapsulating and decapsulating nodes. Sequence numbers are the main meta-data added and can be used to detect packet loss, packet reordering or packet duplication.

3.3.3. Proof of Transit

Proof of transit feature is part of the passive OAM feature set, where the path or the service chain is verified. Service or path verification uses methods like nested hashing or nested encryption of the meta-data. By definition of a service chain or a path, some of the nodes in the domain participate and some do not.
3.4. OAM-layers

OAM-layers is referred to a list of OAM layers above and below that are related to current test point. This allows users to easily navigate up and down to efficiently troubleshoot a connectivity issue. In this model, we have kept level default as 0, as none connectionless network OAM uses layer or level in its mode. Level is provided for scenarios where it might be possible to define layering relationship as it can be used to tie up interworking of fault at related oam layers. For example, there is a defect in the upstream path of the testpoint, the upstream path belongs to server layer LSP and the level is set to "-1", the downstream path of the test point belongs to client layer LSP and the level is set to "1", then we can stitch server layer LSP and client layer LSP at different level and track defect along this path. In another case, if the upstream path and downstream path of the test point is in the same level, the level is set to "0". The snippet below depicts an example of OAM layers.

```yml
list oam-layers {
  key "index";
  leaf index {
    type uint16 {
      range "0..65535";
    }
  }
  leaf level {
    type int32 {
      range "-1..1";
      description "Level";
    }
    ordered-by user;
    description "list of related oam layers.";
  }
}
```

3.5. 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.

In this document, 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, while the path-

discovery rpc command corresponds to IP Traceroute.

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, lsp ping), the base building block
should be extended with corresponding technology specific parameters.

continuity-check:
  rpc continuity-check {
    if-feature continuity-check;
    description
    "Generates continuity-check as per RFC7276.";
    input {
      container destination-tp {
        uses tp-address;
      }
      uses session-type;
      leaf source-interface {
        type if:interface-ref;
        mandatory false;
      }
      leaf outbound-interface {
        type if:interface-ref;
        mandatory false;
      }
      leaf count {
        type uint32;
        default "5";
        mandatory false;
      }
      leaf vrf {
        type rt:routing-instance-ref;
        mandatory false;
      }
      leaf ttl {
        type uint8;
        default "255";
        mandatory false;
      }
      leaf packet-size {
        type uint32 {
          range "64..10000";
        }
        default "64";
        mandatory false;
        description
        "Size of ping echo request packets, in octets";
      }
    }
  }
output {
  list error-code-list {
    key "response-index";
    leaf response-index {
      type uint32;
    }
    leaf status-code {
      type int32;
      description "error code is ";
    }
    leaf status-sub-code {
      type uint8;
      mandatory false;
    }
  }
  leaf tx-packt-count {
    type oam-counter32;
    mandatory false;
    description "Transmitted Packet count";
  }
  leaf rx-packet-count {
    type oam-counter32;
    mandatory false;
    description "Received packet count";
  }
  leaf min-delay {
    type oam-counter32;
    mandatory false;
    units milliseconds;
    description "Delay is specified in milliseconds";
  }
  leaf average-delay {
    type oam-counter32;
    mandatory false;
    units millisecond;
    description "average delay in milliseconds";
  }
  leaf max-delay {
    type oam-counter32;
    mandatory false;
    units millisecond;
  }
}
description
  "Maximum delay in milliseconds";
}
}

Path discovery:
rpc path-discovery {
  description "";
  input {
    container destination-tp {
      uses tp-address;
    }
    uses session-type;
    leaf source-interface {
      type if:interface-ref;
      mandatory false;
    }
    leaf outbound-interface {
      type if:interface-ref;
      mandatory false;
    }
    leaf vrf {
      type rt:routing-instance-ref;
      mandatory false;
    }
    leaf max-ttl {
      type uint8;
      default "255";
      mandatory false;
    }
  }
  output {
    list response-list {
      key "response-index";
      leaf response-index {
        type uint32;
      }
      leaf status-code {
        type int32;
        description "error code is ";
      }
      leaf status-sub-code {
        type uint8;
        mandatory false;
      }
    }
  }
}
leaf hop-cnt {
    type uint8;
    description "";
}
container destination-tp {
    uses tp-address;
}
leaf min-delay {
    type oam-counter32;
    mandatory false;
    units millisecond;
    description "Delay is specified in milliseconds";
}
leaf average-delay {
    type oam-counter32;
    mandatory false;
    units millisecond;
    description "average delay in milliseconds";
}
leaf max-delay {
    type oam-counter32;
    mandatory false;
    units millisecond;
    description "Maximum delay in milliseconds";
}

Snippet of data hierarchy related to rpc calls

3.6. Relation with other OAM YANG Model

In this document we define a generic YANG model for connectionless OAM protocols. The YANG model defined here is generic such that other technologies can extend it for technology specific needs. The Generic YANG model acts as the root for other OAM YANG models. This allows users to traverse between different OAM protocols at ease through a uniform API set. The Generic YANG model for OAM provides a framework where technology-specific YANG models can choose to inherit constructs from the base YANG models without needing to redefine them within the sub-technology.
3.7. Notifications

In this document we define notifications for passive oam. They are passive oam proof of transit triggered notification, and passive oam path packet counters triggered notification.

3.8. Proof of transit notification

It’s generated with node information, ACL name associated with flow, and failure code from the device, and failed count, i.e. number of packet failed to offer proof of transit.

3.9. Path packet counter notification

It’s generated with node information, ACL name associated with flow, and failure code from the device, and failed drop count, failed reorder count failed out of sequence, and failed dup count.

3.10. OAM data hierarchy

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

module: ietf-connectionless-oam
  +-ro oper
    +-ro cc-ipv4-sessions-statistics {continuity-check}?
      +-ro cc-session-statistics
        |   +-ro session-count?          uint32
        |   +-ro session-up-count?       uint32
        |   +-ro session-down-count?     uint32
        |   +-ro session-admin-down-count? uint32
      +-ro cc-ipv6-sessions-statistics {continuity-check}?
        +-ro cc-session-statistics
          |   +-ro session-count?          uint32
          |   +-ro session-up-count?       uint32
          |   +-ro session-down-count?     uint32
          |   +-ro session-admin-down-count? uint32
      +-ro po-sessions-proof-of-transit-statistics {proof-of-transit}?
        +-ro po-per-session-proof-of-transit-statistics
          |   +-ro proved-count?           uint32
          |   +-ro failed-count?           uint32
        +-ro po-sessions-path-trace-statistics {path-trace}?
          +-ro session-delay-jitter-statistics
            |   +-ro timestamp-res?          timestamp-accuracy
            |   +-ro min-delay-value?        uint32
            |   +-ro max-delay-value?        uint32
            |   +-ro average-delay-value?    uint32
            |   +-ro min-jitter-value?       uint32
---ro max-jitter-value?    uint32
---ro average-jitter-value?    uint32
---ro po-sessions-packet-counter-statistics {path-packet-counters}?
   ---ro packet-drops-count?    uint32
   ---ro packet-reorder-count?  uint32
   ---ro packets-out-of-seq-count?  uint32
   ---ro packets-dup-count?    uint32
augment /nd:networks/nd:network/nd:node:
   ---rw test-point-ipv4-location-list {connection-less}?
      ---rw test-point-locations* [ipv4-location]
         ---rw ipv4-location    inet:ipv4-address
         ---rw vrf?    routing-instance-ref
         ---rw (tp-address)?
            ---:mac-address
            +--rw mac-address?    yang:mac-address
            ---:(ipv4-address)
            +--rw ipv4-address?  inet:ipv4-address
            ---:(ipv6-address)
            +--rw ipv6-address?  inet:ipv6-address
            ---:(src-dst-address)
            +--rw src-ip-address?  inet:ip-address
            +--rw dst-ip-address?  inet:ip-address
            +--rw Interface?    if:interface-ref
            ---:(fec)
            +--rw fec-type?    fec-type
            +--rw (fec-value)?
               ---:(ip-prefix)
                  +--rw ip-prefix?  inet:ip-prefix
               ---:(bgp)
                  +--rw bgp?  inet:ip-prefix
               ---:(tunnel)
                  +--rw tunnel-interface?  uint32
               ---:(l3vpn)
                  +--rw l3vpn-id?  uint32
               ---:(pw)
                  +--rw remote-pe-address?  inet:ip-address
                  +--rw pw-id?  uint32
               ---:(vpls)
                  +--rw route-distinguisher?  uint32
                  +--rw sender-ve-id?  uint32
                  +--rw receiver-ve-id?  uint32
               ---:(mpls-mldp)
                  +--rw (root-address)?
                     ---:(ip-address)
                  +--rw source-address?  inet:ip-address
                  +--rw group-ip-address?  IP-Multicast-Group-Addre
---(tlv-address)
  +--rw tlv-type?      int16
  +--rw tlv-len?       int16
  +--rw tlv-value?     binary

---(technology-null)
  +--rw tech-null?     empty

---(technology-string)
  +--rw ipv4-icmp?     string

---(tools-
  +--rw tools-empty
  |  +--rw tools-null?  empty
  |  +--:(tools-ip)
  |     +--rw rfc792?  boolean
  |     +--rw rfc4443? boolean
  |     +--rw rfc4884? boolean
  |     +--rw rfc5837? boolean
  |  +--:(tools-bfd)
  |     +--rw rfc5881? boolean
  |     +--rw rfc5883? boolean
  |     +--rw rfc5884? boolean
  |     +--rw rfc5885? boolean
  |  +--:(tools-mpls)
  |     +--rw rfc4379? boolean
  |     +--rw rfc4687? boolean
  |     +--rw rfc4950? boolean
  |     +--rw mpls-rfc5884? boolean
  |  +--:(tools-mpls-tp)
  |     +--rw rfc6426? boolean
  |     +--rw rfc6435? boolean
  |     +--rw rfc6374? boolean
  |  +--:(tools-pw)
  |     +--rw rfc5085? boolean
  |     +--rw pw_rfc5885? boolean
  |     +--rw rfc6423? boolean
  |     +--rw rfc6310? boolean
  |     +--rw rfc7023? boolean
  +--:(tools-passive-oam)
     +--rw passive-oam-config (path-packet-counters,path-trace,proof-o
         f-transit)?
        +--rw path-packet-counters-config
        |  +--rw flow-classifier* [name]
        |     +--rw name                      string
        |     +--rw access-list?             string
        |  +--rw data-export-profile* [name]
        |     +--rw name                     string
ta-export-type
  +--rw oam-data-export-type?
    da
  +--rw (data-export-types)?
    +--:(flexibleexport)
      +--rw flexible-export-params
        +--rw transport
          +--rw flex-data-export-protocol? export
        +--:(ipfix)
          +--rw ipfix-data-export-protocol? export-t
        +--transport-type
          +--rw flex-data-export-ip? inet:i
        +--p-address
          +--rw flex-data-export-port? inet:p
          +--rw encoding
            +--rw data-export-encoding-type? enumeration
          +--:(ipfix)
            +--rw ipfix-data-export-protocol? export-tp
          +--transp-t-type
            +--rw ipfix-data-export-ip? inet:ip-
            +--rw ipfix-data-export-port? inet:port
          +--rstamp-accuracy
            +--rw sampling-frequency? ti
            +--rw sampling-interval? ui
          +--nt64
            +--rw data-export-trace-time-res? ti
            +--rw data-export-trace-delay-threshold? ui
            +--rw data-export-trace-jitter-threshold? ui
            +--rw data-export-pot-failure-threshold? ui
            +--rw data-export-ppc-drops-failure-threshold? ui
            +--rw data-export-ppc-reorder-failure-threshold? ui
            +--rw data-export-ppc-oos-failure-threshold? ui
            +--rw data-export-ppc-dup-failure-threshold? ui
          +--nt32 {path-trace}?
            +--rw data-export-trace-time-res? ti
            +--rw data-export-trace-delay-threshold? ui
            +--rw data-export-trace-jitter-threshold? ui
            +--rw data-export-pot-failure-threshold? ui
            +--rw data-export-ppc-drops-failure-threshold? ui
            +--rw data-export-ppc-reorder-failure-threshold? ui
            +--rw data-export-ppc-oos-failure-threshold? ui
            +--rw data-export-ppc-dup-failure-threshold? ui
          +--nt32 {path-packet-counters}?
            +--rw transport-encap-profile* [name]
              +--rw name string
              +--rw transport-type? oam-transport-type-def
              +--rw node-id? uint32
              +--rw node-interfaces* [index]
                +--rw index uint32
                +--rw intf-name? if:interface-ref
              +--rw md-ppc-stats? boolean
              +--rw ppc-mode? passive-oa
          +--m-mode
            +--rw notification-ppc? boolean
            +--rw notification-ppc-drops-failure-threshold? uint32
            +--rw notification-ppc-reorder-failure-threshold? uint32
            +--rw notification-ppc-oos-failure-threshold? uint32
            +--rw notification-ppc-dup-failure-threshold? uint32
            +--rw proof-of-transit-config
              +--rw flow-classifier* [name]
                +--rw name string
                +--rw access-list? string
                +--rw data-export-profile* [name]
<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Internet-Draft       Connection-Less OAM YANG model             May 2016

|            |            |            |            |            |            | +--rw flexible-export-params
|            |            |            |            |            |            | +--rw transport
|            |            |            |            |            |            | +--rw flex-data-export-protocol?   export
|            |            |            |            |            |            | -transport-type
|            |            |            |            |            |            | +--rw flex-data-export-ip?   inet:i
|            |            |            |            |            |            | p-address
|            |            |            |            |            |            | +--rw flex-data-export-port?   inet:p
|            |            |            |            |            |            | ort-number
|            |            |            |            |            |            | +--rw encoding
|            |            |            |            |            |            | +--rw data-export-encoding-type?   enumer
|            |            |            |            |            |            | (ipfix)
|            |            |            |            |            |            | +--rw ip-fix-params
|            |            |            |            |            |            | +--rw ipfix-data-export-protocol?   export-type
|            |            |            |            |            |            | address
|            |            |            |            |            |            | +--rw ipfix-data-export-ip?   inet:ip-address
|            |            |            |            |            |            | transport-type
|            |            |            |            |            |            | +--rw ipfix-data-export-port?   inet:port-number
|            |            |            |            |            |            | t-number
|            |            |            |            |            |            | +--rw sampling-frequency?   ti
|            |            |            |            |            |            | mstamp-accuracy
|            |            |            |            |            |            | +--rw sampling-interval?   ui
|            |            |            |            |            |            | nt64
|            |            |            |            |            |            | +--rw data-export-trace-time-res?   ti
|            |            |            |            |            |            | mstamp-accuracy {path-trace}?
|            |            |            |            |            |            | +--rw data-export-trace-delay-threshold?   ui
|            |            |            |            |            |            | nt32 {path-trace}?
|            |            |            |            |            |            | +--rw data-export-trace-jitter-threshold?   ui
|            |            |            |            |            |            | nt32 {path-trace}?
|            |            |            |            |            |            | +--rw data-export-pot-failure-threshold?   ui
|            |            |            |            |            |            | nt32 {proof-of-transit}?
|            |            |            |            |            |            | +--rw data-export-ppc-drops-failure-threshold?   ui
|            |            |            |            |            |            | nt32 {path-packet-counters}?
|            |            |            |            |            |            | +--rw data-export-ppc-reorder-failure-threshold?   ui
|            |            |            |            |            |            | nt32 {path-packet-counters}?
|            |            |            |            |            |            | +--rw data-export-ppc-oos-failure-threshold?   ui
|            |            |            |            |            |            | nt32 {path-packet-counters}?
|            |            |            |            |            |            | +--rw data-export-ppc-dup-failure-threshold?   ui
|            |            |            |            |            |            | nt32 {path-packet-counters}?
|            |            |            |            |            |            | +--rw transport-encap-profile*   [name]
|            |            |            |            |            |            | +--rw name   string
|            |            |            |            |            |            | +--rw transport-type?   oam-transport-type-def
|            |            |            |            |            |            | +--rw node-id?   uint32
|            |            |            |            |            |            | +--rw node-interfaces*   [index]
|            |            |            |            |            |            | +--rw intf-name?   if:interface-ref
|            |            |            |            |            |            | md-proof-of-transit?
|            |            |            |            |            |            | +--rw pot-mode?   passive-oa
m-mode
|            |            |            |            |            |            | +--rw notification-pot?   boolean
|            |            |            |            |            |            | +--rw notification-pot-failure-threshold?   uint32
|            |            |            |            |            |            | path-trace-config
|            |            |            |            |            |            | +--rw flow-classifier*   [name]
|            |            |            |            |            |            | +--rw name   string
|            |            |            |            |            |            | +--rw access-list?   string
|            |            |            |            |            |            | data-export-profile*   [name]
|            |            |            |            |            |            | +--rw name   str
|            |            |            |            |            |            | oam-data-export-type?   da
ta-export-type
|            |            |            |            |            |            | +--rw (data-export-types)?
|            |            |            |            |            |            | +--:(flexibleexport)
<table>
<thead>
<tr>
<th>transport-type</th>
<th>flex-data-export-protocol?</th>
<th>export</th>
</tr>
</thead>
<tbody>
<tr>
<td>flex-data-export-ip?</td>
<td>inet:ip</td>
<td></td>
</tr>
<tr>
<td>flex-data-export-port?</td>
<td>inet:port</td>
<td></td>
</tr>
<tr>
<td>encoding</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
---rw data-export-encoding-type? enumer

---(ipfix)
  ---rw ip-fix-params
    ---rw ipfix-data-export-protocol? export-type

---rw ipfix-data-export-ip? inet:ip-address
---rw ipfix-data-export-port? inet:port-number

---rw sampling-frequency? time-stamp-accuracy
---rw sampling-interval? uint64

---rw data-export-trace-time-res? time-stamp-accuracy {path-trace}?

---rw data-export-trace-delay-threshold? uint32 {path-trace}?
---rw data-export-trace-jitter-threshold? uint32 {path-trace}?

---rw data-export-pot-failure-threshold? uint32 {proof-of-transit}?
---rw data-export-ppc-drops-failure-threshold? uint32 {path-packet-counters}?
---rw data-export-ppc-reorder-failure-threshold? uint32 {path-packet-counters}?
---rw data-export-ppc-oos-failure-threshold? uint32 {path-packet-counters}?
---rw data-export-ppc-dup-failure-threshold? uint32 {path-packet-counters}?

---rw transport-encap-profile* [name]
  ---rw name string
  ---rw transport-type? oam-transport-type-def
  ---rw node-id? uint32
  ---rw node-interfaces* [index]
    ---rw index uint32
    ---rw intf-name? if:interface-ref

---rw md-tracing?

---rw trace-mode passive-oa

---rw trace-length uint8
---rw trace-time-stamp timestamp-accuracy

---rw trace-app-data? uint32

---rw trace-include-interface boolean

---rw oam-layers* [index]
  ---rw index uint16
  ---rw level? int32

---(ip-prefix)
  ---rw mac-address? yang:mac-address
  ---(ipv4-address)
    ---rw ipv4-address? inet:ipv4-address
  ---(ipv6-address)
    ---rw ipv6-address? inet:ipv6-address
  ---(src-dst-address)
    ---rw src-ip-address? inet:ip-address
    ---rw dst-ip-address? inet:ip-address
    ---rw Interface? if:interface-ref
  ---(fec)
    ---rw fec-type? fec-type
    ---rw (fec-value)?
---rw ip-prefix?  inet:ip-prefix
augment /nd:networks/nd:network/nd:node:
  +--rw test-point-ipv6-location-list {connection-less}?
    +--rw ipv6-location inet:ipv6-address
    +--rw vrf? routing-instance-ref
  +--rw (tp-address)?
    +--:(mac-address)
      | +--rw mac-address? yang:mac-address
    +--:(ipv4-address)
      | +--rw ipv4-address? inet:ipv4-address
    +--:(ipv6-address)
      | +--rw ipv6-address? inet:ipv6-address
    +--:(src-dst-address)
      | +--rw src-ip-address? inet:ip-address
      | +--rw dst-ip-address? inet:ip-address
    +--:(mpls-mldp)
      | +--rw (root-address)?
        | +--:(ip-address)
          | +--rw source-address? inet:ip-address
          | +--rw group-ip-address? IP-Multicast-Group-Address
        +--:(vpls)
          | +--rw route-distinguisher? uint32
          | +--rw sender-ve-id? uint32
          | +--rw receiver-ve-id? uint32
        +--:(pw)
          | +--rw remote-pe-address? inet:ip-address
          | +--rw pw-id? uint32
        +--:(tunnel)
          | +--rw tunnel-interface? uint32
        +--:(l3vpn)
          | +--rw l3vpn-id? uint32
        +--:(bgp)
          | +--rw bgp? inet:ip-prefix
        +--:(mpls-mldp)
          | +--rw (root-address)?
            | +--:(ip-address)
              | +--rw source-address? inet:ip-address
              | +--rw group-ip-address? IP-Multicast-Group-Address
    +--:(vnl)
      | +--rw route-distinguisher? uint32
      | +--rw sender-ve-id? uint32
      | +--rw receiver-ve-id? uint32
  +--:(mpls-mldp)
    +--rw (root-address)?
      | +--:(ip-address)
        | +--rw source-address? inet:ip-address
        | +--rw group-ip-address? IP-Multicast-Group-Address
  +--:(l3vpn)
    +--rw l3vpn-id? uint32
  +--:(pw)
    +--rw remote-pe-address? inet:ip-address
    +--rw pw-id? uint32
  +--:(tunnel)
    +--rw tunnel-interface? uint32
  +--:(bgp)
    +--rw bgp? inet:ip-prefix
| +--rw bgp?        inet:ip-prefix
| | (tunnel)
| | +--rw tunnel-interface? uint32
| | (l3vpn)
| +--rw l3vpn-id? uint32
| (pw)
| +--rw remote-pe-address? inet:ip-address
| +--rw pw-id? uint32
| (vpls)
| +--rw sender-ve-id? uint32
+--rw receiver-ve-id? uint32
| (mpls-mldp)
+--rw (root-address)?
| +--rw source-address? inet:ip-address
| +--rw group-ip-address? IP-Multicast-Group-Address
| (ip-address)
| (vpn)
| +--rw as-number? inet:as-number
| (global-id)
+--rw lsp-id? string
| (tlv-address)
+--rw tlv-type? int16
+--rw tlv-len? int16
+--rw tlv-value? binary
| (technology)?
| +--rw (technology-null)
| | +--rw tech-null? empty
| +--rw (technology-string)
| | +--rw ipv4-icmp? string
| +--rw (tools)?
| +--rw (tools-empty)
+--rw tools-null? empty
| +--rw (tools-ip)
| | +--rw rfc792? boolean
| | +--rw rfc4443? boolean
| | +--rw rfc4884? boolean
| | +--rw rfc5837? boolean
| +--rw (tools-bfd)
| | +--rw rfc5881? boolean
| | +--rw rfc5883? boolean
| | +--rw rfc5884? boolean
| | +--rw rfc5885? boolean
| +--rw (tools-mpls)
| | +--rw rfc4379? boolean
| | +--rw rfc4687? boolean
| | +--rw rfc4950? boolean
+--rw mpls-rfc5884? boolean
---: (tools-mpls-tp)
  +--: (rfc)
      +--rw rfc6426? boolean
      +--rw rfc6435? boolean
      +--rw rfc6374? boolean

