Applicability of Generic YANG Data Model for layer Independent OAM Management
draft-zhuang-lime-yang-oam-model-applicability-00

Abstract

A generic YANG data model for Operations, Administration, and Maintenance (OAM) has been defined in [GENYANGOAM], with the intention that technology-specific extensions will be developed to complete the implementation of the model. In this document, we describe the applicability of the generic YANG OAM data model to specific OAM technologies. To be concrete, we also demonstrate the usability and extensibility of the generic YANG OAM model with OAM protocols such as IP Ping, traceroute, and LSP Ping.

Status of This Memo

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

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

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

This Internet-Draft will expire on December 30, 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

1. Introduction ........................................ 3
2. Conventions Used In This Document .................. 3
   2.1. Terminology ........................................ 4
3. Basic Structure of Generic YANG Model for OAM .......... 4
4. Guidelines For Extending the LIME Base Data Model .... 6
   4.1. Importing the Base Model Into the Technology-Specific OAM Model ........................ 7
   4.2. Using "augment" to Insert Additional Data Into the Base Model ................................ 7
   4.3. Using "identity" and "identityref" to Identify New Technology Types ............................ 8
   4.4. Using the Choice and Case Nodes .................... 10
   4.5. Define New RPCs and Notifications .................. 11
5. Applicability of LIME Model to Various Technologies ... 12
   5.1. IP OAM ............................................. 12
      5.1.1. Technology Type Extension ....................... 12
      5.1.2. MEP Configuration Extension ...................... 12
      5.1.3. Connectivity-Context Extension ................. 13
      5.1.4. RPC Extension .................................. 13
      5.1.5. ECMP extension ................................ 13
   5.2. IPPM (IP Performance Measurement) .................. 13
      5.2.1. MEP Configuration Extension ...................... 14
      5.2.2. RPC Extension .................................. 14
   5.3. Ethernet OAM ....................................... 14
      5.3.1. Technology Type Extension ....................... 14
      5.3.2. MEP Configuration Extension ...................... 15
      5.3.3. Connectivity-Context Extension ................. 15
      5.3.4. RPC Extension .................................. 15
      5.3.5. ECMP Extension ................................ 15
      5.3.6. Relation with Service OAM FM Modules .......... 16
   5.4. TRILL OAM ......................................... 16
1. Introduction

The YANG [RFC6020] over NETCONF [RFC6241] data model for Layer independent OAM Management defined in [GENYANGOAM] provides consistent configuration, reporting and representation of OAM mechanisms at any layer for any technology.

This draft describes the applicability of the generic YANG OAM model to various transport technologies and demonstrates that the YANG model(s) in the LIME WG are usable and extensible for those technologies. The demonstration uses IP, Ethernet, NVO3, TRILL, and MPLS as specific examples.

2. Conventions Used In This Document

This document contains no normative language.
2.1. Terminology

MP Maintenance Point [IEEE802.1Q].

MEP Maintenance association End Point [RFC7174] [IEEE802.1Q] [RFC6371].

MIP Maintenance domain Intermediate Point [RFC7174] [IEEE802.1Q] [RFC6371].

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

MD Maintenance Domain [IEEE802.1Q].

OAM Operations, Administration, and Maintenance [RFC6291].

TRILL Transparent Interconnection of Lots of Links [RFC6325].

3. Basic Structure of Generic YANG Model for OAM

As the basis of this document, the generic YANG model for OAM specified as the LIME base model is shown in Figure 1.
The generic YANG OAM model comprises three definitions for configuration and operational state data:

- configuration model definition;
- Remote procedure call (RPC) definition;
- notification definition.

