

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: October 11, 2015

Z. Wang  
Q. Wu  
Huawei  
April 9, 2015

Additional RPC definitions to Generic YANG Data Model for layer  
Independent OAM Management  
draft-wang-lime-rpc-yang-oam-management-03

## Abstract

[I-D.tissa-lime-yang-oam-model] defines a Generic YANG data model for Layer independent OAM Management. This document proposes additional extension to this YANG model which is complementary to the one defined in the [[I-D.tissa-lime-yang-oam-model](#)]. The extension include generic notification and generic rpc command for Unified Management Plane OAM to be used within IETF in a layer independent manner. The generic notification and rpc command described in this document can be applied to various network technologies and includes technology independent configuration data and state data.

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

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

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

This Internet-Draft will expire on October 11, 2015.

## Copyright Notice

Copyright (c) 2015 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

<a href="#">1.</a>	Introduction . . . . .	<a href="#">2</a>
<a href="#">2.</a>	Conventions and Terminology . . . . .	<a href="#">2</a>
<a href="#">2.1.</a>	Tree Diagrams . . . . .	<a href="#">3</a>
<a href="#">3.</a>	Overview of the generic Notification and RPC command . . . . .	<a href="#">4</a>
<a href="#">3.1.</a>	Generic Notification Definition . . . . .	<a href="#">4</a>
<a href="#">3.2.</a>	Generic CC Command Rpc . . . . .	<a href="#">5</a>
<a href="#">3.3.</a>	Generic CV Command Rpc . . . . .	<a href="#">6</a>
<a href="#">3.4.</a>	Generic Path Discovery Command Rpc . . . . .	<a href="#">7</a>
<a href="#">4.</a>	OAM data hierarchy for Abstract Notification and rpc command	8
<a href="#">5.</a>	OAM YANG Module . . . . .	<a href="#">11</a>
<a href="#">6.</a>	Security Considerations . . . . .	<a href="#">29</a>
<a href="#">7.</a>	IANA Considerations . . . . .	<a href="#">29</a>
<a href="#">8.</a>	References . . . . .	<a href="#">29</a>
<a href="#">8.1.</a>	Normative References . . . . .	<a href="#">29</a>
<a href="#">8.2.</a>	Informative References . . . . .	<a href="#">30</a>
	Authors' Addresses . . . . .	<a href="#">30</a>

## [1.](#) Introduction

[I-D.tissa-lime-yang-oam-model] defines a YANG [[RFC6020](#)] data model for Layer independent OAM Management implementations that can be applied to various network technologies. This YANG module describes the generic common core configuration, statistics for Unified Management Plane OAM to be used within IETF in a layer independent manner. This document describes the generic notification and rpc command which is complementary to the one defined in the [I-D.tissa-lime-yang-oam-model]. The generic notification and rpc command includes technology independent configuration data and state data.

## [2.](#) Conventions and Terminology

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.](#) 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 generic Notification and RPC command

[I-D.tissa-lime-yang-oam-model] defines a Generic YANG data model for Layer independent OAM Management. This data model aims to address the problems associated with existing OAM technology deployment described in [[I-D.edprop-opsawg-multi-layer-oam](#)] and adopts IEEE CFM like model or Y.1731 like model and structures it such that it can be adapted to different technologies.

rpc commands, in YANG terms are used between the interface between management plane and data plane. This document defines generic notification and rpc commands providing uniform APIs for common OAM function defined in [section 3 of \[RFC7276\]](#), e.g., CC, CV, Path Discovery.

#### [3.1.](#) Generic Notification Definition

This generic notification is sent whenever defect condition is met.

notifications:

```

+---n defect-condition-notification
+---ro technology          identityref
+---ro md-name-string      string
+---ro md-level?           uint32
+---ro ma-name-string      string
+---ro mep-id?             MP-id
+---ro remote-mepid
|   +---ro (MEP-ID)?
|   +----:(MEP-ID-int)
|   |       +---ro MEP-ID-int ?   int32
+---ro oper-status         enumeration
+---ro sub-oper-status     enumeration
+---ro error-message       string