---: (tools-pw)
  +--: (rfc)
      +--rw rfc5085? boolean
      +--rw pw_rfc5885? boolean
      +--rw rfc6423? boolean
      +--rw rfc6310? boolean
      +--rw rfc7023? boolean

---: (tools-passive-oam)
  +--rw passive-oam-config {path-packet-counters,...}
      ---: (passive-oam-config)
      +--rw path-packet-counters-config
          +--rw flow-classifier* [name]
          |  +--rw name string
          |  +--rw access-list? string
          |  +--rw data-export-profile* [name]
          |     +--rw name string
          |     +--rw oam-data-export-type? data-export-type

---: (flexible-export)
  +--rw flexible-export-params
      +--rw transport
        +--rw flex-data-export-protocol? export-transport

---: (ipfix)
  +--rw ip-fix-params
      +--rw ipfix-data-export-protocol? export-transport
  +--rw ipfix-data-export-ip? inet:ip
  +--rw ipfix-data-export-port? inet:port

---: (sampling)
  +--rw sampling-frequency? timestamp-accuracy
  +--rw sampling-interval? uint64

---: (data-export-trace)
  +--rw data-export-trace-time-res? timestamp-accuracy
  +--rw data-export-trace-delay-threshold? uint32
  +--rw data-export-trace-jitter-threshold? uint32
  +--rw data-export-pot-failure-threshold? uint32
  +--rw data-export-ppc-drops-failure-threshold? uint32
  +--rw data-export-ppc-reorder-failure-threshold? uint32
  +--rw data-export-ppc-oos-failure-threshold? uint32
  +--rw data-export-ppc-dup-failure-threshold? uint32
---rw transport-encap-profile* [name]
  ---rw name string
  ---rw transport-type? oam-transport-type-def
  ---rw node-id? uint32
  ---rw node-interfaces* [index]
++rw index          uint32
++rw intf-name?     if:interface-ref
++rw md-ppc-stats?   boolean
++rw ppc-mode?      passive-oa

m-mode
++rw notification-ppc?  boolean
++rw notification-ppc-drops-failure-threshold?  uint32
++rw notification-ppc-reorder-failure-threshold?  uint32
++rw notification-ppc-oos-failure-threshold?  uint32
++rw notification-ppc-dup-failure-threshold?  uint32
++rw proof-of-transit-config
  ++rw flow-classifier* [name]
    ++rw name            string
    ++rw access-list?    string
  ++rw data-export-profile* [name]
    ++rw name            string

ring
  ++rw oam-data-export-type?  data

ta-export-type
  ++rw (data-export-types)?
    +=:(flexibleexport)
      ++rw flexible-export-params
      ++rw transport
        ++rw flex-data-export-protocol?  export

-transport-type
  ++rw flex-data-export-ip?  inet:ip-address
  ++rw flex-data-export-port?  inet:port-number

transport-type
  ++rw ipfix-data-export-ip?  inet:ip-address
  ++rw ipfix-data-export-port?  inet:port-number

address
  ++rw sampling-frequency?  timestamp-accuracy
  ++rw sampling-interval?   uint64

nt64
  ++rw data-export-trace-time-res?  timestamp-accuracy

mstamp-accuracy {path-trace}?
  ++rw data-export-trace-delay-threshold?  uint32
  ++rw data-export-trace-jitter-threshold?  uint32

nt32 {path-trace}?
  ++rw data-export-pot-failure-threshold?  uint32

nt32 {proof-of-transit}?
  ++rw data-export-ppc-drops-failure-threshold?  uint32

nt32 {path-packet-counters}?
  ++rw data-export-ppc-reorder-failure-threshold?  uint32

nt32 {path-packet-counters}?
  ++rw data-export-ppc-oos-failure-threshold?  uint32

nt32 {path-packet-counters}?
  ++rw data-export-ppc-dup-failure-threshold?  uint32

nt32 {path-packet-counters}?
  ++rw transport-encap-profile* [name]
    ++rw name            string
    ++rw transport-type?  oam-transport-type-def
m-mode
  | ++--rw pot-mode? passive-oa
  | ++--rw notification-pot? boolean
  | ++--rw notification-pot-failure-threshold? uint32
  | ++--rw path-trace-config
    | | ++--rw flow-classifier* [name]
    | |   | ++--rw name string
    | |   | ++--rw access-list? string
    | |   | ++--rw data-export-profile* [name]
    | |   | | ++--rw name string
    | |   | | ++--rw oam-data-export-type? data
    | | ++--rw (data-export-types)?
    | |   | ++--:(flexibleexport)
    | |   | | ++-- rw flexible-export-params
    | |   | | | ++-- rw transport
    | |   | | |   | ++-- rw flex-data-export-protocol? export
    | |   | | |   | | ++-- rw flex-data-export-ip? inet:ip-address
    | |   | | |   | | | ++-- rw flex-data-export-port? inet:port-number
    | |   | | |   | | ++-- rw encoding
    | |   | | |   | | | ++-- rw data-export-encoding-type? enumeration
    | |   | | |   | | | | ++--:(ipfix)
    | |   | | |   | | | | | ++-- rw ip-fix-params
    | |   | | |   | | | | | | ++-- rw ipfix-data-export-protocol? export-type
    | |   | | |   | | | | | | | ++-- rw ipfix-data-export-ip? inet:ip-address
    | |   | | |   | | | | | | | | ++-- rw ipfix-data-export-port? inet:port-number
    | |   | | |   | | | | | | | | | ++-- rw sampling-frequency? timestamp-accuracy
    | |   | | |   | | | | | | | | | | ++-- rw sampling-interval? uint64
    | |   | | |   | | | | | | | | | | | ++-- rw data-export-trace-time-res? timestamp-accuracy {path-trace}?
    | |   | | |   | | | | | | | | | | | | ++-- rw data-export-trace-delay-threshold? uint32 {path-trace}?
    | |   | | |   | | | | | | | | | | | | | ++-- rw data-export-trace-jitter-threshold? uint32 {path-trace}?
    | |   | | |   | | | | | | | | | | | | | | ++-- rw data-export-pot-failure-threshold? uint32 {proof-of-transit}?
    | |   | | |   | | | | | | | | | | | | | | | ++-- rw data-export-ppc-drops-failure-threshold? uint32 {path-packet-counters}?
    | |   | | |   | | | | | | | | | | | | | | | | ++-- rw data-export-ppc-reorder-failure-threshold? uint32 {path-packet-counters}?
    | |   | | |   | | | | | | | | | | | | | | | | | ++-- rw data-export-ppc-oos-failure-threshold? uint32 {path-packet-counters}?
    | |   | | |   | | | | | | | | | | | | | | | | | | ++-- rw data-export-ppc-dup-failure-threshold? uint32 {path-packet-counters}?
    | |   | | |   | | | | | | | | | | | | | | | | | | | | ++-- rw transport-encap-profile* [name]
    | |   | | |   | | | | | | | | | | | | | | | | | | | | | | ++-- rw name string
    | |   | | |   | | | | | | | | | | | | | | | | | | | | | | | | ++-- rw transport-type? oam-transport-type-def
    | |   | | |   | | | | | | | | | | | | | | | | | | | | | | | | | | ++-- rw node-id? uint32
    | |   | | |   | | | | | | | | | | | | | | | | | | | | | | | | | | | ++-- rw node-interfaces* [index]
    | |   | | |   | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | ++-- rw index uint32
    | |   | | |   | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | ++-- rw intf-name? if:interface-ref
    | |   | | |   | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | ++-- rw md-tracing? boolean
    | |   | | |   | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | ++-- rw trace-mode passive-oa
m-mode
|        +--rw trace-length (uint8)
|        +--rw trace-time-stamp (timestamp)
accuracy
|        +--rw trace-app-data? (uint32)
|        +--rw trace-include-interface (boolean)
|        +--rw oam-layers* [index]
++-rw index                   uint16
++-rw level?                  int32
++-rw (tp-address)?
    |   +++-(mac-address)
    |   |   +++-rw mac-address?           yang:mac-address
    |   ++-:(ipv4-address)
    |   |   +++-rw ipv4-address?           inet:ipv4-address
    |   ++-:(ipv6-address)
    |   |   +++-rw ipv6-address?           inet:ipv6-address
    |   ++-:(src-dst-address)
    |   |   +++-rw src-ip-address?         inet:ip-address
    |   |   +++-rw dst-ip-address?         inet:ip-address
    |   |   +++-rw Interface?              if:interface-ref
    |   ++-:(fec)
    |   |   +++-rw fec-type?              fec-type
    |   |   +++-rw (fec-value)?
    |       |   +++-(ip-prefix)
    |       |       |   +++-rw ip-prefix?           inet:ip-prefix
    |       |   +++-:(bgp)
    |       |       |   +++-rw bgp?                   inet:ip-prefix
    |       |   +++-:(tunnel)
    |       |       |   +++-rw tunnel-interface?      uint32
    |       |   +++-:(l3vpn)
    |       |       |   +++-rw l3vpn-id?              uint32
    |       |   +++-:(pw)
    |       |       |   +++-rw remote-pe-address?      inet:ip-address
    |       |       |   +++-rw pw-id?                 uint32
    |       |   +++-:(vpls)
    |       |       |   +++-rw route-distinguisher?     uint32
    |       |       |   +++-rw sender-ve-id?          uint32
    |       |       |   +++-rw receiver-ve-id?        uint32
    |       |   +++-:(mpls-mldp)
    |       |       |   +++-rw (root-address)?
    |       |       |       |   +++-(ip-address)
    |       |       |       |       |   +++-rw source-address?        inet:ip-address
    |       |       |       |       |   +++-rw group-ip-address?      IP-Multicast-Group-Address
    |       |   |       |   +++-(vpn)
    |       |       |       |       |   +++-rw as-number?             inet:as-number
    |       |       |       |   |   +++-(global-id)
    |       |       |       |       |       |   +++-rw lsp-id?                string
    |       |       |   |   +++-(tlv-address)
    |       |       |       |       |   +++-rw tlv-type?             int16
    |       |       |       |       |   +++-rw tlv-len?               int16
    |       |       |       |       |   +++-rw tlv-value?             binary

augment /nd:networks/nd:network/nd:node:
    +++-rw test-point-po-node-list {path-trace,path-packet-counters,proof-of-transit}?
        +++-rw test-point-locations* [node-id]
            +++-rw node-id           uint32
++--rw node-interfaces* [index]
   ++--rw index         uint32
   ++--rw intf-name?    if:interface-ref
++--rw vrf?            routing-instance-ref
++--rw (tp-address)?
   |  +--rw mac-address?       yang:mac-address
   |  ++--:(ipv4-address)      inet:ipv4-address
   |  ++--:(ipv6-address)      inet:ipv6-address
   |  ++--:(src-dst-address)
     |  ++--rw src-ip-address?   inet:ip-address
     |  ++--rw dst-ip-address?   inet:ip-address
   ++--rw Interface?         if:interface-ref
   |  ++--:(fec)
     |  ++--rw fec-type?            fec-type
++--rw (fec-value)?
   |  ++--:(ip-prefix)
     |     |  ++--rw ip-prefix?        inet:ip-prefix
     |  ++--:(bgp)
     |     |  ++--rw bgp?             inet:ip-prefix
     |  ++--:(tunnel)
     |     |  ++--rw tunnel-interface? uint32
     |     ++--:(l3vpn)
     |          |  ++--rw l3vpn-id?          uint32
     |     ++--:(pw)
     |          |  ++--rw remote-pe-address? inet:ip-address
     |          |  ++--rw pw-id?             uint32
     |     ++--:(vpls)
     |          |  ++--rw route-distinguisher? uint32
     |          |  ++--rw sender-ve-id?      uint32
     |          |  ++--rw receiver-ve-id?    uint32
     |     ++--:(mpls-mldp)
     |          |  ++--rw (root-address)?
     |          |     |  ++--:(ip-address)
     |          |     |     |  ++--rw source-address?    inet:ip-address
     |          |     |     ++--rw group-ip-address? IP-Multicast-Group-Addre
     |          |     |  ++--:(vpn)
     |          |     |     |  ++--rw as-number?        inet:as-number
     |          |     |     ++--:(global-id)
     |          |     |          |  ++--rw lsp-id?          string
     |          |  ++--:(tlv-address)
     |          |     |  ++--rw tlv-type?         int16
     |          |     |  ++--rw tlv-len?          int16
     |          |     |  ++--rw tlv-value?        binary
++--rw (technology)?
   |  ++--:(technology-null)
| +--rw tech-null? | empty |
| +--:(technology-string) |
| +--rw ipv4-icmp? | string |
| +--rw (tools)? |
| +--:(tools-empty) |
| +--rw tools-null? | empty |
| +--:(tools-ip) |
| +--rw rfc792? | boolean |
| +--rw rfc4443? | boolean |
| +--rw rfc4884? | boolean |
| +--rw rfc5837? | boolean |
| +--:(tools-bfd) |
| +--rw rfc5881? | boolean |
| +--rw rfc5883? | boolean |
| +--rw rfc5884? | boolean |
| +--rw rfc5885? | boolean |
| +--:(tools-mpls) |
| +--rw rfc4379? | boolean |
| +--rw rfc4687? | boolean |
| +--rw rfc4950? | boolean |
| +--rw mpls-rfc5884? | boolean |
| +--:(tools-mpls-tp) |
| +--rw rfc6426? | boolean |
| +--rw rfc6435? | boolean |
| +--rw rfc6374? | boolean |
| +--:(tools-pw) |
| +--rw rfc5085? | boolean |
| +--rw pw_rfc5885? | boolean |
| +--rw rfc6423? | boolean |
| +--rw rfc6310? | boolean |
| +--rw rfc7023? | boolean |
| +--:(tools-passive-oam) |
| +--rw passive-oam-config |
| | +--rw path-packet-counters-config |
| | | +--rw flow-classifier* [name] |
| | | | +--rw name | string |
| | | | +--rw access-list? | string |
| | | | +--rw data-export-profile* [name] |
| | | | | +--rw name | string |
| | | +--rw oam-data-export-type? | data-export-type |
| | +--rw (data-export-types)? |
| | | +--:(flexibleexport) |
| | | | +--rw flexible-export-params |
| | | | | +--rw transport |
| | | | | | +--rw flex-data-export-protocol? | export |
| | | +--rw flex-data-export-ip? | inet:ipv4 |
| | +--rw flex-data-export-port? | inet:port |
+++rw data-export-encoding-type? enumer

++-:(ipfix)
  ++-rw ip-fix-params
    ++-rw ipfix-data-export-protocol? export-transport-type

++-rw ipfix-data-export-ip? inet:ip-address
++-rw ipfix-data-export-port? inet:port-number

++-rw sampling-frequency? timestamp-accuracy
++-rw sampling-interval? uint64

++-rw data-export-trace-time-res? timestamp-accuracy {path-trace}?
++-rw data-export-trace-delay-threshold? uint32 {path-trace}?
++-rw data-export-trace-jitter-threshold? uint32 {path-trace}?
++-rw data-export-pot-failure-threshold? uint32 {proof-of-transit}?
++-rw data-export-ppc-drops-failure-threshold? uint32 {path-packet-counters}?
++-rw data-export-ppc-reorder-failure-threshold? uint32 {path-packet-counters}?
++-rw data-export-ppc-oos-failure-threshold? uint32 {path-packet-counters}?
++-rw data-export-ppc-dup-failure-threshold? uint32 {path-packet-counters}?

++-rw transport-encap-profile* [name]
  ++-rw name string
  ++-rw transport-type? oam-transport-type-def
  ++-rw node-id? uint32
  ++-rw node-interfaces* [index]
    ++-rw index uint32
    ++-rw intf-name? if:interface-ref

++-rw md-ppc-stats? boolean
++-rw ppc-mode? passive-oam

++-rw notification-ppc? boolean
++-rw notification-ppc-drops-failure-threshold? uint32
++-rw notification-ppc-reorder-failure-threshold? uint32
++-rw notification-ppc-oos-failure-threshold? uint32
++-rw notification-ppc-dup-failure-threshold? uint32
++-rw proof-of-transit-config
  ++-rw flow-classifier* [name]
    ++-rw name string
    ++-rw access-list? string
  ++-rw data-export-profile* [name]
    ++-rw name string

++-rw oam-data-export-type? data-export-type

++-rw (data-export-types)?
  ++-:(flexibleexport)
    ++-rw flexible-export-params
      ++-rw transport
        ++-rw flex-data-export-protocol? export-transport-type

++-rw flex-data-export-ip? inet:ip-address
++-rw flex-data-export-port? inet:port-number
ort-number

|                      |                      |                      |                      | +--rw encoding
|                      |                      |                      |                      |   +--rw data-export-encoding-type? enumer
|                      |                      |                      |                      |   +--:(ipfix)
|                      |                      |                      |                      |     +--rw ip-fix-params

Kumar, et al.          Expires November 27, 2016                     [Page 26]
++rw transport-type
++rw ipfix-data-export-protocol? export-type

++rw ipfix-data-export-ip? inet:ip-address

++rw ipfix-data-export-port? inet:port-number

++rw sampling-frequency? ti

++rw sampling-interval? ui

++rw data-export-trace-time-res? ti

++rw data-export-trace-delay-threshold? ui

++rw data-export-trace-jitter-threshold? ui

++rw data-export-pot-failure-threshold? ui

++rw data-export-ppc-drops-failure-threshold? ui

++rw data-export-ppc-reorder-failure-threshold? ui

++rw data-export-ppc-oos-failure-threshold? ui

++rw data-export-ppc-dup-failure-threshold? ui

++rw transport-encap-profile* [name]
  ++rw name string
  ++rw transport-type? oam-transport-type-def
  ++rw node-id? uint32
  ++rw node-interfaces* [index]
  ++rw index uint32
  ++rw intf-name? if:interface-ref

++rw md-proof-of-transit?
++rw pot-mode? passive-oa

++rw notification-pot? boolean
++rw notification-pot-failure-threshold? uint32

++rw path-trace-config
++rw flow-classifier* [name]
  ++rw name string
  ++rw access-list? string
  ++rw data-export-profile* [name]
    ++rw name string

++rw oam-data-export-type? data-export-type

++rw (data-export-types)?
  ++: (flexibleexport)
    ++rw flexible-export-params
      ++rw transport
        ++rw flex-data-export-protocol? export

++rw flex-data-export-ip? inet:ip-address

++rw flex-data-export-port? inet:port-number

++rw encoding
  ++rw data-export-encoding-type? enumeration

++rw ip-fix-params
  ++rw ipfix-data-export-protocol? export-type
| transport-type | | | | | rw ipfix-data-export-ip? | inet:ip-address
| t-number | | | | | rw ipfix-data-export-port? | inet:port
| mstamp-accuracy | | | | | rw sampling-frequency? | ti
| nt64 | | | | | rw sampling-interval? | ui
| mstamp-accuracy {path-trace} | | | | | rw data-export-trace-time-res? | ti

Kumar, et al. Expires November 27, 2016 [Page 27]
++rw transport-encap-profile* [name]  
  +--rw name              string
  +--rw transport-type?   oam-transport-type-def
     +--rw node-id?       uint32
     +--rw node-interfaces* [index]  
       +--rw index        uint32
       +--rw intf-name?   if:interface-ref