The configuration model definition provides hierarchical structure to describe fault domain (i.e., maintainence domain), test point (i.e., maintainence point), technology type, layering, and session context for trouble-shooting. This basic configuration model enables users to select corresponding layers and nodes serving as anchor points to define their specific technology OAM YANG models.
The RPC definition provides uniform APIs for common OAM functions such as continuity check, connectivity verification, path discovery, performance measurement and their equivalents. These APIs are used by the network management system (NMS) to control and manipulate OAM tools and functionalities on network elements for measuring and monitoring the data plane (e.g., LSP Ping, IP performance measurement protocol). These OAM tools can be activated on a continual basis or manually to detect a specific anomaly.

The notification definition also provides a uniform API to report defects, faults, and network failures at different layers. This API is used by network elements to report to the network management system (NMS). The content of each notification includes the fault domain and the test point(s) that detected the fault and generated the error message. This API must be activated on a continual basis.

4. Guidelines For Extending the LIME Base Data Model

As described in Section 3, the LIME base model provides a hierarchical structure for configuration, notification and RPCs. Each of these three aspects should be extended with technology-specific features and parameters relating to each technology of interest.

YANG allows a module to insert additional nodes into data models, including both the current module (and its submodules) or an external module. This is useful to let specific technologies add specific parameters into the LIME base model.

Here we summarize five steps to extend the LIME base model for specific technologies:

1. Import the base model in the technology specific OAM model to reuse the structure defined in the base model.

2. Use "augment" to insert additional data into the base model (e.g., in the MEP) to specify extensions for the specific technology.

3. Use the "identity" and "identityref" constructs (instead of "enum") to identify new technology types.

4. Use "choice" and "case" nodes instead of "typedef".

5. Use "feature" to define optional functionality within the data model.
6. Define new RPCs and notifications in the technology specific OAM data model.

4.1. Importing the Base Model Into the Technology-Specific OAM Model

YANG allows a module to reference external modules to reuse data already defined in those modules. Therefore a technology-specific model can import data from the LIME base model.

The import statements are used to make definitions available inside other modules [RFC6020]. Users who want to develop a technology-specific OAM model should import the ietf-gen-oam YANG model with the following statements:

```yml
module ietf-xxx-oam {
    namespace "urn:foo:params:xml:ns:yang:ietf-xxx-oam";
    prefix xxxoam;

    import ietf-gen-oam {
        prefix goam;
    }
}
```

4.2. Using "augment" to Insert Additional Data Into the Base Model

As described in [RFC6020], the "augment" statement defines the location in the data model hierarchy where new nodes are inserted. By using the "augment" statement, the hierarchy of the LIME base model can be extended with new data nodes that express technology-specific parameters to meet the requirements of the respective technologies.

The technology-specific model developer must take care to select the right layers and nodes in the LIME base model as anchor points to insert these additional data. For example, assume a technology-specific OAM YANG model A. An "a" node needs to be inserted within the MA (Maintenance Association):
augment /goam:domains/goam:domain/goam:MAs/goam:MA:
   +--a? foo

Corresponding YANG encoding:

augment "/goam:domains/goam:domain/goam:MAs/goam:MA"{
    leaf a{
        type foo
        description
           "foo";
    }
}

4.3. Using "identity" and "identityref" to Identify New Technology Types

[GENYANGOAM] uses the "identity" statement to define a new globally
unique, abstract, and untyped "technology-types" base identity. As
described below, each technology-specific module specifies a
 corresponding concrete identity using this base: ipv4, ipv6, vpls,
nvo3, TRILL, mpls, etc. [GENYANGOAM] itself defines the ipv4 and
ipv6 identities.

[GENYANGOAM] also defines a "command-sub-type" abstract identity for
different RPC commands, e.g., to distinguish the types of IP ping
[RFC792], LSP ping [RFC4379], TRILL OAM [TRILLOAMYANG]. Use of this
identity is optional for most cases.

