

PIM WG
Internet-Draft
Intended status: Standards Track
Obsoletes: RFC1112
Expires: January 13, 2022

S. Deering
Retired
T. Eckert (Ed.)
Futurewei USA
July 12, 2021

Host Extensions IP Multicasting - Any Source Multicast (ASM)
draft-eckert-pim-rfc1112bis-00

ABSTRACT

This memo specifies the extensions required of a host implementation of the Internet Protocol (IP) to support Any Source Multicast (ASM) IP Multicasting or abbreviated IP Multicast. Distribution of this memo is unlimited.

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 <https://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 January 13, 2022.

Copyright Notice

Copyright (c) 2021 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 (<https://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.

2. INTRODUCTION

The host extensions defined in this memo are called Any Source Multicast (ASM) IP multicast or abbreviated IP multicast. The term Any Source Multicast is used to distinguish these extensions from Source Specific Multicast (SSM) IP multicast as defined by [RFC4607]. The abbreviation IP multicast always refers to this memo's extensions.

This memo applies to both IPv4 and IPv6. When it uses the term IP it implies either or both version of the IP protocol. It uses the terms IPv4 and/or IPv6 explicitly when referring to functions applicable to only a specific version of the IP protocol.

This document is a revision of [RFC1112]. See Appendix II. for a detailed list of changes from that memo.

IP multicasting is the transmission of an IP datagram to a "host group", a set of zero or more hosts identified by a single IP destination address. A multicast datagram is delivered to all members of its destination host group with the same "best-efforts" reliability as regular unicast IP datagrams, i.e., the datagram is not guaranteed to arrive intact at all members of the destination group or in the same order relative to other datagrams.

The membership of a host group is dynamic; that is, hosts may join and leave groups at any time. There is no restriction on the location or number of members in a host group. A host may be a member of more than one group at a time. A host need not be a member of a group to send datagrams to it.

A host group may be permanent or transient. A permanent group has a well-known, administratively assigned IP address. It is the address, not the membership of the group, that is permanent; at any time a permanent group may have any number of members, even zero. Those IP multicast addresses that are not reserved for permanent groups are available for dynamic assignment to transient groups which exist only as long as they have members.

Internetwork forwarding of IP multicast datagrams is handled by "multicast routers" which may be co-resident with, or separate from, internet gateways. A host transmits an IP multicast datagram as a local network multicast which reaches all immediately-neighboring members of the destination host group. If the datagram has an IP time-to-live greater than 1, the multicast router(s) attached to the local network take responsibility for forwarding it towards all other networks that have members of the destination group. On those other member networks that are reachable within the IP time-to-live, an attached multicast router completes delivery by transmitting the datagram as a local multicast.

This memo specifies the extensions required of a host IP implementation to support IP multicasting, where a "host" is any internet host or gateway other than those acting as multicast routers. The algorithms and protocols used within and between multicast routers are transparent to hosts and will be specified in separate documents. This memo also does not specify how local network multicasting is accomplished for all types of network, although it does specify the required service interface to an arbitrary local network and gives an Ethernet specification as an example. Specifications for other types of network will be the subject of future memos.

3. LEVELS OF CONFORMANCE

There are three levels of conformance to this specification:

Level 0: no support for IP multicasting.

There is, at this time, no requirement that all IPv4 implementations support IP multicasting. Level 0 hosts will, in general, be

unaffected by multicast activity. The only exception arises on some types of local network, where the presence of level 1 or 2 hosts may cause misdelivery of multicast IP datagrams to level 0 hosts. Such datagrams can easily be identified by the presence of a class D IP address in their destination address field; they should be quietly discarded by hosts that do not support IP multicasting. Class D addresses are described in section 4 of this memo.

Level 1: support for sending but not receiving multicast IP datagrams.

Level 1 allows a host to partake of some multicast-based services, such as resource location or status reporting, but it does not allow a host to join any host groups. An IP implementation may be upgraded from level 0 to level 1 very easily and with little new code. Only sections 4, 5, and 6 of this memo are applicable to level 1 implementations.

Level 2: full support for IP multicasting.

Level 2 allows a host to join and leave host groups, as well as send IP datagrams to host groups. Most IPv6 hosts require Level 2 support because IPv6 Neighbor Discovery ([RFC4861], as used on most link types) depends on multicast and requires that nodes join Solicited Node multicast addresses.

Level 2 requires implementation of the Internet Group Management Protocol (IGMP) for IPv4 and the equivalent Multicast Listener Discovery Protocol (MLDv) for IPv6 and extension of the IP and local network service interfaces within the host.

The current protocol versions are IGMPv3 [RFC3376] and MLDv2 [RFC3810] or lightweight versions of either protocol [RFC5790].

All of the following sections of this memo are applicable to level 2 implementations.

4. HOST GROUP ADDRESSES

IPv4 Host groups are identified by class D IP addresses, i.e., those with "1110" as their high-order four bits. Class E IP addresses, i.e., those with "1111" as their high-order four bits, are reserved for future addressing modes.

In Internet standard "dotted decimal" notation, host group addresses range from 224.0.0.0 to 239.255.255.255. The address 224.0.0.0 is guaranteed not to be assigned to any group, and 224.0.0.1 is assigned to the permanent group of all IPv4 hosts (including gateways). This is used to address all IPv4 multicast hosts on the directly connected network. There is no multicast address (or any other IP address) for all hosts on the total Internet. The addresses of other well-known, permanent groups are to be published in "Assigned Numbers".

IPv6 Host groups are identified by IPv6 addresses as defined in [RFC4291] section 2.7 and updated by [RFC7346], [RFC7371].

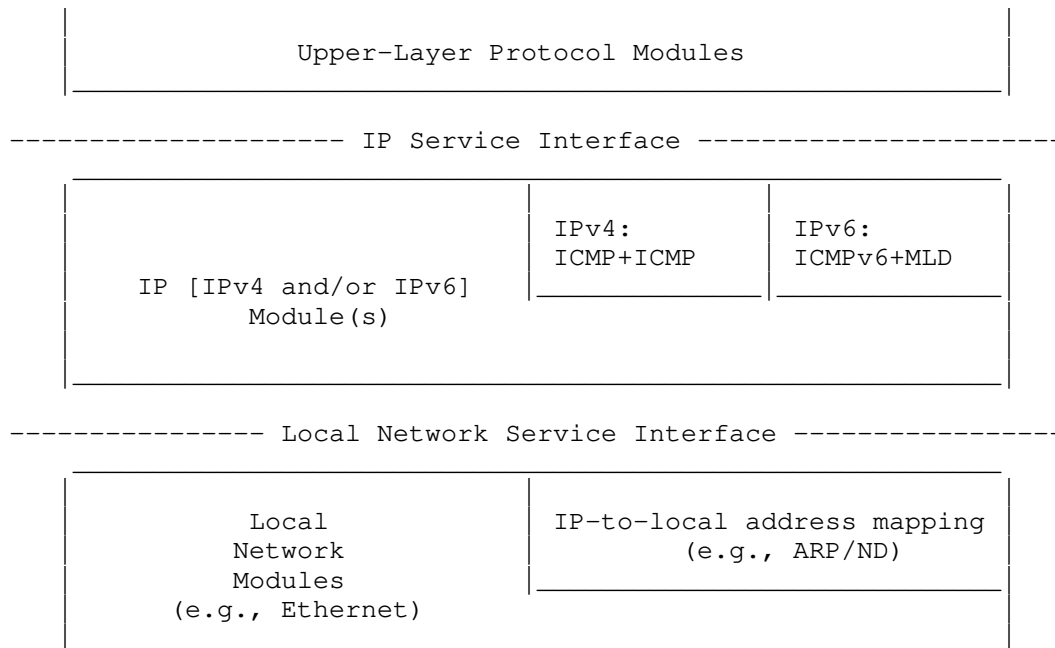
IPv4 and IPv6 addresses as specified in [RFC4607] are not used for ASM IP multicast and are not considered IP host groups. They are instead only the destination address part G of Source Specific Multicast (SSM)

IP multicast (S,G) channels.

