

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 25, 2015

Y. Liu
F. Guo
Huawei
October 22, 2014

Yang Model for Internet Group Management Protocol (IGMP) and Multicast
Listener Discovery (MLD)
draft-liu-pim-igmp-mld-yang-00

Abstract

This document defines a YANG data model that can be used to configure and manage IGMP and MLD.

Requirements Language

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 RFC 2119 [RFC2119].

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 April 25, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

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

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Terminology	2
3. Design of Data Model	3
3.1. Overview	3
3.2. GMP Per-instance Configuration	3
3.2.1. Per-instance Parameters	4
3.2.2. Per-SSM-Mapping Configuration of GMP Instance	4
3.2.3. Per-interface Configuration of GMP Instance	4
4. GMP Yang Module	6
5. IANA Considerations	18
6. Security Considerations	18
7. Acknowledgements	18
8. References	18
8.1. Normative References	18
8.2. Informative References	19
Authors' Addresses	19

1. Introduction

YANG[RFC6020] is a data definition language that was introduced to define the contents of a conceptual data store that allows networked devices to be managed using NETCONF[RFC6241]. YANG is proving relevant beyond its initial confines, as bindings to other interfaces(e.g. REST) and encoding other than XML (e.g. JSON) are being defined. Furthermore, YANG data models can be used as the basis of implementation for other interface, such as CLI and Programmatic APIs.

This document defines a YANG data model that can be used to configure and manage IGMP and MLD. It includes IGMPv1[RFC1112], IGMPv2[RFC2236], IGMPv3[RFC3376] and MLDv1[RFC2710], MLDv2[RFC3810]. In addition, features described in IGMP and MLD standards other than mentioned above RFC are also supported. For convenience, IGMP and MLD are wholly called GMP below.

2. Terminology

- o IGMP: Internet Group Management Protocol
- o MLD: Multicast Listener Discovery

- o GMP: Group Management Protocol
- o SSM: Source-Specific Multicast

3. Design of Data Model

3.1. Overview

The GMP Yang module has only one main container ::

- o gmpInstances : that contains per-instance writable configuration objects.

The figure below describes the overall structure of the GMP Yang module :

```
module: gmp
+--rw gmp
  +--rw gmp-Instances
    +--rw gmp-Instance * [vrfName]
      | ...
    +--rw gmp-SSM-Mappings
      | ...
    +--rw gmp-Interfaces
      +--rw gmp-Interface * [ifName]
        | ...
      +--rw gmp-Static-Groups
        ...
```

Figure 1 The overview of GMP YANG data model

3.2. GMP Per-instance Configuration

GMP per-instance configuration container includes parameters of the public GMP instance or the GMP instance binding a specific VRF. GMP per-instance configuration container is divided into:

- o Per-instance parameters
- o Per-SSM-Mapping configuration of the GMP instance
- o Per-interface configuration of the GMP instance

3.2.1. Per-instance Parameters

The per-instance parameter includes the name of the VRF bound by the GMP instance, and timer parameters such as query interval etc.

```

+--rw gmp-Instances
  +--rw gmp-Instance* [vrfName]
    +--rw vrfName                string
    +--rw addrFamily             enumeration
    +--rw queryInterval?         uint32
    +--rw queryRspInterval?      uint32
    +--rw robustness?            uint32
    +--rw lastMemberQueryInterval? uint32
    +--rw reqRouterAlert?        boolean
    +--rw sendRouterAlert?       boolean
    +--rw othQuerierPstTime?     uint32
    +--ro gmpEntryLimit?         uint32
    +--rw ipsecName?             string
    +--rw ipsecType?             enumeration

