Network Working Group Internet-Draft Intended status: Standards Track Expires: May 31, 2016 Z. Wang Q. Wu Huawei D. Kumar Cisco T. Taylor PT Taylor Consulting November 28, 2015

Generic YANG Data Model for Operations, Administration, and Maintenance (OAM) Performance Management <u>draft-wang-lime-yang-pm-01</u>

Abstract

This document presents a YANG Data model for OAM Performance management support. The YANG Model presented in this document extends the Generic YANG Data Model for OAM with Loss Measurement and Delay measurement to support OAM Performance management.

Status of This Memo

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

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 <u>http://datatracker.ietf.org/drafts/current/</u>.

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 May 31, 2016.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to <u>BCP 78</u> and the IETF Trust's Legal Provisions Relating to IETF Documents (<u>http://trustee.ietf.org/license-info</u>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

Wang, et al.

Expires May 31, 2016

[Page 1]

Gen OAM PM Model

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

$\underline{1}$. Introduction	2
$\underline{2}$. Conventions and Terminology	<u>2</u>
<u>2.1</u> . Terminology	<u>3</u>
$\underline{3}$. YANG Extension	<u>4</u>
<u>3.1</u> . MEP Configuration Extension	<u>4</u>
<u>3.1.1</u> . Loss Measurement Configuration	<u>4</u>
<u>3.2</u> . Delay Measurement Configuration	<u>5</u>
<u>3.3</u> . rpc definitions	<u>8</u>
<u>3.3.1</u> . create-loss-measurement	<u>8</u>
<u>3.3.2</u> . abort-loss-measurement	<u>8</u>
<u>3.3.3</u> . create-delay-measurement	<u>8</u>
<u>3.3.4</u> . abort-delay-measurement	<u>8</u>
$\underline{4}$. PM data hierarchy	<u>8</u>
<u>5</u> . PM YANG Module	<u>14</u>
<u>6</u> . Security Considerations	<u>35</u>
$\underline{7}$. IANA Considerations	<u>35</u>
<u>8</u> . References	<u>35</u>
<u>8.1</u> . Normative References	<u>35</u>
<u>8.2</u> . Informative References	<u>35</u>
Authors' Addresses	<u>36</u>

1. Introduction

Generic OAM Yang model [<u>GENYANGGOAM</u>] presents Generic Yang data model for all OAM technologies.

In this document we extend the YANG model defined in [<u>GENYANGGOAM</u>] with Loss Measurement and Delay measurement to support OAM Performance management. Details are provided in <u>section 4</u> below.

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 [<u>RFC2119</u>].

The following terms are defined in $[\underline{RFC6241}]$ and are not redefined here:

o client

[Page 2]

- o configuration data
- o server
- o state data

The following terms are defined in $[\underline{\mathsf{RFC6020}}]$ and are not redefined here:

- o augment
- o data model
- o data node

The terminology for describing YANG data models is found in [<u>RFC6020</u>].

<u>2.1</u>. Terminology

- MP Maintenance Point [8021.Q]
- MEP Maintenance End Point [RFC7174] [8021Q] [RFC6371]
- MIP Maintenance Intermediate Point [RFC7174] [8021.Q] [RFC6371]
- MA Maintenance Association [8021.Q] [RFC7174]
- MD Maintenance Domain [8021.Q]
- OAM Operations, Administration, and Maintenance [RFC6291]
- DMM Delay Measurement Message
- 1SL One-way Synthetic Loss Measurement message
- 1DM One-way Delay Measurement message
- DMR Delay Measurement Reply
- PM Performance Monitoring
- SLM Synthetic Loss Measurement Message
- SLR Synthetic Loss Measurement Reply

Wang, et al. Expires May 31, 2016 [Page 3]

3. YANG Extension

MEP Addressing is defined in Generic YANG OAM model[GENYANGGOAM]. In this draft we augment MEP Configuration with Performance Management configuration and statistics for Delay Measurement and Performance Measurement. In addtion, we define new rpcs for performance measurement.

3.1. MEP Configuration Extension

3.1.1. Loss Measurement Configuration

This section describes a set of data definitions for Loss Measurement configuration. This set of data definitions augment the MEP configuration defined in the Generic YANG OAM model with parameters related to Loss Measurement and define the role of MEP and measurement method of the loss measurement.

module: ietf-gen-oam-pm

```
/*This set of data definitions defines the role of MEP.*/
   augment /goam:domains/goam:domain/goam:MAs/goam:MA/goam:MEP:
     +--rw loss-responder?
                             Boolean
/*This set of data definitions defines loss measurement configuration*/
   augment /goam:domains/goam:domain/goam:MAs/goam:MA/goam:MEP:
     +--rw loss-measurements?
        +--ro loss-measurements* [session-cookie]
         | +--ro session-cookie
                                   uint32
           +--ro id?
                                   string
                                   Boolean
         | +--ro status?
        +--rw measurement-type
         | +--rw slm?
                                   boolean
           +--rw pm 1sl?
                                      boolean
        +--rw enabled-counters
         +--rw availability-forward-high-loss?
                                                             boolean
         +--rw availability-forward-availables?
                                                             boolean
         +--rw availability-forward-unavailable?
                                                             boolean
         +--rw availability-backward-high-loss?
                                                             boolean
         +--rw availability-backward-consecutive-high-loss? boolean
         +--rw availability-backward-available?
                                                             boolean
           +--rw available-backward-unavailable?
                                                             boolean
         +--rw message-period?
                                          uint32
        +--rw data-pattern?
                                          enumeration
        +--rw measurement-interval?
                                          uint32
        +--rw number-intervals-stored?
                                          uint32
        +--rw availability-measurement-interval? uint32
        +--rw availability-consecutive-intervals-number? unit32
        +--rw session-type?
                                          enumeration
```

