

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: November 27, 2016

D. Kumar
Cisco
M. Wang
Q. Wu
Huawei
R. Rahman
Cisco
May 26, 2016

Generic YANG Data Model for Connection Less Operations, Administration,
and Maintenance(OAM) protocols
draft-kumar-lime-yang-connectionless-oam-03

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

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

Table of Contents

1. Introduction	2
2. Conventions used in this document	3
2.1. Terminology	3
2.2. Tree Diagrams	4
3. Overview of the Connectionless OAM Model	5
3.1. TP address	5
3.2. Tools	5
3.3. Passive OAM	6
3.3.1. Path Trace	6
3.3.2. Path Packet Counters	6
3.3.3. Proof of Transit	6
3.4. OAM-layers	7
3.5. rpc definitions	7
3.6. Relation with other OAM YANG Model	11
3.7. Notifications	12
3.8. Proof of transit notification	12
3.9. Path packet counter notification	12
3.10. OAM data hierarchy	12
4. OAM YANG Module	70
5. Security Considerations	109
6. IANA Considerations	109
7. Normative References	109
Authors' Addresses	110

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 presents 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:

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

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

- o augment
- o data model
- o data node

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

2.1. Terminology

TP - Test Point

MAC - Media Access Control

BFD - Bidirectional Forwarding Detection

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>.

<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 stiched 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 attriburtes associated with test-point.

3.1. TP address

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

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

```
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-packet-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 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";
    }
  }
}
```

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 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
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
ss    | | | |
    | | | | +---:(vpn)

```


ta-export-type					+++rw oam-data-export-type?	da
					+++rw (data-export-types)?	
					+++:(flexibleexport)	
					+++rw flexible-export-params	
					+++rw transport	
-transport-type					+++rw flex-data-export-protocol?	export
p-address					+++rw flex-data-export-ip?	inet:i
ort-number					+++rw flex-data-export-port?	inet:p
					+++rw encoding	
ation					+++rw data-export-encoding-type?	enumer
					+++:(ipfix)	
					+++rw ip-fix-params	
ransport-type					+++rw ipfix-data-export-protocol?	export-t
address					+++rw ipfix-data-export-ip?	inet:ip-
t-number					+++rw ipfix-data-export-port?	inet:por
mestamp-accuracy					+++rw sampling-frequency?	ti
nt64					+++rw sampling-interval?	ui
mestamp-accuracy {path-trace}?					+++rw data-export-trace-time-res?	ti
nt32 {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
					+++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
m-mode					+++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	
					+++rw flow-classifier* [name]	
					+++rw name string	
					+++rw access-list? string	
					+++rw data-export-profile* [name]	

ring				+++rw name	st
ta-export-type				+++rw oam-data-export-type?	da
				+++rw (data-export-types)?	
				+++:(flexibleexport)	

						+--rw flexible-export-params	
						+--rw transport	
-transport-type						+--rw flex-data-export-protocol?	export
p-address						+--rw flex-data-export-ip?	inet:i
ort-number						+--rw flex-data-export-port?	inet:p
						+--rw encoding	
ation						+--rw data-export-encoding-type?	enumer
						+--:(ipfix)	
						+--rw ip-fix-params	
ransport-type						+--rw ipfix-data-export-protocol?	export-t
address						+--rw ipfix-data-export-ip?	inet:ip-
t-number						+--rw ipfix-data-export-port?	inet:por
mestamp-accuracy						+--rw sampling-frequency?	ti
nt64						+--rw sampling-interval?	ui
mestamp-accuracy {path-trace}?						+--rw data-export-trace-time-res?	ti
nt32 {path-trace}?						+--rw data-export-trace-delay-threshold?	ui
nt32 {path-trace}?						+--rw data-export-trace-jitter-threshold?	ui
nt32 {proof-of-transit}?						+--rw data-export-pot-failure-threshold?	ui
nt32 {path-packet-counters}?						+--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
						+--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	
m-mode						+--rw md-proof-of-transit?	boolean
						+--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]	
ring						+--rw name	st
ta-export-type						+--rw oam-data-export-type?	da
						+--rw (data-export-types)?	
						+--:(flexibleexport)	
						+--rw flexible-export-params	

						+-rw transport	
-transport-type						+-rw flex-data-export-protocol?	export
p-address						+-rw flex-data-export-ip?	inet:i
ort-number						+-rw flex-data-export-port?	inet:p
						+-rw encoding	

ation						+++rw data-export-encoding-type?	enumer
						+++:(ipfix)	
						+++rw ip-fix-params	
ransport-type						+++rw ipfix-data-export-protocol?	export-t
address						+++rw ipfix-data-export-ip?	inet:ip-
t-number						+++rw ipfix-data-export-port?	inet:por
mestamp-accuracy						+++rw sampling-frequency?	ti
nt64						+++rw sampling-interval?	ui
mestamp-accuracy {path-trace}?						+++rw data-export-trace-time-res?	ti
nt32 {path-trace}?						+++rw data-export-trace-delay-threshold?	ui
nt32 {path-trace}?						+++rw data-export-trace-jitter-threshold?	ui
nt32 {proof-of-transit}?						+++rw data-export-pot-failure-threshold?	ui
nt32 {path-packet-counters}?						+++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
						+++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
m-mode						+++rw trace-mode	passive-oa
						+++rw trace-length	uint8
accuracy						+++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
--	--	-----------------	----------------

Kumar, et al.

Expires November 27, 2016

[Page 17]