Appendix I contains some background discussion of several issues related to host group addresses.

5. MODEL OF A HOST IP IMPLEMENTATION

The multicast extensions to a host IP implementation are specified in terms of the layered model illustrated below. In this model, ICMP/ICMPv6 and (for level 2 hosts) IGMP/MLD are considered to be implemented within the IP module, and the mapping of IP addresses to local network addresses is considered to be the responsibility of local network modules. This model is for expository purposes only, and should not be construed as constraining an actual implementation.



To provide level 1 multicasting, a host IP implementation must support the transmission of multicast IP datagrams. To provide level 2 multicasting, a host must also support the reception of multicast IP datagrams. Each of these two new services is described in a separate section, below. For each service, extensions are specified for the IP service interface, the IP module, the local network service interface, and an Ethernet local network module. Extensions to local network modules other than Ethernet are mentioned briefly, but are not specified in detail.

6. SENDING MULTICAST IP DATAGRAMS

6.1. Extensions to the IP Service Interface

Multicast IP datagrams are sent using the same "Send IP" operation used to send unicast IP datagrams; an upper-layer protocol module merely specifies an IP host group address, rather than an individual IP address, as the destination. However, a number of extensions may be necessary or desirable.

First, the service interface should provide a way for the upper-layer protocol to specify the IP time-to-live of an outgoing multicast datagram, if such a capability does not already exist. If the upper-layer protocol chooses not to specify a time-to-live, it should

default to 1 for all multicast IP datagrams, so that an explicit choice is required to multicast beyond a single network.

Second, for hosts that may be attached to more than one network, the service interface should provide a way for the upper-layer protocol to identify which network interface is to be used for the multicast transmission. Only one interface is used for the initial transmission; multicast routers are responsible for forwarding to any other networks, if necessary. If the upper-layer protocol chooses not to identify an outgoing interface, a default interface should be used, preferably under the control of system management.

Third (level 2 implementations only), for the case in which the host is itself a member of a group to which a datagram is being sent, the service interface should provide a way for the upper-layer protocol to inhibit local delivery of the datagram; by default, a copy of the datagram is looped back. This is a performance optimization for upper-layer protocols that restrict the membership of a group to one process per host (such as a routing protocol), or that handle loopback of group communication at a higher layer (such as a multicast transport protocol).

IPv6 socket extensions supporting these functions are defined in [RFC3493], section 5.2.

6.2. Extensions to the IP Module

To support the sending of multicast IP datagrams, the IP module must be extended to recognize IP host group addresses when routing outgoing datagrams. Most IP implementations include the following logic:

```
if IP-destination is on the same local network,
    send datagram locally to IP-destination
else
    send datagram locally to GatewayTo( IP-destination )
```

To allow multicast transmissions, the routing logic must be changed to:

```
if IP-destination is on the same local network
or IP-destination is a host group,
    send datagram locally to IP-destination
else
    send datagram locally to GatewayTo( IP-destination )
```

If the sending host is itself a member of the destination group on the outgoing interface, a copy of the outgoing datagram must be looped-back for local delivery, unless inhibited by the sender. (Level 2 implementations only.)

The IP source address of the outgoing datagram must be one of the individual addresses corresponding to the outgoing interface.

A host group address must never be placed in the source address field or anywhere in a source route or record route option of an outgoing IP datagram.

6.3. Extensions to the Local Network Service Interface

No change to the local network service interface is required to support the sending of multicast IP datagrams. The IP module merely specifies an IP host group destination, rather than an individual IP destination, when it invokes the existing "Send Local" operation.

6.4. Extensions to an Ethernet Local Network Module

The Ethernet directly supports the sending of local multicast packets by allowing multicast addresses in the destination field of Ethernet packets. All that is needed to support the sending of multicast IP datagrams is a procedure for mapping IP host group addresses to Ethernet multicast addresses.

An IPv4 host group address is mapped to an Ethernet multicast address by placing the low-order 23-bits of the IP address into the low-order 23 bits of the Ethernet multicast address 01-00-5E-00-00-00 (hex). Because there are 28 significant bits in an IP host group address, more than one host group address may map to the same Ethernet multicast address.

Mapping of IPv6 host group addresses to Ethernet is defined in [RFC2464] and [RFC6085].

6.5. Extensions to Local Network Modules other than Ethernet

Other networks that directly support multicasting, such as rings or buses conforming to the IEEE 802.2 standard, may be handled the same way as Ethernet for the purpose of sending multicast IP datagrams. For a network that supports broadcast but not multicast, such as the Experimental Ethernet, all IP host group addresses may be mapped to a single local broadcast address (at the cost of increased overhead on all local hosts). For a point-to-point link joining two hosts (or a host and a multicast router), multicasts should be transmitted exactly like unicasts. For a store-and-forward network like the ARPANET or a public X.25 network, all IP host group addresses might be mapped to the well-known local address of an IP multicast router; a router on such a network would take responsibility for completing multicast delivery within the network as well as among networks.

7. RECEIVING MULTICAST IP DATAGRAMS

7.1. Extensions to the IP Service Interface

Incoming multicast IP datagrams are received by upper-layer protocol modules using the same "Receive IP" operation as normal, unicast datagrams. Selection of a destination upper-layer protocol is based on the protocol field in the IP header, regardless of the destination IP address. However, before any datagrams destined to a particular group can be received, an upper-layer protocol must ask the IP module to join that group. Thus, the IP service interface must be extended to provide two new operations:

```
JoinHostGroup ( group-address, interface )
```

```
LeaveHostGroup ( group-address, interface )
```

The JoinHostGroup operation requests that this host become a member of the host group identified by "group-address" on the given network interface. The LeaveGroup operation requests that this host give up its membership in the host group identified by "group-address" on the given network interface. The interface argument may be omitted on hosts that support only one interface. For hosts that may be

attached to more than one network, the upper-layer protocol may choose to leave the interface unspecified, in which case the request will apply to the default interface for sending multicast datagrams (see section 6.1).

It is permissible to join the same group on more than one interface, in which case duplicate multicast datagrams may be received. It is also permissible for more than one upper-layer protocol to request membership in the same group.

Both operations should return immediately (i.e., they are non-blocking operations), indicating success or failure. Either operation may fail due to an invalid group address or interface identifier. JoinHostGroup may fail due to lack of local resources. LeaveHostGroup may fail because the host does not belong to the given group on the given interface. LeaveHostGroup may succeed, but the membership persist, if more than one upper-layer protocol has requested membership in the same group.