[Page 4]

```
+--rw thresholds* [id]
  +--rw id
                                 uint32
  +--rw enabled-threshold
  +--rw forward-unavailable-count? boolean
  +--rw backward-unavailable-count? boolean
  +--rw forward-available-ratio?
                                      boolean
 +--rw forward-available-ratio?
                                      boolean
+--rw forward-unavailable-count uint32
  +--rw backward-unavailable-count uint32
+--rw forward-available-ratio? uint32
  +--rw backward-available-ratio?
                                   uint32
+--rw start-time
  +--rw (start-time)?
+--:(immediate)
     | +--rw immediate!
     +--:(absolute)
L
        +--rw absolute? yang:date-and-time
+--rw stop-time
 +--rw (stop-time)?
+--:(none)
     +--rw none!
     +--:(absolute)
+--rw absolute? yang:date-and-time
+--rw availability-forward-status
                                     enumeration
+--rw availability-backward-status
                                     enumeration
+--rw current-stats
+--rw start-time?
                                   yang:date-and-time
 +--rw elapsed-time?
                                             uint32
+--rw suspect-status?
                                             boolean
  +--rw forward-available?
                                   yang:gauge32
+--rw backward-available?
                                   yang:gauge32
+--rw forward-unavailable?
                                   yang:gauge32
+--rw backward-unavailable?
                                   yang:gauge32
+--rw history-stats* [id]
  +--rw id
                                             uint32
  +--rw start-time?
                                 yang:date-and-time
  +--rw elapsed-time?
                                             uint32
  +--rw suspect-status?
                                             Boolean
  +--rw forward-available?
                                   yang:gauge32
  +--rw backward-available?
                                   yang:gauge32
  +--rw forward-unavailable?
                                   yang:gauge32
  +--rw backward-unavailable?
                                   yang:gauge32
```

3.2. Delay Measurement Configuration

This section describes data definition for Delay Measurement configuration. This data definition augments the MEP configuration defined in the Generic YANG OAM model with parameters related to

[Page 5]

Delay Measurement and defines the role of MEP and measurement method of the delay measurement.

Internet-Draft

```
module: ietf-gen-oam-pm
/*This set of data definitions defines the role of MEP.*/
     augment /goam:domains/goam:domain/goam:MAs/goam:MA/goam:MEP:
        +--rw delay-responder?
                                 Boolean
/*This set of data definitions defines delay measurement configuration*/
     augment /goam:domains/goam:domain/goam:MAs/goam:MA/goam:MEP:
     +--rw delay-measurements?
        +--ro delay-measurements* [session-cookie]
         | +--ro session-cookie
                                   uint32
         | +--ro id?
                                   string
                                   boolean
         | +--ro status?
        +--rw measurement-type
                                   boolean
         | +--rw dmm?
         +--rw dm1-transmitted? boolean
         +--rw dm1-received?
                                   boolean
        +--rw measurement-enable
        +--rw message-period?
                                                      uint32
        +--rw data-pattern?
                                                      enumeration
        +--rw measurement-interval?
                                                      uint32
        +--rw number-intervals-stored?
                                                      uint32
        +--rw session-type?
                                                      Enumeration
        +--rw thresholds* [id]
         | +--rw id
                                                      uint32
         +--rw enabled-thresholds?
                                                      bits
         +--rw start-time
         +--rw (start-time)?
              +--:(immediate)
         +--rw immediate!
         +--:(absolute)
         +--rw absolute? yang:date-and-time
         +--rw stop-time
         L
          +--rw (stop-time)?
              +--:(none)
         +--rw none!
              +--:(absolute)
                 +--rw absolute? yang:date-and-time
        +--rw current-stats
          +--rw start-time?
                                               yang:date-and-time
         +--rw elapsed-time?
                                                           uint32
          +--rw suspect-status?
                                                           boolean
        +--rw history-stats* [id]
           +--rw id
                                                           uint32
                                               yang:date-and-time
           +--rw start-time?
           +--rw elapsed-time?
                                                           uint32
           +--rw suspect-status?
                                                           boolean
```

[Page 7]

Gen OAM PM Model

<u>3.3</u>. rpc definitions

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

Configuration data in the <u>Section 3.2</u> and <u>Section 3.1.1</u> provides input extension for Delay and Loss measurement RPCs.

<u>3.3.1</u>. create-loss-measurement

RPC allows scheduling of one-way or two-way on-demand or proactive performance monitoring loss measurement session.