```

dress
|
|   +---:(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-Ad
|   +---:(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-ipv6-location-list {connection-less}?
+---rw test-point-locations* [ipv6-location]
|   +---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
|   |   +---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
ss
| | | | +---:(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 {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                st
ring
    |   |   |   |   |   +---rw oam-data-export-type?        da
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
p-address
    |   |   |   |   |   |   |   |   |   |   |   |   +---rw flex-data-export-port?    inet:p
ort-number
    |   |   |   |   |   |   |   |   |   |   |   |   |   +---rw encoding
    |   |   |   |   |   |   |   |   |   |   |   |   |   |   +---rw data-export-encoding-type?    enumer
ation
    |   |   |   |   |   |   |   |   |   |   |   |   |   |   +---:(ipfix)
    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   +---rw ip-fix-params
    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   +---rw ipfix-data-export-protocol?    export-t
ransport-type
    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   +---rw ipfix-data-export-ip?        inet:ip-
address
    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   +---rw ipfix-data-export-port?    inet:por
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	
m-mode		+--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		
		+--rw flow-classifier* [name]		
		+--rw name	string	
		+--rw access-list?	string	
		+--rw data-export-profile* [name]		
ring		+--rw name		st
		+--rw oam-data-export-type?		da
ta-export-type		+--rw (data-export-types)?		
		+-:(flexibleexport)		
		+--rw flexible-export-params		
		+--rw transport		
-transport-type		+--rw flex-data-export-protocol?	export	
p-address		+--rw flex-data-export-ip?	inet:i	
ort-number		+--rw flex-data-export-port?	inet:p	
		+--rw encoding		
ation		+--rw data-export-encoding-type?	enumer	
		+-:(ipfix)		
		+--rw ip-fix-params		
ransport-type		+--rw ipfix-data-export-protocol?	export-t	
address		+--rw ipfix-data-export-ip?	inet:ip-	
t-number		+--rw ipfix-data-export-port?	inet:por	
mestamp-accuracy		+--rw sampling-frequency?	ti	
nt64		+--rw sampling-interval?	ui	
mestamp-accuracy {path-trace}?		+--rw data-export-trace-time-res?	ti	
nt32 {path-trace}?		+--rw data-export-trace-delay-threshold?	ui	
nt32 {path-trace}?		+--rw data-export-trace-jitter-threshold?	ui	
nt32 {proof-of-transit}?		+--rw data-export-pot-failure-threshold?	ui	
nt32 {path-packet-counters}?		+--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	
		+--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	st
ring			
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			
p-address		+++rw flex-data-export-ip?	inet:i
ort-number		+++rw flex-data-export-port?	inet:p
		+++rw encoding	
ation		+++rw data-export-encoding-type?	enumer
		+++:(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 ipfix-data-export-port?	inet:por
		+++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?	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-Ad
  |   |   +--:(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-trans
it}?
    +--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)?
|   +--:(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
|   |   +--:(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)

```

ss

```

| | +--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-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                  st
ring
| |         +--rw oam-data-export-type? da
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
p-address
| |                 +--rw flex-data-export-port?    inet:p
ort-number
| |                 +--rw encoding

```

ation						+++rw data-export-encoding-type?	enumer
						+++:(ipfix)	
						+++rw ip-fix-params	
ransport-type						+++rw ipfix-data-export-protocol?	export-t
address						+++rw ipfix-data-export-ip?	inet:ip-
t-number						+++rw ipfix-data-export-port?	inet:por
mestamp-accuracy						+++rw sampling-frequency?	ti
nt64						+++rw sampling-interval?	ui
mestamp-accuracy {path-trace}?						+++rw data-export-trace-time-res?	ti
nt32 {path-trace}?						+++rw data-export-trace-delay-threshold?	ui
nt32 {path-trace}?						+++rw data-export-trace-jitter-threshold?	ui
nt32 {proof-of-transit}?						+++rw data-export-pot-failure-threshold?	ui
nt32 {path-packet-counters}?						+++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
						+++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
m-mode						+++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	
						+++rw flow-classifier* [name]	
						+++rw name	string
						+++rw access-list?	string
						+++rw data-export-profile* [name]	
ring						+++rw name	st
ta-export-type						+++rw oam-data-export-type?	da
						+++rw (data-export-types)?	
						+++:(flexibleexport)	
						+++rw flexible-export-params	
						+++rw transport	
-transport-type						+++rw flex-data-export-protocol?	export
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?	enumer
ation						+-:(ipfix)	
						+-rw ip-fix-params	

transport-type						+++rw ipfix-data-export-protocol?	export-t
address						+++rw ipfix-data-export-ip?	inet:ip-
port-number						+++rw ipfix-data-export-port?	inet:por
timestamp-accuracy						+++rw sampling-frequency?	ti
timestamp-accuracy						+++rw sampling-interval?	ui
timestamp-accuracy						+++rw data-export-trace-time-res?	ti
timestamp-accuracy {path-trace}?						+++rw data-export-trace-delay-threshold?	ui
timestamp {path-trace}?						+++rw data-export-trace-jitter-threshold?	ui
timestamp {path-trace}?						+++rw data-export-pot-failure-threshold?	ui
timestamp {proof-of-transit}?						+++rw data-export-ppc-drops-failure-threshold?	ui
timestamp {path-packet-counters}?						+++rw data-export-ppc-reorder-failure-threshold?	ui
timestamp {path-packet-counters}?						+++rw data-export-ppc-oos-failure-threshold?	ui
timestamp {path-packet-counters}?						+++rw data-export-ppc-dup-failure-threshold?	ui
timestamp {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	st
ring						+++rw oam-data-export-type?	da
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:i
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-t

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