```

Snippet of data hierarchy related to Management OAM domains

### [3.2.](#) Generic CC Command Rpc

Rpc model defined here abstracts OAM specific commands in a technology independent manner. Here is rpc model for generic Continuity Check(CC) command.

```

rpcs:
+---x continuity-check
|   +---ro input
|   |   +---ro technology          identityref
|   |   +---ro MD-name-string      MD-name-string
|   |   +---ro MA-name-string?     MA-name-string
|   |   +---ro (flow-entropy)?
|   |   |   +---:(flow-entropy-null)
|   |   |   |       +---ro flow-entropy-null?   empty
|   |   +---ro priority?           uint8
|   |   +---ro ttl?                uint8
|   |   +---ro session-type?       enumeration
|   |   +---ro ecmp-choice?        ecmp-choices
|   |   +---ro sub-type?           identityref
|   |   +---ro outgoing-interfaces* [interface]
|   |   |   +---ro interface      if:interface-ref

```

```

| | +---ro source-mep? MEP-name
| | +---ro destination-mp
| | | +---ro (mp-address)?
| | | | +---:(mac-address)
| | | | +---ro mac-address? yang:mac-address
| | | | +---:(ipv4-address)
| | | | +---ro ipv4-address? inet:ipv4-address
| | | | +---:(ipv6-address)
| | | | +---ro ipv6-address? inet:ipv6-address
| | | +---ro (MEP-ID)?
| | | | +---:(MEP-ID-int)
| | | | +---ro MEP-ID-int? int32
| | | +---ro MEP-ID-format? identityref
| | +---ro count? uint32
| | +---ro interval? Interval
| | +---ro packet-size? uint32
| +---ro output
| | +---ro monitor-stats
| | | --+---:(monitor-null)
| | | +---ro monitor-null? Empty

```

Snippet of data hierarchy related to CC

### [3.3.](#) Generic CV Command Rpc

Rpc model defined here abstracts OAM specific commands in a technology independent manner. Here is rpc model for generic Connectivity Verification (CV) command.

```

rpcs:
  +---x connectivity-verification
  | +---ro input
  | | +---ro technology identityref
  | | +---ro MD-name-string MD-name-string
  | | +---ro MA-name-string? MA-name-string
  | | +---ro (flow-entropy)?
  | | | +---:(flow-entropy-null)
  | | | +---ro flow-entropy-null? empty

```

```

| | +---ro priority?                uint8
| | +---ro ttl?                    uint8
| | +---ro session-type?           enumeration
| | +---ro ecmp-choice?            ecmp-choices
| | +---ro sub-type?               identityref
| | +---ro outgoing-interfaces* [interface]
| | | +---ro interface             if:interface-ref
| | +---ro source-mep?             MEP-name
| | +---ro destination-mp
| | | +---ro (mp-address)?
| | | | +---:(mac-address)
| | | | | +---ro mac-address?      yang:mac-address
| | | | +---:(ipv4-address)
| | | | | +---ro ipv4-address?     inet:ipv4-address
| | | | +---:(ipv6-address)
| | | | | +---ro ipv6-address?     inet:ipv6-address
| | | +---ro (MEP-ID)?
| | | | +---:(MEP-ID-int)
| | | | | +---ro MEP-ID-int?       int32
| | | +---ro MEP-ID-format?       identityref
| | +---ro count?                 uint32
| | +---ro interval?              Interval
| | +---ro packet-size?           uint32
| +---ro output
| | +---ro monitor-stats
| | | --+---:(monitor-null)
| | | | +---ro monitor-null?      Empty

```

Snippet of data hierarchy related to CV

### [3.4.](#) Generic Path Discovery Command Rpc

Rpc model defined here abstracts OAM specific commands in a technology independent manner. Here is rpc model for Generic Path Discovery command.

```

rpcs:
  +---x path-discovery