The corresponding statements are shown as below.
identity technology-types {
  description
  "this is the base identity of technology types such as
  vpls, nvo3, TRILL, ipv4, ipv6, mpls, etc";
}

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, TRILLOAMYANG trill OAM, this is
  optional for most cases";
}

identity icmp-rfc792 {
  base command-sub-type;
  description
  "Defines the command subtypes for ICMP ping";
  reference "RFC 792";
}

With the base identity of technology types and command-sub-type, ietf-gen-oam YANG model users who want to develop a specific OAM technology model can define their own technology-types and/or command-sub-type identity.

For example, if we want to define a specific OAM technology YANG model A, we could define a corresponding technology-types by using following statements:

identity A {
  base goam:technology-types;
  description
  "Technology A";
}
With this definition of a technology type A, the technology "leaf" under the domains list can reference this new type.

4.4. Using the Choice and Case Nodes

YANG allows the data model to segregate incompatible nodes into distinct choices using the "choice" and "case" statements. The "choice" statement contains a set of "case" statements that define sets of schema nodes that cannot appear together. Each "case" may contain multiple nodes, but each node may appear in only one "case" under a "choice". Note that when an element from one case is created, all elements from all other cases are implicitly deleted [RFC6020].

In order to support flexible augment, ietf-gen-oam defined a set of choice and case nodes to serve as anchor points for specific OAM technologies. Note that the default values of these nodes -- MD-name, MA-name, connectivity-context, flow-entropy, et cetera -- are empty.

For example:

```
+--rw MAs
   +--rw MA* [MA-name-string]
   ...
   +--rw (connectivity-context)?
     |    +--:(context-null)
     |    +--rw context-null? Empty
```

Corresponding YANG encoding:

```yang
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 defined";
    }
  }
  description
    "connectivity context";
}
```
ietf-gen-oam YANG model users who want to define a specific OAM technology model can augment the corresponding choice node by defining a new case to carry technology specific extensions.

For example, for a specific OAM technology YANG model A, an "a" node is needed to indicate the connectivity context for this specific OAM technology. To achieve this, it is only necessary to augment the connectivity-context choice node in the ietf-gen-OAM YANG model by defining a "connectivity-context-A" case as:

```
augment /goam:domains/goam:domain/goam:MAs/goam:MA
   /goam:connectivity-context:
       +--:(connectivity-context-A)
           +---a?  foo
```

Corresponding YANG encoding:

```
augment "/goam:domains/goam:domain/goam:MAs/goam:MA"
+"/goam:connectivity-context" {  
    case connectivity-context-A {   
        leaf a{
            type foo;}
    }
}
```

### 4.5. Define New RPCs and Notifications

The LIME base model presents three basic RPCs: continuity check, connectivity verification and path discovery. Technology-specific OAM models can either extend the existing RPCs and notifications defined in the LIME base model or define new RPCs and notifications if generic RPCs and notifications cannot be reused to meet their requirements.

For example, a Multicast Tree Verification (MTV) [TRILLOAMYANG] RPC command is defined in the TRILL OAM model to verify connectivity as well as data-plane and control-plane integrity of TRILL multicast forwarding as follows:
As mentioned above, the ietf-gen-oam model describes the abstract common core configuration, statistics, RPCs, and notifications for layer independent OAM management.

Following guidelines stated in Section 4, ietf-gen-oam YANG model users can augment this base model by defining and adding new data nodes with technology specific functions and parameters into proper anchor points of the ietf-gen-oam model, so as to develop a technology-specific OAM model.

With these guidelines in hand, this section further demonstrates the usability of the ietf-gen-oam YANG model to various OAM technologies. Note that, in this section, we only present several snippets of technology-specific data model extensions for illustrative purposes. The complete model extensions should be worked on in respective protocol working groups.

5.1. IP OAM

5.1.1. Technology Type Extension

The technology types ipv4 and ipv6 have already been defined in the LIME base model. Therefore no technology type extension is required in the IP OAM model.