3.3.2. abort-loss-measurement

RPC allow aborting of currently running or scheduled loss measurement session.

3.3.3. create-delay-measurement

RPC allow scheduling of one-way or two-way on-demand or proactive performance monitoring delay measurement session.

3.3.4. abort-delay-measurement

RPC allow aborting of currently running or scheduled delay measurement session.

<u>4</u>. PM data hierarchy

The complete data hierarchy related to the OAM PM YANG is presented below. The following notations are used within the tree and carry the meaning as noted below.

Each node is printed as:

Wang, et al. Expires May 31, 2016 [Page 8]

```
<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
 module: ietf-gen-oam-pm
 augment /goam:domains/goam:domain/goam:MAs/goam:MA/goam:MEP:
   +--rw delay-responder?
                            Boolean
   +--rw loss-responder?
                            Boolean
augment /goam:domains/goam:domain/goam:MAs/goam:MA/goam:MEP:
  +--rw delay-measurements?
    +--ro delay-measurements* [session-cookie]
     +--ro session-cookie uint32
     | +--ro id?
                               string
                               boolean
     | +--ro status?
    +--rw measurement-type
                               boolean
     | +--rw dmm?
      +--rw dm1-transmitted? boolean
     +--rw dm1-received?
                               boolean
    +--rw measurement-enable
    +--rw message-period?
                                                  uint32
    +--rw data-pattern?
                                                   enumeration
```

[Page 9]

```
+--rw measurement-interval?
                                                 uint32
   +--rw number-intervals-stored?
                                                 uint32
   +--rw session-type?
                                                 enumeration
   +--rw start-time
     +--rw (start-time)?
         +--:(immediate)
         +--rw immediate!
         +--:(absolute)
            +--rw absolute?
                               yang:date-and-time
   +--rw stop-time
     +--rw (stop-time)?
         +--:(none)
         +--rw none!
         +--:(absolute)
            +--rw absolute? yang:date-and-time
   +--rw current-stats
    +--rw start-time?
                                          yang:date-and-time
    +--rw elapsed-time?
                                                      uint32
    +--rw suspect-status?
                                                      Boolean
   +--rw history-stats* [id]
      +--rw id
                                                      uint32
      +--rw start-time?
                                          yang:date-and-time
      +--rw elapsed-time?
                                                      uint32
      +--rw suspect-status?
                                                     boolean
augment /goam:domains/goam:domain/goam:MAs/goam:MA/goam:MEP:
  +--rw loss-measurements?
     +--ro loss-measurements* [session-cookie]
      +--ro session-cookie
                                uint32
      | +--ro id?
                                string
                                boolean
      +--ro status?
     +--rw measurement-type
                                boolean
     | +--rw slm?
       +--rw pm 1sl?
                                   boolean
     +--rw enabled-counters
      +--rw availability-forward-high-loss?
                                                          boolean
      +--rw availability-forward-availables?
                                                          boolean
      +--rw availability-forward-unavailable?
                                                          boolean
      +--rw availability-backward-high-loss?
                                                          Boolean
      +--rw availability-backward-consecutive-high-loss? boolean
     +--rw availability-backward-available?
                                                          boolean
        +--rw available-backward-unavailable?
                                                          boolean
     +--rw message-period?
                                       uint32
     +--rw data-pattern?
                                       enumeration
     +--rw measurement-interval?
                                       uint32
     +--rw number-intervals-stored?
                                       uint32
     +--rw availability-measurement-interval? uint32
     +--rw availability-consecutive-intervals-number? unit32
```

rpcs:

```
+--rw session-type?
                                    enumeration
   +--rw thresholds* [id]
     +--rw id
                                    uint32
      +--rw enabled-threshold
       +--rw forward-unavailable-count? boolean
      +--rw backward-unavailable-count? boolean
      +--rw forward-available-ratio?
                                          boolean
      +--rw forward-available-ratio?
                                          boolean
     +--rw forward-unavailable-count uint32
     +--rw backward-unavailable-count uint32
     +--rw forward-available-ratio? uint32
     +--rw backward-available-ratio? uint32
   +--rw start-time
   +--rw (start-time)?
        +--:(immediate)
        +--rw immediate!
        +--:(absolute)
           +--rw absolute? yang:date-and-time
   +--rw stop-time
     +--rw (stop-time)?
        +--:(none)
         | +--rw none!
        +--:(absolute)
            +--rw absolute? yang:date-and-time
   +--rw current-stats
     +--rw start-time?
                                     yang:date-and-time
   +--rw elapsed-time?
                                                 uint32
     +--rw suspect-status?
                                                 Boolean
   +--rw forward-available?
                                       yang:gauge32
     +--rw backward-available?
                                       yang:gauge32
   +--rw forward-unavailable?
                                       yang:gauge32
   +--rw backward-unavailable?
                                       yang:gauge32
   +--rw history-stats* [id]
     +--rw id
                                                 uint32
      +--rw start-time?
                                     yang:date-and-time
     +--rw elapsed-time?
                                                 uint32
                                                 boolean
      +--rw suspect-status?
     +--rw forward-available?
                                       yang:gauge32
      +--rw backward-available?
                                       yang:gauge32
     +--rw forward-unavailable?
                                       yang:gauge32
     +--rw backward-unavailable?
                                       yang:gauge32
+---x create-loss-measurement
   +--ro input
   +--rw measurement-type
   | | +--rw slm?
                                boolean
      | +--rw pm_1sl?
                                   boolean
      +--rw enabled-counters
```