```

```

+--ro input
|   +--ro technology                identityref
|   +--ro MD-name-string            MD-name-string
|   +--ro MA-name-string            MA-name-string
|   +--ro (flow-entropy)?
|   |   +--:(flow-entropy-null)
|   |   |   +--ro flow-entropy-null?    empty
|   +--ro priority?                uint8
|   +--ro ttl?                     uint8
|   +--ro sub-type?                Identityref
|   +--ro session-type?            enumeration
|   +--ro ecmp-choice?             ecmp-choices
|   +--ro outgoing-interfaces* [interface]
|   |   +--ro interface            if:interface-ref
|   +--ro source-mep?              MEP-name
|   +--ro destination-mp
|   |   +--ro (mp-address)?
|   |   |   +--:(mac-address)
|   |   |   |   +--ro mac-address?      yang:mac-address
|   |   |   +--:(ipv4-address)
|   |   |   |   +--ro ipv4-address?     inet:ipv4-address
|   |   |   +--:(ipv6-address)
|   |   |   |   +--ro ipv6-address?     inet:ipv6-address
|   |   +--ro (MEP-ID)?
|   |   |   +--:(MEP-ID-int)
|   |   |   |   +--ro MEP-ID-int?       int32
|   |   +--ro MEP-ID-format?        identityref
|   +--ro count?                   uint32
|   +--ro interval?                Interval
+--ro output
|   +--ro response* [response-index]
|   |   +--ro response-index        uint8
|   |   +--ro ttl?                 uint8
|   |   +--ro destination-mp
|   |   |   +--ro (mp-address)?
|   |   |   |   +--:(mac-address)
|   |   |   |   |   +--ro mac-address?   yang:mac-address
|   |   |   |   +--:(ipv4-address)
|   |   |   |   |   +--ro ipv4-address?   inet:ipv4-address
|   |   |   |   +--:(ipv6-address)

```



```

|   +---ro (MEP-ID)?
|   |   +---:(MEP-ID-int)
|   |       +---ro MEP-ID-int?           int32
|   +---ro MEP-ID-format?   identityref
|   +---ro monitor-stats
|   |---+---:(monitor-null)
|       +---ro monitor-null?   Empty

```

Snippet of data hierarchy related to Path Discovery

#### 4. OAM data hierarchy for Abstract Notification and rpc command

The complete data hierarchy related to the abstract notification and rpc comand is presented below.

module: ietf-gen-oam (defined in [[I-D.tissa-lime-yang-oam-model](#)])

+---rw domains

```

+---rw domain* [md-name technology]
+---rw technology           identityref
+---rw md-name-format       MD-name-format
+---rw md-name-string       string
+---rw md-level             int32
+---rw MAs!
    .
    .
    .

```

//the generic rpc command and notification are defined in this document

rpcs:

```

+---x continuity-check
|   +---ro input
|   |   +---ro technology           identityref
|   |   +---ro MD-name-string       MD-name-string
|   |   +---ro MA-name-string?      MA-name-string
|   |   +---ro (flow-entropy)?
|   |       +---:(flow-entropy-null)
|   |       |   +---ro flow-entropy-null?   empty
|   |   +---ro priority?            uint8
|   |   +---ro ttl?                 uint8
|   |   +---ro session-type?        enumeration
|   |   +---ro ecmp-choice?          ecmp-choices
|   |   +---ro sub-type?             identityref
|   |   +---ro outgoing-interfaces* [interface]
|   |       |   +---ro interface     if:interface-ref
|   |   +---ro source-mep?           MEP-name
|   |   +---ro destination-mp
|   |       |   +---ro (mp-address)?

