A YANG model for Multicast Protocol for Low power and lossy Networks (MPL)
draft-vanderstok-core-mpl-yang-00

Abstract

This document defines a YANG data model for management of Multicast Protocol for Low power and lossy Networks (MPL) implementations. The data model includes configuration data and state data.

Note

Discussion and suggestions for improvement are requested, and should be sent to roll@ietf.org.

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 September 8, 2016.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents.
1. Introduction

This document defines a YANG [RFC6020] data model for management of Multicast Protocol for Low power and lossy Networks (MPL) [RFC7731] implementations. The data model covers configuration of per-interface MPL parameters. It also provides information about which Multicast addresses are operationally used, and the seeds for which packets are forwarded.

1.1. 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:

- client
- configuration data
- server
- state data
The following terms are defined in [RFC6020] and are not redefined here:

- augment
- data model
- data node

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

Terms like domain, seed, I, k, c are defines in [RFC7731].

1.1.1. Tree Diagrams

A simplified graphical representation of the data model is used in the YANG modules specified in this document. The meaning of the symbols in these diagrams is as follows:

- Brackets "[" and "]" enclose list keys.
- Abbreviations before data node names: "rw" means configuration data (read-write) and "ro" state data (read-only).
- Symbols after data node names: "?" means an optional node, "!" means a presence container, and "*" denotes a list and leaf-list.
- Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").
- Ellipsis ("...") stands for contents of subtrees that are not shown.

2. MPL model

This document defines the YANG module "ietf-yang-mpl", which specifies a data model for MPL servers. The model consists of the following parts: (1) a "mpl-domain" part that describes the MPL-domains and associated Multicast addresses and the interfaces on which the Multicast addresses are enabled, (2) a "mpl-op" part that describes the parameters settings per seed, (3) a "mpl-seeds" part that describes the MPL buffer contents and the Trickle timer values, and (4) a "mpl-statistics" part that describes the number of lost and correctly forwarded messages. The data model has the following structure for MPL configuration per node:

```
  +--ro yang-mpl
```
+--rw mpl-domain  
  +--rw domains* [domainID]  
    +--rw domainID uint16  
    +--rw MClist* yang:ipv6-address  
  +--rw addresses* [MCaddress]  
    +--rw MCaddress yang:ipv6-address  
    +--rw interfaces* string  
+--rw mpl-op  
  +--rw SE_LIFETIME uint16  
  +--rw PROACTIVE_FORWARDING boolean  
  +--rw SEED_SET_ENTRY_LIFETIME uint64  
+--rw mpl-parameter* [domainID]  
  +--rw domainID uint16  
    +--rw DATA_MESSAGE_IMIN uint16  
    +--rw DATA_MESSAGE_IMAX uint16  
    +--rw DATA_MESSAGE_K uint16  
    +--rw DATA_MESSAGE_TIMER_EXPIRATIONS uint16  
  +--rw CONTROL_MESSAGE_IMIN uint16  
  +--rw CONTROL_MESSAGE_IMAX uint16  
  +--rw CONTROL_MESSAGE_K uint16  
  +--rw CONTROL_MESSAGE_TIMER_EXPIRATIONS uint16  
+--rw MC_address* yang:ipv6-address  
+--ro mpl-seeds* [seedID, domainID]  
  +--ro seedID uint64  
  +--ro domainID uint16  
  +--ro local boolean  
  +--ro life-time uint64  
  +--ro min-seqno uint8  
  +--ro data-number uint8  
  +--ro control-number uint8  
  +--ro nr-of-timers uint8  
  +--ro seed_timers* [seqno]  
    +--ro seqno uint8  
    +--ro I uint8  
    +--ro c uint8  
    +--ro e uint8  
    +--ro t uint8  
+--ro mpl-statistics* [seedID, domainID]  
  +--ro seedID uint64  
  +--ro domainID uint16  
  +--ro c-too-high uint64  
  +--ro nr-forwarded uint64  
  +--ro nr-not-forwarded uint64  
  +--ro nr-of-messages-received uint64  
  +--ro nr-of-copies-received uint64  
  +--ro nr-of-messages-forwarded uint64  
  +--ro nr-of-copies-forwarded uint64  
  +--ro nr-of-refused uint64
3. ietf-yang-mpl module

This section contains the ietf-yang-mpl module. The model is based on the MPL specification published in [RFC7731] and the specification of [RFC6206]. The identification of the interfaces follows the specification of ietf-interfaces of [RFC7223].

The data model allows to set values of parameters of the MPL algorithm. This approach requires an active manager process to set the values without use of DHCP as described in: [I-D.ietf-roll-mpl-parameter-configuration].