+--rw availability-forward-high-loss? boolean +--rw availability-forward-availables? boolean +--rw availability-forward-unavailable? boolean +--rw availability-backward-high-loss? boolean +--rw availability-backward-consecutive-high-loss? boolean +--rw availability-backward-available? boolean +--rw available-backward-unavailable? boolean +--ro message-period? uint32 +--ro data-pattern? enumeration +--ro measurement-interval? uint32 +--ro number-intervals-stored? uint32 +--ro session-type? enumeration +--ro start-time +--ro (start-time)? +--:(immediate) +--ro immediate! +--:(absolute) +--ro absolute? yang:date-and-time +--ro stop-time +--ro (stop-time)? +--:(none) | +--ro none! +--:(absolute) +--ro absolute? yang:date-and-time +--ro destination-mep +--ro (mp-address)? +--:(mac-address) +--ro mac-address? yang:mac-address +--:(ipv4-address) +--ro ipv4-address? inet:ipv4-address L +--:(ipv6-address) Τ +--ro ipv6-address? inet:ipv6-address +--ro (MEP-ID)? +--:(MEP-ID-int) +--ro MEP-ID-int? int32 +--ro MEP-ID-format? identityref +--ro output +--ro session-id uint32 +---x abort-loss-measurement +--ro input +--ro technology identityref +--ro MD-name-string MD-name-string +--ro MA-name-string? MA-name-string +--ro destination-mep +--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 L +--ro (MEP-ID)? +--:(MEP-ID-int) +--ro MEP-ID-int? int32 +--ro MEP-ID-format? identityref +--ro session-id uint32 +---x create-delay-measurement +--ro input +--ro measurement-type +--ro dmm? boolean +--ro dm1-transmitted? boolean +--ro dm1-received? boolean +--ro measurement-enable uint32 +--ro message-period? +--ro data-pattern? enumeration +--ro measurement-interval? uint32 +--ro number-intervals-stored? uint32 +--ro session-type? enumeration +--ro start-time +--ro (start-time)? +--:(immediate) +--ro immediate! +--:(absolute) +--ro absolute? yang:date-and-time +--ro stop-time +--ro (stop-time)? +--:(none) | +--ro none! +--:(absolute) +--ro absolute? yang:date-and-time +--ro destination-mep +--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 Τ +--ro MEP-ID-format? identityref +--ro (flow-entropy)? +--:(flow-entropy-null) +--ro flow-entropy-null empty 1 +--ro output

```
+--ro session-id
                      uint32
1
+---x abort-delay-measurement
  +--ro input
     +--ro technology
                            identityref
     +--ro MD-name-string
                          MD-name-string
     +--ro MA-name-string?
                            MA-name-string
     +--ro destination-mep
     +--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 session-id
                            uint32
```

data hierarchy of PM

5. PM YANG Module

```
<CODE BEGINS> file "ietf-gen-oam-pm.yang"
module ietf-gen-oam-pm {
    namespace "urn:ietf:params:xml:ns:yang:ietf-gen-oam-pm";
    prefix goampm;
    import ietf-gen-oam {
        prefix goam;
      }
    import ietf-yang-types {
        prefix yang;
      }
    organization
      "IETF LIME (Layer Independent OAM Management in Multi-Layer
       Environment) Working Group";
    contact
      "zitao wang: wangzitao@huawei.com";
    description
      "This YANG module defines generic loss measurement and
       delay measurement configuration for multi-layer OAM
```

}

```
management to be used within IETF in a protocol independent
  manner.";
revision 2015-01-07 {
    description
      "Initial revision.";
    reference "draft-wang-lime-yang-pm";
 }
   /* features */
feature loss-measurements {
 description
    "This feature indicates that the server supports
     configuration of loss measurement that is used by some OAM
     protocols.
     Servers that do not advertise this feature will not
     configure loss measurement";
}
feature delay-measurements {
  description
    "This feature indicates that the server supports
     configuration of delay measurement that is used by some OAM
     protocols.
     Servers that do not advertise this feature will not
     configure delay measurement";
}
feature create-loss-measurement{
 description
    "This feature indicates that the server supports executing
     the create-loss-measurement OAM command and returning a
     response. Servers that do not advertise this feature will
     not support execution of the create-loss-measurement command
     or the RPC model for the create-loss-measurement command.";
}
feature abort-loss-measurement{
 description
    "This feature indicates that the server supports executing
     the abort-loss-measurement OAM command and returning a
     response. Servers that do not advertise this feature will
     not support execution of the abort-loss-measurement command
     or the RPC model for the abort-loss-measurement command.";
```

```
feature create-delay-measurement{
 description
    "This feature indicates that the server supports executing
    the create-delay-measurement OAM command and returning a
    response. Servers that do not advertise this feature will
    not support execution of the create-delay-measurement
    command or the RPC model for the create-delay-measurement
    command.";
}
feature abort-delay-measurement{
 description
    "This feature indicates that the server supports executing
    the abort-delay-measurement OAM command and returning a
    response. Servers that do not advertise this feature will
    not support execution of the abort-delay-measurement
    command or the RPC model for the abort-delay-measurement
    command.";
}
grouping loss-measurement-configuration-group {
 description
    "This grouping includes configuration objects for the loss
    measurement function defined in GEN-PM.";
  reference
    "draft-wang-lime-yang-pm";
 container measurement-type {
   leaf slm
              {
      type boolean;
      description
        "If true, generate PM SLM Messages and correlate with
         received SLR responses.";
   }
   leaf pm_1sl{
      type boolean;
      description
        "If true, generate PM 1SL Messages.";
   }
   description
      "This object specifies what type of loss measurement
      will be performed.";
 }
 container enabled-counters {
   leaf availability-forward-high-loss {
      type boolean;
```