```

```

| | | | +--:(mac-address)
| | | | | +--ro mac-address?      yang:mac-address
| | | | +--:(ipv4-address)
| | | | | +--ro ipv4-address?    inet:ipv4-address
| | | | +--:(ipv6-address)
| | | | | +--ro ipv6-address?    inet:ipv6-address
| | | +--ro (MEP-ID)?
| | | | +--:(MEP-ID-int)
| | | | | +--ro MEP-ID-int?      int32
| | | +--ro MEP-ID-format?      identityref
| | +--ro count?                uint32
| | +--ro interval?             Interval
| | +--ro packet-size?          uint32
| +--ro output
| | +--ro monitor-stats
| | | --+--:(monitor-null)
| | | | +--ro monitor-null?      Empty
+---x connectivity-verification
| +--ro input
| | +--ro technology              identityref
| | +--ro MD-name-string          MD-name-string
| | +--ro MA-name-string?         MA-name-string
| | +--ro (flow-entropy)?
| | | +--:(flow-entropy-null)
| | | | +--ro flow-entropy-null?  empty
| | +--ro priority?              uint8
| | +--ro ttl?                   uint8
| | +--ro session-type?          enumeration
| | +--ro ecmp-choice?           ecmp-choices
| | +--ro sub-type?              identityref
| | +--ro outgoing-interfaces* [interface]
| | | +--ro interface            if:interface-ref
| | +--ro source-mep?            MEP-name
| | +--ro destination-mp
| | | +--ro (mp-address)?
| | | | +--:(mac-address)
| | | | | +--ro mac-address?      yang:mac-address
| | | | +--:(ipv4-address)
| | | | | +--ro ipv4-address?    inet:ipv4-address
| | | | +--:(ipv6-address)
| | | | | +--ro ipv6-address?    inet:ipv6-address
| | | +--ro (MEP-ID)?
| | | | +--:(MEP-ID-int)
| | | | | +--ro MEP-ID-int?      int32
| | | +--ro MEP-ID-format?      identityref
| | +--ro count?                uint32

```

		+++ro interval?	Interval
		+++ro packet-size?	uint32

```

|  +-ro output
|    |  +-ro monitor-stats
|    |    |--+--:(monitor-null)
|    |    +-ro monitor-null?  Empty
+---x path-discovery
    +-ro input
    |  +-ro technology          identityref
    |  +-ro MD-name-string      MD-name-string
    |  +-ro MA-name-string?     MA-name-string
    |  +-ro (flow-entropy)?
    |    |  +--:(flow-entropy-null)
    |    |    +-ro flow-entropy-null?  empty
    |  +-ro priority?          uint8
    |  +-ro ttl?               uint8
    |  +-ro sub-type?          Identityref
    |  +-ro session-type?      enumeration
    |  +-ro ecmp-choice?       ecmp-choices
    |  +-ro outgoing-interfaces* [interface]
    |    |  +-ro interface      if:interface-ref
    |  +-ro source-mep?        MEP-name
    |  +-ro destination-mp
    |    |  +-ro (mp-address)?
    |    |    |  +--:(mac-address)
    |    |    |    |  +-ro mac-address?  yang:mac-address
    |    |    |    |  +--:(ipv4-address)
    |    |    |    |    |  +-ro ipv4-address?  inet:ipv4-address
    |    |    |    |    |  +--:(ipv6-address)
    |    |    |    |    |    |  +-ro ipv6-address?  inet:ipv6-address
    |    |  +-ro (MEP-ID)?
    |    |    |  +--:(MEP-ID-int)
    |    |    |    |  +-ro MEP-ID-int?  int32
    |    |  +-ro MEP-ID-format?  identityref
    |  +-ro count?              uint32
    |  +-ro interval?          Interval
+---ro output
    +-ro response* [response-index]
    |  +-ro response-index      uint8
    |  +-ro ttl?               uint8
    |  +-ro destination-mp

```

```

|   +--ro (mp-address)?
|   |   +--:(mac-address)
|   |   |   +--ro mac-address?      yang:mac-address
|   |   |   +--:(ipv4-address)
|   |   |   +--ro ipv4-address?     inet:ipv4-address
|   |   |   +--:(ipv6-address)
|   |   |   +--ro ipv6-address?     inet:ipv6-address
|   +--ro (MEP-ID)?
|   |   +--:(MEP-ID-int)

```

```

|   |   +--ro MEP-ID-int?      int32
|   +--ro MEP-ID-format?     identityref
|   +--ro monitor-stats
|   |--+--:(monitor-null)
|       +--ro monitor-null?   Empty

```

notifications:

```

+---n defect-condition-notification
+--ro technology      identityref
+--ro MD-name-string  MD-name-string
+--ro MA-name-string? MA-name-string
+--ro watching-mep-name? MEP-name
+--ro watching-mepid
|   +--ro (MEP-ID)?
|   |   +--:(MEP-ID-int)
|   |   +--ro MEP-ID-int?      int32
|   +--ro MEP-ID-format?     identityref
+--ro active-defects? identityref
+--ro error-message
|   |--+--:(status-null)
|       +--ro status-null?   Empty