```

nt32 {path-trace}? | | | +--rw data-export-trace-delay-threshold? ui
nt32 {path-trace}? | | | +--rw data-export-trace-jitter-threshold? ui
nt32 {proof-of-transit}? | | | +--rw data-export-pot-failure-threshold? ui
nt32 {path-packet-counters}? | | | +--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
| | | +--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

```

```

dress
|
|   +---:(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-Ad
|   +---:(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-tunnel-address-location-list {connection-less}?
+---rw test-point-locations* [tunnel-location]
|   +---rw tunnel-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-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 {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 st
ring
| | | | | +--rw oam-data-export-type? da
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
p-address
| | | | | | | +--rw flex-data-export-port? inet:p
ort-number
| | | | | | | +--rw encoding
| | | | | | | | +--rw data-export-encoding-type? enumer
ation
| | | | | | | +---:(ipfix)
| | | | | | | | +--rw ip-fix-params
| | | | | | | | +--rw ipfix-data-export-protocol? export-t
ransport-type
| | | | | | | +--rw ipfix-data-export-ip? inet:ip-
address
| | | | | | | +--rw ipfix-data-export-port? inet:por
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		st
ring									
							+---rw oam-data-export-type?		da
ta-export-type							+---rw (data-export-types)?		
							+---:(flexibleexport)		
							+---rw flexible-export-params		
							+---rw transport		
-transport-type							+---rw flex-data-export-protocol?	export	
p-address							+---rw flex-data-export-ip?	inet:i	
ort-number							+---rw flex-data-export-port?	inet:p	
							+---rw encoding		
ation							+---rw data-export-encoding-type?	enumer	
							+---:(ipfix)		
							+---rw ip-fix-params		
ransport-type							+---rw ipfix-data-export-protocol?	export-t	
address							+---rw ipfix-data-export-ip?	inet:ip-	
t-number							+---rw ipfix-data-export-port?	inet:por	
mestamp-accuracy							+---rw sampling-frequency?	ti	
nt64							+---rw sampling-interval?	ui	
mestamp-accuracy {path-trace}?							+---rw data-export-trace-time-res?	ti	
nt32 {path-trace}?							+---rw data-export-trace-delay-threshold?	ui	
nt32 {path-trace}?							+---rw data-export-trace-jitter-threshold?	ui	
nt32 {proof-of-transit}?							+---rw data-export-pot-failure-threshold?	ui	
nt32 {path-packet-counters}?							+---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	
							+---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	
m-mode							+---rw md-proof-of-transit?	boolean	
							+---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]		
ring			+-rw name		st


```
|  +--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-Ad
    |   |   |   |   +---:(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
    |   |   |   +---:(src-dst-address)
    |   |   |   |   +--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	
f-transit}?	+(tools-mpls)		
	+--rw rfc4379?	boolean	
	+--rw rfc4687?	boolean	
	+--rw rfc4950?	boolean	
	+--rw mpls- <u>rfc5884</u> ?	boolean	
	+(tools-mpls-tp)		
	+--rw rfc6426?	boolean	
	+--rw rfc6435?	boolean	
	+--rw rfc6374?	boolean	
	+(tools-pw)		
	+--rw rfc5085?	boolean	
	+--rw pw_ufc5885?	boolean	
	+--rw rfc6423?	boolean	
	+--rw rfc6310?	boolean	
	+--rw rfc7023?	boolean	
	+(tools-passive-oam)		
	+--rw passive-oam-config {path-packet-counters,path-trace, <u>proof-of-path</u> }		
	+--rw path-packet-counters-config		
	+--rw flow-classifier* [name]		
	+--rw name	string	
	+--rw access-list?	string	
	+--rw data-export-profile* [name]		
	+--rw name		st
ring			
	+--rw oam-data-export-type?		da
ta-export-type			
	+--rw (data-export-types)?		
	+(flexibleexport)		
	+--rw flexible-export-params		
	+--rw transport		
-transport-type	+--rw flex-data-export-protocol?	export	
p-address	+--rw flex-data-export-ip?	inet:i	
ort-number	+--rw flex-data-export-port?	inet:p	
	+--rw encoding		
aion	+--rw data-export-encoding-type?	enumer	
	+(ipfix)		
	+--rw ip-fix-params		
ransport-type	+--rw ipfix-data-export-protocol?	export-t	
address	+--rw ipfix-data-export-ip?	inet:ip-	
t-number	+--rw ipfix-data-export-port?	inet:por	
mestamp-accuracy	+--rw sampling-frequency?	ti	
nt64	+--rw sampling-interval?	ui	
mestamp-accuracy {path-trace}?}	+--rw data-export-trace-time-res?	ti	
nt32 {path-trace}?}	+--rw data-export-trace-delay-threshold?	ui	
nt32 {path-trace}?}	+--rw data-export-trace-jitter-threshold?	ui	
	+--rw data-export-pot-failure-threshold?	ui	

nt32 {proof-of-transit}?

Kumar, et al.

Expires November 27, 2016

[Page 36]