<CODE BEGINS>
module ietf-yang-mpl {

    yang-version 1;

    namespace

    prefix mpl;

    import ietf-inet-types{
        prefix inet;
    }

    organization
        "IETF ROLL (Routing over Low power and lossy networks) Working Group";

    contact
        "WG Web:   http://tools.ietf.org/wg/roll/
        WG List:   mailto:roll@ietf.org

        WG Chair: Michael Richardson
        mailto:mcr+ietf@sandelman.ca

        WG Chair: Ines Robles
        mailto:maria.ines.robles@ericsson.com

        Editor: Peter van der Stok

}  
</CODE>
This module contains information about the operation of the MPL protocol.

Copyright (c) 2016 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in Section 4.c of the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info).

This version of this YANG module is part of RFC XXXX; see the RFC itself for full legal notices."

revision "2016-01-20" {
    description "Initial revision.";
    reference
        "I-D:draft-vanderstok-roll-mpl-yang: A YANG model for Multicast Protocol for Low power and lossy Networks (MPL)";
}

container yang-mpl {
    description
        "This container describes the data model for MPL servers. The model consists of the following parts: (1) a mpl-domain part that describes the MPL-domains and associated Multicast addresses and the interfaces on which the Multicast addresses are enabled, (2) a mpl-op part that describes the parameters settings per seed, (3) a mpl-seeds part that describes the MPL buffer contents and the Trickle timer values, and (4) a mpl-statistics part that describes the number of lost and correctly forwarded messages.";

    container mpl-domain {
        config true;
        description
            "The entries describe the MPL domains, the associated Multicast addresses and interfaces."

        list domains {
            key "domainID";
            description
                "The entries describe a given domain identified with domainID and the associated Multicast addresses."

            leaf domainID {

type uint16;
description
"Entry uniquely identifies the domain in the
forwarder;"
}

leaf-list MCList{
  type inet:ipv6-address;
  description
    "List of associated IPv6 Addresses."
}
} // domains list

list addresses {
  key "MCaddress";
  description
    "The entries describe the interfaces enabled with a given
    MC address.";
  leaf MCaddress {
    type inet:ipv6-address;
    description
      "MC address belonging to a MPL domain."
  }
  leaf-list interfaces {
    type string;
    description
      "List of names of interfaces enabled for this Multicast address
      . Interface name is defined in [RFC6206]."
  }
} // addresses list
} // container mpl-domain

container mpl-op {
  config true;
  description
    "Parameter settings for each MPL server and for each individual doma
    in of the server.";
  leaf SE_LIFETIME {
    type uint16;
    description
      "lifetime in milliseconds/(mpl timer units),
      equivalent to SEED_SET_ENTRY_LIFETIME/TUNIT as
      specified in ietf-roll-mpl-parameter-
      configuration.";
  }
  leaf PROACTIVE_FORWARDING {
    type boolean;
  }
}
description
"The boolean value indicates whether the MPL forwarder
schedules MPL data message transmission after
receiving them for the first time.";
}

leaf SEED_SET_ENTRY_LIFETIME {
type uint64;
description
"The value indicates the minimum lifetime for an entry
in the Seed set expressed in seconds. Default value
is 30 minutes.";
}

list mpl-parameter{
key "domainID";
description
"Each domain has a set of MPL forwarding parameters
which regulate the forwarding operation.";

leaf domainID{
type uint16;
description
"Each domainID must be present in mpl-parameter list.";
}

leaf DATA_MESSAGE_IMIN{
type uint16;
description
"The minimum Trickle timer interval, as defined in
[RFC6206], for MPL Data Message transmissions.";
}

leaf DATA_MESSAGE_IMAX{
type uint16;
// default DATA_MESSAGE_IMIN;
description
"The maximum Trickle timer interval, as defined in
[RFC6206], for MPL Data Message transmissions.";
}

leaf DATA_MESSAGE_K{
type uint16;
default 1;
description
"The redundancy constant, as defined in [RFC6206], for
MPL Data Message transmissions.";
}
leaf DATA_MESSAGE_TIMER_EXPIRATIONS{
  type uint16;
  default 3;
  description
    "The number of Trickle timer expirations that occur
     before terminating the Trickle algorithm's
     retransmission of a given MPL Data Message.";
}

leaf CONTROL_MESSAGE_IMIN{
  type uint16;
  description
    "The minimum Trickle timer interval, as defined
     in [RFC6206], for MPL Control Message
     transmissions.";
}

leaf CONTROL_MESSAGE_IMAX{
  type uint16;
  //
  default SE_LIFETIME * 300000;
  description
    "The maximum Trickle timer interval, as defined
     in [RFC6206], for MPL Control Message
     transmissions.";
}