Internet-Draft

```
default "false";
  description
    "Enable (true) or disable availability-forward-high-loss
     counters found in the current-stats and history-stats.";
}
leaf availability-forward-available {
  type boolean;
  default "false";
  description
    "Enable (true) or disable availability-forward-available
    counters found in the current-stats and history-stats ";
}
leaf availability-forward-unavailable {
  type boolean;
  default "false";
  description
    "Enable (true) or disable
     availability-forward-unavailable counters found in the
     current-stats and history-stats ";
}
leaf availability-backward-high-loss {
  type boolean;
  default "false";
  description
    "Enable (true) or disable availability-backward-high-loss
     counters found in the current-stats and history-stats ";
}
leaf availability-backward-consecutive-high-loss {
  type boolean;
  default "false";
  description
    "Enable (true) or disable
     availability-backward-consecutive-high-loss counters
     found in the current-stats and history-stats ";
}
leaf availability-backward-available {
  type boolean;
  default "false";
  description
    "Enable (true) or disable availability-backward-available
    counters found in the current-stats and history-stats ";
}
leaf availability-backward-unavailable {
  type boolean;
  default "false";
  description
    "Enable (true) or disable available-backward-unavailable
```

```
counters found in the current-stats and history-stats ";
  }
  description
    "Indicates the types of PM loss measurement counters found
     in the current-stats and history-stats that are enabled.
     Not all counters are supported for all PM Loss Measurement
     types.";
}
/* This terminates the enabled-counters clause */
leaf message-period {
  type uint32;
 units "ms";
 default 1000;
  description
    "This object specifies the interval between loss
    measurement OAM message transmissions.";
}
leaf data-pattern {
  type enumeration {
    enum zeroes {
      description "Indicates the Data TLV contains all Os.";
    }
    enum ones {
      description "Indicates the Data TLV contains all 1s.";
    }
  }
  default zeroes;
  description
    "This object specifies the loss measurement data pattern
     included in the OAM Message.";
}
leaf measurement-interval {
  type uint32;
  units "minutes";
  default 15;
  description
    "This object specifies a measurement interval in
    minutes. Measurements are accumulated in the current-stats
     counters for the duration of this interval, and then
     transferred to the history-stats counters.";
}
leaf number-intervals-stored {
```

```
type uint32 {
    range "2..10";
  }
  default 10;
  description
    "This object specifies the number of completed
     measurement intervals to store in the history statistics
     table.";
}
leaf availability-measurement-interval {
  type uint32 {
    range "1..525600";
  }
  units minutes;
  default 15;
  description
    "This object specifies the availability measurement
     interval in minutes. Measurement interval of 15 minutes
     MUST be supported, other intervals can be supported.";
}
leaf availability-consecutive-intervals-number {
  type uint32 {
    range "1..1000";
  }
  default 10;
  description
    "This object specifies a configurable number of
     consecutive availability indicators to be used to
     determine a change in the availability status as
     indicated by MEF 10.2.1.
     This parameter is equivalent to the Availability
     parameter 'n' as specified by MEF 10.2.1. The number
     range of 1 through 10 MUST be supported. The number
     range of 1 through 1000 may be supported, but is not
     mandatory.";
}
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.";
}
list thresholds {
  key "id";
  description
    "This list contains the list of Loss Measurement
     configuration threshold values for LM Performance
     Monitoring. The main purpose of the threshold
     configuration list is to configure threshold alarm
     notifications indicating that a specific performance
     metric is not being met.";
  leaf id {
    type uint32;
    description
      "The index of the threshold number for the specific LM
       threshold entry. An index value of '1' MUST be
       supported. Other index values can also be supported.";
  }
  container enabled-thresholds {
    leaf forward-unavailable-count {
      type boolean;
      description
        "If true, indicates that the forward-unavailable-count
          is available.";
    }
    leaf forward-available-ratio {
      type boolean;
      description
        "If true, indicates that the forward-available-ratio
         is available.";
    }
    leaf backward-unavailable-count {
      type boolean;
      description
        "If true, indicates that the backward-unavailable-count
         is available.";
    }
    leaf backward-available-ratio {
      type boolean;
      description
```

```
"If true, indicates that the backward-available-ratio
       is available.";
  }
  description
    "A container that indicates the type of OAM loss
     measurement threshold notifications that are enabled.";
}
/* Terminates container enabled-thresholds */
leaf forward-unavailable-count {
  type uint32;
  description
    "This object is used to set the forward unavailability
     threshold value that will be used to determine if a
     threshold notification is generated.";
}
leaf backward-unavailable-count {
  type uint32;
  description
    "This object is used to set the backward unavailability
     threshold value that will be used to determine if a
     threshold notification is generated.";
}
leaf forward-available-ratio {
  type uint32 {
    range "0..100000";
  }
  units milli-percent;
  default 0;
  description
    "This object is used to set the forward
     availability/total time ratio threshold value that will
     be used to determine if a threshold notification is
     generated if the ratio drops below the configured value.
     The ratio value is expressed as a percent with a value
     of 0 (ratio 0.00) through 100000 (ratio 1.00) Units are
     in milli-percent, where 1 indicates 0.001 percent.";
}
leaf backward-available-ratio {
  type uint32 {
    range "0..100000";
  }
  units milli-percent;
  default 0;
  description
    "This object is used to set the backward
     availability/total time ratio threshold value that will
```