```

Figure 2 The YANG data model of GMP instance

3.2.2. Per-SSM-Mapping Configuration of GMP Instance

Per-SSM-Mapping configuration of the GMP instance includes the SSM Mapping rules.IGMPv1/v2 and MLDv1 reports can use these rules to map SG state for PIM SSM[RFC4607].IGMPv3 and MLDv2 can default use PIM SSM , which is described in [RFC4604].

```

+--rw gmp-SSM-Mappings
  |   +--rw gmp-SSM-Mapping
  |   |   +--rw IPV4-ssmmapingGrp    inet:ipv4-address
  |   |   +--rw IPV6-ssmmapingGrp    inet:ipv6-address
  |   |   +--rw isSSMapMask           boolean
  |   |   +--rw IPV4-ssmmapingMask?   inet:ipv4-address
  |   |   +--rw IPV6-ssmmapingMask?   uint32
  |   |   +--rw isMaskLen             boolean
  |   |   +--rw maskLen?              uint32
  |   |   +--rw IPV4-srcAddr          inet:ipv4-address
  |   |   +--rw IPV6-srcAddr          inet:ipv6-address

```

Figure 3 The YANG data model of GMP SSM-Mapping

3.2.3. Per-interface Configuration of GMP Instance

Per-interface configuration of the GMP instance includes the interface name, timer parameters, policies, static groups etc.GMP per-instance configuration container is divided into two containers:

- o Per-interface parameters
- o Per-static-group configuration of the GMP interface

3.2.3.1. Per-interface Parameters

The per-interface parameter includes the name of the interface, and the VRF name bound by the interface, and time parameters, policies etc.

```

+--rw gmp-Interfaces
  +--rw gmp-Interface* [ifName]
    +--rw vrfName                string
    +--rw ifName                 ifName
    +--rw addrFamily             enumeration
    +--rw gmpEnable              boolean
    +--rw ipSourcePly?           boolean
    +--rw ipSrcAclName?          string
    +--rw ipSrcAclIpv6?          string
    +--rw queryInterval?         uint32
    +--rw queryRspInterval?      uint32
    +--rw robustness?            uint32
    +--rw version?               uint32
    +--rw lastMemberQueryInterval? uint32
    +--rw requireRouterAlert?    boolean
    +--rw sendRouterAlert?       boolean
    +--rw othQuerierPresentTime? uint32
    +--rw immediatelyLeave?       boolean
    +--rw immLeaveAclName?         string
    +--rw immLeaveAclIpv6?         string
    +--rw gmpEntryLimit?          uint32
    +--rw exceptAclName?          string
    +--rw exceptAclIpv6?          string
    +--rw ssmapEnable?            boolean
    +--rw groupAclName?           string
    +--rw groupAclIpv6?           string
    +--rw groupAclGMPVer?         uint32
    +--rw queryAclName?           string
    +--rw queryAclIpv6?           string
    +--rw ipsecName?              string
    +--rw ipsecType?              enumeration

```

Figure 4 The YANG data model of GMP interface

3.2.3.2. Per-static-group Configuration of GMP interface

Per-static-group configuration of the GMP interface includes the static group address, and as a option also includes source address, every static group step, and group numbers on the interface.

```

+--rw gmp-Static-Groups
  +--rw gmp-Static-Group
    +--rw vrfName          string
    +--rw addrFamily       enumeration
    +--rw ifName           ifName
    +--rw IPV4-staticGrp   inet:ipv4-address
    +--rw IPV6-staticGrp   inet:ipv6-address
    +--rw isSourceAddr     boolean
    +--rw IPV4-sourceAddr? inet:ipv4-address
    +--rw IPV6-sourceAddr? inet:ipv6-address
    +--rw isStepGrpMask    boolean
    +--rw IPV4-incStepGrpMask? inet:ipv4-address
    +--rw IPV6-incStepGrpMask? inet:ipv6-address
    +--rw isMaskLen        boolean
    +--rw maskLen?         uint32
    +--rw totalNum?        uint32