IPv6 socket extensions supporting these functions are defined in [RFC3493], section 5.2.

[RFC3678] specifies these functions for IPv4 and IPv6 (as well as for SSM).

7.2. Extensions to the IP Module

To support the reception of multicast IP datagrams, the IP module must be extended to maintain a list of host group memberships associated with each network interface. An incoming datagram destined to one of those groups is processed exactly the same way as datagrams destined to one of the host's individual addresses.

Incoming datagrams destined to groups to which the host does not belong are discarded without generating any error report or log entry. On hosts with more than one network interface, if a datagram arrives via one interface, destined for a group to which the host belongs only on a different interface, the datagram is quietly discarded. (These cases should occur only as a result of inadequate multicast address filtering in a local network module.)

An incoming datagram is not rejected for having an IP time-to-live of 1 (i.e., the time-to-live should not automatically be decremented on arriving datagrams that are not being forwarded). An incoming datagram with an IP host group address in its source address field is quietly discarded. An ICMP/ICMPv6 error message (Destination Unreachable, Time Exceeded, Parameter Problem, Source Quench, or Redirect) is never generated in response to a datagram destined to an IP host group.

The list of host group memberships is updated in response to JoinHostGroup and LeaveHostGroup requests from upper-layer protocols. Each membership should have an associated reference count or similar mechanism to handle multiple requests to join and leave the same group. On the first request to join and the last request to leave a group on a given interface, the local network module for that interface is notified, so that it may update its multicast reception filter (see section 7.3).

The IP module must also be extended to implement the IGMP protocol for IPv4 and the MLD protocol for IPv6. IGMP/MLD are used to keep neighboring multicast routers informed of the host group memberships present on a particular local network.

7.3. Extensions to the Local Network Service Interface

Incoming local network multicast packets are delivered to the IP module using the same "Receive Local" operation as local network unicast packets. To allow the IP module to tell the local network module which multicast packets to accept, the local network service interface is extended to provide two new operations:

```
JoinLocalGroup ( group-address )
```

```
LeaveLocalGroup ( group-address )
```

where "group-address" is an IP host group address. The JoinLocalGroup operation requests the local network module to accept and deliver up subsequently arriving packets destined to the given IP host group address. The LeaveLocalGroup operation requests the local network module to stop delivering up packets destined to the given IP host group address. The local network module is expected to map the IP host group addresses to local network addresses as required to update its multicast reception filter. Any local network module is free to ignore LeaveLocalGroup requests, and may deliver up packets destined to more addresses than just those specified in JoinLocalGroup requests, if it is unable to filter incoming packets adequately.

The local network module must not deliver up any multicast packets that were transmitted from that module; loopback of multicasts is handled at the IP layer or higher.

7.4. Extensions to an Ethernet Local Network Module

To support the reception of multicast IP datagrams, an Ethernet module must be able to receive packets addressed to the Ethernet multicast addresses that correspond to the host's IP host group addresses. It is highly desirable to take advantage of any address filtering capabilities that the Ethernet hardware interface may have, so that the host receives only those packets that are destined to it.

Unfortunately, many current Ethernet interfaces have a small limit on the number of addresses that the hardware can be configured to recognize. Nevertheless, an implementation must be capable of listening on an arbitrary number of Ethernet multicast addresses, which may mean "opening up" the address filter to accept all multicast packets during those periods when the number of addresses exceeds the limit of the filter.

For interfaces with inadequate hardware address filtering, it may be desirable (for performance reasons) to perform Ethernet address filtering within the software of the Ethernet module. This is not mandatory, however, because the IP module performs its own filtering based on IP destination addresses.

7.5. Extensions to Local Network Modules other than Ethernet

Other multicast networks, such as IEEE 802.2 networks, can be handled the same way as Ethernet for the purpose of receiving multicast IP datagrams. For pure broadcast networks, such as the Experimental Ethernet, all incoming broadcast packets can be accepted and passed to the IP module for IP-level filtering. On point-to-point or store-and-forward networks, multicast IP datagrams will arrive as local network unicasts, so no change to the local network module

should be necessary.

APPENDIX I. HOST GROUP ADDRESS ISSUES

This appendix is not part of the IP multicasting specification, but provides background discussion of several issues related to IP host group addresses.

Group Address Binding

The binding of IP host group addresses to physical hosts may be considered a generalization of the binding of IP unicast addresses. An IP unicast address is statically bound to a single local network interface on a single IP network. An IP host group address is dynamically bound to a set of local network interfaces on a set of IP networks.

It is important to understand that an IP host group address is NOT bound to a set of IP unicast addresses. The multicast routers do not need to maintain a list of individual members of each host group. For example, a multicast router attached to an Ethernet need associate only a single Ethernet multicast address with each host group having local members, rather than a list of the members' individual IP or Ethernet addresses.

Allocation of Transient Host Group Addresses

This memo does not specify how transient group address are allocated. It is anticipated that different portions of the IP transient host group address space will be allocated using different techniques. For example, there may be a number of servers that can be contacted to acquire a new transient group address. Some higher-level protocols (such as VMTP, specified in RFC-1045) may generate higher-level transient "process group" or "entity group" addresses which are then algorithmically mapped to a subset of the IP transient host group addresses, similarly to the way that IP host group addresses are mapped to Ethernet multicast addresses. A portion of the IP group address space may be set aside for random allocation by applications that can tolerate occasional collisions with other multicast users, perhaps generating new addresses until a suitably "quiet" one is found.

In general, a host cannot assume that datagrams sent to any host group address will reach only the intended hosts, or that datagrams received as a member of a transient host group are intended for the recipient. Misdelivery must be detected at a level above IP, using higher-level identifiers or authentication tokens. Information transmitted to a host group address should be encrypted or governed by administrative routing controls if the sender is concerned about unwanted listeners.

APPENDIX II. Changes from RFC1112

This document updates RFC1112 with the following changes:

- o It removes the claim that these host extensions are "... the recommended standard for IP multicasting in the Internet." (Status of the memo section). This is reflecting the experience of the past 30 years, in which Interdomain and even more so across the Internet has produced no full IETF standard solution for IPv4 and that SSM IP multicast is now considered the preferred standard for Interdomain/Internet deployments. See [draft-ietf-mboned-deprecate-interdomain-asm].

- o It is written to apply to both IPv4 and IPv6 by adding equivalent detail for IPv4 where RFC1112 only covered IPv4: addressing and protocol (MLD vs. IGMP).
- o It introduces the term "ASM IP multicast" as another term for "Host Extensions for IP multicast". This became necessary because RFC4607 introduced another service model for IP Multicast called "Source Specific Multicast" (SSM), and since then, the original service of RFC1112 is more precisely called Any Source Multicast (ASM) IP multicast.
- o It removes the original appendix I of RFC1112 that defined the IGMP version 1 protocol and text in the main document referring to details of IGMPv1 because that protocol is intended to be made obsolete as it is superseded by IGMPv3/IGMPv3.