leaf CONTROL_MESSAGE_K{
  type uint16;
  default 1;
  description
    "The redundancy constant, as defined in [RFC6206],
     for MPL Control Message transmissions.";
}

leaf CONTROL_MESSAGE_TIMER_EXPIRATIONS{
  type uint16;
  default 10;
  description
    "The number of Trickle timer expirations that occur
     before terminating the Trickle algorithm
     for MPL Control Message transmissions.";
}

leaf-list MC_addresses{
  type inet:ipv6-address;
  description
    "All the MPL multicast addresses which belong to this domain.";
}
list mpl-seeds{
  key "seedID domainID";
  description
    "List describes all seeds that are active in the server. Seed information contains the message buffer contents and the operational values of I, c, sequence number and the life-times per message.";

  leaf seedID{
    type uint64;
    description
      "value uniquely identifies the MPL Seed within a MPL domain."
  }

  leaf domainID{
    type uint16;
    description
      "together with seedID uniquely identifies buffer set."
  }

  leaf local {
    type boolean;
    description
      "When local == TRUE, seed is located in this forwarder.
      WHEN local == false, seed is located in different forwarder."
  }

  leaf life-time {
    type uint64;
    description
      "Minimum remaining lifetime of the seed entry in SE_LIFETIME units."
  }

  leaf min-seqno{
    type uint8;
    description
      "Lower bound sequence number for this seed."
  }

  leaf data-number{
    type uint8;
    description
      "Number of currently buffered data messages."
  }
}
leaf control-number{
  type uint8;
  description
    "Number of currently buffered control messages.";
}

leaf nr-of-timers{
  type uint8;
  description
    "number of active trickle timers.";
}

list seed-timers{
  key "seqno";
  description
    "status of timers of the active message identified by seqno.";

  leaf seqno{
    type uint8;
    description
      "Sequence number of message of this seed.";
  }

  leaf I{
    type uint8;
    description
      "Current Trickle timer interval in SE-LIFETIME units.";
  }

  leaf c{
    type uint8;
    description
      "number of times that identical message has been received in this interval.";
  }

  leaf e{
    type uint8;
    description
      "number of Trickle time expirations since last Trickle timer reset.";
  }

  leaf t{
    type uint8;
    description
      "number of time units expressed in SE-LIFETIME units since last Trickle time activation.";
  }
}
list mpl-statistics{
  key "seedID domainID";
  description
   "List describes performance statistics integrated over the messages of a seed. A message identified with a seqno can receive and forward multiple copies";

  leaf seedID{
    type uint64;
    description
     "value uniquely identifies the MPL Seed within a MPL domain.";
  }

  leaf domainID{
    type uint16;
    description
     "together with seed-ID uniquely identifies buffer set.";
  }

  leaf c-too-high {
    type uint64;
    description
     "Number of times that a message copy was not forwarded because c > k.";
  }

  leaf nr-forwarded {
    type uint64;
    description
     "number of times message copies are forwarded.";
  }

  leaf nr-not-forwarded {
    type uint64;
    description
     "number of times no copies of a message are forwarded at all.";
  }

  leaf nr-of-messages-received{
    type uint64;
    description
     "number of messages received, must be smaller than or equal to seq no.";
  }

  leaf nr-of-copies-received{
type uint64;
description
  "number of message copies received.";
}

leaf nr-of-messages-forwarded{
type uint64;
description
  "number of forwarded messages, must be smaller than or equal to nr
  of-messages-received.";
}

leaf nr-of-copies-forwarded{
type uint64;
description
  "number of forwarded copies.";
}

leaf nr-of-refused{
type uint64;
description
  "number of refused messages because seqno too small.";
}

leaf nr-of-notreceived{
type uint64;
description
  "number of messages that are were not received according to control message.";
}

leaf nr-of-missing{
type uint64;
description
  "number of messages that were not received by a neighbour according to control message.";
}

leaf nr-of-inconsistent-data{
type uint64;
description
  "number of inconsistent data messages.";
}

leaf nr-of-consistent-data{
type uint64;
description
  "number of consistent data messages.";
}
4. IANA Considerations

5. Acknowledgements

Andy Bierman has commented on the use of YANG for mpl.

6. Changelog

Changes from version 00 to version 01

- first change

7. References

7.1. Normative References


7.2. Informative References


Authors' Addresses

Peter van der Stok (editor)
consultant

Phone: +31-492474673 (Netherlands), +33-966015248 (France)
Email: consultancy@vanderstok.org
URI: www.vanderstok.org

Pascal Thubert
Cisco Systems, Inc
Building D
45 Allee des Ormes - BP1200
MOUGINS - Sophia Antipolis 06254
FRANCE

Phone: +33 497 23 26 34
Email: pthubert@cisco.com