```

Figure 5 The YANG data model of GMP static group

4. GMP Yang Module

```

module gmp {
  namespace "urn:huawei:params:xml:ns:yang:gmp";
  // replace with IANA namespace when assigned - urn:ietf:params:xml:ns:yang:1
  prefix "gmp";
  import ietf-inet-types {
    prefix inet;
  }
  organization
    "Huawei Technologies Co., Ltd.";
  contact
    "liuyisong@huawei.com
     guofeng@huawei.com ";
  description
    "This YANG module defines the generic configuration
     data for GMP, i.e. IGMP and MLD, which is common across all of the vendor
     implementations of the protocol. It is intended that the module
     will be extended by vendors to define vendor-specific
     GMP configuration parameters.";
  revision 2014-10-21 {
    description
      "Initial revision.";
  }
}

```

```

}

typedef ifName {
    description "ifName is like ethernet1/1/1/1";
    type string {
        length "1..63";
    }
}

container gmp {

    container gmp-Instances {

        list gmp-Instance {

            key "vrfName";
            max-elements "unbounded";
            min-elements "0";
            description "Specifies a list of gmp instances.";

            leaf vrfName {
                description "Name of an gmp instance.If the name string is empty the instance means a public instance whose name is _public_.";
                config "true";
                //default "_public_";
                type string {
                    length "0..32";
                }
            }

            leaf addrFamily {
                description "Specify an address family, which determines whether an address is an IPv4 or IPv6 address.";
                config "true";
                mandatory "true";
                type enumeration {
                    enum ipv4unicast {
                        value "0";
                        description "Specify an address family, which determines whether an address is an IPv4 or IPv6 address.";
                    }
                    enum ipv6unicast {
                        value "1";
                        description "Specify an address family, which determines whether an address is an IPv4 or IPv6 address.";
                    }
                }
            }

            leaf queryInterval {
                description "Specify the interval at which the router sends general query messages. The value is an integer ranging from 1 to 18000, in seconds. The default value for IPv4 is 60, and 125 for IPv6.";
                config "true";
                default "60";
            }
        }
    }
}

```

```

        type uint32 {
            range "1..18000";
        }
    }
    leaf queryRspInterval {
        description "Specify the maximum response time for a query m
message. The value is an integer ranging from 1 to 25, in seconds. The default va
lue is 10.";
        config "true";
        default "10";
        type uint32 {
            range "1..25";
        }
    }
    leaf robustness {
        description "Specify the number of times for retransmitting
a message to avoid the packet loss. The value is an integer ranging from 2 to 5.
The default value is 2.";
        config "true";
        default "2";
        type uint32 {
            range "2..5";
        }
    }
    leaf lastMemberQueryInterval {
        description "Specify the interval at which the querier sends
last-member query messages. The value is an integer ranging from 1 to 5, in sec
onds. The default value is 1. This parameter makes sense only when the current q
uerier runs IGMPv2, IGMPv3 or MLD.";
        config "true";
        default "1";
        type uint32 {
            range "1..5";
        }
    }
    leaf reqRouterAlert {
        description "Configure the router to process only the messag
es whose IP headers contain Router-Alert options globally. By default, the route
r does not check whether the received IGMP messages contain Router-Alert options
.";
        config "true";
        type boolean {
        }
    }
    leaf sendRouterAlert {
        description "Configure the router to send the messages with
Router-Alert options in the IP headers globally. By default, the IP headers cont
ain Router-Alert options. ";
        config "true";
        type boolean {
        }
    }
    leaf othQuerierPstTime {
        description "Set the global Keepalive period for other queri
ers. ";
        config "true";
        type uint32 {
            range "60..300";
        }
    }
    leaf gmpEntryLimit {