```

data hierarchy of Management OAM

## 5. OAM YANG Module

```

<CODE BEGINS> file "ietf-gen-rpc.yang"
module ietf-gen-rpc {
    namespace "urn:ietf:params:xml:ns:yang:ietf-gen-rpc";
    prefix goam-rpc;
    import ietf-interfaces {
    prefix if;
    }

```

```

import ietf-inet-types {
prefix inet;
}
import ietf-yang-types {
prefix yang;
}
organization
"IETF LIME (Layer Independent OAM Management
in Multi-Layer Environment) Working Group";
contact
"wangzitao@huawei.com";
description
"This YANG module defines the generic
notification and rpc command for multi-layer
OAM management to be used
within IETF in a protocol independent manner.";

```

Wang & Wu

Expires October 11, 2015

[Page 11]

---

Internet-Draft

Layer Independent OAM Management

April 2015

```

revision 2014-12-16 {
description
"Initial revision.";
reference "RFC XXXX: A YANG Data Model
for Layer independent OAM management";
}

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

/*identities*/
identity technology-types {
description
"this is the base identity of technology types which are
vpls, nvo3, TRILL, ipv4, ipv6, mpls, etc";
}
identity defect-types {

```

```

    description
        "foo";
}

identity ipv4 {
    base technology-types;
    description
        "technology of ipv4";
}

identity ipv6 {
    base technology-types;
    description
        "technology of ipv6";
}

identity command-sub-type {
    description
        "defines different rpc command subtypes,
        e.g., rfc792 IP ping ,rfc4379 LSP ping,
        rfc6905 trill OAM, this is optional for most cases";
}

```

```

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

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

identity identifier-format {
    description
        "identifier-format identity can be augmented to define other

```

```

        format identifiers used in MEPD-ID etc";
    }

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

/*typedefs*/
    typedef MEP-direction {
        type enumeration {
            enum "Up" {
                value 0;
            }
            description
                "MEP direction up.";
        }
        enum "Down" {
            value 1;
        }
        description
            "MEP direction down.";
    }
    description
        "Describes the direction of MEP.";

}

    typedef MEP-name {

```

```

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

    typedef Interval {
        type uint32;
        units "milliseconds";
        default "1000";
        description
            "Interval between packets in milliseconds.

```

```

        0 means no packets are sent.";
    }

    typedef ecmp-choices {
        type enumeration {
            enum "ecmp-use-platform-hash" {
                value 0;
description
    "use platform hash.";
            }
            enum "ecmp-use-round-robin" {
                value 1;
description
    "use round robin.";
            }
        }
description
    "describes the algorithm of ecmp";
    }

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

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

    typedef oam-counter32 {
        type yang:zero-based-counter32;

```

```

        description
            "defines 32 bit counter for OAM";
    }

```



```

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

/*groupings*/
grouping topology {
    choice topology {
        case topo-null {
            description
                "this is a placeholder when no topology is needed";
            leaf topo-null {
                type empty;
                description
                    "there is no topology define, it will be defined
                    in technology specific model.";
            }
        }
    }
    description
        "describes the topology choice";
}
description
    "This grouping describes the topology";
}

grouping error-message {
    choice error {
        case error-null {
            description
                "this is a placeholder when no error status is needed";
            leaf error-null {
                type empty;
                description
                    "there is no error define, it will be defined in
                    technology specific model.";
            }
        }
    }
    description
        "describes the error message";
}