++rw md-tracing?  boolean
++rw trace-mode    passive-oa
m-mode
  +--rw trace-length  uint8
  +--rw trace-time-stamp  timestamp-accuracy
accuracy
  +--rw trace-app-data?  uint32
  +--rw trace-include-interface  boolean
++rw oam-layers* [index]  
  +--rw index                 uint16
  +--rw level?                 int32
  +--rw (tp-address)?
     +--:(mac-address)
        +--rw mac-address?   yang:mac-address
     +--:(ipv4-address)
        +--rw ipv4-address?  inet:ipv4-address
     +--:(ipv6-address)
        +--rwipv6-address?  inet:ipv6-address
     +--:(src-dst-address)
        +--rw src-ip-address?  inet:ip-address
        +--rw dst-ip-address?  inet:ip-address
        +--rw Interface?      if:interface-ref
     +--:(fec)
        +--rw fec-type?     fec-type
        +--rw (fec-value)?
           +--:(ip-prefix)
              +--rw ip-prefix?  inet:ip-prefix
           +--:(bgp)
              +--rw bgp?   inet:ip-prefix
           +--:(tunnel)
              +--rw tunnel-interface?  uint32
           +--:(l3vpn)
              +--rw l3vpn-id?  uint32
           +--:(pw)
              +--rw remote-pe-address?  inet:ip-address
              +--rw pw-id?  uint32

```plaintext
augment /nd:networks/nd:network/nd:node:
  - rw test-point-tunnel-address-location-list {connection-less}?
  - rw test-point-locations* [tunnel-location]
    - rw tunnel-location uint32
  - rw vrf? routing-instance-ref
  - rw (tp-address)?
    - rw mac-address? yang:mac-address
    - (ipv4-address)
    - (ipv6-address)
    - (src-dst-address)
      - rw src-ip-address? inet:ip-address
      - rw dst-ip-address? inet:ip-address
      - rw Interface? if:interface-ref
    - rw (fec)
      - rw fec-type? fec-type
      - rw (fec-value)?
        - rw (ip-prefix)
        - rw bgp?
        - rw (tunnel)
          - rw tunnel-interface? uint32
          - rw (l3vpn)
            - rw l3vpn-id? uint32
            - rw (pw)
              - rw remote-pe-address? inet:ip-address
              - rw pw-id? uint32
```
+--rw route-distinguisher?  uint32
+--rw sender-ve-id?         uint32
+--rw receiver-ve-id?       uint32
++--:(mpls-mldp)
   +--rw (root-address)?
      ++--:(ip-address)
         |   +--rw source-address?     inet:ip-address
         |   +--rw group-ip-address?   IP-Multicast-Group-Address
      ++--:(vpn)
         |   +--rw as-number?          inet:as-number
      ++--:(global-id)
         |   +--rw lsp-id?             string
++--:(tlv-address)
   +--rw tlv-type?            int16
   +--rw tlv-len?             int16
   +--rw tlv-value?           binary
++--rw (technology)?
   ++--:(technology-null)
      |   +--rw tech-null?         empty
   ++--:(technology-string)
      +--rw ipv4-icmp?           string
++--rw (tools)?
   ++--:(tools-empty)
      |   +--rw tools-null?        empty
   ++--:(tools-ip)
      |   +--rw rfc792?            boolean
      |   +--rw rfc4443?           boolean
      |   +--rw rfc4884?           boolean
      |   +--rw rfc5837?           boolean
   ++--:(tools-bfd)
      |   +--rw rfc5881?           boolean
      |   +--rw rfc5883?           boolean
      |   +--rw rfc5884?           boolean
      |   +--rw rfc5885?           boolean
   ++--:(tools-mpls)
      |   +--rw rfc4379?           boolean
      |   +--rw rfc4687?           boolean
      |   +--rw rfc4950?           boolean
      |   +--rw mpls-rfc5884?      boolean
   ++--:(tools-mpls-tp)
      |   +--rw rfc6426?           boolean
      |   +--rw rfc6435?           boolean
      |   +--rw rfc6374?           boolean
   ++--:(tools-pw)
      |   +--rw rfc5085?           boolean
      |   +--rw pw_rfc5085?        boolean
      |   +--rw rfc6423?           boolean
      |   +--rw rfc6310?           boolean
++rw rfc7023? boolean
+++:(tools-passive-oam)
++rw passive-oam-config {path-packet-counters,path-trace,proof-of-transit}?
  ++rw path-packet-counters-config
    ++rw flow-classifier* [name]
      ++rw name string
      ++rw access-list? string
    ++rw data-export-profile* [name]
      ++rw name string
  ++rw oam-data-export-type?
    ++rw data-export-encoding-type? enumeration
++rw transport-type
  ++rw (flexible-exports)
    ++rw transport
      ++rw flex-data-export-protocol? export
  ++rw transport-encap-profile* [name]
    ++rw name string
    ++rw transport-type? oam-transport-type-definition
    ++rw node-id? uint32
    ++rw node-interfaces* [index]
      ++rw index uint32
      ++rw intf-name? if:interface-ref
++rw md-ppc-stats? boolean
++rw ppc-mode? passive-oa
---rw proof-of-transit-config
  ---rw flow-classifier* [name]
    ---rw name               string
    ---rw access-list?       string
    ---rw data-export-profile* [name]
      ---rw name             string
  |---rw oam-data-export-type?  da
  |---rw (data-export-types)?
  |   ---:(flexibleexport)
  |     ---rw flexible-export-params
  |       ---rw transport
  |          ---rw flex-data-export-protocol?  export
  |          ---rw flex-data-export-ip?  inet:ip-address
  |          ---rw flex-data-export-port?  inet:port-number
  |       +--rw encoding
  |          ---rw data-export-encoding-type?  enumeration
  |          ---:(ipfix)
  |             ---rw ip-fix-params
  |             ---rw ipfix-data-export-protocol?  export-type
  |             ---rw ipfix-data-export-ip?  inet:ip-address
  |             ---rw ipfix-data-export-port?  inet:port-number
  |       +--rw sampling-frequency?  timestamp-accuracy
  |          ---rw sampling-interval?  uint64
  |          ---rw data-export-trace-time-res?  timestamp-accuracy {path-trace}?
  |          ---rw data-export-trace-delay-threshold?  uint32 {path-trace}?
  |          ---rw data-export-trace-jitter-threshold?  uint32 {path-trace}?
  |          ---rw data-export-pot-failure-threshold?  uint32 {proof-of-transit}?
  |          ---rw data-export-ppc-drops-failure-threshold?  uint32 {path-packet-counters}?
  |          ---rw data-export-ppc-reorder-failure-threshold?  uint32 {path-packet-counters}?
  |          ---rw data-export-ppc-oos-failure-threshold?  uint32 {path-packet-counters}?
  |          ---rw data-export-ppc-dup-failure-threshold?  uint32 {path-packet-counters}?
  |       +--rw node-interfaces* [index]
  |         +--rw index        uint32
  |         +--rw intf-name?   if:interface-ref
  |       +--rw md-proof-of-transit?  boolean
  |          ---rw transport-type?  oam-transport-type-definition
  |          ---rw node-id?        uint32
  |          ---rw node-interfaces* [index]
  |             +--rw index        uint32
  |             +--rw intf-name?   if:interface-ref
  |       +--rw notification-pot?  boolean
  |          ---rw notification-pot-failure-threshold?  uint32
  |       +--rw path-trace-config
ta-export-type
  +--rw oam-data-export-type?  da
  +--rw (data-export-types)?
    +--:(flexibleexport)
      +--rw flexible-export-params
        +--rw transport
          +--rw flex-data-export-protocol?  export
  +--transport-type
    +--rw flex-data-export-ip?  inet:i
    +--rw flex-data-export-port?  inet:p
  +--p-address
    +--rw encoding
      +--rw data-export-encoding-type?  enumeration
      +--:(ipfix)
        +--rw ip-fix-params
          +--rw ipfix-data-export-protocol?  export-type
          +--rw ipfix-data-export-ip?  inet:ip
    +--t-number
      +--rw sampling-frequency?  ti
      +--rw sampling-interval?  ui
    +--nt64
      +--rw data-export-trace-time-res?  ti
      +--rw data-export-trace-delay-threshold?  ui
      +--rw data-export-trace-jitter-threshold?  ui
      +--rw data-export-pot-failure-threshold?  ui
      +--rw data-export-ppc-drops-failure-threshold?  ui
      +--rw data-export-ppc-reorder-failure-threshold?  ui
      +--rw data-export-ppc-oos-failure-threshold?  ui
      +--rw data-export-ppc-dup-failure-threshold?  ui
    +--nt32 {path-trace}?
      +--rw data-export-trace-time-stamp  timestamp
      +--rw data-export-trace-app-data?  ui
      +--rw data-export-trace-include-interface  boolean
      +--rw transport-encap-profile* [name]
        +--rw name  string
        +--rw transport-type?  oam-transport-type-def
      +--rw node-id?  uint32
      +--rw node-interfaces* [index]
        +--rw index  uint32
        +--rw intf-name?  if:interface-ref
    +--rw md-tracing?  boolean
    +--rw trace-mode  passive-oas
    +--rw trace-length  uint8
    +--rw trace-time-stamp  timestamp
    +--rw trace-app-data?  ui
    +--rw trace-include-interface  boolean
  +--nt32 {path-packet-counters}?
    +--rw transport-encap-profile* [name]
    +--rw name  string
    +--rw transport-type?  oam-transport-type-def
    +--rw node-id?  uint32
    +--rw node-interfaces* [index]
      +--rw index  uint32
      +--rw intf-name?  if:interface-ref
    +--rw md-tracing?  boolean
    +--rw trace-mode  passive-oas
    +--rw trace-length  uint8
    +--rw trace-time-stamp  timestamp
    +--rw trace-app-data?  ui
    +--rw trace-include-interface  boolean
    +--rw oam-layers* [index]
      +--rw index  uint16
      +--rw level?  int32
      +--rw (tp-address)?
        +--:(mac-address)
| +--rw mac-address?                        | yang:mac-address |
| +--:(ipv4-address)                          |
| | +--rw ipv4-address?                      | inet:ipv4-address |
| | +--:(ipv6-address)                          |
| | | +--rw ipv6-address?                      | inet:ipv6-address |

Kumar, et al. Expires November 27, 2016 [Page 33]
++-:(src-dst-address)
  ++-rw src-ip-address?      inet:ip-address
  ++-rw dst-ip-address?      inet:ip-address
  ++-rw Interface?           if:interface-ref
++-:(fec)
  ++-rw fec-type?            fec-type
  ++-rw (fec-value)?
    ++-:(ip-prefix)
      |  ++-rw ip-prefix?        inet:ip-prefix
    ++-:(bgp)
      |  ++-rw bgp?             inet:ip-prefix
    ++-:(tunnel)
      |  ++-rw tunnel-interface? uint32
    ++-:(l3vpn)
      |  ++-rw l3vpn-id?        uint32
    ++-:(pw)
      |  ++-rw remote-pe-address? inet:ip-address
      |  ++-rw pw-id?           uint32
    ++-:(vpls)
      |  ++-rw route-distinguisher? uint32
      |  ++-rw sender-ve-id?     uint32
      |  ++-rw receiver-ve-id?   uint32
    ++-:(mpls-mldp)
      |  ++-rw (root-address)?
        |   ++-:(ip-address)
        |       |  ++-rw source-address?    inet:ip-address
        |       |  ++-rw group-ip-address?  IP-Multicast-Group-Address
        |   ++-:(ip-address)
        |       |  ++-rw as-number?         inet:as-number
        |   ++-:(global-id)
        |       |  ++-rw lsp-id?            string
      |   ++-:(tlv-address)
      |     |  ++-rw tlv-type?         int16
      |     |  ++-rw tlv-len?          int16
      |     |  ++-rw tlv-value?        binary
augment /nd:networks/nd:network/nd:node:
  ++-rw test-point-mac-address-location-list {connection-less}?
  ++-rw test-point-locations* [mac-address-location]
    ++-rw mac-address-location       yang:mac-address
    ++-rw vrf?                       routing-instance-ref
    ++-rw (tp-address)?
      |   ++-:(mac-address)
      |      |  ++-rw mac-address?      yang:mac-address
      |   ++-:(ipv4-address)
      |      |  ++-rw ipv4-address?     inet:ipv4-address
      |   ++-:(ipv6-address)
      |      |  ++-rw ipv6-address?     inet:ipv6-address
  ++-:(src-dst-address)
---rw rfc5881?  boolean
---rw rfc5883?  boolean
---rw rfc5884?  boolean
---rw rfc5885?  boolean
+-:(tools-mpls)
  ---rw rfc4379?  boolean
  ---rw rfc4687?  boolean
  ---rw rfc4950?  boolean
  ---rw mpls-rfc5884?  boolean
+-:(tools-mpls-tp)
  ---rw rfc6426?  boolean
  ---rw rfc6435?  boolean
  ---rw rfc6374?  boolean
+-:(tools-pw)
  ---rw rfc5085?  boolean
  ---rw pw_rfc5885?  boolean
  ---rw rfc6423?  boolean
  ---rw rfc6310?  boolean
  ---rw rfc7023?  boolean
+-:(tools-passive-oam)
  ---rw passive-oam-config {path-packet-counters,path-trace,proof-of-transit}?
    ---rw path-packet-counters-config
      ---rw flow-classifier* [name]
        ---rw name                       string
        ---rw access-list?               string
        ---rw data-export-profile* [name]
          ---rw name                                         string
          ---rw oam-data-export-type?                     data-export-type
    ---:(flexibleexport)
      ---rw flexible-export-params
        ---rw transport
          ---rw flex-data-export-protocol?  export
          ---rw flex-data-export-ip?       inet:ip-address
          ---rw flex-data-export-port?     inet:port-number
    ---:(ipfix)
      ---rw ip-fix-params
        ---rw ipfix-data-export-protocol?  export
        ---rw ipfix-data-export-ip?       inet:ip-address
        ---rw ipfix-data-export-port?     inet:port-number
    ---rw sampling-frequency?  timestamp-accuracy
    ---rw sampling-interval?    uint64
    ---rw data-export-trace-time-res?  timestamp-accuracy {path-trace}?
    ---rw data-export-trace-delay-threshold?  uint32 {path-trace}?
    ---rw data-export-trace-jitter-threshold?  uint32 {path-trace}?
    ---rw data-export-pot-failure-threshold?  uint32 {path-trace}?
nt32 {proof-of-transit}?
Internet-Draft       Connection-Less OAM YANG model             May 2016

|        |  |  |  ++rw data-export-ppc-drops-failure-threshold?     ui
|        |  |  |  ++rw data-export-ppc-reorder-failure-threshold?     ui
|        |  |  |  ++rw data-export-ppc-oos-failure-threshold?         ui
|        |  |  |  ++rw data-export-ppc-dup-failure-threshold?         ui
|        |  |  |  ++rw transport-encap-profile* [name]
|        |  |  |     ++rw name              string
|        |  |  |     ++rw transport-type?   oam-transport-type-def
|        |  |  |     ++rw node-id?           uint32
|        |  |  |     ++rw node-interfaces* [index]
|        |  |  |        ++rw index          uint32
|        |  |  |        ++rw intf-name?     if:interface-ref
|        |  |  |  ++rw md-ppc-stats?         boolean
|        |  |  |  ++rw ppc-mode?             passive-oa

m-mode

++rw notification-ppc?     boolean
++rw notification-ppc-drops-failure-threshold?     uint32
++rw notification-ppc-reorder-failure-threshold?   uint32
++rw notification-ppc-oos-failure-threshold?       uint32
++rw notification-ppc-dup-failure-threshold?       uint32
++rw proof-of-transit-config

++rw flow-classifier* [name]

++rw name              string
++rw access-list?       string
++rw data-export-profile* [name]

++rw oam-data-export-type?     da
++rw (data-export-types)?
++:(flexibleexport)
++rw flexible-export-params
++rw transport
++rw flex-data-export-protocol?   export

-transport-type

++rw flex-data-export-ip?     inet:ip
++rw flex-data-export-port?   inet:port

p-address

++rw encoding
++rw data-export-encoding-type?   enumeration
++:(ipfix)
++rw ip-fix-params
++rw ip-fix-data-export-protocol? export

transport-type

++rw ipfix-data-export-ip?     inet:ip
++rw ipfix-data-export-port?   inet:port

address

++rw sampling-frequency?     ti
++rw sampling-interval?       ui

nt64

++rw data-export-trace-time-res?     ti
++rw data-export-trace-delay-threshold?   ui
++rw data-export-trace-jitter-threshold?   ui
++rw data-export-trace-time-res?     ti
++rw data-export-trace-delay-threshold?   ui
++rw data-export-trace-jitter-threshold?   ui
++rw data-export-trace-time-res?     ti
++rw data-export-trace-delay-threshold?   ui
++rw data-export-trace-jitter-threshold?   ui
Kumar, et al. Expires November 27, 2016 [Page 37]
++-rw data-export-ppc-dup-failure-threshold? ui

nt32 {path-packet-counters}?
  +++-rw transport-encap-profile* [name]
    +++-rw name string
    +++-rw transport-type? oam-transport-type-def
  +++-rw node-id? uint32
  +++-rw node-interfaces* [index]
    +++-rw index uint32
  +++-rw intf-name? if:interface-ref
  +++-rw md-proof-of-transit? boolean
  +++-rw md-pot-mode? passive-oa

m-mode
  +++-rw notification-pot? boolean
  +++-rw notification-pot-failure-threshold? uint32
  +++-rw path-trace-config
    +++-rw flow-classifier* [name]
      +++-rw name string
      +++-rw access-list? string
    +++-rw data-export-profile* [name]
      +++-rw name string
  +++-rw data-export-profile* [name]
    +++-rw oam-data-export-type? da

  +++-rw (data-export-types)?
    +++-:(flexibleexport)
      +++-rw flex-data-export-params
        +++-rw transport
          +++-rw flex-data-export-protocol? export

  +++-:(ipfix)
    +++-rw ip-fix-params
      +++-rw ipfix-data-export-protocol? export-t

-t-transport-type
  +++-rw flex-data-export-ip? inet:ip

p-address
  +++-rw flex-data-export-port? inet:port

port-number
  +++-rw encoding
    +++-rw data-export-encoding-type? enumeration

  +++-:(ipfix)
    +++-rw ip-fix-params
      +++-rw ipfix-data-export-protocol? export-t

transport-type

address
  +++-rw ipfix-data-export-port? inet:port

t-number
  +++-rw sampling-frequency? ti

mestamp-accuracy
  +++-rw sampling-interval? ui

nt64
  +++-rw data-export-trace-time-res? ti

mestamp-accuracy {path-trace}?
  +++-rw data-export-trace-delay-threshold? ui

nt32 {path-trace}?
  +++-rw data-export-trace-jitter-threshold? ui

nt32 {path-trace}?
  +++-rw data-export-pot-failure-threshold? ui

nt32 {proof-of-transit}?
  +++-rw data-export-ppc-drops-failure-threshold? ui

nt32 {path-packet-counters}?
  +++-rw data-export-ppc-reorder-failure-threshold? ui

nt32 {path-packet-counters}?
  +++-rw data-export-ppc-oos-failure-threshold? ui

nt32 {path-packet-counters}?
  +++-rw data-export-ppc-dup-failure-threshold? ui
nt32 {path-packet-counters}?

+---rw transport-encap-profile* [name]
    +---rw name   string
    +---rw transport-type? oam-transport-type-def
    +---rw node-id? uint32
    +---rw node-interfaces* [index]
---rw index         uint32
   ---rw md-tracing?   if:interface-ref
   +--rw trace-mode    passive-oa
m-mode
   +--rw trace-length  uint8
   +--rw trace-time-stamp  timestamp-accuracy
accuracy
   +--rw trace-app-data?  uint32
   +--rw trace-include-interface boolean
---rw oam-layers* [index]
   ---rw index         uint16
   ---rw level?        int32
   ---rw (tp-address)?
      +--:(mac-address)
         | +--rw mac-address?           yang:mac-address
      +--:(ipv4-address)
         | +--rw ipv4-address?           inet:ipv4-address
      +--:(ipv6-address)
         | +--rw ipv6-address?           inet:ipv6-address
      +--:(src-dst-address)
         | +--rw src-ip-address?         inet:ip-address
         +--rw dst-ip-address?         inet:ip-address
      +--rw Interface?             if:interface-ref
      +--:(fec)
      +--rw fec-type?              fec-type
      +--rw (fec-value)?
         +--:(ip-prefix)
            | +--rw ip-prefix?           inet:ip-prefix
         +--:(bgp)
            | +--rw bgp?                   inet:ip-prefix
         +--:(tunnel)
            | +--rw tunnel-interface?      uint32
         +--:(l3vpn)
            | +--rw l3vpn-id?              uint32
         +--:(pw)
            | +--rw remote-pe-address?     inet:ip-address
               | +--rw pw-id?                 uint32
         +--:(vpls)
            | +--rw route-distinguisher?    uint32
               | +--rw sender-ve-id?          uint32
               | +--rw receiver-ve-id?        uint32
         +--:(mpls-mldp)
            | +--rw (root-address)?
               | +--:(ip-address)
                  | +--rw source-address?         inet:ip-address
                  | +--rw group-ip-address?       IP-Multicast-Group-Address
         +--:(vnp)
            | +--rw as-number?              inet:as-number
            +--:(global-id)
Internet-Draft       Connection-Less OAM YANG model             May 2016

|                     +--rw lsp-id? string
+--:(tlv-address)
    +--rw tlv-type? int16
    +--rw tlv-len? int16
    +--rw tlv-value? binary

augment /nd:networks/nd:network/nd:node:
    +--rw test-point-ip-prefix-location-list {connection-less}?
        +--rw test-point-locations* [ip-prefix-location]
            +--rw ip-prefix-location inet:ip-prefix
            +--rw vrf? routing-instance-ref
            +--rw (tp-address)?
                +--:(mac-address)
                    |  +--rw mac-address? yang:mac-address
                +--:(ipv4-address)
                    +--rw ipv4-address? inet:ipv4-address
                +--:(ipv6-address)
                    +--rw ipv6-address? inet:ipv6-address
                +--:(src-dst-address)
                    +--rw src-ip-address? inet:ip-address
                    +--rw dst-ip-address? inet:ip-address
                    +--rw Interface? if:interface-ref
                +--:(fec)
                    +--rw fec-type? fec-type
                    +--rw (fec-value)?
                        +--:(ip-prefix)
                            |  +--rw ip-prefix? inet:ip-prefix
                        +--:(bgp)
                            |  +--rw bgp? inet:ip-prefix
                        +--:(tunnel)
                            |  +--rw tunnel-interface? uint32
                        +--:(l3vpn)
                            |  +--rw l3vpn-id? uint32
                        +--:(pw)
                            |  +--rw remote-pe-address? inet:ip-address
                            |  +--rw pw-id? uint32
                        +--:(vpls)
                            |  +--rw route-distinguisher? uint32
                            |  +--rw sender-ve-id? uint32
                            |  +--rw receiver-ve-id? uint32
                        +--:(mpls-mldp)
                            |  +--rw (root-address)?
                                +--:(ip-address)
                                    |  +--rw source-address? inet:ip-address
                                    |  +--rw group-ip-address? IP-Multicast-Group-Addre

Kumar, et al.       Expires November 27, 2016 [Page 40]
---: (tlv-address)
  ---rw tlv-type? int16
  ---rw tlv-len? int16
  ---rw tlv-value? binary

---rw (technology)?
  ---: (technology-null)
    | ---rw tech-null? empty
  ---: (technology-string)
    +--rw ipv4-icmp? string