```



```

        description "Set the maximum number of entries that can be c
reated for the current instance. The value is an integer ranging from 1 to 49152
. The default value is 49152.";
        config "false";
        type uint32 {
            range "1..49152";
        }
    }
    leaf ipsecName {
        description "SA name. The value is a string of 1 to 15 chara
acters.";
        config "true";
        type string {
            length "1..15";
            pattern "^[^ ]+$";
        }
    }
    leaf ipsecType {
        description "IPsec type used as a query option.";
        config "true";
        type enumeration {
            enum forAll {
                value "0";
                description "IPsec type used as a query option.";
            }
            enum forQuery {
                value "1";
                description "IPsec type used as a query option.";
            }
        }
    }
}

container gmp-SSM-Mappings {
    container gmp-SSM-Mapping {
        leaf IPV4-ssmmapingGrp {
            description "Specify the address of a multicast grou
p.";
            config "true";
            mandatory "true";
            type inet:ipv4-address;
        }
        leaf IPV6-ssmmapingGrp {
            description "Specify the address of a multicast grou
p.";
            config "true";
            mandatory "true";
            type inet:ipv6-address;
        }
        leaf isSSMapMask {
            description "Whether a mask is configured for a mult
icast group address.";
            config "true";

```

```

        mandatory "true";
        type boolean {
        }
    }
    leaf IPV4-ssmmapingMask {
        description "Specify the mask of a multicast group a
ddress.";
        config "true";
        type inet:ipv4-address;
    }
    leaf IPV6-ssmmapingMask {
        description "Specify the mask of a multicast group a
ddress.";
        config "true";
        type uint32 {
            range "0..128";
        }
    }

    leaf isMaskLen {
        description "Whether the length is set for the mask
of a multicast group address.";
        config "true";
        mandatory "true";
        type boolean {
        }
    }
    leaf maskLen {
        description "Specify the mask length of a multicast
group address. In the case of an IPv4 address, the mask length ranges from 4 to
32. In the case of an IPv6 address, the mask length is 16/32/64/128.";
        config "true";
        type uint32 {
            range "0..128";
        }
    }
    leaf IPV4-srcAddr {
        description "Specify the address of a multicast sour
ce.";
        config "true";
        mandatory "true";
        type inet:ipv4-address;
    }
    leaf IPV6-srcAddr {
        description "Specify the address of a multicast sour
ce.";
        config "true";
        mandatory "true";
        type inet:ipv6-address;
    }
}

container gmp-Interfaces {

```

```

list gmp-Interface {
    key "ifName";
    max-elements "unbounded";
    min-elements "0";
    description "Specifies an gmp interface.";

    leaf vrfName {
        description "Name of an gmp instance. If the name st
ring is empty the instance means a public instance whose name is _public_.";
        config "true";
        mandatory "true";
        //default "_public_";
        type string {
            length "0..32";
        }
    }
    leaf ifName {
        description "Interface name.";
        config "true";
        type ifName;
    }
    leaf addrFamily {
        description "Specify an address family, which determ
ines whether an address is an IPv4 or IPv6 address.";
        config "true";
        mandatory "true";
        type enumeration {
            enum ipv4unicast {
                value "0";
                description "Specify an address family, whic
h determines whether an address is an IPv4 or IPv6 address.";
            }
            enum ipv6unicast {
                value "1";
                description "Specify an address family, whic
h determines whether an address is an IPv4 or IPv6 address.";
            }
        }
    }
    leaf gmpEnable {
        description "Enable protocols on an interface.";
        config "true";
        mandatory "true";
        type boolean {
        }
    }
    leaf ipSourcePly {
        description "Configure a policy for filtering IGMP R
eport messages based on host addresses. By default, no policy is configured for
filtering IGMP Report messages based on host addresses.";
        config "true";
        type boolean {
        }
    }
}