```

```
description
  "this grouping describes the error message";
}

grouping mp-address {
  choice mp-address {
    case mac-address {
      leaf mac-address {
        type yang:mac-address;
      }
      description
        "mac address";
    }
    case ipv4-address {
      leaf ipv4-address {
        type inet:ipv4-address;
      }
      description
        "ipv4 address";
    }
    case ipv6-address {
      leaf ipv6-address {
        type inet:ipv6-address;
      }
      description
        "ipv6 address";
    }
  }
  description
    "describes the mp-address";
}
description
  "describes the mp-address";
}

grouping maintenance-domain-id {
  //status current;
  description
    "Grouping containing leaves sufficient to identify an MD";
  leaf technology {
    type identityref {
      base technology-types;
    }
  }
  //status current;
  mandatory true;
  description
    "Defines the technology";
}
```

```
leaf MD-name-string {
```

```
    //status current;
    type MD-name-string;
    mandatory true;
description
    "Defines the generic administrative
    maintenance domain name";
}
}
grouping MD-name {
    leaf MD-name-format {
        type identityref {
            base name-format;
        }
description
    "Defines the md name format";
}
    choice MD-name {
        case MD-name-null {
            leaf MD-name-null {
                when "../..../MD-name-format = name-format-null"{
                    description
                        "describe condition met by MD-name-format";
                }
                type empty;
description
    "there is no MD NAME define, it will be defined in
    technology specific model.";
            }
        }
description
    "Defines the MD-name choice";
    }
description
    "Defines the MD-name grouping";
}
grouping ma-identifier {
    description
        "Grouping containing leaves sufficient to identify an MA";
    leaf MA-name-string {
        type MA-name-string;
    }
}
```

```

description
"define the MA name";
}
}
grouping MA-name {
  leaf MA-name-format {
    type identityref {
      base name-format;

```

```

    }
  description
  "define the MA name format";
}
choice MA-name {
  case MA-name-null {
    leaf MA-name-null {
      when "../..../MA-name-format = name-format-null"{
        description
        "describe condition met by MA-name-format";
      }
      type empty;
    }
  }
  description
  "there is no MA-name-format define,
  it will be defined in technology specific model";
}
}
description
"define the MA NAME";
}
description
"define the MA name grouping";
}

grouping MEP-ID {
  choice MEP-ID {
    default "MEP-ID-int";
    case MEP-ID-int {
      leaf MEP-ID-int {
        type int32;
      }
    }
  }
  description
  "define the MEP id int";
}

```

```

    }
description
  "define the MEP ID";
}
leaf MEP-ID-format {
  type identityref {
    base identifier-format;
  }
  description
    "define the MEP ID format";
}
description
  "define the MEP ID grouping";
}
grouping MEP {

```

```

    // status current;
    description
      "Defines elements within the MEP";
    leaf mep-name {
      type MEP-name;
mandatory true;
      //status current;
      description
        "Generic administrative name of the MEP";
    }
    uses MEP-ID;
    uses mp-address;
    uses connectivity-context;
    leaf Interface {
      type if:interface-ref;
      description
        "Interface name as defined by ietf-interfaces";
    }
    uses topology;
  }
  grouping session-type {
    leaf session-type{
      type enumeration {
        enum proactive {
          description

```

```

        "The current session is 'proactive.'";
    }
    enum on-demand {
        description
            "The current session is on-demand.";
    }
}
description
    "This object indicates whether the current
    session is defined to be proactive or on-demand.";
}
description
    "define the session type grouping";
}

```

```

grouping monitor-stats {
    description
        "grouping for monitoring statistics, this will be augmented
        by others who use this component";
    choice monitor-stats {
        default "monitor-null";
    }
}

```

```

        case monitor-null {
            description
                "this is a place holder when
                no monitoring statistics is needed";
            leaf monitor-null {
                type empty;
                description
                    "there is no monitoring statistics to be defined";
            }
        }
    }
description
    "define the monitor stats";
}
}
grouping MIP {
    description
        "defines MIP";
    leaf interface {
        type if:interface-ref;
    }
}

```

```

description
"define the interface";
}

grouping related-oam-layer {
  leaf offset {
    type int32 {
      range "-255..255";
    }
    description
      "defines offset (in MD levels) to a related OAM layer
      +1 is the layer immediately above
      -1 is the layer immediately below";
  }
  uses maintenance-domain-id;
  uses ma-identifier;
description
"define the related oam layer";
}

grouping interface-status {
  description
    "collection of interface related status";
  leaf admin-status {
    type leafref {
      path "/if:interfaces-state/if:interface/if:admin-status";
    }
  }
  config false;
}

```

```

    description
      "oper status from ietf-interface module";
  }
  leaf oper-status {
    type leafref {
      path "/if:interfaces-state/if:interface/if:oper-status";
    }
    config false;
    description
      "oper status from ietf-interface module";
  }
}