5.1.2. MEP Configuration Extension

MEP configuration in the LIME base model already supports configuring the interface on which the MEP is located with an IP address. There is no additional MEP configuration extension needed for IP OAM.
However, IP Ping, traceroute and session monitoring do not use the MEPID in their message headers. Therefore it is important to have a method to derive the MEPID in an automatic manner with no user intervention.

5.1.3. Connectivity-Context Extension

In IP OAM, the connectivity-context is a 12 bit VLAN ID. The LIME base model defines a placeholder for connectivity-context. This allows other technologies to easily augment it to include technology specific extensions. The snippet below depicts an example of augmenting context-id to include VLAN ID.

```plaintext
  ++-(context-id-vlan)
    ++-rw context-id-vlan?   vlan
  ++-(context-id-vlan)
    ++-rw context-id-vlan?   vlan
```

5.1.4. RPC Extension

Technology type in the RPC definition has already been defined in the LIME OAM base model. Therefore no technology type extension is required in the RPC definition. For IP OAM, IP Ping and IP Traceroute RPCs need to be supported. For the IP OAM model, the continuity-check RPC with IPv4 or IPv6 as technology type can be mapped to the IP Ping RPC, while the path-discovery RPC with IPv4 or IPv6 as technology type can be mapped to IP Traceroute.

5.1.5. ECMP extension

The flow-entropy parameter in the LIME OAM configuration model is an optional parameter. Since standard IP OAM protocols, e.g., IP Ping and Traceroute, don’t support ECMP path selection, the flow-entropy parameter does not need to be supported in the IP OAM model.

5.2. IPPM (IP Performance Measurement)

Editor Note: IP performance measurement (PM) and IP Ping and Traceroute are discussed separately based on the RFC7276 classification of OAM technologies. Although IPPM and IP OAM are both applied to the IP network, based on Table 4 of RFC7276, IP OAM does not support performance measurement. It is necessary to use OWAMP and TWAMP, defined in IPPM, for that purpose.
5.2.1. MEP Configuration Extension

To support IP performance measurement, MEP configuration in the LIME base model needs to be extended with:

- loss-stats-group: grouping object for loss measurement session statistics.
- measurement-timing-group: grouping object used for proactive and on-demand scheduling of PM measurement sessions.
- delay-measurement-configuration-group: grouping configuration object for the delay measurement function.
- delay-measurement-stats-group: grouping object for delay measurement session statistics.

5.2.2. RPC Extension

To support IP performance measurement, four RPCs need to be defined in the IPPM model:

- create-loss-measurement RPC: allows scheduling of one-way or two-way on-demand or proactive performance monitoring loss measurement sessions.
- abort-loss-measurement RPC: allows aborting of currently running or scheduled loss measurement session.
- create-delay-measurement RPC: allows scheduling of one-way or two-way on-demand or proactive performance monitoring delay measurement sessions.
- abort-delay-measurement RPC: allows aborting of currently running or scheduled delay measurement sessions.

5.3. Ethernet OAM

5.3.1. Technology Type Extension

No Ethernet technology type is defined in the LIME base model. Therefore a technology type extension for Ethernet technology is required in the Ethernet OAM model as follows:
identity ethernet {
    base technology-types;
    description
        "ethernet technology";
}

5.3.2. MEP Configuration Extension

MEP configuration definition in the LIME base model already supports configuring the interface on which MEP is located with MAC address. No additional MEP configuration extension is required.

5.3.3. Connectivity-Context Extension

In Ethernet OAM, the connectivity-context is a 12 bit VLAN ID. The LIME base model defines a placeholder for context-id. This allows other technologies to easily augment that to include technology specific extensions. The snippet below depicts an example of augmenting context-id to include VLAN ID.

```
augment /goam:domains/goam:domain/goam:MAs
    /goam:MA/goam:MEP/goam:connectivity-context:
        +--:(context-id-vlan)
            +--rw context-id-vlan?   vlan
    /goam:session/goam:connectivity-context:
        +--:(context-id-vlan)
            +--rw context-id-vlan?   vlan
```

5.3.4. RPC Extension

Technology type in the RPC definition reuses the ethernet identity defined earlier. For Ethernet OAM, Connectivity check, Loopback and linktrace RPCs need to be supported. The connectivity-check RPC in the LIME base model with Ethernet as the technology type can be used to represent Ethernet connectivity check. The base model connectivity-verification RPC with Ethernet as the technology type can be mapped to Ethernet loopback. The base model path-discovery RPC with Ethernet as the technology type can be mapped to Ethernet linktrace.