```

```
    }
    leaf ipSrcAclName {
        description "Configure an ACL that defines a host ad-
dresses range. The value is an integer ranging from 2000 to 2999, or a case-sens-
itive string with a maximum of 32 characters. By default, the ACL is not configu-
red.";
        config "true";
        type string {
            length "1..32";
            pattern "^[^ ]+$";
        }
    }
    leaf ipSrcAclIpv6 {
        description "Configure an ACL that defines a host ad-
dresses range. The value is an integer ranging from 2000 to 2999, or a case-sens-
itive string with a maximum of 32 characters. By default, the ACL is not configu-
red.";
        config "true";
        type string {
            length "1..32";
            pattern "^[^ ]+$";
        }
    }
    leaf queryInterval {
        description "Specify the interval at which the route
r sends general query messages. The value is an integer ranging from 1 to 18000,
in seconds. The default value for IPv4 is 60, and 125 for IPv6.";
        config "true";
        default "60";
        type uint32 {
            range "1..18000";
        }
    }
    leaf queryRspInterval {
        description "Specify the maximum response time for a
query message. The value is an integer ranging from 1 to 25, in seconds. The de-
fault value is 10. This time is used to control the deadline of mainframe feed b-
ack the relation-ship of group members.";
        config "true";
        default "10";
        type uint32 {
            range "1..25";
        }
    }
    leaf robustness {
        description "Specify the number of times for retrans-
mitting messages to avoid packet loss. The value is an integer ranging from 2 to
5. The default value is 2.";
        config "true";
        default "2";
        type uint32 {
            range "2..5";
        }
    }
    leaf version {
        description "Specify the version of IGMP or MLD runn-
ing on an interface. By default, IGMPv2 or MLDv2 is used.";
        config "true";
        default "2";
        type uint32 {
            range "1..3";
        }
    }
}
```



```

    }
    leaf lastMemberQueryInterval {
        description "Specify the interval at which the queri
er sends last-member query messages. The value is an integer ranging from 1 to 5
, in seconds. The default value is 1. This parameter makes sense only when the c
urrent querier runs IGMPv2, IGMPv3 or MLD.";
        config "true";
        default "1";
        type uint32 {
            range "1..5";
        }
    }
    leaf requireRouterAlert {
        description "Configure an interface to process only
messages whose IP headers contain Router-Alert options. By default, the interfac
e does not check whether the received messages contain Router-Alert options.";
        config "true";
        type boolean {
        }
    }
    leaf sendRouterAlert {
        description "Configure an interface to send the mess
ages with Router-Alert options in the IP headers. By default, the IP header cont
ain Router-Alert options.";
        config "true";
        default "true";
        type boolean {
        }
    }
    leaf othQuerierPresentTime {
        description "Set the Keepalive period for other quer
iers on an interface. The value ranges from 60 to 300, in second. By default, no
Keepalive period is set for other queriers.";
        config "true";
        type uint32 {
            range "60..300";
        }
    }
    leaf immediatelyLeave {
        description "Configure an interface that receives a
Leave message of a certain group to immediately delete the corresponding group r
ecords, without sending a last-member query message.";
        config "true";
        type boolean {
        }
    }
    leaf immLeaveAclName {
        description "Configure an ACL that defines a multica
st group range. The basic ACL number ranges from 2000 to 2999, and the advanced
ACL number ranges from 3000 to 3999. The name is a string with a maximum of 32 c
ase-sensitive characters. By default, the ACL is not con";
        config "true";
        type string {
            length "1..32";
            pattern "^[^ ]+$";
        }
    }
    leaf immLeaveAclIpv6 {
        description "Configure an ACL that defines a multica
st group range. The basic ACL number ranges from 2000 to 2999, and the advanced
ACL number ranges from 3000 to 3999. The name is a string with a maximum of 32 c
ase-sensitive characters. By default, the ACL is not con";
        config "true";