Internet-Draft

```
be used to determine if a thresh-old notification is
         generated if the ratio drops below the configured value.
         The ratio value is expressed as a percent with a value
         of 0 (ratio 0.00) through 100000 (ratio 1.00) Units are
         in milli-percent, where 1 indicates 0.001 percent.";
   }
 }
 /* This terminates list thresholds */
}
/* This terminates loss-measurement-configuration-group */
grouping loss-stats-group {
 description
    "This grouping includes statistics objects for a loss
     measurement session.";
 leaf suspect-status {
    type boolean;
    description
        "true means statistics for this measurement interval are
         not valid.";
 }
}
grouping measurement-timing-group {
 description
    "This grouping includes objects used for proactive and
     on-demand scheduling of PM measurement sessions.";
 container start-time {
    description
      "This container defines the session start time.";
    choice start-time {
      description
        "Measurement session start time can be immediate,
         relative, or absolute.";
      container immediate {
        description
          "Start the measurement session immediately.";
      }
      leaf relative {
     type yang:timeticks;
      description
     "This object specifies the relative start time.";
     }
      leaf absolute {
```

```
type yang:date-and-time;
        description
          "This object specifies the scheduled start time to
           perform the on-demand monitoring operations.";
      }
   }
 }
 container stop-time {
    description
      "This container defines the session stop time.";
    choice stop-time {
      description
        "Measurement session stop time can be none, or
         absolute.";
      container none {
        description
          "Never end the measurement session.";
      }
      leaf absolute {
        type yang:date-and-time;
        description
          "This object specifies the scheduled stop time
           to perform the on-demand monitoring operations.";
      }
   }
 }
}
/* End of measurement-timing-group */
grouping delay-measurement-configuration-group {
 description
    "This grouping includes configuration objects for Delay
     Measurement function defined in PM.";
 reference
    "draft-wang-lime-yang-pm";
 container measurement-type {
    description
      "This container defines the measurement type.";
    leaf dmm {
      type boolean;
      description
        "If true, generate DMM Message, correlate with DMR
```

```
responses.";
  }
  leaf dm1-transmitted {
    type boolean;
    description
      "If true, generate 1DM Message.";
  }
  leaf dm1-received {
    type boolean;
    description
      "Receive 1DM PDU and generate measurement.";
 }
}
container measurement-enable {
  description
    "Indicates the types of DM counters that are enabled. Not
     all DM counters are supported for all DM types.";
}
leaf message-period {
 type uint32;
  default 100;
  description
    "This object specifies the interval between delay
      measurement OAM message transmissions.";
}
leaf data-pattern {
  type enumeration {
    enum zeroes {
      description "Indicates the Data TLV contains all Os.";
    }
    enum ones {
      description "Indicates the Data TLV contains all 1s.";
    }
  }
  default zeroes;
  description
    "This object specifies the delay measurement data pattern
     included in the OAM packet.";
}
leaf measurement-interval {
 type uint32;
 units minutes;
  default 15;
```

```
description
          "This object specifies a Measurement Interval in minutes.";
      }
      leaf number-intervals-stored {
        type uint32 {
          range "2..10";
        }
        default 10;
        description
          "This object specifies the number of completed measurement
           intervals to store in the history statistics table.";
      }
      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.";
      }
    }
    /* End of delay-measurement-configuration-group */
    grouping delay-measurement-stats-group {
      description
        "This grouping includes statistics objects for a delay
         measurement session.";
      leaf suspect-status {
        type boolean;
        description
          "true means statistics for this measurement interval are
           not valid.";
     }
    }
    /* End of delay-measurement-stats-group */
/*This set of data definitions defines the role of MEP.*/
    augment "/goam:domains/goam:domain"
```

```
+"/goam:MAs/goam:MA/goam:MEP" {
        description
          "This set of data definitions extends the MEP as described
           in goam";
        leaf delay-responder {
          type boolean;
          default true;
          description
            "This object specifies whether Delay Measurement (DMM)
             single ended Responder is enabled. The value 'false'
             indicates the Delay measurement responder is disabled
             and received DMM will be discarded.";
        }
        leaf loss-responder {
          type boolean;
          default true;
          description
            "This object specifies whether Loss Measurement (LMM)
             single ended Responder is enabled. The value 'false'
             indicates the Loss measurement responder is disabled and
             received LMM will be discarded.";
        }
    }
/*This set of data definitions defines performance measurement */
/*configuration.*/
    augment "/goam:domains/goam:domain"
      +"/goam:MAs/goam:MA/goam:MEP" {
        description
          "This set of data definitions extends the MEP as described
           in goam, specially with regard to delay measurements.";
        container delay-measurements {
          if-feature delay-measurements;
            description
              "This container contains a collection of data
               definitions related to Delay Measurements as defined
               in PM.";
          list delay-measurements {
            key "session-cookie";
            config false;
            description
              "List of Delay Measurement PM Sessions where each
               instance is uniquely identified by an session-cookie
```

```
attribute.";
 leaf session-cookie {
    type uint32;
   config false;
   description
      "cookie to identify delay measurement session.";
 }
 leaf id {
    type string;
   description
      "This object uniquely identifies a scheduled
       measurement time.";
 }
 leaf status {
   type boolean;
   config false;
   description
      "This object indicates DM session status, true means
       active, false means not active.";
 }
}
/* End of list delay-measurements */
uses delay-measurement-configuration-group;
uses measurement-timing-group;
container current-stats {
 description
    "This container contains result of the current
     Measurement Interval in a delay measurement
     session gathered during the interval indicated by
     measurement-interval.";
 leaf start-time {
    type yang:date-and-time;
   description
      "Start time for current measurement interval.";
 }
 leaf elapsed-time {
    type uint32;
   units "tens of ms";
    description
      "Elapsed time for current measurement interval in
      0.01 seconds.";
```