5.3.5. ECMP Extension

The flow-entropy parameter in the LIME OAM configuration model is an optional parameter. Standard Ethernet OAM protocols, i.e., connectivity check, loopback and linktrace do not support ECMP path selection, so the flow-entropy parameter does not need to be supported in the Ethernet OAM model.
5.3.6. Relation with Service OAM FM Modules

Note that Service OAM Fault Management YANG Modules defined in [MEF-38] are one example of an Ethernet OAM model, but were developed before the LIME base model was proposed. As a result, the Service OAM Fault Management YANG Modules defined in [MEF-38] are not compliant with LIME base model nor do they extend from LIME base model.

5.4. TRILL OAM

5.4.1. Technology Type Extension

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

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

5.4.2. MEP Configuration Extension

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

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

5.4.3. Connectivity-Context Extension

In TRILL OAM, connectivity-context is either a 12 bit VLAN ID or a 24 bit Fine Grain Label. The LIME base model defines a placeholder for context-id. This allows other technologies to easily augment that to include technology specific extensions. The snippet below depicts an example of augmenting connectivity-context to include either VLAN ID or Fine Grain Label.
augment /goam:domains/goam:domain/goam:MA\n/goam:MA /goam:connectivity-context:
  ---:(connectivity-context-vlan)
      ---rw connectivity-context-vlan?   vlan
  ---:(connectivity-context-fgl)
      ---rw connectivity-context-fgl?    fgl

augment /goam:domains/goam:domain/goam:MA\n/goam:session/goam:connectivity-context:
  ---:(connectivity-context-vlan)
      ---rw connectivity-context-vlan?   vlan
  ---:(connectivity-context-fgl)
      ---rw connectivity-context-fgl?    fgl

5.4.4. RPC Extension

In the TRILL OAM YANG model, the continuity-check and path-discovery
RPC commands are extended with TRILL specific requirements. The
snippet below illustrates the TRILL OAM RPC extension.
The flow-entropy parameter in the LIME base model is an optional parameter. Since TRILL supports ECMP path selection, flow-entropy in TRILL is defined as a 96 octet field. The snippet below illustrates its extension.
5.4.6. Performance Management (PM) Extension

5.4.6.1. MEP Configuration Extension

To support performance measurement for TRILL, MEP configuration in the LIME base model needs to be extended with:

- loss-stats-group: grouping statistics object for TRILL Loss measurement sessions;
- measurement-timing-group: grouping object used for proactive and on-demand scheduling of PM measurement sessions;
- delay-measurement-configuration-group: grouping configuration object for TRILL delay measurement function;
- delay-measurement-stats-group: grouping statistics object for TRILL delay measurement sessions.

5.4.6.2. RPC Extension

To support performance measurement for TRILL, four new RPCs need to be defined in the TRILL OAM PM model:

- create-loss-measurement RPC: allows scheduling of one-way or two-way on-demand or proactive performance monitoring loss measurement sessions.
- abort-loss-measurement RPC: allows aborting of currently running or scheduled loss measurement sessions.
- create-delay-measurement RPC: allows scheduling of one-way or two-way on-demand or proactive performance monitoring delay measurement sessions.
- abort-delay-measurement RPC: allows aborting of currently running or scheduled delay measurement sessions.

5.4.7. Usage example

This part gives a simple example of implementing the TRILL OAM model onto network devices.