APPENDIX III. Discussion and Explanations

[RFC-editor: Please remove this section]

Intention:

This document is intended to be an update to RFC1112 for the following reasons:

RFC1112 is at the time of this writing the only FULL INTERNET STANDARD describing the ASM IP Multicast Service Model for IPv4. There is no equivalent document for IPv6. RFC6434 section 5.10 is a stand-in, but it only describes MLD protocol version aspects but not the overall aspects of the host extensions and specifically the ASM IP multicast service. Instead it just refers to the term ASM and RFC1112.

RFC1112 includes the specification of IGMPv1. PIM WG would like to make IGMPv1 historic, but not the ASM host extensions / service. Instead, it ideally wants a FULL INTERNET STANDARD normative reference for ASM that applies to IPv4 and IPv6. This RFC1112bis is the attempt to do this.

Open Issues:

Has any document after RFC1112 (re-)defined the mapping of IPv4 multicast group addresses to ethernet multicast MAC ? If so, then we should include a reference to it and update the appropriate text in this rfc1112bis.

This document uses/defines the term "host group", which really is only a term relevant in this traditional ASM service model, but not in SSM. Therefore new text stating that IP multicast group addresses from RFC4607 are not included in this ASM definition. This is hopefully aligned with the text in RFC4607.

Is it appropriate to have included text to refer to socket API for ASM eg: rfc3678. These socket APIs are primarily about UDP sockets, and only rarely for IP level. This document only specifies an IP Service Interface. Note that RFC4607 (SSM) also refers to the socket interface extensions for SSM under a section called "Extensions to the IP Module Interface".

Discuss: Why would this document still need to be a standards track document ? IETF typically does not assign standard track to pure API/service-interface document. With IGMPv1 removed from the document, what does it still make it standards track, aka: define in a normative fashion interoperability impacting behavior of nodes ?

Answer1: Definition of the IPv4 ASM address space, Definition of the IP Multicast group to ethernet MAC address mapping for IPv4. The document now also contains references to these standards aspects for IPv6, but those are references to prior standards track documents.

Answer2: This document is similar in scope to RFC4607 (SSM) which is standards track. The newly defined interop impacting behavior on the wire are also

limited: address ranges for IPv4/IPv6 SSM, and it is referring to RFC1112.

Discuss: Which documents should we claim this document is updating ? Hopefully none other than rfc1112 - rfc1112 itself is referenced by > 60 RFCs.
si

Author's Addresses

Stephen E. Deering
Retired
Vancouver, British Columbia
Canada
Email: bob.hinden@gmail.com (email secretary)

Toerless Eckert (editor)
Futurewei Technologies Inc. USA
2220 Central Expy
Santa Clara, CA 95050
United States of America
Email: tte@cs.fau.de

PIM Working Group
Internet Draft
Intended status: Standards Track
Expires: February 28, 2022

H. Zhao
Ericsson
X. Liu
Volta
Y. Liu
China Mobile
M. Panchanathan
Cisco
M. Sivakumar
Juniper

August 30, 2021

A Yang Data Model for IGMP/MLD Proxy
draft-ietf-pim-igmp-mld-proxy-yang-06.txt

Abstract

This document defines a YANG data model that can be used to configure and manage Internet Group Management Protocol (IGMP) or Multicast Listener Discovery (MLD) proxy devices. The YANG module in this document conforms to Network Management Datastore Architecture (NMDA).

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), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>

This Internet-Draft will expire on February 28, 2022.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

| | |
|--|----|
| 1. Introduction..... | 3 |
| 1.1. Terminology..... | 3 |
| 1.2. Conventions Used in This Document..... | 3 |
| 1.3. Tree Diagrams..... | 4 |
| 1.4. Prefixes in Data Node Names..... | 4 |
| 2. Design of Data Model..... | 4 |
| 2.1. Overview..... | 5 |
| 2.2. Optional Capabilities..... | 5 |
| 2.3. Position of Address Family in Hierarchy..... | 5 |
| 3. Module Structure..... | 6 |
| 3.1. IGMP Proxy Configuration and Operational State..... | 6 |
| 3.2. MLD Proxy Configuration and Operational State..... | 7 |
| 4. IGMP/MLD Proxy YANG Module..... | 8 |
| 5. Security Considerations..... | 15 |
| 6. IANA Considerations..... | 16 |
| 6.1. XML Registry..... | 16 |
| 6.2. YANG Module Names Registry..... | 16 |
| 7. References..... | 17 |
| 7.1. Normative References..... | 17 |
| 7.2. Informative References..... | 18 |
| Appendix. Data Tree Example..... | 19 |
| Authors' Addresses..... | 22 |

1. Introduction

This document defines a YANG [RFC7950] data model for the management of Internet Group Management Protocol (IGMP) or Multicast Listener Discovery (MLD) Proxy [RFC4605] devices.

The YANG module in this document conforms to the Network Management Datastore Architecture defined in [RFC8342]. The "Network Management Datastore Architecture" (NMDA) adds the ability to inspect the current operational values for configuration, allowing clients to use identical paths for retrieving the configured values and the operational values.

1.1. Terminology

The terminology for describing YANG data models is found in [RFC6020] and [RFC7950], including:

- * augment
- * data model
- * data node
- * identity
- * module

The following abbreviations are used in this document and defined model:

IGMP: Internet Group Management Protocol [RFC3376].

MLD: Multicast Listener Discovery [RFC3810].

PIM: Protocol Independent Multicast [RFC7761].

1.2. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.3. Tree Diagrams

Tree diagrams used in this document follow the notation defined in [RFC8340].

1.4. Prefixes in Data Node Names

In this document, names of data nodes, and other data model objects are often used without a prefix, as long as it is clear from the context in which YANG module each name is defined. Otherwise, names are prefixed using the standard prefix associated with the corresponding YANG module, as shown in Table 1.

| Prefix | YANG module | Reference |
|----------|--------------------|-----------------------|
| inet | ietf-inet-types | [RFC6991] |
| if | ietf-interfaces | [RFC8343] |
| rt | ietf-routing | [RFC8349] |
| rt-types | ietf-routing-types | [RFC8294] |
| pim-base | ietf-pim-base | [draft-ietf-pim-yang] |

Table 1: Prefixes and Corresponding YANG Modules

2. Design of Data Model

The model covers Considerations for Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD) - Based Multicast Forwarding ("IGMP/MLD Proxying") [RFC4605].

The goal of this document is to define a data model that provides a common user interface to IGMP/MLD Proxy. This document provides freedom for vendors to adapt the data model to their product implementations.

2.1. Overview

The model defined in this document has all the common building blocks for the IGMP/MLD Proxy devices. It can be used to configure IGMP/MLD Proxy. The operational state data and statistics can also be retrieved by it.

The model contains all the basic configuration parameters. The occasionally implemented parameters are modeled as optional features in this model, while the rarely implemented parameters are not included in this model and left for augmentation.

2.2. Optional Capabilities

This model is designed to represent the basic capability subsets of IGMP / MLD Proxy. The main design goals of this document are that the basic capabilities described in the model are supported by any major now-existing implementation, and that the configuration of all implementations meeting the specifications is easy to express through some combination of the optional features in the model and simple vendor augmentations.

There is also value in widely supported features being standardized, to provide a standardized way to access these features, to save work for individual vendors, and so that mapping between different vendors' configuration is not needlessly complicated. Therefore, this model declares a number of features representing capabilities that not all deployed devices support.