```

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

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

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

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

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

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

ransport-type
| | | | | | +--rw ipfix-data-export-ip? inet:ip-

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

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-proof-of-transit?	boolean
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 st	
ring								+--rw oam-data-export-type?	da
ta-export-type								+--rw (data-export-types)?	
								+---:(flexibleexport)	
								+--rw flexible-export-params	
								+--rw transport	
-transport-type								+--rw flex-data-export-protocol?	export
p-address								+--rw flex-data-export-ip?	inet:i
ort-number								+--rw flex-data-export-port?	inet:p
								+--rw encoding	
ation								+--rw data-export-encoding-type?	enumer
								+---:(ipfix)	
								+--rw ip-fix-params	
ransport-type								+--rw ipfix-data-export-protocol?	export-t
address								+--rw ipfix-data-export-ip?	inet:ip-
t-number								+--rw ipfix-data-export-port?	inet:por
mestamp-accuracy								+--rw sampling-frequency?	ti
nt64								+--rw sampling-interval?	ui
mestamp-accuracy {path-trace}?								+--rw data-export-trace-time-res?	ti
nt32 {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]
    |

```

		uint32	
	if:interface-ref		
	md-tracing?		boolean
m-mode	trace-mode		passive-oa
	trace-length		uint8
accuracy	trace-time-stamp		timestamp-
	trace-app-data?		uint32
	trace-include-interface		boolean
	+--rw oam-layers* [index]		
	index	uint16	
	level?	int32	
	+--rw (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	
	+--rw (fec-value)?		
	+--:(ip-prefix)		
	ip-prefix?	inet:ip-prefix	
	+--:(bgp)		
	bgp?	inet:ip-prefix	
	+--:(tunnel)		
	tunnel-interface?	uint32	
	+--:(l3vpn)		
	l3vpn-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)		
	+--rw (root-address)?		
	+--:(ip-address)		
	source-address?	inet:ip-address	
dress	group-ip-address?	IP-Multicast-Group-Ad	
	+--:(vpn)		
	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-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
  |   |   +---:(vpn)
  |   |   |   +-rw as-number?       inet:as-number
  |   |   +---:(global-id)
  |   |   |   +-rw lsp-id?          string

```

ss

```

    +---:(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-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                    st
ring
  | | +---rw oam-data-export-type?      da
ta-export-type
  | | +---rw (data-export-types)?
  | | | +---:(flexibleexport)

```

							+--rw flexible-export-params	
							+--rw transport	
-transport-type							+--rw flex-data-export-protocol?	export
p-address							+--rw flex-data-export-ip?	inet:i
ort-number							+--rw flex-data-export-port?	inet:p
							+--rw encoding	
ation							+--rw data-export-encoding-type?	enumer
							+---:(ipfix)	
							+--rw ip-fix-params	
ransport-type							+--rw ipfix-data-export-protocol?	export-t
address							+--rw ipfix-data-export-ip?	inet:ip-
t-number							+--rw ipfix-data-export-port?	inet:por
mestamp-accuracy							+--rw sampling-frequency?	ti
nt64							+--rw sampling-interval?	ui
mestamp-accuracy {path-trace}?							+--rw data-export-trace-time-res?	ti
nt32 {path-trace}?							+--rw data-export-trace-delay-threshold?	ui
nt32 {path-trace}?							+--rw data-export-trace-jitter-threshold?	ui
nt32 {proof-of-transit}?							+--rw data-export-pot-failure-threshold?	ui
nt32 {path-packet-counters}?							+--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
							+--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	
m-mode							+--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	
							+--rw flow-classifier* [name]	
							+--rw name string	
							+--rw access-list? string	
							+--rw data-export-profile* [name]	
ring							+--rw name	st
ta-export-type							+--rw oam-data-export-type?	da