---rw (tools)?
  ---: (tools-empty)
  | ---rw tools-null? empty
  ---: (tools-ip)
  | ---rw rfc792? boolean
  | ---rw rfc4443? boolean
  | ---rw rfc4884? boolean
  | ---rw rfc5837? boolean
  ---: (tools-bfd)
  | ---rw rfc5881? boolean
  | ---rw rfc5883? boolean
  | ---rw rfc5884? boolean
  | ---rw rfc5885? boolean
  ---: (tools-mpls)
  | ---rw rfc4379? boolean
  | ---rw rfc4687? boolean
  | ---rw rfc4950? boolean
  | ---rw mpls-rfc5884? boolean
  ---: (tools-mpls-tp)
  | ---rw rfc6426? boolean
  | ---rw rfc6435? boolean
  | ---rw rfc6374? boolean
  ---: (tools-pw)
  | ---rw rfc5085? boolean
  | ---rw pw_rfc5885? boolean
  | ---rw rfc6423? boolean
  | ---rw rfc6310? boolean
  | ---rw rfc7023? boolean
  ---: (tools-passive-oam)
    +--rw passive-oam-config {path-packet-counters, path-trace, proof-of-transit}?
      +--rw path-packet-counters-config
        | ---rw flow-classifier* [name]
        |   ---rw name string
        |   ---rw access-list? string
        |   ---rw data-export-profile* [name]
        |     | ---rw name string
        |     | ---rw oam-data-export-type? data-export-type
        |     |   | ---rw (data-export-types)?
        |     |     | ---: (flexibleexport)
++rw flexible-export-params
  +--rw transport
    +--rw flex-data-export-protocol?  export
-transport-type
p-address
  |  |  |  |  |  |  +--rw flex-data-export-ip?  inet:i
|  |  |  |  |  |  |  +--rw flex-data-export-port?  inet:p
ort-number
  |  |  |  |  |  |  +--rw encoding
    |  |  |  |  |  |  +--rw data-export-encoding-type?  enum
ratiotion
  |  |  |  |  |  |  +--:(ipfix)
    |  |  |  |  |  |  +--rw ip-fix-params
    |  |  |  |  |  |  +--rw ipfix-data-export-protocol?  export-t
ransport-type
address
  |  |  |  |  |  |  +--rw ipfix-data-export-ip?  inet:ip-
t-number
  |  |  |  |  |  |  +--rw sampling-frequency?  ti
mestamp-accuracy
  |  |  |  |  |  |  +--rw sampling-interval?  ui
nt64
  |  |  |  |  |  |  +--rw data-export-trace-time-res?  ti
mestamp-accuracy {path-trace}?
  |  |  |  |  |  |  +--rw data-export-trace-delay-threshold?  ui
nt32 {path-trace}?
  |  |  |  |  |  |  +--rw data-export-trace-jitter-threshold?  ui
nt32 {path-trace}?
  |  |  |  |  |  |  +--rw data-export-pot-failure-threshold?  ui
nt32 {proof-of-transit}?
  |  |  |  |  |  |  +--rw data-export-ppc-drops-failure-threshold?  ui
nt32 {path-packet-counters}?
  |  |  |  |  |  |  +--rw data-export-ppc-reorder-failure-threshold?  ui
nt32 {path-packet-counters}?
  |  |  |  |  |  |  +--rw data-export-ppc-oos-failure-threshold?  ui
nt32 {path-packet-counters}?
  |  |  |  |  |  |  +--rw data-export-ppc-dup-failure-threshold?  ui
nt32 {path-packet-counters}?
  |  |  |  |  |  |  +--rw transport-encap-profile* [name]
    |  |  |  |  |  |  +--rw name  string
    |  |  |  |  |  |  +--rw transport-type?  oam-transport-type-def
    |  |  |  |  |  |  +--rw node-id?  uint32
    |  |  |  |  |  |  +--rw node-interfaces* [index]
    |  |  |  |  |  |  +--rw index  uint32
    |  |  |  |  |  |  +--rw intf-name?  if:interface-ref
    |  |  |  |  |  |  +--rw md-ppc-stats?  boolean
    |  |  |  |  |  |  +--rw ppc-mode?  passive-oa
m-mode
  |  |  |  |  |  |  +--rw notification-ppc?  boolean
  |  |  |  |  |  |  +--rw notification-ppc-drops-failure-threshold?  uint32
  |  |  |  |  |  |  +--rw notification-ppc-reorder-failure-threshold?  uint32
  |  |  |  |  |  |  +--rw notification-ppc-oos-failure-threshold?  uint32
  |  |  |  |  |  |  +--rw notification-ppc-dup-failure-threshold?  uint32
  |  |  |  |  |  |  +--rw proof-of-transit-config
    |  |  |  |  |  |  +--rw flow-classifier* [name]
    |  |  |  |  |  |  +--rw name  string
    |  |  |  |  |  |  +--rw access-list?  string
    |  |  |  |  |  |  +--rw data-export-profile* [name]
    |  |  |  |  |  |  |  +--rw name  string
    |  |  |  |  |  |  |  +--rw data-export-type?  da
ringleg
  |  |  |  |  |  |  +--rw oam-data-export-type?  da
p-address
   |       |       |       |       |       | +--rw flex-data-export-port? inet:p
port-number
   |       |       |       |       |       | +--rw encoding
   |       |       |       |       |       |   +--rw data-export-encoding-type? enumer
ation
   |       |       |       |       |       | +--:(ipfix)
   |       |       |       |       |       |   +--rw ip-fix-params
---rw ipfix-data-export-protocol? export-type
---rw ipfix-data-export-ip? inet:ip-address
---rw ipfix-data-export-port? inet:port-number
---rw sampling-frequency? time-interval
---rw sampling-interval? untyped
---rw data-export-trace-time-res? time-interval
---rw data-export-trace-delay-threshold? untyped
---rw data-export-trace-jitter-threshold? untyped
---rw data-export-pot-failure-threshold? untyped
---rw data-export-ppc-drops-failure-threshold? untyped
---rw data-export-ppc-reorder-failure-threshold? untyped
---rw data-export-ppc-oos-failure-threshold? untyped
---rw data-export-ppc-dup-failure-threshold? untyped
---rw transport-encap-profile* [name]
  ---rw name string
  ---rw transport-type? oam-transport-type-definition
  ---rw node-id? uint32
  ---rw node-interfaces* [index]
    ---rw index uint32
    ---rw intf-name? if:interface-ref
  ---rw md-tracing? boolean
  ---rw trace-mode passive-oam
  ---rw trace-length uint8
  ---rw trace-time-stamp timestamp
  ---rw trace-app-data? uint32
  ---rw trace-include-interface boolean
---rw oam-layers* [index]
  ---rw index uint16
  ---rw level? int32
  ---rw (tp-address)?
    ---:(mac-address)
      ---rw mac-address? yang:mac-address
    ---:(ipv4-address)
      ---rw ipv4-address? inet:ipv4-address
    ---:(ipv6-address)
      ---rw ipv6-address? inet:ipv6-address
    ---:(src-dst-address)
      ---rw src-ip-address? inet:ip-address
      ---rw dst-ip-address? inet:ip-address
      ---rw Interface? if:interface-ref
    ---:(fec)
      ---rw fec-type? fec-type
      ---rw (fec-value)?
        ---:(ip-prefix)
          ---rw ip-prefix? inet:ip-prefix
        ---:(bgp)
          ---rw bgp? inet:ip-prefix
        ---:(tunnel)
augment /nd:networks/nd:network/nd:node:
  +--rw test-point-route-dist-location-list {connection-less}?
  |  +--rw test-point-locations* [route-dist-location]
  |     +--rw route-dist-location uint32
  |     +--rw vrf? routing-instance-ref
  +--rw (tp-address)?
     +--:(mac-address)
        |  +--rw mac-address? yang:mac-address
     +--:(ipv4-address)
        |  +--rw ipv4-address? inet:ipv4-address
     +--:(ipv6-address)
        |  +--rw ipv6-address? inet:ipv6-address
     +--:(src-dst-address)
        |  +--rw src-ip-address? inet:ip-address
        |  +--rw dst-ip-address? inet:ip-address
        |  +--rw Interface? if:interface-ref
     +--:(fec)
        |  +--rw fec-type? fec-type
        |  +--rw (fec-value)?
        |     +--:(ip-prefix)
        |     |  +--rw ip-prefix? inet:ip-prefix
        |     +--:(bgp)
        |        |  +--rw bgp? inet:ip-prefix
        |        +--:(tunnel)
        |     |  +--rw tunnel-interface? uint32
---:(l3vpn)
  |  ---:rw l3vpn-id?  uint32
---:(pw)
  |  ---:rw remote-pe-address?  inet:ip-address
  |  ---:rw pw-id?  uint32
---:(vpls)
  |  ---:rw route-distinguisher?  uint32
  |  ---:rw sender-ve-id?  uint32
  |  ---:rw receiver-ve-id?  uint32
---:(mpls-mldp)
  |  ---:rw (root-address)?
  |  |  ---::(ip-address)
  |  |  |  ---:rw source-address?  inet:ip-address
  |  |  |  ---:rw group-ip-address?  IP-Multicast-Group-Address
  |  |  ---::(vpn)
  |  |  |  ---:rw as-number?  inet:as-number
  |  |  |  ---::(global-id)
  |  |  |  ---:rw lsp-id?  string
---:(tlv-address)
  |  ---:rw tlv-type?  int16
  |  ---:rw tlv-len?  int16
  |  ---:rw tlv-value?  binary
---:rw (technology)?
  |  ---::(technology-null)
  |  |  ---:rw tech-null?  empty
  |  |  ---::(technology-string)
  |  |  |  ---:rw ipv4-icmp?  boolean
---:rw (tools)?
  |  ---::(tools-empty)
  |  |  ---:rw tools-null?  empty
  |  ---::(tools-ip)
  |  |  ---:rw rfc792?  boolean
  |  |  ---:rw rfc4443?  boolean
  |  |  ---:rw rfc4884?  boolean
  |  |  ---:rw rfc5837?  boolean
  |  ---::(tools-bfd)
  |  |  ---:rw rfc5881?  boolean
  |  |  ---:rw rfc5883?  boolean
  |  |  ---:rw rfc5884?  boolean
  |  |  ---:rw rfc5885?  boolean
  |  ---::(tools-mpls)
  |  |  ---:rw rfc4379?  boolean
  |  |  ---:rw rfc4687?  boolean
  |  |  ---:rw rfc4950?  boolean
  |  |  ---:rw mpls-rfc5884?  boolean
---:rw (tools-mpls-tp)
  |  ---:rw rfc6426?  boolean
  |  ---:rw rfc6435?  boolean
| +--rw rfc6374?          boolean |
| +--:(tools-pw)          |
| +--rw rfc5085?          boolean |
| +--rw pw_rfc5885?       boolean |
| +--rw rfc6423?          boolean |
| +--rw rfc6310?          boolean |
| +--rw rfc7023?          boolean |
| +--:(tools-passive-oam) |
| +--rw passive-oam-config (path-packet-counters, path-trace, proof-of-transit)? |
| | +--rw path-packet-counters-config |
| | | +--rw name                        string |
| | | +--rw access-list?                string |
| | | +--rw data-export-profile* [name] |
| | | | +--rw name                       string |
| | +--rw oam-data-export-type?        data-export-type |
| +--:(flexibleexport) |
| | | +--rw flexible-export-params    |
| | | | +--rw transport                 |
| | | | | +--rw flex-data-export-protocol? export |
| | | | | +--rw flex-data-export-ip?      inet:ip-address |
| | | | | | +--rw flex-data-export-port?    inet:port-number |
| | | | | | | +--rw encoding                |
| | | | | | | | +--rw data-export-encoding-type? enumeration |
| | | | | | | | | +--:(ipfix)                |
| | | | | | | | | | +--rw ipfix-params           |
| | | | | | | | | | | +--rw ipfix-data-export-protocol? export-type |
| | | | | | | | | | | | +--rw ipfix-data-export-ip?    inet:ip-address |
| | | | | | | | | | | | | +--rw ipfix-data-export-port?    inet:port-number |
| | | | | | | | | | | | | | +--rw sampling-frequency?       ti |
| | | | | | | | | | | | | | | +--rw sampling-interval?         ui |
| | | | | | | | | | | | | | | | +--rw data-export-trace-time-res? ti |
| | | | | | | | | | | | | | | | | +--rw data-export-trace-delay-threshold? ui |
| | | | | | | | | | | | | | | | | | +--rw data-export-trace-jitter-threshold? ui |
| | | | | | | | | | | | | | | | | | | +--rw data-export-pot-failure-threshold? ui |
| | | | | | | | | | | | | | | | | | | | +--rw data-export-ppc-drops-failure-threshold? ui |
| | | | | | | | | | | | | | | | | | | | | +--rw data-export-ppc-reorder-failure-threshold? ui |
| | | | | | | | | | | | | | | | | | | | | | +--rw data-export-ppc-oos-failure-threshold? ui |
| | | | | | | | | | | | | | | | | | | | | | | +--rw data-export-ppc-dup-failure-threshold? ui |
| | | | | | | | | | | | | | | | | | | | | | | | +--rw transport-encap-profile* [name] |
| | | | | | | | | | | | | | | | | | | | | | | | | +--rw name                        string |
| | | | | | | | | | | | | | | | | | | | | | | | | | +--rw transport-type?           oam-transport-type-def |
+-rw node-id?        uint32
+-rw node-interfaces* [index]
   |  +--rw index     uint32
   |      +--rw intf-name? if:interface-ref
   +--rw md-ppc-stats? boolean
---rw ppc-mode? passive-oa

m-mode
  +--rw notification-ppc? boolean
  +--rw notification-ppc-drops-failure-threshold? uint32
  +--rw notification-ppc-reorder-failure-threshold? uint32
  +--rw notification-ppc-oos-failure-threshold? uint32
  +--rw notification-ppc-dup-failure-threshold? uint32
  +--rw proof-of-transit-config
    +--rw flow-classifier* [name]
      +--rw name string
      +--rw access-list? string
      +--rw data-export-profile* [name]
        +--rw name string

ring
  | | +--rw oam-data-export-type? data-export-type

ta-export-type
  | | +--rw (data-export-types)?
    +--:(flexibleexport)
      +--rw flexible-export-params
        +--rw transport
          +--rw flex-data-export-protocol? oam-transport-def
          | +--rw encoding
            +--rw data-export-encoding-type? enumeration

transport-type
  | | | +--rw flex-data-export-ip? inet:ip-address
  | | | +--rw flex-data-export-port? inet:port-number
  | | +--rw sampling-frequency? timestamp-accuracy
  | | +--rw sampling-interval? uint64
  | +--rw data-export-trace-time-res? timestamp-accuracy {path-trace}?
  | | +--rw data-export-trace-delay-threshold? uint32
  | +--rw data-export-trace-jitter-threshold? uint32 {path-trace}?
  +--rw data-export-pot-failure-threshold? uint32 {proof-of-transit}?
  +--rw data-export-ppc-drops-failure-threshold? uint32 {path-packet-counters}?
  +--rw data-export-ppc-reorder-failure-threshold? uint32 {path-packet-counters}?
  +--rw data-export-ppc-oos-failure-threshold? uint32 {path-packet-counters}?
  +--rw data-export-ppc-dup-failure-threshold? uint32 {path-packet-counters}?
  +--rw transport-encap-profile* [name]
    +--rw name string
    +--rw transport-type? oam-transport-type-def
    +--rw node-id? uint32
    +--rw node-interfaces* [index]
      +--rw index uint32
| m-mode | ---rw intf-name? if:interface-ref | boolean |
|        | ---rw md-proof-of-transit?       | passive-oa |
|        | ---rw pot-mode?                  |          |
|        | ---rw notification-pot?          | boolean |
|        | ---rw notification-pot-failure-threshold? | uint32 |

++rw path-trace-config
  +--rw flow-classifier* [name]
    +--rw name                      string
    +--rw access-list?              string
  +--rw data-export-profile* [name]
    +--rw name                     string
  +--rw oam-data-export-type?
  +--:(flexibleexport)
    +--rw flexible-export-params
      +--rw transport
        +--rw flex-data-export-protocol? export
      +--:(ipfix)
        +--rw ip-fix-params
          +--rw ipfix-data-export-protocol? export-type
    +--rw transport-type
    +--rw p-address
      +--rw flex-data-export-ip? inet:ip-address
    +--rw port-number
      +--rw flex-data-export-port? inet:port-number
  +--rw md-tracing?
    +--rw trace-mode                passive-oa
    +--rw trace-length              uint8
    +--rw trace-time-stamp          timestamp-accuracy

++--:(mac-address)
|  +--rw mac-address?       yang:mac-address
++--:(ipv4-address)
|  +--rw ipv4-address?      inet:ipv4-address
++--:(ipv6-address)
|  +--rw ipv6-address?      inet:ipv6-address
++--:(src-dst-address)
|  +--rw src-ip-address?     inet:ip-address
|  +--rw dst-ip-address?     inet:ip-address
|  +--rw Interface?          if:interface-ref
++--:(fec)
|  +--rw fec-type?           fec-type
++--rw (fec-value)?
  ++--:(ip-prefix)
   |    +--rw ip-prefix?       inet:ip-prefix
  ++--:(bgp)
   |    +--rw bgp?             inet:ip-prefix
  ++--:(tunnel)
   |    +--rw tunnel-interface? uint32
  ++--:(l3vpn)
   |    +--rw l3vpn-id?        uint32
  ++--:(pw)
   |    +--rw remote-pe-address? inet:ip-address
   |    +--rw pw-id?           uint32
++--:(vpls)
|    +--rw route-distinguisher? uint32
|    +--rw sender-ve-id?      uint32
|    +--rw receiver-ve-id?      uint32
++--:(mpls-mldp)
|    +--rw (root-address)?
     ++--:(ip-address)
      |      +--rw source-address?     inet:ip-address
      |      +--rw group-ip-address?    IP-Multicast-Group-Address
|    ++--:(vpn)
     |      +--rw as-number?         inet:as-number
     ++--:(global-id)
      |        +--rw lsp-id?          string
++--:(tlv-address)
|    +--rw tlv-type?          int16
|    +--rw tlv-len?           int16
|    +--rw tlv-value?         binary

augment /nd:networks/nd:network/nd:node:
  +--rw test-point-group-ip-address-location-list {connection-less}?
  +--rw test-point-locations* [group-ip-address-location]
   +--rw group-ip-address-location     IP-Multicast-Group-Address
   +--rw vrf?                         routing-instance-ref
   +--rw (tp-address)?
    |    +--:(mac-address)
<no text content for this page>
t-number
mestamp-accuracy
nt64
mestamp-accuracy {path-trace}?

---

Kumar, et al. Expires November 27, 2016 [Page 53]
---rw data-export-trace-delay-threshold?  ui
nt32 {path-trace}?  |  ---rw data-export-trace-jitter-threshold?  ui
nt32 {path-trace}?  |  ---rw data-export-pot-failure-threshold?  ui
nt32 {proof-of-transit}?  |  ---rw data-export-ppc-drops-failure-threshold?  ui
nt32 {path-packet-counters}?  |  ---rw data-export-ppc-reorder-failure-threshold?  ui
nt32 {path-packet-counters}?  |  ---rw data-export-ppc-oos-failure-threshold?  ui
nt32 {path-packet-counters}?  |  ---rw data-export-ppc-dup-failure-threshold?  ui
nt32 {path-packet-counters}?  |  ---rw transport-encap-profile* [name]
  |  +--rw name              string
  |  +--rw transport-type?   oam-transport-type-def
  +--rw node-id?           uint32
  +--rw node-interfaces* [index]
    +--rw index        uint32
    +--rw intf-name?   if:interface-ref
  +--rw md-proof-of-transit?  boolean
  +--rw pot-mode?         passive-oa
m-mode  |  ---rw notification-pot?  boolean
  ---rw notification-pot-failure-threshold?  uint32
  ---rw path-trace-config
    +--rw flow-classifier* [name]
      +--rw name              string
      +--rw access-list?     string
    +--rw data-export-profile* [name]
      +--rw name              string
      +--rw oam-data-export-type?  da
      +--:(flexibleexport)
        +--rw flexible-export-params
          +--rw transport
            +--rw flex-data-export-protocol?  export
-trans-transport-type
p-address  |  |  +--rw flex-data-export-ip?  inet:i
port-number  |  |  +--rw flex-data-export-port?  inet:p
+--rw encoding
  +--rw data-export-encoding-type?  enumer
ation
  +--:(ipfix)
    +--rw ip-fix-params
      +--rw ipfix-data-export-protocol?  export-t
transport-type
  +--rw ipfix-data-export-ip?  inet:ip
address  |  +--rw ipfix-data-export-port?  inet:por
t-number  |  +--rw sampling-frequency?  ti
nt64  |  +--rw sampling-interval?  ui
mestamp-accuracy  |  +--rw data-export-trace-time-res?  ti
mestamp-accuracy {path-trace}?
  |  +--rw data-export-trace-delay-threshold?  ui
nt32 {path-trace}?
    |        |  |  |  +--rw data-export-trace-jitter-threshold?     ui
nt32 {path-trace}?
    |        |  |  |  +--rw data-export-pot-failure-threshold?      ui
nt32 {proof-of-transit}?
    |        |  |  |  +--rw data-export-ppc-drops-failure-threshold? ui
nt32 {path-packet-counters}?
    |        |  |  |  +--rw data-export-ppc-reorder-failure-threshold? ui
nt32 {path-packet-counters}?
    |        |  |  |  +--rw data-export-ppc-oos-failure-threshold?   ui
nt32 {path-packet-counters}?
Internet-Draft       Connection-Less OAM YANG model             May 2016

---rw data-export-ppc-dup-failure-threshold?       ui
nt32 {path-packet-counters}?
  |  +--rw transport-encap-profile* [name]
  |     +--rw name              string
  |     +--rw transport-type?   oam-transport-type-def
  |     +--rw node-id?           uint32
  |     +--rw node-interfaces* [index]
  |        +--rw index        uint32
  |        +--rw intf-name?   if:interface-ref
  |        +--rw md-tracing?                                   boolean
  |        +--rw trace-mode
  |        m-mode
    +--rw trace-length
    +--rw trace-time-stamp
    +--rw trace-app-data?                               uint32
    +--rw trace-include-interface
    |  +--rw index                  uint16
    |  +--rw level?                 int32
    ++--rw (tp-address)?
      |      +--:(mac-address)
      |      |  +--rw mac-address?           yang:mac-address
      |      +--:(ipv4-address)
      |      |  +--rw ipv4-address?          inet:ipv4-address
      |      +--:(ipv6-address)
      |      |  +--rw ipv6-address?          inet:ipv6-address
      |      +--:(src-dst-address)
      |      |  +--rw src-ip-address?        inet:ip-address
      |      +--rw dst-ip-address?        inet:ip-address
      |      +--rw Interface?
      |      |  +--rw fec-type?              fec-type
      |      ++--rw (fec-value)?
      |      |      +--:(ip-prefix)
      |      |      |  +--rw ip-prefix?             inet:ip-prefix
      |      |      +--:(bgp)
      |      |      |  +--rw bgp?                   inet:ip-prefix
      |      |      +--:(tunnel)
      |      |      |  +--rw tunnel-interface?       uint32
      |      |      +--:(l3vpn)
      |      |      |  +--rw l3vpn-id?              uint32
      |      |      +--:(pw)
      |      |      |  +--rw remote-pe-address?      inet:ip-address
      |      |      +--rw pw-id?               uint32
      |      +--:(vpls)
      |      |  +--rw route-distinguisher?    uint32
      |      +--rw sender-ve-id?         uint32
      |      +--rw receiver-ve-id?       uint32
      |      +--:(mpls-mldp)
      |      ++--rw (root-address)?
augment /nd:networks/nd:network/nd:node:
  +--:(tlv-address)
      +--:(global-id)
          +--rw lsp-id?                string
      +--:(vpn)
          |  +--rw as-number?             inet:as-number
      +--:(test-point-as-number-location-list) {connection-less}?
      +--rw test-point-locations* [as-number-location]
          +--rw as-number-location inet:as-number
      +--rw vrf?                   routing-instance-ref
      +--rw tp-address?      inet:ip-address
      +--:(mac-address)  yang:mac-address
          |  +--rw mac-address?           yang:mac-address
      +--:(ipv4-address)  inet:ipv4-address
          |  +--rw ipv4-address?          inet:ipv4-address
      +--:(ipv6-address)  inet:ipv6-address
          |  +--rw ipv6-address?          inet:ipv6-address
      +--:(src-dst-address)
          +--rw src-ip-address?        inet:ip-address
          +--rw dst-ip-address?        inet:ip-address
          +--rw Interface?             if:interface-ref
      +--:(fec)
          +--rw fec-type?              fec-type
      +--rw (fec-value)?
          |  +--:(ip-prefix)
              |      +--rw ip-prefix?             inet:ip-prefix
          |  +--:(bgp)
              |      +--rw bgp?                   inet:ip-prefix
          |  +--:(tunnel)
              |      +--rw tunnel-interface?      uint32
          |  +--:(l3vpn)
              |      +--rw l3vpn-id?              uint32
          |  +--:(pw)
              |      +--rw remote-pe-address?     inet:ip-address
              |      +--rw pw-id?                 uint32
          |  +--:(vpls)
              |      +--rw route-distinguisher?    uint32
              |      +--rw sender-ve-id?          uint32
              |      +--rw receiver-ve-id?        uint32
          |  +--:(mpls-mldp)
              |      +--rw (root-address)?
              |      +--:(ip-address)
| | | | | +--rw source-address? inet:ip-address
| | | | +--rw group-ip-address? IP-Multicast-Group-Address
| | | | | | --: (vpn)
| | | | | | | +--rw as-number? inet:as-number
| | | | | | | --: (global-id)
| | | | | | | | +--rw lsp-id? string
| | | | | | | +--: (tlv-address)
| | | | | | | | +--rw tlv-type? int16
| | | | | | | | +--rw tlv-len? int16
| | | | | | | | +--rw tlv-value? binary
| | | | | | | +--: (technology)
| | | | | | | | +--: (technology-null)
| | | | | | | | | +--rw tech-null? empty
| | | | | | | | +--: (technology-string)
| | | | | | | | | +--rw ipv4-icmp? string
| | | | | | | +--: (tools)
| | | | | | | | +--: (tools-empty)
| | | | | | | | | +--rw tools-null? empty
| | | | | | | | +--: (tools-ip)
| | | | | | | | | +--rw rfc7922? boolean
| | | | | | | | | +--rw rfc4443? boolean
| | | | | | | | | +--rw rfc4884? boolean
| | | | | | | | | +--rw rfc5837? boolean
| | | | | | | | +--: (tools-bfd)
| | | | | | | | | +--rw rfc5881? boolean
| | | | | | | | | +--rw rfc5883? boolean
| | | | | | | | | +--rw rfc5884? boolean
| | | | | | | | | +--rw rfc5885? boolean
| | | | | | | | +--: (tools-mpls)
| | | | | | | | | +--rw rfc4379? boolean
| | | | | | | | | +--rw rfc4687? boolean
| | | | | | | | | +--rw rfc4950? boolean
| | | | | | | | | +--rw mpls-rfc5884? boolean
| | | | | | | | +--: (tools-mpls-tp)
| | | | | | | | | +--rw rfc6426? boolean
| | | | | | | | | +--rw rfc6435? boolean
| | | | | | | | | +--rw rfc6374? boolean
| | | | | | | | +--: (tools-pw)
| | | | | | | | | +--rw rfc5085? boolean
| | | | | | | | | +--rw pw_rfc5085? boolean
| | | | | | | | | +--rw rfc4233? boolean
| | | | | | | | | +--rw rfc6519? boolean
| | | | | | | | | +--rw rfc7023? boolean
| | | | | | | | +--: (tools-passive-oam)
| | | | | | | | | +--rw passive-oam-config {path-packet-counters, path-trace, proof-of-transit}
| | | | | | | | | | +--rw path-packet-counters-config
| | | | | | | | | | | +--rw flow-classifier* [name]
| | | | | | | | | | | | +--rw name string