```

```

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

grouping priority {
    description
        "Priority used in transmitted packets; for example, in the
        TOS/DSCP field in IP or the Traffic Class field in MPLS";
    leaf priority {
        type uint8;
    }
    description
        "define the priority which be used in
        transmitted packets.";
}

grouping flow-entropy {
    description

```

```

        "defines the grouping statement for flow-entropy";
    choice flow-entropy {
        default "flow-entropy-null";
        case flow-entropy-null {
            description

```



```

        "this is a place holder when no flow entropy is needed";
    leaf flow-entropy-null {
        type empty;
        description
            "there is no flow entropy defined";
    }
}
description
    "define the flow entropy";
}
}

notification defect-condition-notification {
    description
        "When defect condition is met this notificiation is sent";
    uses maintenance-domain-id {
        description
            "defines the MD (Maintenance Domain) identifier,
which is the Generic MD-name-string
and the technology.";
    }
    uses ma-identifier;
    leaf mep-name {
        type MEP-name;
        description
            "Indicate which MEP is seeing the error";
    }
    container remote-mepid {
        uses MEP-ID;
        description
            "Who is seeing the error (if known)
if unknown make it 0.";
    }
    leaf defect-type{
        type identityref {
            base defect-types;
        }
    }
    description
        "The currently active defects on the specific MEP.";}
    uses error-message {
        description
            "Error message to indicate more details.";
    }
}

```

```

}

rpc continuity-check {
  description
    "Generates continuity check and return response";
  input {
    uses maintenance-domain-id {
      description
        "defines the MD (Maintenance Domain) identifier,
        which is the generic MD-name-string
        and the technology.";
    }

    uses ma-identifier {
      description
        "identifies the Maintenance association";
    }
    uses flow-entropy;
    uses priority;
    leaf ttl {
      type uint8;
      default "255";
    }
    description
      "define the ttl";
  }
  uses session-type;
  leaf ecmp-choice {
    type ecmp-choices;
    description
      "0 means use the specified interface
      1 means use round robin";
  }
  leaf sub-type {
    type identityref {
      base command-sub-type;
    }
    description
      "defines different command types";
  }
  list outgoing-interfaces {
    key "interface";
    description
      "list of outgoing interface";
    leaf interface {
      type if:interface-ref;
      description
        "define the interface";
    }
  }
}

```

Internet-Draft

Layer Independent OAM Management

April 2015

```
    }
  }
  leaf source-mep {
    type MEP-name;
    description
    "define the source mep";
  }
  container destination-mp {
    description
    "this container collect a set of parameters
    of destination mep";
    uses mp-address;
    uses MEP-ID {
      description
      "Only applicable if the destination is a MEP";
    }
  }
  leaf count {
    type uint32;
    default "3";
    description

    "Number of ping echo request message to send";
  }
  leaf interval {
    type Interval;
    description
    "Interval between echo requests";
  }
  leaf packet-size {
    type uint32 {
      range "64..10000";
    }
    default "64";
    description
    "Size of ping echo request packets, in octets";
  }
}
output {
  uses monitor-stats {
    description
    "Stats of continuity check is same as
```

```

        that of monitor sessions";
    }
}
}
rpc connectivity-verification {
    if-feature connectivity-verification;

```

```

description
    "Generates connectivity verification and return response";
input {
    uses maintenance-domain-id {
        description
            "defines the MD (Maintenance Domain) identifier,
            which is the generic MD-name-string and
            the technology.";
    }

    uses ma-identifier {
        description
            "identifies the Maintenance association";
    }

    uses flow-entropy;
    uses priority;
    leaf ttl {
        type uint8;
        default "255";
    }
    description
        "define the ttl leaf";
    }

    uses session-type;
    leaf ecmp-choice {
        type ecmp-choices;
        description
            "0 means use the specified interface
            1 means use round robin";
    }
    leaf sub-type {
        type identityref {
            base command-sub-type;
        }
        description
            "defines different command types";
    }

```

```

    }
    list outgoing-interfaces {
        key "interface";
    description
    "a list of outgoing interface";
        leaf interface {
            type if:interface-ref;
            description
            "define the interface";
        }
    }
    leaf source-mep {
        type MEP-name;
    }