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

p-address							+++rw flex-data-export-ip?	inet:i
ort-number							+++rw flex-data-export-port?	inet:p
ation							+++rw encoding +++rw data-export-encoding-type?	enumer
ransport-type							+++:(ipfix) +++rw ip-fix-params +++rw ipfix-data-export-protocol?	export-t
address							+++rw ipfix-data-export-ip?	inet:ip-
t-number							+++rw ipfix-data-export-port?	inet:por
mestamp-accuracy							+++rw sampling-frequency?	ti
nt64							+++rw sampling-interval?	ui
mestamp-accuracy {path-trace}?							+++rw data-export-trace-time-res?	ti
nt32 {path-trace}?							+++rw data-export-trace-delay-threshold?	ui
nt32 {path-trace}?							+++rw data-export-trace-jitter-threshold?	ui
nt32 {proof-of-transit}?							+++rw data-export-pot-failure-threshold?	ui
nt32 {path-packet-counters}?							+++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
							+++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	
m-mode							+++rw md-proof-of-transit? +++rw pot-mode?	boolean passive-oa
							+++rw notification-pot? +++rw notification-pot-failure-threshold? +++rw path-trace-config +++rw flow-classifier* [name] +++rw name string +++rw access-list? string +++rw data-export-profile* [name] +++rw name	boolean uint32
ring								st
ta-export-type							+++rw oam-data-export-type?	da
							+++rw (data-export-types)? +++:(flexibleexport) +++rw flexible-export-params +++rw transport	
-transport-type							+++rw flex-data-export-protocol?	export
							+++rw flex-data-export-ip?	inet:i

```
p-address      |          |   |   |   |       |    +--rw flex-data-export-port?           inet:p
ort-number     |          |   |   |   |       |    +--rw encoding                          enumer
               |          |   |   |   |       |    +--rw data-export-encoding-type?
ation         |          |   |   |   |       |    +--:(ipfix)
               |          |   |   |   |       |    +--rw ip-fix-params
```

transport-type					+++rw ipfix-data-export-protocol?	export-t
address					+++rw ipfix-data-export-ip?	inet:ip-
t-number					+++rw ipfix-data-export-port?	inet:por
mestamp-accuracy				+++rw sampling-frequency?		ti
nt64				+++rw sampling-interval?		ui
mestamp-accuracy {path-trace}?				+++rw data-export-trace-time-res?		ti
nt32 {path-trace}?				+++rw data-export-trace-delay-threshold?		ui
nt32 {path-trace}?				+++rw data-export-trace-jitter-threshold?		ui
nt32 {proof-of-transit}?				+++rw data-export-pot-failure-threshold?		ui
nt32 {path-packet-counters}?				+++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
				+++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
m-mode				+++rw trace-mode		passive-oa
				+++rw trace-length		uint8
accuracy				+++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)		


```

| | | +---:(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
ss | | |
| | | | +---:(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 {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                st
ring
| | | | +--rw oam-data-export-type?    da
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
p-address
| | | | | | | | | +--rw flex-data-export-port?       inet:p
ort-number
| | | | | | | | | +--rw encoding
| | | | | | | | | +--rw data-export-encoding-type?   enumer
ation
| | | | | | | | | +---:(ipfix)
| | | | | | | | | | +--rw ip-fix-params
| | | | | | | | | | +--rw ipfix-data-export-protocol?  export-t
ransport-type
| | | | | | | | | | +--rw ipfix-data-export-ip?       inet:ip-
address
| | | | | | | | | | +--rw ipfix-data-export-port?     inet:por
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

m-mode		+++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	
		+++rw flow-classifier* [name]	
		+++rw name	string
		+++rw access-list?	string
		+++rw data-export-profile* [name]	
		+++rw name	st
ring			
ta-export-type		+++rw oam-data-export-type?	da
		+++rw (data-export-types)?	
		+++:(flexibleexport)	
		+++rw flexible-export-params	
		+++rw transport	
-transport-type		+++rw flex-data-export-protocol?	export
p-address		+++rw flex-data-export-ip?	inet:i
ort-number		+++rw flex-data-export-port?	inet:p
		+++rw encoding	
ation		+++rw data-export-encoding-type?	enumer
		+++:(ipfix)	
		+++rw ip-fix-params	
ransport-type		+++rw ipfix-data-export-protocol?	export-t
address		+++rw ipfix-data-export-ip?	inet:ip-
t-number		+++rw ipfix-data-export-port?	inet:por
mestamp-accuracy		+++rw sampling-frequency?	ti
nt64		+++rw sampling-interval?	ui
mestamp-accuracy {path-trace}?		+++rw data-export-trace-time-res?	ti
nt32 {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

m-mode			+-rw intf-name? if:interface-ref	
			+-rw md-proof-of-transit?	boolean
			+-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		st
ring									
							+--rw oam-data-export-type?		da
ta-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	
ort-number							+--rw flex-data-export-port?	inet:p	
							+--rw encoding		
ation							+--rw data-export-encoding-type?	enumer	
							+++:(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 ipfix-data-export-port?	inet:por	
mestamp-accuracy							+--rw sampling-frequency?	ti	
nt64							+--rw sampling-interval?	ui	
mestamp-accuracy {path-trace}?							+--rw data-export-trace-time-res?	ti	
nt32 {path-trace}?							+--rw data-export-trace-delay-threshold?	ui	
nt32 {path-trace}?							+--rw data-export-trace-jitter-threshold?	ui	
nt32 {proof-of-transit}?							+--rw data-export-pot-failure-threshold?	ui	
nt32 {path-packet-counters}?							+--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	
							+--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	
m-mode							+--rw trace-mode	passive-oa	
							+--rw trace-length	uint8	
accuracy							+--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)
|       |   +--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-Ad
|       +---:(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)