The scenario is shown in Figure 2, in which there are two companies: A and B. Both have departments in City 1 and City 2. Meanwhile, different departments within the same company should be able to communicate with each other. However, the communication services of these two companies should be segregated from each other.
To meet the requirements above, two Ethernet Lease line, E-Line-1 and E-Line-2, are set between NE1 and NE3: to isolate the communication traffic between two companies. VLAN 100 associates port 3-EFF8-1 of NE1 facing with company A while VLAN 200 associates port 3-EF8-2 of NE1 facing with company B. For network maintenance, NE1, NE2 and NE3 are within a same maintenance domain: MD1. Two maintenance associations MA1 and MA2 are configured and stand for E-Line-1 and E-Line-2 under MD1. The MAC addresses of NE1, NE2, NE3 are 00-1E-4C-84-22-F1, 00-1E-4C-84-22-F2, 00-1E-4C-84-22-F3 respectively.

![Figure 2: TRILL OAM scenario](image)

5.4.7.1. Ethernet OAM Extension

To fulfill the Ethernet OAM configuration, the lIME base model should be extended by augmenting the connectivity-context and inserting a port node in the MEP list. The snippet below illustrates the Ethernet OAM model extension.

```plaintext
augment /goam:domains/goam:domain/goam:MAs
    /goam:MA/goam:MEP /goam:mep-address:
        +--:( mep-address-trill)
            | +--rw mep-address-trill? tril-rb-nickname
augment /goam:domains/goam:domain/goam:MAs/goam:MA
    /goam:connectivity-context:
        +--:(connectivity-context-vlan)
            | +--rw connectivity-context-vlan? vlan
        +--:(connectivity-context-fgl)
            +--rw connectivity-context-fgl? fgl
    /goam:session/goam:connectivity-context:
        +--:(connectivity-context-vlan)
            | +--rw connectivity-context-vlan? vlan
        +--:(connectivity-context-fgl)
```

5.4.7.2. Corresponding XML Instance Example

This section gives an example of the corresponding XML instance for devices to implement the example Ethernet OAM data models in Section 5.4.7.1.

```xml
<domains>
    <domains>
```

---rw connectivity-context-fgl? fgl
augment /goam:domains/goam:domain/goam:MAAs/goam:MA
/goam:MEP/goam:flow-entropy:
    +---:(flow-entropy-trill)
    +---rw flow-entropy-trill? flow-entropy-trill
/goam:session/goam:flow-entropy:
    +---:(flow-entropy-trill)
    +---rw flow-entropy-trill? flow-entropy-trill
augment /goam:continuity-check/goam:input:
    +---ro (out-of-band)?
        | +---:(ipv4-address)
        |    | +---ro ipv4-address? inet:ipv4-address
        | +---:(ipv6-address)
        |    | +---ro ipv6-address? inet:ipv6-address
        | +---:(trill-nickname)
        |    +---ro trill-nickname? tril-rb-nickname
    +---ro diagnostic-vlan? boolean
augment /goam:continuity-check/goam:input/goam:flow-entropy:
    +---:(flow-entropy-trill)
    +---ro flow-entropy-trill? flow-entropy-trill
augment /goam:continuity-check/goam:output:
    +---ro upstream-rbridge? tril-rb-nickname
    +---ro next-hop-rbridge* tril-rb-nickname
augment /goam:path-discovery/goam:input:
    +---ro (out-of-band)?
        | +---:(ipv4-address)
        |    | +---ro ipv4-address? inet:ipv4-address
        | +---:(ipv6-address)
        |    | +---ro ipv6-address? inet:ipv6-address
        | +---:(trill-nickname)
        |    +---ro trill-nickname? tril-rb-nickname
    +---ro diagnostic-vlan? boolean
augment /goam:path-discovery/goam:input/goam:flow-entropy:
    +---:(flow-entropy-trill)
    +---ro flow-entropy-trill? flow-entropy-trill
augment /goam:path-discovery/goam:output/goam:response:
    +---ro upstream-rbridge? tril-rb-nickname
    +---ro next-hop-rbridge* tril-rb-nickname
<technology> ethernet </technology>
<MD-name-string> MD1 </MD-name-string>
<MA>
  <MA-name-string> MA1 </MA-name-string>
  <connectivity-context>
    <connectivity-context-vlan> 100 </connectivity-context-vlan>
  </connectivity-context>
  <MEP>
    <mep-name> NE1 </mep-name>
    <mp-address>
      <mac-address> 00-1E-4C-84-22-F1 </mac-address>
    </mp-address>
  </MEP>
</MA>