```

```

    description
    "define the source mep";
    }
    container destination-mp {
    description
    "this container collect a set of
    destination mp parameters";
        uses mp-address;
        uses MEP-ID {
            description
            "Only applicable if the destination is a MEP";
        }
    }
    leaf count {
        type uint32;
        default "3";
        description

        "Number of ping echo request message to send";
    }
    leaf interval {
        type Interval;
        description
        "Interval between echo requests";
    }
    leaf packet-size {
        type uint32 {
            range "64..10000";
        }
    }

```

```

    }
    default "64";
    description
        "Size of ping echo request packets, in octets";
    }
}
output {
    uses monitor-stats {
        description
            "Stats of connectivity verification is
            same as that of monitor sessions";
    }
}
}
rpc path-discovery {
    description
        "Generates path discovery and return response.
        Starts with TTL of one and increment by one at each hop.
        Untill destination reached or TTL reach max valune";
    input {

```

```

    uses maintenance-domain-id {
        description
            "defines the MD (Maintenance Domain) identifier,
            which is the generic MD-name-string and
            the technology.";
    }
    uses ma-identifier {
        description
            "identfies the Maintenance association";
    }
    uses flow-entropy;
    uses priority;
    leaf ttl {
        type uint8;
        default "255";
    }
    description
        "define the ttl";
    }
    leaf command-sub-type {
        type identityref {
            base command-sub-type;

```

```

    }
    description
        "defines different command types";
    }
    uses session-type;
    leaf ecmp-choice {
        type ecmp-choices;
        description
            "0 means use the specified interface
            1 means use round robin";
    }
    list outgoing-interfaces {
        key "interface";
    }
    description
        "list of outgoing interface";
        leaf interface {
            type if:interface-ref;
        }
        description
            "define the interface";
    }
    leaf source-mep {
        type MEP-name;
    }
    description
        "define the source mep";
    }
    container destination-mp {

```

```

description
    "this container collect a set of
    destination mp";
    uses mp-address;
    uses MEP-ID {
        description
            "Only applicable if the destination is a MEP";
    }
    }
    leaf count {
        type uint32;
        default "1";
    }
    description
        "Number of traceroute probes to send.

```

```

        In protocols where a separate message is sent at each TTL,
        this is the number of packets to send at each TTL.";
    }
    leaf interval {
        type Interval;
        description
            "Interval between echo requests";
    }
}
output {
    list response {
        key "response-index";
        description
            "a list of path discovery response";
        leaf response-index {
            type uint8;
            description
                "Arbitrary index for the response.
                In protocols which guarantee there is only
                a single response at each TTL
                (e.g., IP Traceroute), the TTL can be used
                as the response index.";
        }
        leaf ttl {
            type uint8;
            description
                "define the ttl leaf";
        }
        container destination-mp {
            description
                "MP from which the response has been received";
            uses mp-address;
            uses MEP-ID {
                description

```

```

        "Only applicable if the destination is a MEP";
    }
}
uses monitor-stats {
    description
        "If count is 1,
        there is a single delay value reported.";

```



```
    }  
  }  
}
```

<CODE ENDS>

## [6.](#) Security Considerations

TBD.

## [7.](#) IANA Considerations

TBD.

## [8.](#) References

### [8.1.](#) Normative References

- [I-D.edprop-opsawg-multi-layer-oam]  
Wu, Q., "Problem Statement for Layer and Technology Independent OAM in a Multi- Layer Environment", ID <http://tools.ietf.org/html/draft-edprop-opsawg-multi-layer-oam-02>, September 2014.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", [RFC 6020](#), October 2010.
- [RFC6241] Enns, R., Bjorklund, M., Schoenwaelder, J., and A. Bierman, "Network Configuration Protocol (NETCONF)", [RFC 6241](#), June 2011.

## 8.2. Informative References

[I-D.tissa-lime-yang-oam-model]

Senevirathne , T. and Q. Wu, "YANG Data Model for Generic Operations, Administration, and Maintenance (OAM)", ID <http://tools.ietf.org/html/draft-tissa-lime-yang-oam-management-01>, October 2014.

### Authors' Addresses

Zitao 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