```

		+-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?	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-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                st
ring
    |   |   |   +---rw oam-data-export-type?         da
ta-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:i
p-address
    |   |   |   |   |   |   |   +---rw flex-data-export-port?       inet:p
ort-number
    |   |   |   |   |   |   |   +---rw encoding
    |   |   |   |   |   |   |   |   +---rw data-export-encoding-type?   enumer
ation
    |   |   |   |   |   |   |   +---:(ipfix)
    |   |   |   |   |   |   |   |   +---rw ip-fix-params
    |   |   |   |   |   |   |   |   |   +---rw ipfix-data-export-protocol?   export-t
ransport-type
    |   |   |   |   |   |   |   |   |   +---rw ipfix-data-export-ip?       inet:ip-
address
    |   |   |   |   |   |   |   |   |   +---rw ipfix-data-export-port?       inet:por
t-number

```


mestamp-accuracy							+++rw sampling-frequency?	ti
nt64							+++rw sampling-interval?	ui
mestamp-accuracy {path-trace}?							+++rw data-export-trace-time-res?	ti
nt32 {path-trace}?							+++rw data-export-trace-delay-threshold?	ui
nt32 {path-trace}?							+++rw data-export-trace-jitter-threshold?	ui
nt32 {proof-of-transit}?							+++rw data-export-pot-failure-threshold?	ui
nt32 {path-packet-counters}?							+++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
							+++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	
m-mode							+++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	
							+++rw flow-classifier* [name]	
							+++rw name string	
							+++rw access-list? string	
							+++rw data-export-profile* [name]	
ring							+++rw name	st
ta-export-type							+++rw oam-data-export-type?	da
							+++rw (data-export-types)?	
							+++:(flexibleexport)	
							+++rw flexible-export-params	
							+++rw transport	
-transport-type							+++rw flex-data-export-protocol?	export
p-address							+++rw flex-data-export-ip?	inet:i
ort-number							+++rw flex-data-export-port?	inet:p
							+++rw encoding	
ation							+++rw data-export-encoding-type?	enumer
							+++:(ipfix)	
							+++rw ip-fix-params	
ransport-type							+++rw ipfix-data-export-protocol?	export-t
address							+++rw ipfix-data-export-ip?	inet:ip-

t-number					+++rw ipfix-data-export-port?	inet:por
mestamp-accuracy					+++rw sampling-frequency?	ti
nt64					+++rw sampling-interval?	ui
mestamp-accuracy					+++rw data-export-trace-time-res?	ti
{path-trace}?						

```

nt32 {path-trace}? | | | +--rw data-export-trace-delay-threshold? ui
nt32 {path-trace}? | | | +--rw data-export-trace-jitter-threshold? ui
nt32 {proof-of-transit}? | | | +--rw data-export-pot-failure-threshold? ui
nt32 {path-packet-counters}? | | | +--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 st
ring | | | +--rw oam-data-export-type? da
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
p-address | | | | +--rw flex-data-export-port? inet:p
ort-number | | | | +--rw encoding
| | | | +--rw data-export-encoding-type? enumer
ation | | | | +--:(ipfix)
| | | | | +--rw ip-fix-params
| | | | | +--rw ipfix-data-export-protocol? export-t
ransport-type | | | | +--rw ipfix-data-export-ip? inet:ip-
address | | | | +--rw ipfix-data-export-port? inet:por
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}?

```

```

nt32 {path-packet-counters}?
|
| | | | +--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-tracing?          boolean
| | | | |   +--rw trace-mode          passive-oa
| | | |
| | | | +--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)
| | | | |   |   |   |   |   +--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)?

```

```

dress
|
|         +---:(ip-address)
|         |   +---rw source-address?      inet:ip-address
|         |   +---rw group-ip-address?    IP-Multicast-Group-Ad
|
|         +---:(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-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)?
|   +---:(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
ss
| | | | +---:(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 {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		st
ring						
				+---rw oam-data-export-type?		da
ta-export-type						
				+---rw (data-export-types)?		
				+---:(flexibleexport)		
				+---rw flexible-export-params		
				+---rw transport		
-transport-type				+---rw flex-data-export-protocol?	export	
p-address				+---rw flex-data-export-ip?	inet:i	
ort-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	
ransport-type						
				+---rw ipfix-data-export-ip?	inet:ip-	
address						
				+---rw ipfix-data-export-port?	inet:por	
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	
m-mode				+---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		