Kumar, et al. Expires November 27, 2016 [Page 57]
---rw access-list? string
---rw data-export-profile* [name]
  +--rw name string

---rw data-export-profile* [name]
  +--rw name string

ta-export-type
  +--rw data-export-profile? [name]

---rw transport-encap-profile* [name]
  +--rw name string
  +--rw transport-type? oam-transport-type-def
  +--rw node-id? uint32
  +--rw node-interfaces* [index]
    +--rw index uint32
    +--rw intf-name? if:interface-ref

---rw md-ppc-stats? boolean
---rw ppc-mode? passive-oa

---rw notification-ppc? boolean
---rw notification-ppc-drops-failure-threshold? uint32
---rw notification-ppc-reorder-failure-threshold? uint32
---rw notification-ppc-oos-failure-threshold? uint32
---rw notification-ppc-dup-failure-threshold? uint32
---rw proof-of-transit-config
ta-export-type
| | | +-rw oam-data-export-type? da

+--rw (data-export-types)?
| | +-:(flexibleexport)
| | | +-rw flexible-export-params
| | | | +-rw transport
| | | | | +-rw flex-data-export-protocol? export

-data-export-type
| | | | +-rw flex-data-export-ip? inet:i

p-address
| | | | +-rw flex-data-export-port? inet:p

ort-number
| | | | +-rw encoding
| | | | | +-rw data-export-encoding-type? enumeration

+-:(ipfix)
| | +--rw ip-fix-params
| | | +-rw ipfix-data-export-protocol? export-t

ransport-type

tnumber
| | | +-rw sampling-frequency? timestamp-accuracy
| | | | +-rw sampling-interval? uint64
| | | | | +-rw data-export-trace-time-res? timestamp-accuracy
| | | | | | +-rw data-export-trace-delay-threshold? uint32
| | | | | | | +-rw data-export-trace-jitter-threshold? uint32
| | | | | | | | +-rw data-export-pot-failure-threshold? uint32
| | | | | | | | | +-rw path-trace-config
| | | | | | | | | | +-rw flow-classifier* [name]
| | | | | | | | | | | +-rw name string
| | | | | | | | | | | | +-rw transport-type? oam-transport-type-def
| | | | | | | | | | | | | +-rw node-id? uint32
| | | | | | | | | | | | | | +-rw node-interfaces* [index]
| | | | | | | | | | | | | | | +-rw index uint32
| | | | | | | | | | | | | | | | +-rw intf-name? if:interface-ref
| | | | | | | | | | | | | | | | | +-rw md-proof-of-transit? boolean
| | | | | | | | | | | | | | | | | | +-rw pot-mode? passive-oa

m-mode
| +--rw notification-pot? boolean
| | +--rw notification-pot-failure-threshold? uint32
| +--rw path-trace-config
| | +--rw flow-classifier* [name]
| | | +-rw name string
| | | | +-rw access-list? string
| | | | | +--rw data-export-profile* [name]
| | | | | | +--rw name string
ta-export-type
  |        |  |  |  +--rw (data-export-types)?
  |        |  |  |  |  +--:(flexibleexport)
  |        |  |  |  |  |  +--rw flexible-export-params
  |        |  |  |  |  |  |  +--rw transport
  |        |  |  |  |  |  |  |  +--rw flex-data-export-protocol? export
-transport-type
| | | | | | +--rw flex-data-export-ip? inet:i-ip
p-address | | | | | | +--rw flex-data-export-port? inet:i-port
| | | | | +--rw encoding
     +--rw data-export-encoding-type? enumeration
ort-number | | | | | | +--:(ipfix)
     +--rw ip-fix-params
         +--rw ipfix-data-export-protocol? export:
transport-type | | | | | +--rw ipfix-data-export-ip? inet:i-ip-
t-number | | | | | +--rw sampling-frequency? ti
mestamp-accuracy | | | | | +--rw sampling-interval? ui
nt64 | | | | | +--rw data-export-trace-time-res? ti
mestamp-accuracy {path-trace}?
    +--rw data-export-trace-delay-threshold? ui
nt32 {path-trace}?
    +--rw data-export-trace-jitter-threshold? ui
nt32 {path-trace}?
    +--rw data-export-pot-failure-threshold? ui
nt32 {proof-of-transit}?
    +--rw data-export-ppc-drops-failure-threshold? ui
nt32 {path-packet-counters}?
    +--rw data-export-ppc-reorder-failure-threshold? ui
nt32 {path-packet-counters}?
    +--rw data-export-ppc-oos-failure-threshold? ui
nt32 {path-packet-counters}?
    +--rw data-export-ppc-dup-failure-threshold? ui
nt32 {path-packet-counters}?
    +--rw transport-encap-profile* [name]
        +--rw name              string
        +--rw transport-type?   oam-transport-type-def
    +--rw node-id?           uint32
    +--rw node-interfaces* [index]
        +--rw index        uint32
        +--rw intf-name?   if:interface-ref
        +--rw md-tracing?
        +--rw trace-mode    passive-oa
m-mode
    +--rw trace-length       uint8
    +--rw trace-time-stamp   timestamp-accuracy
    +--rw trace-app-data?    uint32
    +--rw trace-include-interface boolean
     +--rw oam-layers* [index]
         +--rw index        uint16
         +--rw level?       int32
     +--rw (tp-address)?
         +--:(mac-address)
             +--rw mac-address? yang:mac-address
         +--:(ipv4-address)
             +--rw ipv4-address? inet:ipv4-address
         +--:(ipv6-address)
             +--rw ipv6-address? inet:ipv6-address
         +--:(src-dst-address)
             +--rw src-ip-address? inet:ip-address
             +--rw dst-ip-address? inet:ip-address
---rw (fec-value)
  ---:(ip-prefix)
    ---rw ip-prefix? inet:ip-prefix
  ---:(bgp)
  ---:(tunnel)
    ---rw tunnel-interface? uint32
  ---:(l3vpn)
    ---rw l3vpn-id? uint32
  ---:(pw)
    ---rw remote-pe-address? inet:ip-address
    ---rw pw-id? uint32
  ---:(vpls)
    ---rw route-distinguisher? uint32
    ---rw sender-ve-id? uint32
    ---rw receiver-ve-id? uint32
  ---:(mpls-mldp)
    ---rw (root-address)
      ---:(ip-address)
        ---rw source-address? inet:ip-address
        ---rw group-ip-address? IP-Multicast-Group-Address
      ---:(vpn)
        ---rw as-number? inet:as-number
  ---:(global-id)
    ---rw lsp-id? string
---:(tlv-address)
  ---rw tlv-type? int16
  ---rw tlv-len? int16
  ---rw tlv-value? binary
augment /nd:networks/nd:network/nd:node:
  ---rw test-point-lsp-id-location-list {connection-less}?
  ---rw test-point-locations* [lsp-id-location]
    ---rw lsp-id-location string
    ---rw vrf? routing-instance-ref
  ---rw (tp-address)
    ---:(mac-address)
      ---rw mac-address? yang:mac-address
    ---:(ipv4-address)
      ---rw ipv4-address? inet:ipv4-address
    ---:(ipv6-address)
      ---rw ipv6-address? inet:ipv6-address
    ---:(src-dst-address)
      ---rw src-ip-address? inet:ipv6-address
      ---rw dst-ip-address? inet:ipv6-address
      ---rw Interface? if:interface-ref
    ---:(fec)
      ---rw fec-type? fec-type
    ---rw (fec-value)?
---rw rfc4687?  boolean
---rw rfc4950?  boolean
---rw mpls-rfc5884?  boolean
---:(tools-mpls-tp)
  ---rw rfc6426?  boolean
  ---rw rfc6435?  boolean
  ---rw rfc6374?  boolean
---:(tools-pw)
  ---rw rfc5085?  boolean
  ---rw pw_rfc5885?  boolean
  ---rw rfc6423?  boolean
  ---rw rfc6310?  boolean
  ---rw rfc7023?  boolean
---:(tools-passive-oam)
  ---rw passive-oam-config {path-packet-counters,path-trace,proof-of-transit}?
    ---rw path-packet-counters-config
      ---rw flow-classifier* [name]
        ---rw name                       string
        ---rw access-list?               string
        ---rw data-export-profile* [name]
          ---rw name                       string
    ---rw oam-data-export-type?  data-export-type
---:(flexibleexport)
  ---rw flex-data-export-params
    ---rw transport
    ---rw flex-data-export-protocol?  export-transport-type
---:(ipfix)
  ---rw ip-fix-params
    ---rw ipfix-data-export-protocol?  export-transport-type
---:(ipfix)
  ---rw ipfix-data-export-ip?  inet:ip-address
---:(ipfix)
  ---rw ipfix-data-export-port?  inet:port-number
---rw sampling-frequency?  timestamp-accuracy
---rw sampling-interval?  uint64
---rw data-export-trace-time-res?  timestamp-accuracy {path-trace}?
---rw data-export-trace-delay-threshold?  uint32 {path-trace}?
---rw data-export-trace-jitter-threshold?  uint32 {path-trace}?
---rw data-export-pot-failure-threshold?  uint32 {proof-of-transit}?
---rw data-export-ppc-drops-failure-threshold?  uint32 {path-packet-counters}?
---rw data-export-ppc-reorder-failure-threshold?  uint32 {path-packet-counters}?
---rw data-export-ppc-oos-failure-threshold?  uint32
nt32 {path-packet-counters}?
        |        |  |  |  +--rw data-export-ppc-dup-failure-threshold?   ui
nt32 {path-packet-counters}?
        |        |  |  |  +--rw transport-encap-profile* [name]
        |        |  |  |  +--rw name              string

Kumar, et al.           Expires November 27, 2016              [Page 63]
Internet-Draft       Connection-Less OAM YANG model             May 2016

|     +--rw transport-type?   oam-transport-type-def
|     +--rw node-id?           uint32
|     +--rw node-interfaces* [index]
|     |     +--rw index        uint32
|     |     +--rw intf-name?   if:interface-ref
|     +--rw md-ppc-stats?       boolean
|     +--rw ppc-mode?           passive-oa

m-mode

|     +--rw notification-ppc?   boolean
|     +--rw notification-ppc-drops-failure-threshold?   uint32
|     +--rw notification-ppc-reorder-failure-threshold?  uint32
|     +--rw notification-ppc-oos-failure-threshold?      uint32
|     +--rw notification-ppc-dup-failure-threshold?      uint32

++rw proof-of-transit-config

|     +--rw flow-classifier* [name]
|     |     +--rw name                       string
|     |     +--rw access-list?               string
|     |     +--rw data-export-profile* [name]
|     |     |     +--rw name                                         string
|     |     |     +--rw data-export-type?                        da
|     |     |     +--:(flexibleexport)
|     |     |     |     +--rw flexible-export-params
|     |     |     |     |     +--rw transport
|     |     |     |     |     |     +--rw flex-data-export-protocol?   export
|     |     |     |     |     |     +--rw encoding
|     |     |     |     |     |     |     +--rw data-export-encoding-type?   enumeration
|     |     |     |     |     |     +--:(ipfix)
|     |     |     |     |     |     |     +--rw ip-fix-params
|     |     |     |     |     |     |     |     +--rw ipfix-data-export-protocol?   export-type
|     |     |     |     |     |     |     +--rw address
|     |     |     |     |     |     |     |     +--rw ipfix-data-export-ip?   inet:ip-address
|     |     |     |     |     |     |     |     +--rw t-number
|     |     |     |     |     |     |     |     +--rw sampling-frequency?
|     |     |     |     |     |     |     |     +--rw sampling-interval?
|     |     |     |     |     |     |     +--rw data-export-trace-time-res?
|     |     |     |     |     |     |     |     +--rw data-export-trace-delay-threshold?
|     |     |     |     |     |     |     |     +--rw data-export-trace-jitter-threshold?
|     |     |     |     |     |     |     |     +--rw data-export-pot-failure-threshold?
|     |     |     |     |     |     |     |     +--rw data-export-ppc-drops-failure-threshold?
|     |     |     |     |     |     |     |     +--rw data-export-ppc-reorder-failure-threshold?
|     |     |     |     |     |     |     |     +--rw data-export-ppc-oos-failure-threshold?
|     |     |     |     |     |     |     |     +--rw data-export-ppc-dup-failure-threshold?
---rw transport-encap-profile* [name]
  +--rw name              string
  +--rw transport-type?    oam-transport-type-def
  +--rw node-id?           uint32
  +--rw node-interfaces*   [index]
+--rw trace-app-data?                      uint32
+--rw trace-include-interface           boolean

+-rw oam-layers* [index]
  +--rw index                  uint16
  +--rw level?                 int32
  +--rw (tp-address)?
    +--:(mac-address)
      |  +--rw mac-address?       yang:mac-address
    +--:(ipv4-address)
      |  +--rw ipv4-address?      inet:ipv4-address
    +--:(ipv6-address)
      |  +--rw ipv6-address?      inet:ipv6-address
    +--:(src-dst-address)
      |  +--rw src-ip-address?    inet:ip-address
      +--rw dst-ip-address?     inet:ip-address
    +--rw Interface?             if:interface-ref
    +--:(fec)
      |  +--rw fec-type?          fec-type
      +--rw (fec-value)?
        +--:(ip-prefix)
          |  +--rw ip-prefix?         inet:ip-prefix
        +--:(bgp)
          |  +--rw bgp?               inet:ip-prefix
        +--:(tunnel)
          |  +--rw tunnel-interface?  uint32
        +--:(l3vpn)
          |  +--rw l3vpn-id?          uint32
        +--:(pw)
          |  +--rw remote-pe-address? inet:ip-address
          |  +--rw pw-id?             uint32
        +--:(vpls)
          |  +--rw route-distinguisher? uint32
          |  +--rw sender-ve-id?      uint32
          |  +--rw receiver-ve-id?    uint32
        +--:(mpls-mldp)
          |  +--rw (root-address)?
            +--:(ip-address)
              |  +--rw source-address?     inet:ip-address
              |  +--rw group-ip-address?   IP-Multicast-Group-Address
            +--:(vpn)
              |  +--rw as-number?          inet:as-number
              +--:(global-id)
                |  +--rw lsp-id?             string
            +--:(tlv-address)
              +--rw tlv-type?           int16
              +--rw tlv-len?            int16
              +--rw tlv-value?          binary

rpcs:
++-x continuity-check {continuity-check}?
| ++-w input
| | ++-w destination-tp
| | | ++-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
| | | | | | | | ++-:(l3vpn)
| | | | | | | | | ++-w l3vpn-id?  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
| | | | | | | | | | | | | | | ++-w session-type-enum?  enumeration
| | | | | | | | | | | | | | | | ++-w source-interface?  if:interface-ref
| | | | | | | | | | | | | | | | | ++-w outbound-interface?  if:interface-ref
++-w count?                uint32
++++-w vrf?                  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 tx-packet-count?     oam-counter32
|  +++-ro rx-packet-count?     oam-counter32
|  +++-ro min-delay?           oam-counter32
|  +++-ro average-delay?       oam-counter32
|  +++-ro max-delay?           oam-counter32

++++-x path-discovery
++++-w input
|  +++-w destination-tp
|     +++-w (tp-address)?
|     |     +++-:(mac-address)
|     |     |     +++-w mac-address? yang:mac-address
|     |     +++-:(ipv4-address)  inet:ipv4-address
|     |     |     +++-w ipv4-address? inet:ipv4-address
|     |     +++-:(ipv6-address)  inet: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
|     |     |     +++-:(l3vpn)
|     |     |     |     |     +++-w l3vpn-id?   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)?
4. OAM YANG Module

<CODE BEGINS> file "ietf-connectionless-oam.yang"

    module ietf-connectionless-oam {

Kumar, et al. Expires November 27, 2016 [Page 70]
namespace "urn:ietf:params:xml:ns:yang:ietf-connectionless-oam";
prefix coam;

import ietf-network{
  prefix nd;
}
import ietf-yang-types {
  prefix yang;
}
import ietf-interfaces {
  prefix if;
}
import ietf-inet-types {
  prefix inet;
}
import ietf-routing {
  prefix rt;
}
import ietf-packet-fields {
  prefix pfs;
  revision-date 2015-06-11;
}
organization "IETF LIME Working Group";
contact
  "Deepak Kumar dekumar@cisco.com
  Qin Wu       bill.wu@huawei.com";
description
  "This YANG module defines the generic configuration,
   statistics and rpc 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 2015-12-22 {
  description
    "Initial revision. - 01 version";
  reference "";
}
/* features */
feature connection-less {
  description
    "this feature indicates that OAM solution is connection less.";
}
feature continuity-check {
  description
    "This feature indicates that the server supports
executing continuity check OAM command and returning a response. Servers that do not advertise this feature will not support executing continuity check command or rpc model for continuity check command.

}

feature path-discovery {

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

}

feature path-trace {

description
"This feature indicates that the server supports configuration and execution of the in-band passive path tracing feature. Servers that do not advertise this feature will not support executing in-band path trace and the relevant configuration.

Path-trace feature is part of the passive-oam feature set, where meta-data is expected to be collected at every hop that a packet traverses, i.e. in a typical deployment all nodes in an passive-oam domain would participate in the domain and thus be transit nodes, encapsulating or decapsulating nodes. The network diameter of the domain is assumed to be known. The passive-oam path-trace decapsulating node removes the meta-data and process and/or export the meta-data.

Some examples of the meta-data include node identification, incoming interface identification, outgoing interface identification, timestamp or generic application specific meta-data."

}

feature path-packet-counters {

description
"This feature indicates that the server supports executing passive OAM via path packet counter options. Servers that do not advertise this feature will not support executing passive OAM path packet counters.

Path-packet-counters feature is part of the passive-oam feature set, where the meta-data carried is to be interpreted by the passive-oam domain’s encapsulating and decapsulating
nodes. Sequence numbers are the main meta-data added and can be used to detect packet loss, packet reordering or packet duplication.

feature proof-of-transit {
  description
  "This feature indicates that the server supports executing passive proof of transit OAM options. Servers that do not advertise this feature will not support executing proof-of-transit.

  Proof-of-transit feature is part of the passive-oam feature set, where the path or the service chain is verified. Service or path verification uses methods like nested hashing or nested encryption of the meta-data. By definition of a service chain or a path, some of the nodes in the domain participate and some do not."
}

/* Identities */
/* typedefs */
typedef routing-instance-ref {
  type leafref {
    path "/rt:routing/rt:routing-protocols/rt:routing-protocol/rt:name";
  }
  description
  "This type is used for leafs that reference a routing instance configuration."
}

typedef IPv4-Multicast-Group-Address {
  type string {
    pattern ’2((2[4-9])|(3[0-9]))\.’
    ’+(‘{[0-9]||1-9][0-9]|1[0-9][0-9]|0-9)’
    ’+’{0-4}’{0-9}’{25}’{0-5})’;
  }
  description
  "The IPv4-Multicast-Group-Address type represents an IPv4 multicast address in dotted-quad notation.";
  reference "RFC4607";
} // typedef IPv4-Multicast-Group-Address
typedef IPv6-Multicast-Group-Address {
  type string {
    pattern
 Pace
typedef IP-Multicast-Group-Address {
  type union {
    type IPv4-Multicast-Group-Address;
    type IPv6-Multicast-Group-Address;
  }
  description "The IP-Multicast-Group-Address type represents an IP multicast address and is IP version neutral. The format of the textual representations implies the IP version."
} // typedef IP-Multicast-Group-Address

identity fec-types {
  description "This is base identity of fec types which are ip-prefix, bgp, tunnel, l3vpn, pwe3, vpls, etc.";
}

typedef fec-type {
  type identityref {
    base fec-types;
  }
  description "Target FEC type."
}

typedef oam-counter32 {
  type yang:zero-based-counter32;
  description "defines 32 bit counter for OAM";
}
typedef passive-oam-mode {
    type enumeration {
        enum none {
            value 0;
            description "passive-oam mode not enabled.";
        }
        enum encap {
            value 1;
            description "passive-oam encap mode enabled.";
        }
        enum transit {
            value 2;
            description "passive-oam transit mode enabled.";
        }
        enum decap {
            value 3;
            description "passive-oam decap mode enabled.";
        }
    }
    description
        "OAM adds, updates or deletes data in-band based
         on a per-flow and per-node basis.";
}

typedef timestamp-accuracy {
    type enumeration {
        enum none {
            value 0;
            description "Do not collect Timestamp.";
        }
        enum seconds {
            value 1;
            description "Timestamp in seconds.";
        }
        enum milliseconds {
            value 2;
            description "Timestamp in milli seconds.";
        }
        enum microseconds {
            value 3;
            description "Timestamp in micro seconds.";
        }
        enum nanoseconds {
            value 4;
            description "Timestamp in nano seconds.";
        }
    }
}
typedef data-export-type {
    type enumeration {
        enum flexibleexport {
            value 1;
            description "raw format export";
        }
        enum ipfix {
            value 2;
            description "IPFIX formatted export";
        }
    }
}

typedef export-transport-type {
    type enumeration {
        enum tcp {
            value 6;
            description "TCP transport protocol.";
        }
        enum udp {
            value 17;
            description "UDP transport protocol.";
        }
    }
}