The extensive use of feature declarations should also substantially simplify the capability negotiation process for a vendor's IGMP / MLD Proxy implementations.

2.3. Position of Address Family in Hierarchy

IGMP Proxy only supports IPv4, while MLD Proxy only supports IPv6. The data model defined in this document can be used for both IPv4 and IPv6 address families.

This document defines IGMP Proxy and MLD Proxy as separate schema branches in the structure. The benefits are:

- * The model can support IGMP Proxy (IPv4), MLD Proxy (IPv6), or both optionally and independently. Such flexibility cannot be achieved cleanly with a combined branch.

* The structure is consistent with other YANG data models such as [RFC8652], which uses separate branches for IPv4 and IPv6.

* Having separate branches for IGMP Proxy and MLD Proxy allows minor differences in their behavior to be modelled more simply and cleanly. The two branches can better support different features and node types.

3. Module Structure

This model augments the core routing data model specified in [RFC8349].

```
+--rw routing
  +--rw router-id?
  +--rw control-plane-protocols
    |   +--rw control-plane-protocol* [type name]
    |   |   +--rw type
    |   |   +--rw name
    |   |   +--rw igmp-proxy <= Augmented by this Model
    |   |   ...
    |   +--rw mld-proxy <= Augmented by this Model
```

The "igmp-proxy" container instantiates IGMP Proxy. The "mld-proxy" container instantiates MLD Proxy.

The YANG data model defined in this document conforms to the Network Management Datastore Architecture (NMDA) [RFC8342]. The operational state data is combined with the associated configuration data in the same hierarchy [RFC8407].

3.1. IGMP Proxy Configuration and Operational State

The YANG module augments /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol to add the igmp-proxy container.

All the IGMP Proxy related attributes are defined in the igmp-proxy container. The read-write attributes represent configurable data. The read-only attributes represent state data.

The igmp-version represents version of IGMP protocol, and default value is 2. If the value of enable is true, it means IGMP Proxy is enabled.

The interface list under igmp-proxy contains upstream interfaces for IGMP proxy. There is also a constraint to make sure the upstream interface for IGMP proxy should not be configured PIM.

To configure a downstream interface for IGMP proxy, it is needed to enable IGMP on that interface. This is defined in the YANG Data Model

for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) [RFC8652].

```

augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol:
    +--rw igmp-proxy {igmp-proxy}?
      +--rw interfaces
        +--rw interface* [interface-name]
          +--rw interface-name          if:interface-ref
          +--rw igmp-version?           uint8
          +--rw enable?                 boolean
          +--rw sender-source-address?  inet:ipv4-address
          +--ro group* [group-address]
            +--ro group-address
              | rt-types:ipv4-multicast-group-address
            +--ro up-time?               uint32
            +--ro filter-mode            enumeration
            +--ro source* [source-address]
              +--ro source-address       inet:ipv4-address
              +--ro up-time?             uint32
              +--ro downstream-interface* [interface-name]
                +--ro interface-name    if:interface-ref

```

3.2. MLD Proxy Configuration and Operational State

The YANG module augments /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol to add the mld-proxy container.

All the MLD Proxy related attributes are defined in the mld-proxy container. The read-write attributes represent configurable data. The read-only attributes represent state data.

The mld-version represents version of MLD protocol, and default value is 2. If the value of enable is true, it means MLD Proxy is enabled.

The interface list under mld-proxy contains upstream interfaces for MLD proxy. There is also a constraint to make sure the upstream interface for MLD proxy should not be configured PIM.

To configure a downstream interface for MLD proxy, enable MLD on that interface. This is defined in the YANG Data Model for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) [RFC8652].

```

augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol:
    +--rw mld-proxy {mld-proxy}?
      +--rw interfaces
        +--rw interface* [interface-name]

```

```

+--rw interface-name          if:interface-ref
+--rw mld-version?            uint8
+--rw enable?                 boolean
+--rw sender-source-address?  inet:ipv6-address
+--ro group* [group-address]
  +--ro group-address
    |   rt-types:ipv6-multicast-group-address
  +--ro up-time?              uint32
  +--ro filter-mode           enumeration
  +--ro source* [source-address]
    +--ro source-address      inet:ipv6-address
    +--ro up-time?            uint32
    +--ro downstream-interface* [interface-name]
      +--ro interface-name    if:interface-ref

```

4. IGMP/MLD Proxy YANG Module

This module references [RFC4605], [RFC6991], [RFC8294], [RFC8343], [RFC8349] and [draft-ietf-pim-yang].

```

<CODE BEGINS> file ietf-igmp-mld-proxy@2021-04-21.yang
module ietf-igmp-mld-proxy {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-igmp-mld-proxy";
  // replace with IANA namespace when assigned
  prefix igmp-mld-proxy;

  import ietf-inet-types {
    prefix inet;
  }
  import ietf-interfaces {
    prefix if;
  }
  import ietf-routing {
    prefix rt;
  }
  import ietf-routing-types {
    prefix rt-types;
  }
  import ietf-pim-base {
    prefix pim-base;
  }

  organization
    "IETF PIM Working Group";

  contact
    "WG Web:  <http://tools.ietf.org/wg/pim/>
    WG List:  <mailto:pim@ietf.org>

```

Editors: Hongji Zhao
<mailto:hongji.zhao@ericsson.com>

Xufeng Liu
<mailto:xufeng.liu.ietf@gmail.com>

Yisong Liu
<mailto:liuyisong@chinamobile.com>

Mani Panchanathan
<mailto:mapancha@cisco.com>

Mahesh Sivakumar
<mailto:sivakumar.mahesh@gmail.com>

";

description

"The module defines a collection of YANG definitions common for all Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Proxy devices.

Copyright (c) 2021 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 2021-04-21 {  
  description  
    "Initial revision.";  
  reference  
    "RFC XXXX: A YANG Data Model for IGMP and MLD Proxy";  
}
```

```
/*  
 * Features  
 */
```

```
feature igmp-proxy {  
  description  
    "Support IGMP Proxy protocol.";  
  reference  
    "RFC 4605";
```

```
}

feature mld-proxy {
  description
    "Support MLD Proxy protocol.";
  reference
    "RFC 4605";
}

/*
 * Identities
 */

identity igmp-proxy {
  base rt:control-plane-protocol;
  description
    "IGMP Proxy protocol";
}

identity mld-proxy {
  base rt:control-plane-protocol;
  description
    "MLD Proxy protocol";
}

/*
 * Groupings
 */

grouping per-interface-config-attributes {
  description "Config attributes under interface view";
  leaf enable {
    type boolean;
    default false;
    description
      "Set the value to true to enable IGMP/MLD proxy";
  }
} // per-interface-config-attributes

grouping state-group-attributes {
  description
    "State group attributes";
  leaf up-time {
    type uint32;
    units seconds;
    description
      "The elapsed time for (S,G) or (*,G).";
  }
  leaf filter-mode {
    type enumeration {
      enum "include" {
```

```
        description
            "In include mode, reception of packets sent
            to the specified multicast address is requested
            only from those IP source addresses listed in the
            source-list parameter";
    }
    enum "exclude" {
        description
            "In exclude mode, reception of packets sent
            to the given multicast address is requested
            from all IP source addresses except those
            listed in the source-list parameter.";
    }
}
mandatory true;
description
    "Filter mode for a multicast group,
    may be either include or exclude.";
}
} // state-group-attributes