			+-rw flow-classifier* [name]		
			+-rw name	string	
			+-rw access-list?	string	
			+-rw data-export-profile* [name]		
ring			+-rw name		st

ta-export-type				+++rw oam-data-export-type?	da
				+++rw (data-export-types)?	
				+++:(flexibleexport)	
				+++rw flexible-export-params	
				+++rw transport	
-transport-type				+++rw flex-data-export-protocol?	export
p-address				+++rw flex-data-export-ip?	inet:i
ort-number				+++rw flex-data-export-port?	inet:p
				+++rw encoding	
ation				+++rw data-export-encoding-type?	enumer
				+++:(ipfix)	
				+++rw ip-fix-params	
ransport-type				+++rw ipfix-data-export-protocol?	export-t
address				+++rw ipfix-data-export-ip?	inet:ip-
t-number				+++rw ipfix-data-export-port?	inet:por
mestamp-accuracy				+++rw sampling-frequency?	ti
nt64				+++rw sampling-interval?	ui
mestamp-accuracy {path-trace}?				+++rw data-export-trace-time-res?	ti
nt32 {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
				+++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
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]	
ring				+++rw name	st
				+++rw oam-data-export-type?	da

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

p-address							+++rw flex-data-export-ip?	inet:i
ort-number							+++rw flex-data-export-port?	inet:p
ation							+++rw encoding	
							+++rw data-export-encoding-type?	enumer
							+++:(ipfix)	
							+++rw ip-fix-params	
ransport-type							+++rw ipfix-data-export-protocol?	export-t
address							+++rw ipfix-data-export-ip?	inet:ip-
t-number							+++rw ipfix-data-export-port?	inet:por
mestamp-accuracy							+++rw sampling-frequency?	ti
nt64							+++rw sampling-interval?	ui
mestamp-accuracy							+++rw data-export-trace-time-res?	ti
nt32 {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?	boolean
m-mode							+++rw trace-mode	passive-oa
							+++rw trace-length	uint8
accuracy							+++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)
        | +--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-Ad
dress
          +---:(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: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
SS
| | | |   +---:(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

```

[illegible]


```

nt32 {path-packet-counters}?
      | | | | |--rw data-export-ppc-dup-failure-threshold?      ui
nt32 {path-packet-counters}?
      | | | |--rw transport-encap-profile* [name]
      | | | |--rw name                                           string

```

[illegible]

```
| | |   +--rw transport-encap-profile* [name]
| | |   |   +--rw name                string
| | |   |   +--rw transport-type?    oaam-transport-type-def
| | | +--rw node-id?                uint32
| | | +--rw node-interfaces* [index]
```

						+--rw index	uint32	
						+--rw intf-name?	if:interface-ref	
m-mode						+--rw md-proof-of-transit?	boolean	
						+--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]		
ring						+--rw name	st	
ta-export-type						+--rw oam-data-export-type?	da	
						+--rw (data-export-types)?		
						+---:(flexibleexport)		
						+--rw flexible-export-params		
-transport-type						+--rw transport		
p-address						+--rw flex-data-export-protocol?	export	
ort-number						+--rw flex-data-export-ip?	inet:i	
						+--rw flex-data-export-port?	inet:p	
ation						+--rw encoding		
						+--rw data-export-encoding-type?	enumer	
						+---:(ipfix)		
ransport-type						+--rw ip-fix-params		
address						+--rw ipfix-data-export-protocol?	export-t	
t-number						+--rw ipfix-data-export-ip?	inet:ip-	
mestamp-accuracy						+--rw ipfix-data-export-port?	inet:por	
nt64						+--rw sampling-frequency?	ti	
mestamp-accuracy {path-trace}?						+--rw sampling-interval?	ui	
nt32 {path-trace}?						+--rw data-export-trace-time-res?	ti	
nt32 {path-trace}?						+--rw data-export-trace-delay-threshold?	ui	
nt32 {path-trace}?						+--rw data-export-trace-jitter-threshold?	ui	
nt32 {proof-of-transit}?						+--rw data-export-pot-failure-threshold?	ui	
nt32 {path-packet-counters}?						+--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	
						+--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
m-mode			+-rw trace-mode	passive-oa
			+-rw trace-length	uint8
accuracy			+-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)
|       |     | +---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-Ad
|       |   +---:(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
dress
rpccs:

```

```

+---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-Ad
|   |   +---:(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
dress

```

```

| | +---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)
          | +---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)?