}

```
}
          uses delay-measurement-stats-group;
      }
      /* End of current-stats */
      list history-stats {
        key id;
        description
          "This list contains the result for historic measurement
           intervals for performance measurement session.";
        leaf id {
          type uint32;
          description
            "This id can be used to identify different history
             stats.";
        }
        leaf start-time {
          type yang:date-and-time;
          description
            "Start time for measurement interval.";
        }
        leaf elapsed-time {
          type uint32;
          units "tens of ms";
          description
            "Elapsed time for measurement interval in 0.01
             seconds.";
         }
        uses delay-measurement-stats-group;
      }
      /* End of history-stats */
    }
    /* End of container delay-measurements */
/* End of augment clause */
augment "/goam:domains/goam:domain"
 +"/goam:MAs/goam:MA/goam:MEP" {
    description
      "This set of data definitions extends the MEP as described
       in goam, specially with regards to loss measurements.";
    container loss-measurements {
```

```
if-feature loss-measurements;
description
  "This container contains a collection of data definitions
   related to loss measurements as defined in this document.";
list loss-measurements {
  key "session-cookie";
 config false;
 description
    "List of Loss Measurement PM Sessions where each
     instance is uniquely identified by an session-cookie
     attribute.";
 leaf session-cookie {
    type uint32;
   config false;
   description
      "Cookie to identify loss measurement session.";
 }
 leaf id {
    type string;
   description
      "This object uniquely identifies a scheduled measurement
       time.";
 }
 leaf status {
    type boolean;
   config false;
   description
      "This object indicates loss measurement session status,
       true means active, false means not active.";
 }
}
/* End of list loss-measurements */
uses loss-measurement-configuration-group;
uses measurement-timing-group;
leaf availability-forward-status {
  type enumeration {
   enum available {
      description
        "Indicates the MEP is available.";
    }
    enum unavailable {
      description
```

```
"Indicates the MEP is unavailable.";
   }
   enum unknown {
      description
        "Indicates the availability is not known.";
   }
  }
  description
    "This object indicates the availability status in the
     forward direction.";
}
leaf availability-backward-status {
  type enumeration {
    enum available {
      description
        "Indicates the MEP is available.";
    }
    enum unavailable {
      description
        "Indicates the MEP is unavailable.";
    }
   enum unknown {
      description
        "Indicates the availability is not known.";
   }
 }
 description
    "This object indicates the availability status
     in the backward direction.";
}
container current-stats {
  description
    "This container contains result of the current
     measurement interval in a PM loss measurement session
     gathered during the interval indicated by
     measurement-interval.";
 leaf start-time {
    type yang:date-and-time;
   description
      "Start time for current measurement interval.";
 }
 leaf elapsed-time {
    type uint32;
    units "tens of ms";
    description
```

```
"Elapsed time for current measurement
               interval in 0.01 seconds.";
 }
 leaf forward-available {
    type yang:gauge32;
   description
      "The current value of forward-available for the
       referenced loss measurement session.";
  }
 leaf backward-available {
    type yang:gauge32;
    description
      "The current value of backward-available for the
       referenced loss measurement session.";
  }
 leaf forward-unavailable {
  type yang:gauge32;
  description
    "The current value of forward-unavailable for the
     referenced loss measurement session.";
 }
 leaf backward-unavailable {
    type yang:gauge32;
    description
      "The current value of backward-unavailable for the
       referenced loss measurement session.";
 }
 uses loss-stats-group;
}
/* End of container current-stats */
list history-stats {
 key id;
 description
    "This list contains the result for historic measurement
     intervals for PM loss measurement session.";
 leaf id {
    type uint32;
    description
      "This leaf can be used to select different
      history-stats intervals.";
 }
 leaf start-time {
    type yang:date-and-time;
    description
      "Start time for measurement interval.";
```

```
}
        leaf elapsed-time {
          type uint32;
         units "tens of ms";
          description
            "Elapsed time for measurement interval in 0.01
             seconds.";
        }
        leaf forward-available {
          type yang:gauge32;
         description
            "The value of forward-available for the
             referenced loss measurement session and interval.";
        }
        leaf backward-available {
          type yang:gauge32;
         description
            "The value of backward-available for the
             referenced loss measurement session and interval.";
        }
        leaf forward-unavailable {
          type yang:gauge32;
         description
            "The value of forward-unavailable for the
             referenced loss measurement session and interval.";
        }
        leaf backward-unavailable {
          type yang:gauge32;
          description
            "The value of backward-unavailable for the
             referenced loss measurement session and interval.";
        }
        uses loss-stats-group;
      }
      /* End of list history-stats */
   }
   /* End of container loss-measurements */
/* End of augments clause */
       //RPCs related to Generic PM
rpc create-loss-measurement {
 if-feature create-loss-measurement;
 description
 "Schedule a one-way or two-way on-demand or proactive
  performance monitoring loss measurement session on a specific
  MEP and flow.
```