/* augments */

augment "/rt:routing/rt:control-plane-protocols"+
    "/rt:control-plane-protocol" {
    when
        "derived-from-or-self(rt:type, 'igmp-mld-proxy:igmp-proxy')" {
            description
                "This augmentation is only valid for IGMP Proxy.";
        }
    description
        "IGMP Proxy augmentation to routing control plane protocol
        configuration and state.";
    container igmp-proxy {
        if-feature "igmp-proxy";
        description "IGMP proxy";
        container interfaces {
            description
                "Containing a list of upstream interfaces.";
            list interface {
                key "interface-name";
                description
                    "List of upstream interfaces.";
                leaf interface-name {
                    type if:interface-ref;
                    must "not( current() = /rt:routing"+
                        "/rt:control-plane-protocols/pim-base:pim"+
                        "/pim-base:interfaces/pim-base:interface"+
                        "/pim-base:name )" {
                        description
```

```
        "The upstream interface for IGMP proxy
        should not be configured PIM.";
    }
    description "The upstream interface name.";
}
leaf igmp-version {
    type uint8 {
        range "1..3";
    }
    default 2;
    description "IGMP version.";
}
uses per-interface-config-attributes;
leaf sender-source-address {
    type inet:ipv4-address;
    description
        "The sender source address of
        IGMP membership report or leave.";
}
list group {
    key "group-address";
    config false;
    description
        "Multicast group membership information
        that joined on the interface.";
    leaf group-address {
        type rt-types:ipv4-multicast-group-address;
        description
            "Multicast group address.";
    }
    uses state-group-attributes;
    list source {
        key "source-address";
        description
            "List of multicast source information
            of the multicast group.";
        leaf source-address {
            type inet:ipv4-address;
            description
                "Multicast source address";
        }
    }
    leaf up-time {
        type uint32;
        units seconds;
        description
            "The elapsed time for (S,G) or (*,G).";
    }
    list downstream-interface {
        key "interface-name";
        description "The downstream interfaces list.";
        leaf interface-name {
```

```

        type if:interface-ref;
        description
            "Downstream interfaces
             for each upstream-interface";
    }
    }
    } // list source
    } // list group
    } // interface
    } // interfaces
}
}

augment "/rt:routing/rt:control-plane-protocols"+
    "/rt:control-plane-protocol" {
    when
        "derived-from-or-self(rt:type, 'igmp-mld-proxy:mld-proxy')" {
        description
            "This augmentation is only valid for MLD Proxy.";
        }
    description
        "MLD Proxy augmentation to routing control plane protocol
         configuration and state.";
    container mld-proxy {
        if-feature "mld-proxy";
        description "MLD proxy";
        container interfaces {
            description
                "Containing a list of upstream interfaces.";
            list interface {
                key "interface-name";
                description
                    "List of upstream interfaces.";
                leaf interface-name {
                    type if:interface-ref;
                    must "not( current() = /rt:routing"+
                        "/rt:control-plane-protocols/pim-base:pim"+
                        "/pim-base:interfaces/pim-base:interface"+
                        "/pim-base:name )" {
                        description
                            "The upstream interface for MLD proxy
                             should not be configured PIM.";
                    }
                }
                description "The upstream interface name.";
            }
            leaf mld-version {
                type uint8 {
                    range "1..2";
                }
                default 2;
                description "MLD version.";
            }
        }
    }
}

```



```
    }
    uses per-interface-config-attributes;
    leaf sender-source-address {
        type inet:ipv6-address;
        description
            "The sender source address of
             MLD membership report or leave.";
    }
    list group {
        key "group-address";
        config false;
        description
            "Multicast group membership information
             that joined on the interface.";
        leaf group-address {
            type rt-types:ipv6-multicast-group-address;
            description
                "Multicast group address.";
        }
    }
    uses state-group-attributes;
    list source {
        key "source-address";
        description
            "List of multicast source information
             of the multicast group.";
        leaf source-address {
            type inet:ipv6-address;
            description
                "Multicast source address";
        }
        leaf up-time {
            type uint32;
            units seconds;
            description
                "The elapsed time for (S,G) or (*,G).";
        }
        list downstream-interface {
            key "interface-name";
            description "The downstream interfaces list.";
            leaf interface-name {
                type if:interface-ref;
                description
                    "Downstream interfaces
                     for each upstream-interface";
            }
        }
    } // list source
} // list group
} // interface
} // interfaces
}
```