/* groupings */
grouping cc-session-statistics {
    description "Grouping for session statistics.";
    container cc-session-statistics {
        description "cc session counters";
        leaf session-count {
            type uint32;
            description "Number of cc sessions.";
        }
        leaf session-up-count {
            type uint32;
            description "Number of sessions which are up.";
        }
    }
}
leaf session-down-count {
  type uint32;
  description "Number of sessions which are down.";
}
leaf session-admin-down-count {
  type uint32;
  description "Number of sessions which are admin-down.";
}

grouping cc-per-session-statistics {
  description "Grouping for per session statistics";
container cc-per-session-statistics {
  description "per session statistics.";
  leaf create-time {
    type yang:date-and-time;
    description "Time and date when session is created.";
  }
  leaf last-down-time {
    type yang:date-and-time;
    description "Time and date last time session is down.";
  }
  leaf last-up-time {
    type yang:date-and-time;
    description "Time and date last time session is up.";
  }
  leaf down-count {
    type uint32;
    description "Total down count.";
  }
  leaf admin-down-count {
    type uint32;
    description "Total down count.";
  }
  leaf rx-packet-count {
    type uint32;
    description "Total received packet count.";
  }
  leaf tx-packet-count {
    type uint32;
    description "Total transmitted packet count.";
  }
  leaf rx-bad-packet {
    type uint32;
    description "Total received bad packet.";
  }
  leaf tx-packet-failed {
type uint32;
description "Total send packet failed.";
}
}

grouping session-error-statistics {
description "Grouping for per session error statistics";
container session-error-statistics {
description "Per session error statistics.";
leaf packet-drops-count {
type uint32;
description "Total received packet drops count.";
}
leaf packet-reorder-count {
type uint32;
description "Total received packet reordered count.";
}
leaf packets-out-of-seq-count {
type uint32;
description "Total received out of sequence count.";
}
leaf packets-dup-count {
type uint32;
description "Total received packet duplicates count.";
}
}

grouping session-delay-jitter-statistics {
description "Grouping for per session delay and jitter statistics";
container session-delay-jitter-statistics {
description "Session delay and jitter summarised information.";
leaf timestamp-res {
type timestamp-accuracy;
description "Timestamp accuracy among choice of s,ms,ns etc.";
}
leaf min-delay-value {
type uint32;
description "Minimum delay value observed.";
}
leaf max-delay-value {
type uint32;
description "Maximum delay value observed.";
}
leaf average-delay-value {
type uint32;
description "Average delay value observed.";
}
leaf min-jitter-value {
    type uint32;
    description "Minimum jitter value observed.";
}
leaf max-jitter-value {
    type uint32;
    description "Maximum jitter value observed.";
}
leaf average-jitter-value {
    type uint32;
    description "Average jitter value observed.";
}

grouping po-per-session-proof-of-transit-statistics {
    description "Grouping for per session proof of transit statistics";
    container po-per-session-proof-of-transit-statistics{
        description "OAM per session proof of transit statistics.";
        leaf proved-count {
            type uint32;
            description "Total number of packets the offered proof of transit.";
        }
        leaf failed-count {
            type uint32;
            description "Total number of packets failed to offer proof of transit.";
        }
    }
}

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

grouping oam-data-export-encoding-types {
  description
    "This specifies the types of data-export encoding types.";
  leaf data-export-encoding-type {
    type enumeration {
      enum json {
        description
          "The data encoding is done in the json format.";
        }
      enum protobuf {
        description
          "The data encoding is done as protobuf.";
        }
    }
  }
  description
    "Encoding type enum";
}


grouping oam-data-export-freq-config {
  description
    "Data export frequency configuration.";
  leaf sampling-frequency {
    type timestamp-accuracy;
    description
      "Data export sampling frequency accuracy.";
  }
  leaf sampling-interval {
    type uint64;
    description
      "If sampling-interval is set to 0, the data export becomes event
      based. In future revisions, the publish-subscribe model may be
      used."
  }
}

grouping oam-data-export-threshold-config {
  description
    "Data export threshold related configuration.";

  leaf data-export-trace-time-res {
    if-feature path-trace;
    type timestamp-accuracy;
    description
      "...";
  }
}
"Time accuracy among choice of s, ms, ns etc.";

leaf data-export-trace-delay-threshold {
    if-feature path-trace;
    type uint32;
    description "Threshold of number of packets suffering delay to trigger failure data export.";
}

leaf data-export-trace-jitter-threshold {
    if-feature path-trace;
    type uint32;
    description "Threshold of number of packets suffering jitter to trigger failure data export.";
}

leaf data-export-pot-failure-threshold {
    if-feature proof-of-transit;
    type uint32;
    description "Threshold of number of packets failing proof of transit method to trigger failure data export.";
}

leaf data-export-ppc-drops-failure-threshold {
    if-feature path-packet-counters;
    type uint32;
    description "Threshold of number of packets failing path packet counter drops or lost to trigger failure data export.";
}

leaf data-export-ppc-reorder-failure-threshold {
    if-feature path-packet-counters;
    type uint32;
    description "Threshold of number of packets failing path packet counter reorders to trigger failure data export.";
}

leaf data-export-ppc-oos-failure-threshold {
    if-feature path-packet-counters;
    type uint32;
    description "Threshold of number of packets failing path packet counter

out of sequence to trigger failure data export.

leaf data-export-ppc-dup-failure-threshold {
    if-feature path-packet-counters;
    type uint32;
    description
        "Threshold of number of packets failing path packet counter
duplicates to trigger failure data export."
}


grouping oam-data-export-config {
    description "OAM data export configuration details.";
    leaf oam-data-export-type {
        type data-export-type;
        description "Desired data export type."
    }
}

choice data-export-types {
    description "Data export configuration options.";
    case flexibleexport {
        when "oam-data-export-type = flexible-export";
        container flexible-export-params {
            description "Flexible data export configuration options.";
            container transport {
                description "Transport for exporting the data";
                leaf flex-data-export-protocol {
                    type export-transport-type;
                    description
                        "Internet Protocol number. TCP/UDP etc.";
                }
                leaf flex-data-export-ip {
                    type inet:ip-address;
                    description
                        "The server’s IP (v4 or v6) address for data export.";
                }
                leaf flex-data-export-port {
                    type inet:port-number;
                    description
                        "The server’s UDP or TCP port for data export.";
                }
            }
            container encoding {
                description "Encoding of the data exported."
                uses oam-data-export-encoding-types;
            }
        }
    }
}

Kumar, et al. Expires November 27, 2016 [Page 82]
case ipfix {
    when "oam-data-export-type = ip-fix";
    container ip-fix-params {
        description "IPFIX data export configuration options.";
        leaf ipfix-data-export-protocol {
            type export-transport-type;
            description "Internet Protocol number. TCP/UDP etc.,";
        }
        leaf ipfix-data-export-ip {
            type inet:ip-address;
            description "The ipfix server’s IP (v4 or v6) address for data export.";
        }
        leaf ipfix-data-export-port {
            type inet:port-number;
            description "The ipfix server’s UDP or TCP port for data export.";
        }
    }
}

grouping oam-data-export-profile {
    description "Data export profile configuration.";
    list data-export-profile {
        key "name";
        ordered-by system;
        description "Set of oam data export profiles that group parameters for data export";
        leaf name {
            type string {
                length "0..255";
            }
            mandatory true;
            description "";
        }
    }
}
typedef oam-transport-type-def {
  type enumeration {
    enum ipv4 {
      description "The transport type is ipv4.";
    }
    enum ipv6 {
      description "The transport type is ipv6.";
    }
    enum nsh {
      description "The transport type is nsh.";
    }
    enum vxlan-gpe {
      description "The transport type is programmable vxlan.";
    }
    enum gre {
      description "The transport type is GRE.";
    }
    enum mpls {
      description "The transport type is MPLS.";
    }
    enum other {
      description "The transport type is not defined.";
    }
  }
  description "Transport type enum.";
}

grouping oam-transport-encap-profile {
  description "Place holder for related transport encap configurations.";
  list transport-encap-profile {
    key "name";
    ordered-by system;
  }
}
description
  "Set of passive oam transport encap related profiles";
leaf name {
  type string {
    length "0..255";
  }
  mandatory true;
  description
    "Unique identifier for each profile";
}

leaf transport-type {
  type oam-transport-type-def;
  description
    "Transport type.";
}

grouping tp-flow-classifier {
  description
    "Classifier grouping which represents the ACL being applied. Going forward, attachment point and any relevant fields associated with that ACL can be added.";
list flow-classifier {
  key "name";
  description
    "A list that holds all classifiers";
  leaf name {
    type string;
    description
      "Classification name";
  }
}/***/ From ietf-access-control-list.yang /***/
leaf access-list {
  type string;
  description
    "The ACL name associated with this classifier";
}

uses oam-data-export-profile;
uses oam-transport-encap-profile;
}

grouping node-info {
  description "Node id and related interfaces information.";
  leaf node-id {

type uint32;
description
  "Node id assigned to this node";
}

list node-interfaces {
  key "index";
  ordered-by system;
  description
    "List of node's interfaces.";

  leaf index {
    type uint32;
    description
      "Index for the interfaces list";
  }

  leaf intf-name {
    type if:interface-ref;
    description
      "Instance of ietf-interfaces:interface-ref";
  }
}

grouping tp-address {
  choice tp-address {
    case mac-address {
      leaf mac-address {
        type yang:mac-address;
        description
          "MAC Address";
      }
      description
        "MAC Address based MP Addressing.";
    }
    case ipv4-address {
      leaf ipv4-address {
        type inet:ipv4-address;
        description
          "Ipv4 Address";
      }
      description
        "Ip Address based MP Addressing.";
    }
    case ipv6-address {
      leaf ipv6-address {
        type inet:ipv6-address;
    }
description
"Ipv6 Address";
}
description
"ipv6 Address based MP Addressing.";
} case src-dst-address {
  leaf src-ip-address {
    type inet:ip-address;
    description
    "source ip address.";
  }
  leaf dst-ip-address {
    type inet:ip-address;
    description
    "destination ip address.";
  }
  leaf Interface {
    type if:interface-ref;
    description
    "interface.";
  }
}
}
case fec {
  leaf fec-type {
    type fec-type;
    description
    "fec type.";
  }
  choice fec-value {
    description
    "fec value.";
    case ip-prefix {
      leaf ip-prefix {
        type inet:ip-prefix;
        description
        "ip prefix.";
      }
    }
    case bgp {
      leaf bgp {
        type inet:ip-prefix;
        description
        "BGP Labeled Prefix ";
      }
    }
    case tunnel {
      leaf tunnel-interface {

type uint32;
    description
    "VPN Prefix ";
  )
}
case l3vpn {
  leaf l3vpn-id {
    type uint32;
    description
    "FEC layer 3 vpn.";
  }
}
case pw {
  leaf remote-pe-address{
    type inet:ip-address;
    description
    "remote pe address.";
  }
  leaf pw-id {
    type uint32;
    description
    "Pseudowire id.";
  }
}
case vpls {
  leaf route-distinguisher {
    type uint32;
    description
    "Route Distinguisher(8 octets).";
  }
  leaf sender-ve-id{
    type uint32;
    description
    "Sender’s VE ID.";
  }
  leaf receiver-ve-id{
    type uint32;
    description
    "Receiver’s VE ID.";
  }
}
case mpls-mldp{
  choice root-address{
    description
    "root address choice.";
  case ip-address{
    leaf source-address{
      type inet:ip-address;
description
  "ip address.";
}
leaf group-ip-address{
type IP-Multicast-Group-Address;
description
  "group ip address.";
}
}
case vpn{
leaf as-number{
type inet:as-number;
description
  "AS number.";
}
}
case global-id{
leaf lsp-id{
type string;
description
  "lsp id.";
}
}
}
}
case tlv-address {
leaf tlv-type {
type int16;
description
  "Type of MEP-ID";
}
leaf tlv-len {
type int16;
description
  "Length of MEP-ID value";
}
leaf tlv-value {
type binary {
  length "12..255";
}
description
  "Value please refer RFC6428 (Figure 4,5,6).";
}
description
  "MEP-ID";
}
description
"TP Addressing.";
}
description
"TP Address";
}
grouping connectionless-oam-layers {
    list oam-layers {
        key "index";
        leaf index {
            type uint16 {
                range "0..65535";
            }
        }
        leaf level {
            type int32 {
                range "-1..1";
            }
        }
    }
}
uses tp-address;
ordered-by user;
description
"list of related oam layers.
0 means they are in same level, especially
interworking scenarios of stitching multiple
technology at same layer.
-1 means server layer, for eg:- in case of
Overlay and Underlay, Underlay is server layer for
Overlay Test Point.
+1 means client layer, for eg:- in case of
Service OAM and Transport OAM, Service OAM is client
layer to Transport OAM.";
}
description
"connectionless related OAM layer";
}
grouping tp-technology {
    choice technology {
        default technology-null;
        case technology-null {
            description
        }
    }
}

Kumar, et al. Expires November 27, 2016 [Page 90]
leaf tech-null {
    type empty;
    description
        "there is no technology define";
}

description
    "technology choice null";

case technology-string {
    description
        "oam technology string";
    leaf ipv4-icmp {
        type string;
        description
            "name to identify oam technology";
    }
}

description
    "OAM Technology";
}

grouping path-packet-counters-config {
    description "Passive OAM path packet counters sub-feature configuration.";

    container path-packet-counters-config {
        description "Container for config information.";
        /*** selects the data traffic flow for which the config is active ***/
        uses tp-flow-classifier;

        /*** selects the time range between which the config is active ***/
        uses pfs:timerange;

        /*** Assigns node-id and also selects which interfaces this config ***
         *** is active ***/  
        uses node-info;
    }

    leaf md-ppc-stats {
        type boolean;
        description
            "OAM path packet counter method for maintenance domain supported.";
    }

    leaf ppc-mode {
        type passive-oam-mode;
        description "Encap, intermediate or decap type of node.";
    }
}
leaf notification-ppc {
    type boolean;
    description
        "Enable notifications from decap MEP nodes to help aid triggered OAM";
}

leaf notification-ppc-drops-failure-threshold {
    type uint32;
    description
        "Threshold of number of packets failing path packet counter
drops or lost to trigger failure notifications.";
}

leaf notification-ppc-reorder-failure-threshold {
    type uint32;
    description
        "Threshold of number of packets failing path packet counter reorders
to trigger failure notifications.";
}

leaf notification-ppc-oos-failure-threshold {
    type uint32;
    description
        "Threshold of number of packets failing path packet counter
out of sequence to trigger failure notifications.";
}

leaf notification-ppc-dup-failure-threshold {
    type uint32;
    description
        "Threshold of number of packets failing path packet counter
duplicates to trigger failure notifications.";
}

grouping proof-of-transit-config {
    description "Passive OAM proof of transit sub-feature configuration.";
    container proof-of-transit-config {
        description "Container for config information.";
        /*** selects the data traffic flow for which the config is active ***/
        uses tp-flow-classifier;
        /*** selects the time range between which the config is active ***/
        uses pfs:timerange;
    }
}
/** Assigns node-id and also selects which interfaces this config ***/
/** is active ***/
uses node-info;
}

leaf md-proof-of-transit {
  type boolean;
  description
  "OAM proof of transit method for maintenance domain supported.";
}

leaf pot-mode {
  type passive-oam-mode;
  description "Encap, intermediate or decap type of node.";
}

leaf notification-pot {
  type boolean;
  description
  "Enable notifications from decap MEP nodes to help aid triggered OAM";
}

leaf notification-pot-failure-threshold {
  type uint32;
  description
  "Threshold of number of packets failing proof of transit method to tri
  gger
  failure notifications.";
}

grouping path-trace-config {
  description "Passive OAM path trace sub-feature configuration.";
  container path-trace-config {
    description "Container for config information.";
    /** select the data traffic flow for which the config is active ***/
    uses tp-flow-classifier;
    /** select the time range between which the config is active ***/
    uses pfs:timerange;
    /** Assigns node-id and also selects which interfaces this config ***/
    /** is active ***/
    uses node-info;
  }
  leaf md-tracing {
    type boolean;
  }
}
description
   "OAM tracing method for maintenance domain supported.";
}

leaf trace-mode {
    type passive-oam-mode;
    mandatory true;
    description "Encap, intermediate or decap type of node.";
}

leaf trace-length {
    type uint8;
    mandatory true;
    description "Length of the path trace to be collected, in terms of number of nodes.";
}

leaf trace-time-stamp {
    type timestamp-accuracy;
    mandatory true;
    description "This field depicts the delay domain of the trace path, whether it is to be collected, if yes units in seconds, milliseconds, microseconds or nanoseconds.";
}

leaf trace-app-data {
    type uint32;
    description "Application specific data to be added by the node";
}

leaf trace-include-interface {
    type boolean;
    mandatory true;
    description "Specifies to include/exclude interface information in the trace";
}

}

grouping tp-tools {
    description "Test Point OAM Toolset.";
    choice tools {
        default tools-empty;
description
"choice of test point tools.
Empty tools means based on Test Point it’s implicit
all OAM tools are present and no further configuration
is supported."

case tools-empty {
  description
  "this is a placeholder when oam toolset is not needed."
  leaf tools-null {
    type empty;
    description
    "there is no oam toolset defined.";
  }
}
case tools-ip{
  description
  "Oam Toolset for Ip"
  leaf rfc792 {
    type boolean;
    description
    "rfc792 (icmpv4) supported.";
  }
  leaf rfc4443 {
    type boolean;
    description
    "rfc4443 supported.";
  }
  leaf rfc4884 {
    type boolean;
    description
    "rfc4884 supported.";
  }
  leaf rfc5837 {
    type boolean;
    description
    "rfc5837 supported.";
  }
}
case tools-bfd {
  leaf rfc5881 {
    type boolean;
    description
    "rfc5881 supported.";
  }
  leaf rfc5883 {
    type boolean;
    description
    "rfc5883 supported.";
  }
}
leaf rfc5884 {
    type boolean;
    description
        "rfc5884 supported.";
}
leaf rfc5885 {
    type boolean;
    description
        "rfc5885 supported.";
}

case tools-mpls {
    description
        "Oam Toolset for mpls";
    leaf rfc4379 {
        type boolean;
        description
            "rfc4379 supported.";
    }
    leaf rfc4687 {
        type boolean;
        description
            "rfc4687 supported.";
    }
    leaf rfc4950 {
        type boolean;
        description
            "rfc4950 supported.";
    }
    leaf mpls-rfc5884 {
        type boolean;
        description
            "rfc5884 supported.";
    }
}

case tools-mpls-tp {
    description
        "Oam Toolset for mpls TP.";
    leaf rfc6426 {
        type boolean;
        description
            "rfc6426 supported.";
    }
    leaf rfc6435 {
        type boolean;
        description
            "rfc6435 supported.";
    }
case tools-pw {
    description
        "Oam Toolset for pw oam.";
leaf rfc5085 {
    type boolean;
    description
        "rfc5085 supported.";
}
leaf pw_rfc5885 {
    type boolean;
    description
        "rfc5885 supported.";
}
leaf rfc6423 {
    type boolean;
    description
        "rfc6423 supported.";
}
leaf rfc6310 {
    type boolean;
    description
        "rfc6310 supported.";
}
leaf rfc7023 {
    type boolean;
    description
        "rfc7023 supported.";
}
}

case tools-passive-oam {
    container passive-oam-config {
        if-feature path-packet-counters;
        if-feature path-trace;
        if-feature proof-of-transit;
        description "Passive OAM configuration for PPC, PoT and Trace options";
        uses path-packet-counters-config;
        uses proof-of-transit-config;
        uses path-trace-config;
    }
}
grouping test-point-location {
    leaf vrf {
        type routing-instance-ref;
        description
            "The vrf is used to describe the corresponding network instance";
    }
    uses tp-address;
    uses tp-technology;
    uses tp-tools;
    uses connectionless-oam-layers;
    description
        "Test point Address";
}

augment "/nd:networks/nd:network/nd:node"{
    description
        "Augment test points of connectionless oam.";
    container test-point-ipv4-location-list {
        if-feature connection-less;
        list test-point-locations {
            key "ipv4-location";
            leaf ipv4-location {
                type inet:ipv4-address;
                description
                    "Ipv4 Address.";
            }
            uses test-point-location;
            ordered-by user;
            description
                "list of test point locations.";
        }
        description
            "Serves as top-level container for test point location list.";
    }
}

augment "/nd:networks/nd:network/nd:node"{
    description
        "Augment test points of connectionless oam.";
    container test-point-ipv6-location-list {
        if-feature connection-less;
        list test-point-locations {
            key "ipv6-location";
            leaf ipv6-location {
                type inet:ipv6-address;
                description
            }
            uses test-point-location;
            ordered-by user;
            description
                "list of test point locations.";
        }
        description
            "Serves as top-level container for test point location list.";
    }
}
"Ipv6 Address."
)
uses test-point-location;
ordered-by user;
description
"list of test point locations."
}
description
"Serves as top-level container for test point location list.";
}

augment "/nd:networks/nd:network/nd:node"{

description
"Augment test points of passive oam."
container test-point-po-node-list {
  if-feature path-trace;
  if-feature path-packet-counters;
  if-feature proof-of-transit;
  list test-point-locations {
    key "node-id";
    ordered-by user;
    uses node-info;
    uses test-point-location;
    description
    "list of test point locations by nodeid."
  }
description
"Serves as top-level container for test point location list.";
}

augment "/nd:networks/nd:network/nd:node"{

description
"Augment test points of connectionless oam."
container test-point-tunnel-address-location-list {
  if-feature connection-less;
  list test-point-locations {
    key "tunnel-location";
    leaf tunnel-location {
      type uint32;
      description
      "VPN Prefix ";
    }
    uses test-point-location;
    ordered-by user;
    description
  }
}

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

augment "/nd:networks/nd:network/nd:node"{

description
"Augment test points of connectionless oam.";
container test-point-mac-address-location-list {
  if-feature connection-less;
  list test-point-locations {
    key "mac-address-location";
    leaf mac-address-location {
      type yang:mac-address;
      description
      "MAC Address";
    }
    uses test-point-location;
    ordered-by user;
    description
    "list of test point locations.";
  }
  description
  "Serves as top-level container for test point location list.";
}
}

augment "/nd:networks/nd:network/nd:node"{

description
"Augment test points of connectionless oam.";
container test-point-ip-prefix-location-list {
  if-feature connection-less;
  list test-point-locations {
    key "ip-prefix-location";
    leaf ip-prefix-location {
      type inet:ip-prefix;
      description
      "ip prefix.";
    }
    uses test-point-location;
    ordered-by user;
    description
    "list of test point locations.";
  }
  description
  "Serves as top-level container for test point location list.";
}
}
augment "/nd:Networks/nd:Network/nd:Node"{
  description
  "Augment test points of connectionless oam.";
  container test-point-route-dist-location-list {
    if-feature connection-less;
    list test-point-locations {
      key "route-dist-location";
      leaf route-dist-location {
        type uint32;
        description
        "Route Distinguisher(8 octets).";
      }
      uses test-point-location;
      ordered-by user;
      description
      "list of test point locations.";
    }
    description
    "Serves as top-level container for test point location list.";
  }
}

augment "/nd:Networks/nd:Network/nd:Node"{
  description
  "Augment test points of connectionless oam.";
  container test-point-group-ip-address-location-list {
    if-feature connection-less;
    list test-point-locations {
      key "group-ip-address-location";
      leaf group-ip-address-location {
        type IP-Multicast-Group-Address;
        description
        "group ip address.";
      }
      uses test-point-location;
      ordered-by user;
      description
      "list of test point locations.";
    }
    description
    "Serves as top-level container for test point location list.";
  }
}

augment "/nd:Networks/nd:Network/nd:Node"{
  description
  "Augment test points of connectionless oam.";
}
"Augment test points of connectionless oam."
container test-point-as-number-location-list {
  if-feature connection-less;
  list test-point-locations {
    key "as-number-location";
    leaf as-number-location {
      type inet:as-number;
      description
      "AS number.";
    }
    uses test-point-location;
    ordered-by user;
    description
    "list of test point locations.";
    description
    "Serves as top-level container for test point location list.";
  }
}

augment "/nd:nets/nd:network/nd:node"{
  description
  "Augment test points of connectionless oam.";
  container test-point-lsp-id-location-list {
    if-feature connection-less;
    list test-point-locations {
      key "lsp-id-location";
      leaf lsp-id-location{
        type string;
        description
        "lsp id.";
      }
    }
    uses test-point-location;
    ordered-by user;
    description
    "list of test point locations.";
    description
    "Serves as top-level container for test point location list.";
  }
}

container oper {
  config "false";
  description "session operational information.";
  container cc-ipv4-sessions-statistics {
    if-feature continuity-check;
description "cc ipv4 sessions";
uses cc-session-statistics;
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
type uint32;
default "5";

description
"Specifies the number of packets that will be sent.";
}
leaf vrf {
  type 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.";
  }
}
leaf tx-packt-count {
  type oam-counter32;
  description  "Transmitted Packet count";
}

leaf rx-packet-count {
  type oam-counter32;
  description  "Received packet count";
}

leaf min-delay {
  type oam-counter32;
  units milliseconds;
  description  "Delay is specified in milliseconds";
}

leaf average-delay {
  type oam-counter32;
  units milliseconds;
  description  "average delay in milliseconds";
}

leaf max-delay {
  type oam-counter32;
  units milliseconds;
  description  "Maximum delay in milliseconds";
}

rpc path-discovery {
  description  "Generates path discovery as per RFC7276.";
  input {
    container destination-tp {
      uses tp-address;
      description  "destination test point.";
    }
    uses session-type;
    leaf source-interface {
      type if:interface-ref;
leaf outbound-interface {
    type if:interface-ref;
    description "outbound interface.";
}
leaf vrf {
    type 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 ";
        }
        leaf hop-cnt {
            type uint8;
            description "hop count.";
        }
    }
}

container destination-tp {
    uses tp-address;
    description
    "destination test point.";
}
leaf min-delay {
    type oam-counter32;
    units milliseconds;
    description
    "Delay is specified in milliseconds";
}
leaf average-delay {
    type oam-counter32;
    units millisecond;
    description
    "average delay in milliseconds";
}
leaf max-delay {
    type oam-counter32;
    units millisecond;
    description
    "Maximum delay in milliseconds";
}
}

notification passive-oam-pot-triggered-oam {
    if-feature proof-of-transit;
    description
    "A failure notification based on a configured trigger has been detected";
    