Internet-Draft

```
A list entry associated with the newly created session will be
  created in the loss-measurements container and the assigned
   session identifier will be returned in the output parameter.";
 input {
   uses loss-measurement-configuration-group;
   uses measurement-timing-group;
   container destination-mep {
      uses goam:mp-address;
      uses goam:MEP-ID;
              description
              "destination mep";
   }
 }
 output {
   leaf session-id {
      type uint32;
      mandatory true;
      description
        "The session identifier of the newly created loss
        measurement session.";
   }
 }
} //end of rpc
rpc abort-loss-measurement {
 if-feature abort-delay-measurement;
 description
    "Abort a currently running or scheduled single-ended
     on-demand PM loss measurement function.";
 input {
   uses goam:maintenance-domain-id;
   uses goam:ma-identifier;
   container destination-mep {
      uses goam:mp-address;
      uses goam:MEP-ID;
              description
              "destination mep";
   }
   leaf session-id {
      type uint32;
      mandatory true;
      description
        "The session Id of the measurement session to
        be aborted.";
```

```
}
  }
} //end of RPC
rpc create-delay-measurement {
 if-feature create-delay-measurement;
 description
    "Schedule a one-way or two-way on-demand or proactive
     performance monitoring delay measurement session on a
     specific MEP and flow. A list entry associated with the
     newly created session will be created in the
     delay-measurements container and the assigned session
     identifier will be returned in the output parameter.";
 input {
    uses delay-measurement-configuration-group;
    uses measurement-timing-group;
    container destination-mep {
      uses goam:mp-address;
      uses goam:MEP-ID;
              description
              "description mep.";
   }
    uses goam:flow-entropy;
  }
 output {
    leaf session-id {
    type uint32;
    mandatory true;
    description
      "The session identifier of the newly created
       delay measurement session.";
    }
  }
} //end of rpc
rpc abort-delay-measurement {
 if-feature abort-delay-measurement;
 description
    "Abort a currently running or scheduled single-ended
     on-demand PM function.";
 input {
    uses goam:maintenance-domain-id;
    uses goam:ma-identifier;
```

```
container destination-mep {
          uses goam:mp-address;
          uses goam:MEP-ID;
                  description
                  "destination mep";
        }
        leaf session-id {
          type uint32;
          mandatory true;
          description
            "The session Id of the measurement session to
             be aborted.";
        }
      }
    } //end of RPC
  }
<CODE ENDS>
```

6. Security Considerations

TBD.

7. IANA Considerations

TBD.

8. References

8.1. Normative References

[GENYANGGOAM]

Senevirathne , T. and Q. Wu, "Generic YANG Data Model for Operations, Administration, and Maintenance (OAM)", ID <u>http://tools.ietf.org/html/</u> <u>draft-tissa-lime-yang-oam-management-04</u>, April 2015.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", <u>BCP 14</u>, <u>RFC 2119</u>, DOI 10.17487/RFC2119, March 1997, <<u>http://www.rfc-editor.org/info/rfc2119</u>>.

8.2. Informative References

[IEEE802.1Q]

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

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

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

Email: dekumar@cisco.com

Tom Taylor PT Taylor Consulting Ottawa Canada

Email: tom.taylor.stds@gmail.com

Wang, et al. Expires May 31, 2016 [Page 36]