<MA>
  <MA-name-string> MA2 </MA-name-string>
  <connectivity-context>
    <connectivity-context-vlan> 200 </connectivity-context-vlan>
  </connectivity-context>
  <MEP>
    <mep-name> NE1 </mep-name>
    <port> 3-EFF8-2 </port>
    <mp-address>
      <mac-address> 00-1E-4C-84-22-F1 </mac-address>
    </mp-address>
  </MEP>
</MA>

<connection-context>
  <connection-context-vlan> 100 </connection-context-vlan>
</connection-context>

<MEP>
  <mep-name> NE3 </mep-name>
  <mp-address>
    <mac-address> 00-1E-4C-84-22-F3 </mac-address>
  </mp-address>
</MEP>
</MA>

<MEP>
  <mep-name> NE1 </mep-name>
  <port> 3-EFF8-1 </port>
  <mp-address>
    <mac-address> 00-1E-4C-84-22-F1 </mac-address>
  </mp-address>
</MEP>
</MA>

<MEP>
  <mep-name> NE3 </mep-name>
</MEP>
</MIP>
5.5. NVO3 OAM

5.5.1. Technology Type Extension

No NVO3 technology type has been defined in the LIME base model. Therefore technology type extension is required in the NVO3 OAM model. The technology type "nvo3" is defined as an identity that augments the base "technology-types" defined in the LIME base model:

```plaintext
identity nvo3{
  base goam:technology-types;
  description
    "nvo3 type";
}
```

5.5.2. Sub Technology Type Extension

In NVO3, since different overlay encapsulation types such as VxLAN, NVGRE can be employed, the "technology-sub-type" data node is defined and added into the LIME base model to further identify the overlay types within the NVO3 model. Based on it, we also define a technology sub-type for VxLAN encapsulation. NVGRE and GENEVE, sub-types can be defined in the same way.
identity technology-sub-type {
  description
  "certain implementations such as nvo3 can have different 
  encapsulation types such as vxlan, nvgre and so on. 
  Instead of defining separate models for each 
  encapsulation, we define a technology sub-type to 
  further identify different encapsulations. Technology 
  sub-type is associated at the MA level";
}

identity technology-sub-type-vxlan {
  base technology-sub-type;
  description
  "technology sub-type is vxlan";
}

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

5.5.3. MEP Configuration Extension

In NVO3, the MEP address is either an IPv4 or IPV6 address. In the 
LIME base model, MEP address is defined as an IP address and the same 
definition can be used for NVO3 with no further modification.

5.5.4. Connectivity-Context Extension

In NVO3 context-id is a 24 bit virtual network identifier (VNI). The 
LIME base model defines a placeholder for context-id. This allows 
other technologies to easily augment that to include technology 
specific extensions. The snippet below depicts an example of 
augmenting context-id to include VNI.

augment "/goam:domains/goam:domain/goam:MAs/goam:MA 
/goam:connectivity-context"
  {
    case connectivity-context-nvo3 {
      leaf vni {
        type vni;
      }
    }
  }
5.5.5. RPC Extension

In the NVO3 OAM YANG model, the End-Station-Locator RPC command is defined. This command locates an end-station within the NVO3 deployment. [PTT -- what other tools are applicable??? Presumably one can use ICMP Ping, LSP Ping for CV, and the PM extensions, per RFC 7276 Table 4.]