    //*** Node-id and interfaces concerned ***/
    uses node-info;
    
    //*** flow information that triggered the failure ***/
    leaf access-list {
        type string;
        description
        "The ACL name associated with the flow.";
    }
    
    //*** Failure code ***/
    leaf failure-code {
        type uint32;
    }
}
description
    "Failure code from the devices that could provide more information as needed.";
}

/*** Number of packets failing the threshold ***/
leaf failed-count {
    type uint32;
    description "Total number of packets failed to offer proof of transit.";
}

notification passive-oam-ppc-triggered-oam {
    if-feature path-packet-counters;
    description
        "A failure notification based on a configured trigger has been detected";

   /*** Node-id and interfaces concerned ***/
    uses node-info;

   /*** flow information that triggered the failure ***/
    leaf access-list {
        type string;
        description
            "The ACL name associated with the flow.";
    }

   /*** Failure code ***/
    leaf failure-code {
        type uint32;
        description
            "Failure code from the devices that could provide more information as needed.";
    }

   /*** Number of packets failing the threshold ***/
    leaf failed-drop-count {
        type uint32;
        description "Total number of packets dropped or lost.";
    }

    leaf failed-reorder-count {
        type uint32;
        description "Total number of packets reordered.";
    }

    leaf failed-oos-count {
        type uint32;
    }
description "Total number of packets out of sequence.";
}
leaf failed-dup-count {
  type uint32;
  description "Total number of duplicate packets.";
}
}

YANG module of OAM

<CODE ENDS>

5. Security Considerations

TBD.

6. IANA Considerations

This document registers a URI in the IETF XML registry [RFC3688] [RFC3688]. Following the format in RFC 3688, the following registration is requested to be made:


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].

  prefix: goam reference: RFC XXXX

7. Normative References


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

Kumar, et al. Expires November 27, 2016 [Page 110]
Generic YANG Data Model for Connection Less Operations, Administration, and Maintenance (OAM) protocols
draft-kumar-lime-yang-connectionless-oam-05

Abstract

This document presents a base YANG Data model for connectionless OAM protocols. It provides a technology-independent abstraction of key OAM constructs for connectionless protocols. The Base model presented here can be extended to include technology specific details. This is leading to uniformity between OAM protocols and support nested OAM workflows (i.e., performing OAM functions at different or 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 January 20, 2017.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.
1. Introduction

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

1. Monitor networks connections (Reachability Verification, Continuity Check).
2. Troubleshoot failures (Fault verification and localization).
3. Monitor Performance

An overview of OAM tools is presented at [RFC7276].
Ping and Traceroute [RFC792], [RFC4443] are well-known fault verification and isolation tools, respectively, for IP networks. Over the years, different technologies have developed similar tools for similar purposes.

In this document, we present two modules, one to represent the base independent and stand-alone YANG data model for connectionless OAM protocols and the other one focuses on data retrieval procedures like RPCs. The split module approach avoids mixing the models for the retrieved-data from the retrieval procedures. It is expected that retrieval procedures would evolve faster than the data model and will allow new procedures to be defined for retrieval of the same data defined by the base data model. This also allows the data model to change at its own pace.

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

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
BFD - Bidirectional Forwarding Detection
TLV - Type Length Value
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
x for deprecated
o for obsolete

<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>.
3. Overview of the Connectionless OAM Model

At the top of the Model, there is an oper container for session statistics. Grouping is also defined for common session statistics and these are applicable for proactive OAM sessions. Multiple test-point-locations keyed using technology specific keys (e.g., IPv4 address for IPv4 locations) are possible by augmented network topology nodes. Each test-point-location is chosen based on location-type which when chosen, leads to a container that includes a list of test-point-locations keyed by technology specific keys. Each test point location includes a test-point-location-info. The test-point-location-info includes tp-technology, tp-tools, and connectionless-oam-layers. The groupings of tp-address and tp-address-vrf are kept out of test-point-location-info to make it addressing agnostic and allow varied composition. Depending upon the choice of the location-type (determined by the tp-address-vrf), the containers differ in its composition of test-point-locations while the test-point-location-info, is a common aspect of every test-point-location. The vrf is used to describe the corresponding network instance. The tp-technology indicate oam technology details. The tp-tools describe the oam tools supported. The connectionless-oam-layers is used to describe the relationship of one test point with other test points. The level in oam-layers indicate whether related oam test point is client layer, server layer or same or stiched layer. The Model is augmented to /nd:networks/nd:network/nd:node using Test Point Locations defined below.

3.1. TP Address

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

- MAC address
- IPv4 or IPv6 address
- a pair of source, destination addresses, and interface (Useful for BFD)
o FEC

o TLV address (RFC6428 (Figure 4, 5, and 6))

o System-id to represent the device or node.

3.2. Tools

In connectionless OAM, the tools attribute is used to describe a toolset for fault detection and isolation, and for performance measurement. And it can serve as a constraint condition when the base model be extended to specific OAM technology. For example, to fulfill the icmp ping configuration, the ".../coam:tools-ip/coam:rfc792" should be set to "true", and then the lime base model should be augmented with icmp ping specific details.

3.3. OAM-layers

As typical networks have a multi-layer architecture, the set of OAM protocols similarly take a multi-layer structure; each layer has its own OAM protocols [RFC7276] and is corresponding to specific network portion or pathand has associated test points. OAM-layers is referred to a list of upper layer, lower layer that are related to current test point. This allow users to easily navigate up and down to efficiently troubleshoot a connectivity issue at different layer. In this model, we have kept level default as 0, when all test points are located at the same layer. Level is provided for scenarios where it might be possible to define layering relationship as it can be used to stitching fault at related oam layers. For example, there is a network in which data traffic between two customer edges is transported over three consecutive network portions, the current test point is located in the second network portion. If there is a defect in the first network portion is located at the upstream of the second network portion, the level of the first network portion is set to "-1". If the third network portion is located at the downstream of the second network portion and the level is set to "1". In another case, if the first network portion and the third network portion is in the same level of thesecond network portion, the level is set to "0". The snippet below depicts an example of OAM layers.
list oam-layers {
  key "index";
  leaf index {
    type uint16 {
      range "0..65535";
    }
  }
  leaf level {
    type int32 {
      range "-1..1";
    }
    description
      "Level";
  }
  ordered-by user;
  description
    "list of related oam layers.";
}

3.4. Test Point Locations Information

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

3.5. Test Point Locations

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

3.6. Path discovery data

This is a generic grouping for path discovery data model that can be retrieved by any data retrieval methods including RPCs. Path discovery data output from methods, includes src-test-point, dst-test-point, sequence-number, hop-cnt, session statistics of various kinds, path verification and path trace related information. Path discovery includes data to be retrieved on a per-hop basis via a list of path-trace-info-list which includes information like timestamps, ingress-interface, egress-interface and app-meta-data. The path discovery data model is made generic enough to allow active, passive and hybrid OAMs to do the retrieval. None of the fields are made mandatory for that reason.
3.7. Continuity check data

This is a generic grouping for continuity check data model that can be retrieved by any data retrieval methods including RPCs. Continuity check data output from methods, includes src-test-point, dst-test-point, sequence-number, hop-cnt and session statistics of various kinds. The continuity check data model is made generic enough to allow active, passive and hybrid OAMs to do the retrieval. None of the fields are made mandatory for that reason.

3.8. RPC definitions

The rpc model described here as a separate model outside of the data 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, while the path-discovery rpc command corresponds to IP Traceroute.

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, lsp ping), 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.

```yaml
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
      "status code.";
  }
}

"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 {

3.9. Relation with other OAM YANG Model

In this document we define a generic YANG data model for connectionless OAM protocols. The other model defined adds generic data-retrieval methods. The YANG data model defined here is generic such that other technologies can extend it for technology specific needs. The Generic YANG model acts as the root for other OAM YANG models. This allows users to traverse between different OAM protocols at ease through a uniform API set. The Generic YANG model for OAM provides a framework where technology-specific YANG models can choose to inherit constructs from the base YANG models without needing to redefine them within the sub-technology.

3.10. OAM data hierarchy

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

module: ietf-connectionless-oam
  +--ro oper {continuity-check}?
+++ro cc-ipv4-sessions-statistics
  +++ro cc-session-statistics
   +++ro session-count?              uint32
   +++ro session-up-count?           uint32
   +++ro session-down-count?         uint32
   +++ro session-admin-down-count?   uint32
+++ro cc-ipv6-sessions-statistics
  +++ro cc-session-statistics
   +++ro session-count?              uint32
   +++ro session-up-count?           uint32
   +++ro session-down-count?         uint32
   +++ro session-admin-down-count?   uint32
augment /nd:networks/nd:network/nd:node:
  +++rw tp-address-type-value?       identityref
  +++rw (location-type)?
   ++:(ipv4-location-type)
    +++rw test-point-ipv4-location-list
     +++rw ipv4-location inet:ipv4-address
     +++rw vrf? routing-instance-ref
    +++rw (technology)?
     ++:(technology-null)
      ++rw tech-null? empty
    +++rw ipv4-icmp? string
   +++rw (tools)?
    ++:(tools-empty)
      ++rw tools-null? empty
    ++:(tools-ip)
      ++rw rfc792? boolean
      ++rw rfc4443? boolean
      ++rw rfc4884? boolean
      ++rw rfc5837? boolean
    ++:(tools-bfd)
      ++rw rfc5881? boolean
      ++rw rfc5883? boolean
      ++rw rfc5884? boolean
      ++rw rfc5885? boolean
    ++:(tools-mpls)
      ++rw rfc4339? boolean
      ++rw rfc4687? boolean
      ++rw rfc4950? boolean
      ++rw mpls-rfc5884? boolean
    ++:(tools-mpls-tp)
      ++rw rfc6426? boolean
      ++rw rfc6435? boolean
      ++rw rfc6374? boolean
    ++:(tools-pw)
typedef uint16 index

typedef inet:ipv6-address ipv6-location

typedef routing-instance-ref vrf

typedef boolean rfc5085?
typedef boolean pw_rfc5885?
typedef boolean rfc6423?
typedef boolean rfc6310?
typedef boolean rfc7023?
typedef boolean test-point-ipv6-location-
location-list
typedef inet:ipv6-address ipv6-location
typedef routing-instance-ref vrf
typedef boolean rfc5085?
typedef boolean pw_rfc5885?
typedef boolean rfc6423?
typedef boolean rfc6310?
typedef boolean rfc7023?

define leaf oam-layers* { index }
  rpc index uint16
  rpc level? int32
  define leaf (ipv6-location-type) {
    define leaf test-point-ipv6-location-list {
      define leaf ipv6-location inet:ipv6-address
      define leaf vrf? routing-instance-ref
      define leaf (technology)? {
        define leaf (technology-null) {
          define leaf tech-null? empty
        }
        define leaf (technology-string) {
          define leaf ipv4-icmp? string
        }
      }
      define leaf (tools)? {
        define leaf (tools-empty) {
          define leaf tools-null? empty
        }
        define leaf (tools-ip) {
          define leaf rfc792? boolean
          define leaf rfc4443? boolean
          define leaf rfc4884? boolean
          define leaf rfc5837? boolean
        }
        define leaf (tools-bfd) {
          define leaf rfc5881? boolean
          define leaf rfc5883? boolean
          define leaf rfc5884? boolean
          define leaf rfc5885? boolean
        }
        define leaf (tools-mpls) {
          define leaf rfc4379? boolean
          define leaf rfc4687? boolean
          define leaf rfc4950? boolean
          define leaf mpls-rfc5884? boolean
        }
        define leaf (tools-mpls-tp) {
          define leaf rfc6426? boolean
          define leaf rfc6435? boolean
          define leaf rfc6374? boolean
        }
        define leaf (tools-pw) {
          define leaf rfc5085? boolean
          define leaf pw_rfc5885? boolean
          define leaf rfc6423? boolean
          define leaf rfc6310? boolean
          define leaf rfc7023? boolean
        }
      }
  }

--rw level?  int32
---:(mac-location-type)
  --rw test-point-mac-address-location-list
    ---rw test-point-locations* [mac-address-location]
    ---rw mac-address-location  yang:mac-address
    ---rw (technology)?
      ---:(technology-null)
      |  ---rw tech-null?  empty
      ---:(technology-string)
      |  ---rw ipv4-icmp?  string
    ---rw (tools)?
      ---:(tools-empty)
      |  ---rw tools-null?  empty
      ---:(tools-ip)
      |  ---rw rfc792?  boolean
      |  ---rw rfc4443?  boolean
      |  ---rw rfc4884?  boolean
      |  ---rw rfc5837?  boolean
      ---:(tools-bfd)
      |  ---rw rfc5881?  boolean
      |  ---rw rfc5883?  boolean
      |  ---rw rfc5884?  boolean
      |  ---rw rfc5885?  boolean
      ---:(tools-mpls)
      |  ---rw rfc4379?  boolean
      |  ---rw rfc4687?  boolean
      |  ---rw rfc4950?  boolean
      |  ---rw mpls-rfc5884?  boolean
      ---:(tools-mpls-tp)
      |  ---rw rfc6426?  boolean
      |  ---rw rfc6435?  boolean
      |  ---rw rfc6374?  boolean
      ---:(tools-pw)
      |  ---rw rfc5085?  boolean
      |  ---rw pw_rfc5085?  boolean
      |  ---rw rfc6423?  boolean
      |  ---rw rfc6310?  boolean
      |  ---rw rfc7023?  boolean
    ---rw oam-layers* [index]
    |  ---rw index  uint16
    |  ---rw level?  int32
---:(tunnel-location-type)
  --rw test-point-tunnel-address-location-list
    ---rw test-point-locations* [tunnel-location]
    ---rw tunnel-location  uint32
    ---rw vrf?  routing-instance-ref
    ---rw (technology)?
      |  ---:(technology-null)
| +--:(tools-mpls-tp)                                      boolean
|    +--rw rfc6426?                                      boolean
|    +--rw rfc6435?                                      boolean
|    +--rw rfc6374?                                      boolean
| +--:(tools-pw)                                         boolean
|    +--rw rfc5085?                                      boolean
|    +--rw pw_rfc5885?                                   boolean
|    +--rw rfc6423?                                      boolean
|    +--rw rfc6310?                                      boolean
|    +--rw rfc7023?                                      boolean
| +--rw oam-layers* [index]                              boolean
|    +--rw index                                        uint16
|    +--rw level?                                       int32
+--:(group-as-number-location-type)                      boolean
+--rw test-point-as-number-location-list                 boolean
| +--rw test-point-locations* [as-number-location]       boolean
|    +--rw as-number-location                            inet:as-number
|    +--rw vrf?                                         routing-instance-ref
| +--rw (technology)?                                    boolean
|    +--:(technology-null)                               boolean
|        | +--rw tech-null?                                    empty
|    +--:(technology-string)                             boolean
|        | +--rw ipv4-icmp?                                    string
| +--rw (tools)?                                        boolean
|    +--:(tools-empty)                                   boolean
|        | +--rw tools-null?                                   empty
|    +--:(tools-ip)                                      boolean
|        | +--rw rfc792?                                       boolean
|        | +--rw rfc4443?                                      boolean
|        | +--rw rfc4884?                                      boolean
|        | +--rw rfc5837?                                      boolean
|    +--:(tools-bfd)                                     boolean
|        | +--rw rfc5881?                                      boolean
|        | +--rw rfc5883?                                      boolean
|        | +--rw rfc5884?                                      boolean
|        | +--rw rfc5885?                                      boolean
|    +--:(tools-mpls)                                    boolean
|        | +--rw rfc4379?                                      boolean
|        | +--rw rfc4687?                                      boolean
|        | +--rw rfc4950?                                      boolean
|        | +--rw mpls-rfc5884?                                 boolean
|    +--:(tools-mpls-tp)                                 boolean
|        | +--rw rfc6426?                                      boolean
|        | +--rw rfc6435?                                      boolean
|        | +--rw rfc6374?                                      boolean
|    +--:(tools-pw)                                      boolean
|        | +--rw rfc5085?                                      boolean
|        | +--rw pw_rfc5885?                                   boolean

| +--rw rfc6423?  | boolean      |
| +--rw rfc6310?  | boolean      |
| +--rw rfc7023?  | boolean      |
| +--rw oam-layers* [index] |
|   +--rw index   | uint16       |
|   +--rw level?  | int32        |
| +--:(group-lsp-id-location-type) |
|   +--rw test-point-lsp-id-location-list |
|     +--rw test-point-locations* [lsp-id-location] |
|       +--rw lsp-id-location | string        |
|       +--rw vrf? | routing-instance-ref |
|     +--rw (technology)? |
|       +--:(technology-null) |
|         | +--rw tech-null? | empty |
|       +--:(technology-string) |
|         | +--rw ipv4-icmp?  | string |
|     +--rw (tools)? |
|       +--:(tools-empty) |
|         | +--rw tools-null? | empty |
|       +--:(tools-ip) |
|         | +--rw rfc792?    | boolean     |
|         | +--rw rfc4443?   | boolean     |
|         | +--rw rfc4884?   | boolean     |
|         | +--rw rfc5837?   | boolean     |
|       +--:(tools-bfd) |
|         | +--rw rfc5881?   | boolean     |
|         | +--rw rfc5883?   | boolean     |
|         | +--rw rfc5884?   | boolean     |
|         | +--rw rfc5885?   | boolean     |
|       +--:(tools-mpls) |
|         | +--rw rfc4379?   | boolean     |
|         | +--rw rfc4687?   | boolean     |
|         | +--rw rfc4950?   | boolean     |
|         | +--rw mpls-rfc5884? | boolean   |
|       +--:(tools-mpls-tp) |
|         | +--rw rfc6426?   | boolean     |
|         | +--rw rfc6435?   | boolean     |
|         | +--rw rfc6374?   | boolean     |
|       +--:(tools-pw) |
|         | +--rw rfc5085?   | boolean     |
|         | +--rw pw_rfc5085? | boolean     |
|         | +--rw rfc6423?   | boolean     |
|         | +--rw rfc6310?   | boolean     |
|         | +--rw rfc7023?   | boolean     |
| +--rw oam-layers* [index] |
|   +--rw index   | uint16       |
|   +--rw level?  | int32        |
| +--:(group-system-id-location-type) |
module: ietf-connectionless-oam-methods
rpcs:
    +---x continuity-check {coam:continuity-check}?
        +---w input
            |     +---w destination-tp
            |         +---w (tp-address)?
|  +--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)?
        +--:mac-address)
        |  +--:mac-address?       yang:mac-address
        +--:(ipv4-address)
        |  +--:ipv4-address?      inet:ipv4-address
        +--:(ipv6-address)
        |  +--:ipv6-address?      inet:ipv6-address
        +--:(src-dst-address)
        |  +--:src-ip-address?    inet:ip-address
        |  +--:dst-ip-address?    inet:ip-address
        |  +--:Interface?         if:interface-ref
        +--:(fec)
        |  +--:fec-type?          fec-type
      +--ro (ip-prefix)
        +--:ip-prefix?          inet:ip-prefix
      +--:(bgp)
        +--:bgp?                inet:ip-prefix
      +--:(tunnel)
        +--:tunnel-interface?  uint32
      +--:(13vpn)
        +--:13vpn-id?           uint32
      +--:(pw)
        +--:remote-pe-address?  inet:ip-address
        +--:pw-id?              uint32
      +--:(vpls)
        +--:route-distinguisher? uint32
        +--:sender-ve-id?       uint32
        +--:receiver-ve-id?     uint32
      +--:(mpls-mldp)
        +--:root-address)?
          +--:ip-address)
          |  +--:source-address?    inet:ip-address
          |  +--:group-ip-address?  IP-Multicast-Group-Address
        +--:(vpn)
          |  +--:as-number?         inet:as-number
        +--:(global-id)
          +--: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)?
  +-:(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
   |     +-:(l3vpn)
   |        +--ro l3vpn-id? 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)
   |     +-:(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
+++w input
  +++w destination-tp
    +++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
+---:(l3vpn)
  | +---w l3vpn-id?  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)?
  |     | +---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 (ip-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
|       +--(l3vpn)
|         |   +--ro l3vpn-id?   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)?
|     +--(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)?       |
|   | | ++--ro (ip-prefix) |
|   | | | | ++--ro ip-prefix?      inet:ip-prefix |
|   | | | | ++--ro bgp?            inet:ip-prefix |
|   | | | | ++--ro tunnel?         |
|   | | | | | | ++--ro tunnel-interface? uint32 |
|   | | | | | | | | ++--ro (l3vpn) |
|   | | | | | | | | | | ++--ro l3vpn-id?   uint32 |
|   | | | | | | | | | | ++--ro (pw) |
|   | | | | | | | | | | | | ++--ro remote-pe-address? inet:ip-address |
|   | | | | | | | | | | | | ++--ro pw-id?   uint32 |
|   | | | | | | | | | | | | | | ++--ro (vpls) |
|   | | | | | | | | | | | | | | | | ++--ro route-distinguisher? uint32 |
|   | | | | | | | | | | | | | | | | ++--ro sender-ve-id?  uint32 |
|   | | | | | | | | | | | | | | | | ++--ro receiver-ve-id? uint32 |
|   | | | | | | | | | | | | | | | | | | | | ++--ro (mpls-mldp) |
|   | | | | | | | | | | | | | | | | | | | | | | ++--ro (root-address)? |
|   | | | | | | | | | | | | | | | | | | | | | |   | | ++--ro (ip-address) |
|   | | | | | | | | | | | | | | | | | | | | | |     | | ++--ro source-address?     inet:ip-address |
|   | | | | | | | | | | | | | | | | | | | | | |     | | ++--ro group-ip-address?  IP-Multicast-Group-Adress |
|   | | | | | | | | | | | | | | | | | | | | | |     +--ro as-number?     inet:as-number |
|   | | | | | | | | | | | | | | | | | | | | | |     | | ++--ro (global-id) |
|   | | | | | | | | | | | | | | | | | | | | | |     | |     | | ++--ro lsp-id?     string |
|   | | | | | | | | | | | | | | | | | | | | | |     ++--ro (tlv-address) |
|   | | | | | | | | | | | | | | | | | | | | | |     | | ++--ro tlv-type?      int16 |
|   | | | | | | | | | | | | | | | | | | | | | |     | | ++--ro tlv-len?        int16 |
|   | | | | | | | | | | | | | | | | | | | | | |     | | ++--ro tlv-value?      binary |
|   | | | | | | | | | | | | | | | | | | | | | |     ++--ro (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)?           
| | | ++--:(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
| | | | | ++--:(l3vpn)
| | | | | | ++--ro l3vpn-id?           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

4. OAM YANG Module

<CODE BEGINS> file "ietf-connectionless-oam.yang"

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

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

  organization "IETF LIME Working Group";
  contact
    "Deepak Kumar dekumar@cisco.com
    Qin Wu       bill.wu@huawei.com
    S Raghavan   srihari@cisco.com";
  description

Kumar, et al.           Expires January 20, 2017

Kumar, et al.           Expires January 20, 2017
"This YANG module defines the generic configuration, data model, statistics for connectionless OAM to be used within IETF in a protocol independent manner. Functional level abstraction is independent 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 2015-12-22 {
    description
    "Initial revision. - 01 version";
    reference "";
}

/* features */
feature connection-less {
    description
    "this feature indicates that OAM solution is connection less.";
}

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

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

/* Identities */
/* typedefs */
typedef routing-instance-ref {
    type leafref {
        path "/ni:network-instances/ni:network-instance/ni:name";
    }
    description
    "This type is used for leaves that reference a routing instance configuration.";
}
typedef IPv4-Multicast-Group-Address {
    type string {
        pattern '2((2[4-9])|(3[0-9]))\.'
        +'((0-9)|[1-9][0-9]|1[0-9][0-9]|' +'2[0-4][0-9]|25[0-5])\.(2)' +'((0-9)|[1-9][0-9]|1[0-9][0-9]|' +'2[0-4][0-9]|25[0-5])';
    }
    description
    "The IPv4-Multicast-Group-Address type
represents an IPv4 multicast address
in dotted-quad notation.";
    reference "RFC4607";
} // typedef IPv4-Multicast-Group-Address
typedef IPv6-Multicast-Group-Address {
    type string {
        pattern
        '(((FF|ff)\[0-9a-fA-F]\{2\})\{0-9a-fA-F\}' +'\{0,5\}(((0-9a-fA-F)\{0,4\})\?' +':((0-9a-fA-F)\{0,4\})|((25[0-5]\{2\})' +'}\{0-9\}|[01]?[0-9]?[0-9])\.\{3\}25[0-5]|' +'2[0-4]\{0-9\}|[01]?[0-9]?[0-9])\{0-9\})\'';
        pattern
        '([^:].+)\{6\}(([^:].+)[^:]^)+' +'\.(.*)\.(.*)|([^:]^)*[^:]^+'' +'?:(\[^:]^)\*[^:]^+'';
    }
    description
    "The IPv6-Multicast-Group-Address
type represents an IPv6 address in full,
mixed, shortened, and shortened-mixed
notation.";
    reference "RFC4291 2.7.
ietf-inet-types:ipv6-address";
} typedef IP-Multicast-Group-Address {
    type union {
        type IPv4-Multicast-Group-Address;
        type IPv6-Multicast-Group-Address;
    }
    description
    "The IP-Multicast-Group-Address type
represents an IP multicast address and
is IP version neutral. The format of the
textual representations implies the IP version.";
} // typedef IP-Multicast-Group-Address

identity fec-types {
typedef fec-type {
  type identityref {
    base fec-types;
  }
  description "Target FEC type.";
}

typedef oam-counter32 {
  type yang:zero-based-counter32;
  description "defines 32 bit counter for OAM";
}

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

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

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

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

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

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

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

/* groupings */
grouping cc-session-statistics {
    description "Grouping for session statistics.";
    container cc-session-statistics {
        description "cc session counters";
        leaf session-count {
            type uint32;
            description "Number of cc sessions.";
        }
        leaf session-up-count {
            type uint32;
            description "Number of sessions which are up.";
        }
        leaf session-down-count {
            type uint32;
            description "Number of sessions which are down.";
        }
        leaf session-admin-down-count {
            type uint32;
            description "Number of sessions which are admin-down.";
        }
    }
}

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

} 

} 

} 

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

} 

grouping session-error-statistics {
  description "Grouping for per session error statistics";
  container session-error-statistics {
    description "Per session error statistics.";
    leaf packet-drops-count {
      type uint32;
      description "Total received packet drops count.";
    }
    leaf packet-reorder-count {

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

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

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

leaf max-jitter-value {
  type uint32;
  description "Maximum jitter value observed.";
}

leaf average-jitter-value {
  type uint32;
  description "Average jitter value observed.";
}

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

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

"session type enum";
}

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

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

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

identity ipv6-address-type {
    base tp-address-type;
    description
        "IPv6 address type";
}

identity src-dst-address-type {
    base tp-address-type;
    description
        "Source/Dest address type";
}

identity fec-address-type {
    base tp-address-type;
    description
        "FEC address type";
}

identity tlv-address-type {
    base tp-address-type;
    description
        "TLV address type";
}

identity system-id-address-type {
    base tp-address-type;
    description

"System id address type";
}

identity lsp-id-address-type {
    base tp-address-type;
    description
        "LSP ID address type";
}

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

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

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

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

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

grouping tp-address {
    leaf tp-address-type-value {
        type identityref {
            base tp-address-type;
        }
        description "Test point address type.";
    }

    choice tp-address {

case mac-address {
  when "tp-address-type-value = mac-address-type" {
    description "MAC address type";
  }
  leaf mac-address {
    type yang:mac-address;
    description "MAC Address";
  }
  description "MAC Address based MP Addressing.";
}
case ipv4-address {
  when "tp-address-type-value = ipv4-address-type" {
    description "IPv4 address type";
  }
  leaf ipv4-address {
    type inet:ipv4-address;
    description "Ipv4 Address";
  }
  description "Ip Address based MP Addressing.";
}
case ipv6-address {
  when "tp-address-type-value = ipv6-address-type" {
    description "IPv6 address type";
  }
  leaf ipv6-address {
    type inet:ipv6-address;
    description "Ipv6 Address";
  }
  description "ipv6 Address based MP Addressing.";
}
case src-dst-address {
  when "tp-address-type-value = src-dst-address-type" {
    description "Src dest address type for BFD";
  }
  leaf src-ip-address {
    type inet:ip-address;
    description "source ip address.";
  }
  leaf dst-ip-address {
    type inet:ip-address;
    description
"destination ip address."
}
leaf Interface {
    type if:interface-ref;
    description
        "interface.";
}
}

case fec {
    when "tp-address-type-value = fec-address-type" {
        description "FEC address type";
    }
    leaf fec-type {
        type fec-type;
        description
            "fec type.";
    }
    choice fec-value {
        description
            "fec value.";
        case ip-prefix {
            leaf ip-prefix {
                type inet:ip-prefix;
                description
                    "ip prefix.";
            }
        }
        case bgp {
            leaf bgp {
                type inet:ip-prefix;
                description
                    "BGP Labeled Prefix ";
            }
        }
        case tunnel {
            leaf tunnel-interface {
                type uint32;
                description
                    "VPN Prefix ";
            }
        }
        case l3vpn {
            leaf l3vpn-id {
                type uint32;
                description
                    "FEC layer 3 vpn.";
            }
        }
    }
}
case pw {
  leaf remote-pe-address{
    type inet:ip-address;
    description
    "remote pe address.";
  }
  leaf pw-id {
    type uint32;
    description
    "Pseudowire id.";
  }
}

case vpls {
  leaf route-distinguisher {
    type uint32;
    description
    "Route Distinguisher(8 octets).";
  }
  leaf sender-ve-id{
    type uint32;
    description
    "Sender’s VE ID.";
  }
  leaf receiver-ve-id{
    type uint32;
    description
    "Receiver’s VE ID.";
  }
}

case mpls-mldp{
  choice root-address{
    description
    "root address choice.";
  }
  case ip-address{
    leaf source-address{
      type inet:ip-address;
      description
      "ip address.";
    }
    leaf group-ip-address{
      type IP-Multicast-Group-Address;
      description
      "group ip address.";
    }
  }
  case vpn{
    leaf as-number{
      type inet:as-number;
description
"AS number.";
}
}
case global-id{
  leaf lsp-id{
    type string;
    description
    "lsp id.";
  }
}
}
case tlv-address {
  when "tp-address-type-value = tlv-address-type" {
    description "TLV address type";
  }
  leaf tlv-type {
    type int16;
    description
    "Type of MEP-ID";
  }
  leaf tlv-len {
    type int16;
    description
    "Length of MEP-ID value";
  }
  leaf tlv-value {
    type binary {
      length "12..255";
    }
    description
    "Value please refer RFC6428 (Figure 4,5,6).";
  }
  description
  "MEP-ID";
}
case system-info {
  when "tp-address-type-value = system-id-address-type" {
    description "System id address type";
  }
  leaf system-id {
    type inet:uri;
    description
    "System ID assigned to this node.";
  }
}
grouping tp-address-vrf {
  description
  "Test point address with VRF.";
  leaf vrf {
    type routing-instance-ref;
    description
    "The vrf is used to describe the corresponding network instance";
  }
}

uses tp-address;
}

grouping connectionless-oam-layers {
  list oam-layers {
    key "index";
    leaf index {
      type uint16 {
        range "0..65535";
      }
      description
      "Index";
    }
    leaf level {
      type int32 {
        range "-1..1";
        default 0;
        description
        "Level 0 indicates default level, -1 means server and +1 means client layer. In relationship 0 means same layer.";
      }
      ordered-by user;
      description
      "List of related OAM layers. 0 means they are in same level, especially interworking scenarios of sticking multiple technology at same layer. -1 means server layer, for eg:- in case of
Overlay and Underlay, Underlay is server layer for Overlay Test Point.
+1 means client layer, for eg:- in case of Service OAM and Transport OAM, Service OAM is client layer to Transport OAM.