```

```
type string {  
    length "1..32";
```



```

        pattern "^[^ ]+$";
    }
}
leaf gmpEntryLimit {
    description "Specify the maximum number of entries t
hat the current interface can create. It is an integer ranging from 1 to 16384.
The default value is 16384.";
    config "true";
    type uint32 {
        range "1..16384";
    }
}
leaf exceptAclName {
    description "Specify the range of multicast groups,
the number of IGMP entries corresponding to which needs not be limited. The basi
c ACL number ranges from 2000 to 2999. The basic ACL filters group addresses onl
y, without distinguishing (*, G) entries and (S, G) entr";
    config "true";
    type string {
        length "1..32";
        pattern "^[^ ]+$";
    }
}
leaf exceptAclIpv6 {
    description "Specify the range of multicast groups,
the number of IGMP entries corresponding to which needs not be limited. The basi
c ACL number ranges from 2000 to 2999. The basic ACL filters group addresses onl
y, without distinguishing (*, G) entries and (S, G) entr";
    config "true";
    type string {
        length "1..32";
        pattern "^[^ ]+$";
    }
}
leaf ssmapEnable {
    description "Enable SSM mapping on an interface.";
    config "true";
    type boolean {
    }
}
leaf groupAclName {
    description "Set a filter for multicast groups on an
interface to control the range of multicast groups that hosts can join. The bas
ic ACL number ranges from 2000 to 2999, and the advanced ACL number ranges from
3000 to 3999. The name is a string with a maximum of 32 ";
    config "true";
    type string {
        length "1..32";
        pattern "^[^ ]+$";
    }
}
leaf groupAclIpv6 {
    description "Set a filter for multicast groups on an
interface to control the range of multicast groups that hosts can join. The bas
ic ACL number ranges from 2000 to 2999, and the advanced ACL number ranges from
3000 to 3999. The name is a string with a maximum of 32 ";
    config "true";
    type string {
        length "1..32";
        pattern "^[^ ]+$";
    }
}

```



```

        leaf groupAclGMPVer {
            description "Forbids hosts that use a specified vers
ion to join the multicast group. The version can be specified in this command on
ly after a multicast filtering policy is configured.";
            config "true";
            type uint32 {
                range "1..3";
            }
        }
        leaf queryAclName {
            description "Configure an ACL that defines a host ad
dresses range. The value is an integer ranging from 2000 to 2999, or a case-sens
itive string with a maximum of 32 characters. By default, the ACL is not configu
red.";
            config "true";
            type string {
                length "1..32";
                pattern "^[^ ]+$";
            }
        }
        leaf queryAclIpv6 {
            description "Configure an ACL that defines a host ad
dresses range. The value is an integer ranging from 2000 to 2999, or a case-sens
itive string with a maximum of 32 characters. By default, the ACL is not configu
red.";
            config "true";
            type string {
                length "1..32";
                pattern "^[^ ]+$";
            }
        }
        leaf ipsecName {
            description "SA name. The value is a string of 1 to
15 characters.";
            config "true";
            type string {
                length "1..15";
                pattern "^[^ ]+$";
            }
        }
        leaf ipsecType {
            description "IPsec type used as a query option.";
            config "true";
            type enumeration {
                enum forAll {
                    value "0";
                    description "IPsec type used as a query opti
on.";
                }
                enum forQuery {
                    value "1";
                    description "IPsec type used as a query opti
on.";
                }
            }
        }
    }

    container gmp-Static-Groups {

```



```

        container gmp-Static-Group {
            leaf vrfName {
                description "Name of an GMP instance. If the
name string is empty the instance means a public instance whose name is _public
_.";
                config "true";
                mandatory "true";
                //default "_public_";
                type string {
                    length "0..32";
                }
            }
            leaf addrFamily {
                description "Specify an address family, whic
h determines whether an IPv4 or IPv6 address is used.";
                config "true";
                mandatory "true";
                type enumeration {
                    enum ipv4unicast {
                        value "0";
                        description "Specify an address fami
ly, which determines whether an IPv4 or IPv6 address is used.";
                    }
                    enum ipv6unicast {
                        value "1";
                        description "Specify an address fami
ly, which determines whether an IPv4 or IPv6 address is used.";
                    }
                }
            }
            leaf ifName {
                description "Interface name.";
                config "true";
                mandatory "true";
                type ifName;
            }
            leaf IPV4-staticGrp {
                description "Specifies the address of a stat
ic group.";
                config "true";
                mandatory "true";
                type inet:ipv4-address;
            }
            leaf IPV6-staticGrp {
                description "Specifies the address of a stat
ic group.";
                config "true";
                mandatory "true";
                type inet:ipv6-address;
            }
            leaf isSourceAddr {
                description "Whether an address is configure
d for the multicast source.";
                config "true";
                mandatory "true";
            }
        }