```
    }  
  }  
<CODE ENDS>
```

5. Security Considerations

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

The Network Configuration Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

Under /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol:/

igmp-mld-proxy:igmp-proxy

igmp-mld-proxy:mld-proxy

Unauthorized access to any data node of these subtrees can adversely affect the IGMP / MLD Proxy subsystem of both the local device and the network. This may lead to network malfunctions, delivery of packets to inappropriate destinations, and other problems.

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

Under /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol:/

igmp-mld-proxy:igmp-proxy

igmp-mld-proxy:mld-proxy

Unauthorized access to any data node of these subtrees can disclose the operational state information of IGMP / MLD Proxy on this device. The group/source information may expose multicast group memberships.

6. IANA Considerations

RFC Ed.: In this section, replace all occurrences of 'XXXX' with the actual RFC number (and remove this note).

6.1. XML Registry

This document registers the following namespace URIs in the IETF XML registry [RFC3688]:

URI: urn:ietf:params:xml:ns:yang:ietf-igmp-mld-proxy
Registrant Contact: The IETF.
XML: N/A, the requested URI is an XML namespace.

6.2. YANG Module Names Registry

This document registers the following YANG modules in the YANG Module Names registry [RFC7950]:

| | |
|------------|---|
| name: | ietf-igmp-mld-proxy |
| namespace: | urn:ietf:params:xml:ns:yang:ietf-igmp-mld-proxy |
| prefix: | igmp-mld-proxy |
| reference: | RFC XXXX |

7. References

7.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.
- [RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, October 2002.
- [RFC3688] Mealling, M., "The IETF XML Registry", RFC 3688, January 2004.
- [RFC3810] Vida, R. and L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, June 2004.
- [RFC4605] B. Fenner, H. He, B. Haberman and H. Sandick, "Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD) - Based Multicast Forwarding ("IGMP/MLD Proxying")", RFC 4605, August 2006.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.
- [RFC6241] R. Enns, Ed., M. Bjorklund, Ed., J. Schoenwaelder, Ed., A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, June 2011.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, June 2011.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, July 2013.
- [RFC7950] M. Bjorklund, Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, August 2016.
- [RFC8040] A. Bierman, M. Bjorklund, K. Watsen, "RESTCONF Protocol", RFC 8040, January 2017.
- [RFC8174] B. Leiba, "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", RFC 8174, May 2017.
- [RFC8294] X. Liu, Y. Qu, A. Lindem, C. Hopps, L. Berger, "Common YANG Data Types for the Routing Area", RFC 8294, December 2017.
- [RFC8340] M. Bjorklund, and L. Berger, Ed., "YANG Tree Diagrams", RFC 8340, March 2018.

- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", RFC 8341, March 2018.
- [RFC8342] M. Bjorklund and J. Schoenwaelder, "Network Management Datastore Architecture (NMDA)", RFC 8342, March 2018.
- [RFC8343] M. Bjorklund, "A YANG Data Model for Interface Management", RFC 8343, March 2018.
- [RFC8349] L. Lhotka, A. Lindem, Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", RFC 8349, March 2018.
- [RFC8407] A. Bierman, "Guidelines for Authors and Reviewers of Documents Containing YANG Data Models", RFC 8407, October 2018.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, August 2018.
- [RFC8652] X. Liu, F. Guo, M. Sivakumar, P. McAllister, A. Peter, "A YANG Data Model for the Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD)", RFC 8652, November 2019.
- [draft-ietf-pim-yang] X. Liu, P. McAllister, A. Peter, M. Sivakumar, Y. Liu, F. Hu, "A YANG Data Model for Protocol Independent Multicast (PIM)", draft-ietf-pim-yang-17 (RFC Editor state is MISSREF), May 2018.

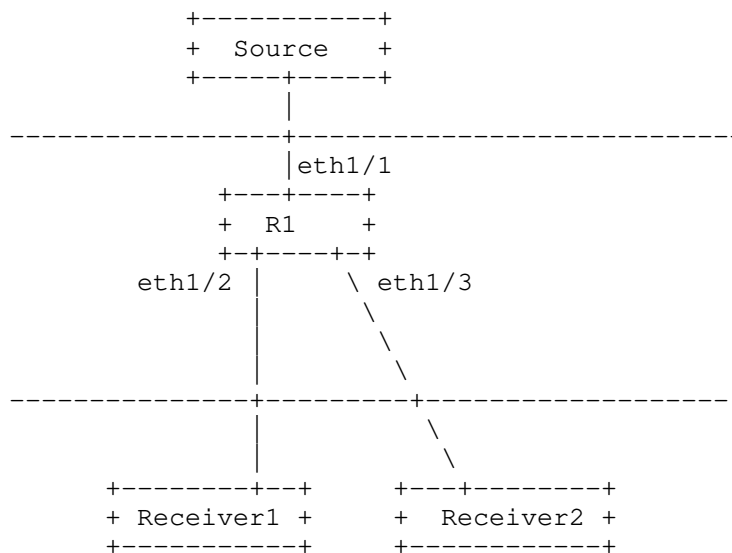
7.2. Informative References

- [RFC7761] B. Fenner, M. Handley, H. Holbrook, I. Kouvelas, R. Parekh, Z. Zhang, L. Zheng, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", RFC 7761, March 2016.
- [RFC7951] L. Lhotka, "JSON Encoding of Data Modeled with YANG", RFC 7951, August 2016.

Appendix. Data Tree Example

This section contains an example for IGMP Proxy in the JSON encoding [RFC7951], containing both configuration and state data. In the example IGMP Proxy is enabled on interface eth1/1.

It is also needed to enable IGMP on eth1/2 and eth1/3. The configuration details are omitted here because this document is focused on IGMP/MLD Proxy.



The configuration data for R1 in the above figure could be as follows:

```

{
  "ietf-interfaces:interfaces": {
    "interface": [
      {
        "name": "eth1/1",
        "type": "iana-if-type:ipForward",
        "ietf-ip:ipv4": {
          "address": [
            {
              "ip": "11.0.0.1",
              "prefix-length": 24
            }
          ]
        }
      }
    ]
  }
}

```

```

    ]
  },
  "ietf-routing:routing": {
    "control-plane-protocols": {
      "control-plane-protocol": [
        {
          "type": "ietf-igmp-mld-proxy:igmp-proxy",
          "name": "proxy1",
          "ietf-igmp-mld-proxy:igmp-proxy": {
            "interfaces": {
              "interface": [
                {
                  "interface-name": "eth1/1",
                  "igmp-version": 3,
                  "enable": true
                }
              ]
            }
          }
        }
      ]
    }
  }
}

```

The corresponding operational state data for R1 could be as follows:

```

{
  "ietf-interfaces:interfaces": {
    "interface": [
      {
        "name": "eth1/1",
        "type": "iana-if-type:ipForward",
        "admin-status": "up",
        "oper-status": "up",
        "if-index": 25678136,
        "statistics": {
          "discontinuity-time": "2021-05-23T10:34:56-06:00"
        },
        "ietf-ip:ipv4": {
          "address": [
            {
              "ip": "11.0.0.1",
              "prefix-length": 24
            }
          ]
        }
      }
    ]
  }
}

```