```



```

dress
|
|
|
|      +---:(ip-address)
|      |   +---w source-address?      inet:ip-address
|      |   +---w group-ip-address?   IP-Multicast-Group-Ad
|
|      +---:(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 vrf?                    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 hop-cnt?                      uint8
+--ro destination-tp
|   +--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)
|       |   +--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
+--ro min-delay?                  oam-counter32
+--ro average-delay?              oam-counter32
+--ro max-delay?                  oam-counter32
notifications:
+---n passive-oam-pot-triggered-oam {proof-of-transit}?
|   +--ro node-id?                uint32
|   +---ro node-interfaces* [index]
|       |   +--ro index            uint32
|       |   +--ro intf-name?       if:interface-ref
|   +--ro access-list?            string
|   +--ro failure-code?           uint32
|   +--ro failed-count?           uint32
+---n passive-oam-ppc-triggered-oam {path-packet-counters}?
|   +--ro node-id?                uint32
|   +---ro node-interfaces* [index]
|       |   +--ro index            uint32
|       |   +--ro intf-name?       if:interface-ref
|   +--ro access-list?            string
|   +--ro failure-code?           uint32
|   +--ro failed-drop-count?      uint32
|   +--ro failed-reorder-count?   uint32
|   +--ro failed-oos-count?       uint32
|   +--ro failed-dup-count?       uint32

```

data hierarchy of OAM

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-routing {
  prefix rt;
}
import ietf-packet-fields {
  prefix pfs;
  revision-date 2015-06-11;
}
organization "IETF LIME Working Group";
contact
  "Deepak Kumar dekkumar@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 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.";
}
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]|'
                +'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]'
        +'{0,4}):){0,5}((([0-9a-fA-F]{0,4}):)?'
        +'(:|[0-9a-fA-F]{0,4}))|(((25[0-5]|2[0-4]'
        +'[0-9]|[01]?[0-9]?[0-9])\.){3}(25[0-5]|'
        +'2[0-4][0-9]|[01]?[0-9]?[0-9])))'';
    pattern
    '(([^:]+:){6}(([^:]+:[^:]+)|'
    +'(.*\..*)))|((([^:]+:)*[^^:]+)'
    +'?::(([^:]+:)*[^^:]+)?)';
    }
    description
        "The IPv6-Multicast-Group-Address
        type represents an IPv6 address in full,
        mixed, shortened, and shortened-mixed
        notation.";
    reference "RFC4291 2.7.
    ietf-inet-types:ipv6-address";
}
typedef IP-Multicast-Group-Address {
    type union {
        type IPv4-Multicast-Group-Address;
        type IPv6-Multicast-Group-Address;
    }
    description
        "The IP-Multicast-Group-Address type
        represents an IP multicast address and
        is IP version neutral. The format of the
        textual representations implies the IP version.";
} // 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.";
    }
  }
}
```

```
    description
      "Timestamp accuracy options.";
  }

  typedef data-export-type {
    type enumeration {
      enum flexibleexport {
        value 1;
        description
          "raw format export";
      }
      enum ipfix {
        value 2;
        description
          "IPFIX formatted export";
      }
    }
    description
      "Determines the data export format type.";
  }

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

  /* groupings */
  grouping cc-session-statsitics {
    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 tri
gger
             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 opti
ons.";
            }
        }

        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;
        }
    }
}

```

```

    }
    /*** container:end ***/
  }
  /*** case:end ***/
}
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.";
    }
  }
  /*** container:end ***/
}
/*** case:end ***/
}
}
/*** grouping:end ***/
}

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

```

```
        "Unique identifier for each profile";
    }
    uses oam-data-export-config;
    uses oam-data-export-freq-config;
    uses oam-data-export-threshold-config;
}

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";
            }
            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.";
        }
    }
    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
```

```
        "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 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.";
        /*** 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-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
      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.";
    }
}
```

```

    }
    leaf rfc6374 {
        type boolean;
        description
            "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.";
    }
}
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 option
s";
        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
```

```
        "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.";
  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:networks/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;
  }
  container cc-ipv6-sessions-statistics {
    if-feature continuity-check;
    description "cc ipv6 sessions";
    uses cc-session-statistics;
  }
  container po-sessions-proof-of-transit-statistics {
    if-feature proof-of-transit;
    description "for oam proof of transit stats of sessions";
    uses po-per-session-proof-of-transit-statistics;
  }
  container po-sessions-path-trace-statistics {
    if-feature path-trace;
    description "for passive oam path trace of sessions";
    uses session-delay-jitter-statistics;
  }
  container po-sessions-packet-counter-statistics {
    if-feature path-packet-counters;
    description "for passive oam path packet counter of sessions";
    uses session-error-statistics;
  }
}

rpc continuity-check {
  if-feature continuity-check;
  description
    "Generates continuity-check as per RFC7276.";
  input {
    container destination-tp {
      uses tp-address;
      description
        "destination test point.";
    }
    uses 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 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-packet-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;
        }
    }
}
```

```
        description
            "source interface.";
    }
    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:

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

Registrant Contact: The IESG.

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

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

name: ietf-connectionless-oam namespace: urn:ietf:params:xml:ns:yang:ietf-connectionless-oam
prefix: goam reference: RFC XXXX

7. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<http://www.rfc-editor.org/info/rfc6020>>.

- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<http://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<http://www.rfc-editor.org/info/rfc6242>>.
- [RFC792] Postel, J., "Internet Control Message Protocol", RFC 792, September 1981.

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