```yaml
grouping tp-technology {
  choice technology {
    default technology-null;
    case technology-null {
      description "this is a placeholder when no technology is needed.";
      leaf tech-null {
        type empty;
        description "there is no technology define";
      }
    }
    description "technology choice null";
    case technology-string {
      description "oam technology string";
      leaf ipv4-icmp {
        type string;
        description "name to identify oam technology";
      }
    }
  }
  description "OAM Technology";
}

grouping tp-tools {
  description "Test Point OAM Toolset.";
  choice tools {
    default tools-empty;
    description "choice of test point tools. Empty tools means based on Test Point it’s implicit all OAM tools are present and no further configuration";
  }
}
```
is supported.");
case tools-empty {
  description
  "this is a placeholder when oam toolset is not needed.";
  leaf tools-null {
    type empty;
    description
    "there is no oam toolset defined.";
  }
}
case tools-ip{
  description
  "Oam Toolset for Ip";
  leaf rfc792 {
    type boolean;
    description
    "rfc792 (icmpv4) supported.";
  }
  leaf rfc4443 {
    type boolean;
    description
    "rfc4443 supported.";
  }
  leaf rfc4884 {
    type boolean;
    description
    "rfc4884 supported.";
  }
  leaf rfc5837 {
    type boolean;
    description
    "rfc5837 supported.";
  }
}
case tools-bfd {
  leaf rfc5881 {
    type boolean;
    description
    "rfc5881 supported.";
  }
  leaf rfc5883 {
    type boolean;
    description
    "rfc5883 supported.";
  }
  leaf rfc5884 {
    type boolean;
    description
  }
case tools-mpls {
    description "Oam Toolset for mpls";
    leaf rfc4379 {
        type boolean;
        description "rfc4379 supported.";
    }
    leaf rfc4687 {
        type boolean;
        description "rfc4687 supported.";
    }
    leaf rfc4950 {
        type boolean;
        description "rfc4950 supported.";
    }
    leaf mpls-rfc5884 {
        type boolean;
        description "rfc5884 supported.";
    }
}
case tools-mpls-tp {
    description "Oam Toolset for mpls TP.";
    leaf rfc6426 {
        type boolean;
        description "rfc6426 supported.";
    }
    leaf rfc6435 {
        type boolean;
        description "rfc6435 supported.";
    }
    leaf rfc6374 {
        type boolean;
        description "rfc6374 supported.";
    }
}
"rfc6374 supported."
)

case tools-pw {
  description
    "Oam Toolset for pw oam.";
  leaf rfc5085 {
    type boolean;
    description
      "rfc5085 supported.";
  }
  leaf pw_rfc5885 {
    type boolean;
    description
      "rfc5885 supported.";
  }
  leaf rfc6423 {
    type boolean;
    description
      "rfc6423 supported.";
  }
  leaf rfc6310 {
    type boolean;
    description
      "rfc6310 supported.";
  }
  leaf rfc7023 {
    type boolean;
    description
      "rfc7023 supported.";
  }
}


grouping test-point-location-info {
  uses tp-technology;
  uses tp-tools;
  uses connectionless-oam-layers;
  description
    "Test point Location";
}


grouping test-point-locations {
  description "Group of test point locations.";
  leaf tp-address-type-value {
type identityref {
  base tp-address-type;
} description "Test point address type.";
}
choice location-type {
  case ipv4-location-type {
    when "tp-address-type-value = ipv4-address-type" {
      description "when test point address is equal to ipv4 address.";
    }
    container test-point-ipv4-location-list {
      list test-point-locations {
        key "ipv4-location";
        leaf ipv4-location {
          type inet:ipv4-address;
          description "Ipv4 Address.";
        }
        leaf vrf {
          type routing-instance-ref;
          description "The vrf is used to describe the corresponding network instance";
        }
        uses test-point-location-info;
        ordered-by user;
        description "list of test point locations.";
      }
      description "Serves as top-level container for test point location list.";
    }
  }
  case ipv6-location-type {
    when "tp-address-type-value = ipv6-address-type" {
      description "when test point address is equal to ipv6 address";
    }
    container test-point-ipv6-location-list {
      list test-point-locations {
        key "ipv6-location";
        leaf ipv6-location {
          type inet:ipv6-address;
          description "Ipv6 Address.";
        }
        leaf vrf {
          type routing-instance-ref;
          description "The vrf is used to describe the corresponding network instance";
        }
        uses test-point-location-info;
        ordered-by user;
        description "list of test point locations.";
      }
      description "Serves as top-level container for test point location list.";
    }
  }
}
type routing-instance-ref;
  description
    "The vrf is used to describe the
    corresponding network instance";
}
uses test-point-location-info;
ordered-by user;
description
  "list of test point locations.";
}
description
  "Serves as top-level container for test point location list.";
}
}
case mac-location-type {
  when "tp-address-type-value = mac-address-type" {
    description
      "when test point address is equal to mac address.";
  }
}
container test-point-mac-address-location-list {
  list test-point-locations {
    key "mac-address-location";
    leaf mac-address-location {
      type yang:mac-address;
      description
        "MAC Address";
    }
    uses test-point-location-info;
    ordered-by user;
    description
      "list of test point locations.";
  }
  description
    "Serves as top-level container for test point location list.";
}
}
case tunnel-location-type {
  when "tp-address-type-value = tunnel-address-type" {
    description
      "when test point address is equal to tunnel type.";
  }
}
container test-point-tunnel-address-location-list {
  list test-point-locations {
    key "tunnel-location";
    leaf tunnel-location {
      type uint32;
      description
        "VPN Prefix";
leaf vrf {
  type routing-instance-ref;
  description "The vrf is used to describe the corresponding network instance";
}
uses test-point-location-info;
ordered-by user;
description "list of test point locations.";
}
description "Serves as top-level container for test point location list.";
}
case ip-prefix-location-type {
  when "tp-address-type-value = ip-prefix-address-type" {
    description "when test point address is equal to ip prefix.";
  }
  container test-point-ip-prefix-location-list {
    list test-point-locations {
      key "ip-prefix-location";
      leaf ip-prefix-location {
        type inet:ip-prefix;
        description "IP Prefix";
      }
    }
    leaf vrf {
      type routing-instance-ref;
      description "The vrf is used to describe the corresponding network instance";
    }
    uses test-point-location-info;
    ordered-by user;
    description "list of test point locations.";
  }
description "Serves as top-level container for test point location list.";
}
}
case route-distinguisher-location-type {
  when "tp-address-type-value = route-distinguisher-address-type" {
    description "when test point address is equal to route distinguisher.";
  }
}
container test-point-route-dist-location-list {
  list test-point-locations {
    key "route-dist-location";
    leaf route-dist-location {
      type uint32;
      description
      "Route Distinguisher(8 octets).";
    }
    leaf vrf {
      type routing-instance-ref;
      description
      "The vrf is used to describe the corresponding network instance";
    }
    uses test-point-location-info;
    ordered-by user;
    description
    "list of test point locations.";
    description
    "Serves as top-level container for test point location list.";
  }
}

case group-ip-address-location-type {
  when "tp-address-type-value = group-ip-address-type" {
    description "when test point address is equal to group ip address.";
  }
}

container test-point-group-ip-address-location-list {
  list test-point-locations {
    key "group-ip-address-location";
    leaf group-ip-address-location {
      type IP-Multicast-Group-Address;
      description
      "Group IP address.";
    }
    leaf vrf {
      type routing-instance-ref;
      description
      "The vrf is used to describe the corresponding network instance";
    }
    uses test-point-location-info;
    ordered-by user;
    description
    "list of test point locations.";
  }
}
case group-as-number-location-type {
  when "tp-address-type-value = as-number-address-type" {
    description "when test point address is equal to
                as-number.";
  }
}
case group-lsp-id-location-type {
  when "tp-address-type-value = lsp-id-address-type" {
    description "when test point address is equal to lspid.";
  }
}
container test-point-as-number-location-list {
  list test-point-locations {
    key "as-number-location";
    leaf as-number-location {
      type inet:as-number;
      description "AS number.";
    }
    leaf vrf {
      type routing-instance-ref;
      description "The vrf is used to describe the
                  corresponding network instance";
    }
  }
}
container test-point-lsp-id-location-list {
  list test-point-locations {
    key "lsp-id-location";
    leaf lsp-id-location {
      type string;
      description "LSP Id.";
    }
    leaf vrf {
      type routing-instance-ref;
      description "The vrf is used to describe the
corresponding network instance";
}
uses test-point-location-info;
ordered-by user;
description
"list of test point locations.";
}
description
"Serves as top-level container for test point location list.";
}
}
case group-system-id-location-type {
  when "tp-address-type-value = system-id-address-type" {
    description "when test point address is equal to
    system info.";
  }
}
container test-point-system-info-location-list {
  list test-point-locations {
    key "system-id-location";
    leaf system-id-location {
      type inet:uri;
      description
        "System Id.";
    }
    leaf vrf {
      type routing-instance-ref;
      description
        "The vrf is used to describe the
        corresponding network instance";
    }
  }
uses test-point-location-info;
ordered-by user;
description
"list of test point locations.";
}
description
"Serves as top-level container for test point location list.";
}

description
"Choice of address types.";
}

augment "/nd:networks/nd:network/nd:node"{
  description
    "Augment test points of connectionless oam.";
  uses test-point-locations;
grouping path-discovery-data {
    description "Path discovery related data output from nodes.";
    container src-test-point {
        description "Source test point.";
        uses tp-address-vrf;
    }
    container dest-test-point {
        description "Destination test point.";
        uses tp-address-vrf;
    }
    leaf sequence-number {
        type uint64;
        description "Sequence number in data packets.";
    }
    leaf hop-cnt {
        type uint8;
        description "hop count.";
    }
    uses session-packet-statistics;
    uses session-error-statistics;
    uses session-delay-statistics;
    uses session-jitter-statistics;
}

container path-verification {
    description "Optional path verification related information.";
    leaf flow-info {
        type string;
        description "ACL name that refers to the flow, if any.";
    }
    uses session-path-verification-statistics;
}

container path-trace-info {
    description "Optional path trace per-hop test point information. The list has typically a single element for per-hop cases like path-discovery RPC but allows a list of hop related information for other types of data retrieval methods.";
    list path-trace-info-list {
        key "index";
        description "Path trace information list.";
        leaf index {
            type uint32;
        }
    }
}
description "Trace information index.";
}

uses tp-address-vrf;

leaf timestamp-val {
  type yang:date-and-time;
  description "Timestamp value";
}

leaf ingress-intf-name {
  type if:interface-ref;
  description "Ingress interface name";
}

leaf egress-intf-name {
  type if:interface-ref;
  description "Egress interface name";
}

leaf app-meta-data {
  type uint32;
  description "Application specific data added by node.";
}
}
}

grouping continuity-check-data {
  description "Continuity check data output from nodes.";
  container src-test-point {
    description "Source test point.";
    uses tp-address-vrf;

    leaf egress-intf-name {
      type if:interface-ref;
      description "Egress interface name";
    }
  }
  container dest-test-point {
    description "Destination test point.";
    uses tp-address-vrf;

    leaf ingress-intf-name {
      type if:interface-ref;
      description "Ingress interface name";
    }
  }
}
leaf sequence-number {
    type uint64;
    description "Sequence number.";
}
leaf hop-cnt {
    type uint8;
    description "hop count.";
}

uses session-packet-statistics;
uses session-error-statistics;
uses session-delay-statistics;
uses session-jitter-statistics;
}

container oper {
    if-feature continuity-check;
    config "false";
    description "cc operational information.";
    container cc-ipv4-sessions-statistics {
        description "cc ipv4 sessions";
        uses cc-session-statsitics;
    }
    container cc-ipv6-sessions-statistics {
        description "cc ipv6 sessions";
        uses cc-session-statsitics;
    }
}

module 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";
}
This YANG module defines the RPCs for connectionless OAM to be used within IETF in a protocol Independent manner. Functional level abstraction is independent 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 2015-12-22 {
  description
    "Initial revision. - 01 version";
  reference ""
}

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;
5. Security Considerations

TBD.

6. IANA Considerations

This document registers a URI in the IETF XML registry [RFC3688] [RFC3688]. Following the format in RFC 3688, the following registration is requested to be made:


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].

prefix: goam reference: RFC XXXX

7. Normative References


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