```

```

        type boolean {
        }
    }
    leaf IPV4-sourceAddr {
        description "Specifies the IPv4 address of a
remote neighbor.";
        config "true";
        type inet:ipv4-address;
    }
    leaf IPV6-sourceAddr {
        description "Specifies the IPv6 address of a
remote neighbor.";
        config "true";
        type inet:ipv6-address;
    }
    leaf isStepGrpMask {
        description "Determine whether to specify the
step mask in batch configuration mode.";
        config "true";
        mandatory "true";
        type boolean {
        }
    }
    leaf IPV4-incStepGrpMask {
        description "Specify the step mask of a group
address in batch configuration mode.";
        config "true";
        type inet:ipv4-address;
    }
    leaf IPV6-incStepGrpMask {
        description "Specify the step mask of a group
address in batch configuration mode.";
        config "true";
        type inet:ipv6-address;
    }
    leaf isMaskLen {
        description "Determine whether to set the length
for the step mask of a multicast group address in batch
configuration mode.";
        config "true";
        mandatory "true";
        type boolean {
        }
    }
    leaf maskLen {
        description "Specify the mask length of a multicast
group address. In the case of an IPv4 address, the mask
length ranges from 4 to 32. In the case of an IPv6 address,
the mask length is 16/32/64/128.";
        config "true";
        type uint32 {
            range "0..128";
        }
    }
    leaf totalNum {
        description "Specify the number of multicast
group addresses in batch configuration mode. It is an
integer ranging from 2 to 512.";
        config "true";
        default "2";
    }

```



```

    type uint32 {
        range "2..512";
    }
}
}
}
}
}
}
}
}
}
}
```

5. IANA Considerations

This draft includes no request to IANA.

6. Security Considerations

The data model defined does not create any security implications. This draft does not change any underlying security issues inherent in [I-D.ietf-netmod-routing-cfg].

7. Acknowledgements

TBD

8. References

8.1. Normative References

```
[I-D.ietf-netmod-routing-cfg]
    Lhotka, L., "A YANG Data Model for Routing Management",
    draft-ietf-netmod-routing-cfg-15 (work in progress), May
    2014.

[RFC1112] Deering, S., "Host extensions for IP multicasting", STD 5,
    RFC 1112, August 1989.

[RFC2236] Fenner, W., "Internet Group Management Protocol, Version
    2", RFC 2236, November 1997.
```


- [RFC2710] Deering, S., Fenner, W., and B. Haberman, "Multicast Listener Discovery (MLD) for IPv6", RFC 2710, October 1999.
- [RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, October 2002.
- [RFC3810] Vida, R. and L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, June 2004.
- [RFC4604] Holbrook, H., Cain, B., and B. Haberman, "Using Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Protocol Version 2 (MLDv2) for Source-Specific Multicast", RFC 4604, August 2006.
- [RFC4607] Holbrook, H. and B. Cain, "Source-Specific Multicast for IP", RFC 4607, August 2006.
- [RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.
- [RFC6241] Enns, R., Bjorklund, M., Schoenwaelder, J., and A. Bierman, "Network Configuration Protocol (NETCONF)", RFC 6241, June 2011.

8.2. Informative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

Authors' Addresses

Yisong Liu
Huawei
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095
China

Email: liuyisong@huawei.com

Feng Guo
Huawei
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095
China

Email: guofeng@huawei.com