5.5.6. ECMP Extension

In NVO3, flow-entropy depends on the technology sub-type, e.g., VxLAN. Technology sub-type is used to extend the base model to specific usage. The snippet below illustrates the extension for VxLAN.

```y additive
augment "/goam:domains/goam:domain/goam:MAAs/goam:MA
/goam:flow-entropy" {
  case flow-entropy-vxlan {
    leaf flags-vxlan {
      type flags-vxlan;
    }
    leaf flow-entropy-vxlan {
      type flow-entropy-vxlan;
    }
  }
}
```

5.6. MPLS OAM

5.6.1. Technology Type Extension

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

```y additive
identity mpls{
  base goam:technology-types;
  description
    "mpls type";
}
```

5.6.2. MEP Configuration Extension

In MPLS, the MEP address is either an IPv4 or IPV6 address. MEP-ID is either a 2 octet unsigned integer value or a variable length label value. In the LIME base model, MEP-ID is defined as a variable
length label value and the same definition can be used for MPLS with no further modification.

5.6.3. Connectivity-Context Extension

In MPLS context-id is a 24 bit MPLS label. The LIME base model defines a placeholder for context-id. This allows other technologies to easily augment that to include technology specific extensions. The snippet below depicts an example of augmenting context-id to include per VRF MPLS labels in IP VPN or per CE-PE MPLS labels in IP VPN.

```yang
augment "/goam:domains/goam:domain/goam:MAAs/goam:MA
/goam:connectivity-context"
{
    case connectivity-context-mpls {
        leaf vrf-label {
            type vrf-label;
        }
    }
}
```

5.6.4. ECMP Extension

Since MPLS supports ECMP path selection, the flow-entropy should be defined in MPLS OAM model. Technology type is used to extend the YANG model to specific usage.

```yang
augment "/goam:domains/goam:domain/goam:MAAs/goam:MA
/goam:flow-entropy" {
    case flow-entropy-mpls {
        leaf flags-mpls {
            type flags-mpls;
        }
        leaf flow-entropy-mpls {
            type flow-entropy-mpls;
        }
    }
}
```

5.6.5. Usage Example

In the MPLS tunnel scenario (see Figure 3): tunnel_1 is a static LSP tunnel passing through NE1-NE2-NE4. It is used to perform LSP PING. tunnel_3 is another static LSP tunnel passing through NE4-NE2-NE1, used to bring back the LSP PING result. tunnel_2 is a third static LSP tunnel passing through NE1-NE3-NE4, used to perform LSP
Traceroute. tunnel_4 is a fourth static LSP tunnel passing through NE4-NE3-NE1, used to bring back the LSP Traceroute result.

Figure 3: MPLS OAM scenario

5.6.5.1. MPLS OAM Model Extension

TBD.

5.6.5.2. Corresponding XML Instance Example

TBD.

6. Open Issues

Not all usage examples of technology-specific OAM model are specified.

How technology specific OAM model is used to provide consistent configuration, reporting is not specified.

How LIME based model is used to provide consistent configuration and reporting is not specified.
7. Security Considerations

TBD.

8. IANA Considerations

This document registers the following namespace URI in the IETF XML registry.

URI:TBD

9. References

9.1. Normative References

[GENYANGOAM]

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


9.2. Informative References


Authors’ Addresses

Yan Zhuang
Huawei Technologies Co., Ltd
101 Software Avenue, Yuhua District
Nanjing 210012
China

Email: zhuangyan.zhuang@huawei.com

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

Email: wangzitao@huawei.com

Daniel King
Lancaster University
UK

Email: daniel@olddog.co.uk
Qin Wu
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: bill.wu@huawei.com

Tom Taylor
PT Taylor Consulting
Ottawa
Canada

Email: tom.taylor.stds@gmail.com