```

"ietf-routing:routing": {
  "control-plane-protocols": {
    "control-plane-protocol": [
      {
        "type": "ietf-igmp-mld-proxy:igmp-proxy",
        "name": "proxyl",
        "ietf-igmp-mld-proxy:igmp-proxy": {
          "interfaces": {
            "interface": [
              {
                "interface-name": "eth1/1",
                "igmp-version": 3,
                "enable": true,
                "group": [
                  {
                    "group-address": "225.0.0.1",
                    "filter-mode": "include",
                    "source": [
                      {
                        "source-address": "1.1.1.1",
                        "downstream-interface": [
                          {
                            "interface-name": "eth1/2"
                          },
                          {
                            "interface-name": "eth1/3"
                          }
                        ]
                      }
                    ]
                  }
                ]
              }
            ]
          }
        }
      }
    ]
  }
}

```


Authors' Addresses

Hongji Zhao
Ericsson (China) Communications Company Ltd.
Ericsson Tower, No. 5 Lize East Street,
Chaoyang District Beijing 100102, China
Email: hongji.zhao@ericsson.com

Xufeng Liu
Volta Networks
USA
EMail: Xufeng.liu.ietf@gmail.com

Yisong Liu
China Mobile
China
Email: liuyisong@chinamobile.com

Mani Panchanathan
Cisco
India
Email: mapancha@cisco.com

Mahesh Sivakumar
Juniper Networks
1133 Innovation Way
Sunnyvale, California
USA
EMail: sivakumar.mahesh@gmail.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 13, 2022

Z. Li
Huawei Technologies
G. Mishra
Verizon Inc.
Z. Qin
China Unicom
July 12, 2021

Gap Analysis of IPv6 Multicast Source Routing (MSR6)
draft-li-spring-ipv6-msr-gap-analysis-00

Abstract

This document analyses the gaps of the existing IPv6 multicast solutions under discussion in IETF based on the requirements.

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 <https://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 January 13, 2022.

Copyright Notice

Copyright (c) 2021 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 (<https://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. Gap Analysis | 3 |
| 2.1. BIERin6 | 3 |
| 2.2. BIERv6 | 4 |
| 2.3. Historical Review | 5 |
| 3. Summary | 5 |
| 4. IANA Considerations | 6 |
| 5. Security Considerations | 6 |
| 6. Acknowledgements | 6 |
| 7. Normative References | 6 |
| Authors' Addresses | 7 |

1. Introduction

Multicast could provide efficient P2MP service without bandwidth waste. The increasing amount of live video traffic in the network bring new requirements for multicast solutions. The existing multicast solutions request multicast tree-building on control plane and maintaining end-to-end tree state per flow, which impacts router state capacity and network convergence time. There has been a lot of work in IETF to simplify service deployment, in which Source Routing is a very important technology, including SRv6, BIER, etc. Source routing is able to reduce the state of intermediate nodes and indicate multicast forwarding in the ingress nodes, which could simplify multicast deployment. Source routing requires sufficient flexibility on the forwarding plane and IPv6 has the advantage with good scalability. Therefore, it is important to simplify multicast deployment and meet high quality service requirements with IPv6 Source Routing based multicast.

Based on the design consideration defined in [I-D.cheng-spring-ipv6-msr-design-consideration], this document analyses the gaps of the existing IPv6 multicast solutions under discussion in IETF. The definition of the new IPv6 multicast source routing solution is out of the scope of this document.

2. Gap Analysis

2.1. BIERin6

The solution described in [I-D.zhang-bier-bierin6] is called BIERin6. The encapsulation of BIERin6 is as follows:

| | | |
|-------------------------------------|--|---|
| IPv6 header Ethertype= 0xAB37 | BIER Header defined in RFC8296 Protocol = X | X type of C-multicast packet (IPv4/IPv6/Ethernet) |
| <--IPv6 header--> | <--BIER Header--> | <--BIERin6 Payload--> |

BIERin6 proposes to encapsulate the IPv6 header before the BIER header in order to transit the BIER header through IPv6 nodes by adding IPv6 header between BIER forwarding nodes. The "next header" codepoint of the IPv6 header is set to the value that means that "the next header is non-MPLS BIER".

BIERin6 implements P2MP forwarding follows the BIER architecture defined in [RFC8279].

There are some issues to be considered with the classic Layered architecture of BIER used by BIERin6 in supporting IPv6 Multicast Source Routing:

1. Support non-native IPv6 scenarios : In BIERin6, IPv6 is only used as the transport tunnel to transit the IPv4 domain. In fact this method can also be used in the IPv4 to traverse the IPv4 domain. Moreover the service layer above the transport tunnel can be non-IPv6. For example, in BIERin6 MVPN could use MPLS label for VPN identification. Unlike BIERin6, IPv6 Multicast Source Routing is supposed to be a native IPv6 solution.

2. Architecture Considerations:

- 1) When the BIER layer is treated as an independent layer to support features mentioned in section 2, the new encapsulation for fragment, security, network slicing, DetNet, IOAM has to be defined in the BIER layer. Moreover, this will also cause redundancy and conflicts when BIER is used with IPv6 layer or MPLS layer since the encapsulations for these functionalities are also defined for the IPv6 layer and MPLS layer;

2) When the BIER encapsulation is treated as a separate layer and the rest of the functionalities are realized in the IPv6 layer, it also causes problems. For example, if encryption is supported in the IPv6 data plane, it makes the contents of the BIER layer encrypted and unprocessable; if IOAM and RH are supported in the IPv6 data plane which cause the more overhead, it would make it difficult to get information on the BIER layer and has much effect on the forwarding performance. In addition, if IOAM is engaged, this leads to information loss when IPv6 encapsulation is switched in the middle nodes and cannot be implemented end-to-end.

3. BIERin6 is hard to complete end-to-end service to support host initiating source routing. If BIERin6 is used in the host, the source address information can be lost in the middle nodes. Moreover, the BIER layer as an independent network layer above IPv6, just as TCP or UDP, will cause conflictions with the transport layer, and have more impact on the Internet architecture.

4. Maintenance of tunnel state: BIERin6 needs to maintain the state of the tunnel in the middle nodes when traversing IPv6 domains, which adds complexity to service deployment. When new functionalities mentioned in the sectioned such as network slicing, Detnet, IOAM, etc. are applied when traverse the IPv6 domains, it will cause more complexity in service provisioning and more network state maintenance.

5. Based on existing functionalities, MPVPN/TE/Fragmentation/ESP need be supported by BIERin6.

2.2. BIERv6

The solution described in [I-D.xie-bier-ipv6-encapsulation] is called BIERv6. The encapsulation of BIERv6 is as follows:

| | | |
|---------------------------|------------------------------------|---------------------------------|
| IPv6 header | IPv6 DO Header with BIER Option | X type of C-multicast packet |
| Next Hdr = 60 | Nxt Hdr = X | (IPv4/IPv6/Ethernet) |
| <-----BIERv6 header-----> | | <---BIERv6 payload---> |

BIERv6 proposes to define a new type of destination options header for BIER. The destination address of the IPv6 header indicates the BIER forwarding nodes and changed to the next BIER forwarding nodes based on the result of BIER forwarding table lookup.

BIERv6 implements P2MP forwarding follows the BIER architecture defined in [RFC8279].

BIERv6 uses Native IPv6 extension header to carry BIER info. And BIERv6 could support MVPN by defining MVPN indication in source address of the outer IPv6 header, which doesn't change along the P2MP tunnel. The MVPN mechanism for BIERv6 is defined in [I-D.xie-bier-ipv6-mvpn]. BIERv6 is able to directly reuse the new functionalities supported by IPv6 and SRv6 without the problems associated with Layered Architecture. In addition, BIERv6 uses Native IPv6 and can be started directly at the Host without conflicts with the transport layer.

BIERv6 has the following challenges:

1. Non-MPLS BIER header defined in [RFC8296] is used, but the BIER header is designed as a separate layer, leaving some fields unused and redundant in IPv6.
2. BIERv6 needs to support MVPN and Traffic Engineering.

2.3. Historical Review

In the field of IPv6 unicast source routing, there has been SR over IP ([RFC8663]) and SRv6. SR over IP takes the layered architecture while SRv6 adopts native IPv6 design. Both solutions has different application scenarios though there is the common functionality to traverse IPv6 domain. SR over IP controls the scope to support more new functionalities in the IPv6 data plane. SRv6 is being developed combing with the new functionalities based on the IPv6 extension.

3. Summary

MSR6 is supposed to have:

- o Native IPv6 design to reduce header layers and enable unified processing
- o Reuse existing IPv6 capabilities and SRv6 capabilities for multicast
- o BIER is able to implement network programming at the ingress nodes in Best Effort scenarios. MSR6 needs to take advantage of the capabilities in the existing BIER mechanism
- o MVPN and Traffic Engineering support are requested

The existing multicast solutions have their own limitations which constrains the deployment and development of multicast. New multicast solution is expected in IETF.

4. IANA Considerations

This document makes no request of IANA.

5. Security Considerations

TBD

6. Acknowledgements

TBD

7. Normative References

[I-D.cheng-spring-ipv6-msr-design-consideration]

Cheng, W., Mishra, G., Li, Z., Wang, A., Qin, Z., and C. Fan, "Design Consideration of IPv6 Multicast Source Routing (MSR6)", draft-cheng-spring-ipv6-msr-design-consideration-00 (work in progress), July 2021.

[I-D.xie-bier-ipv6-encapsulation]

Xie, J., Geng, L., McBride, M., Asati, R., Dhanaraj, S., Zhu, Y., Qin, Z., Shin, M., Mishra, G., and X. Geng, "Encapsulation for BIER in Non-MPLS IPv6 Networks", draft-xie-bier-ipv6-encapsulation-10 (work in progress), February 2021.

[I-D.xie-bier-ipv6-mvpn]

Xie, J., McBride, M., Dhanaraj, S., Geng, L., and G. Mishra, "Use of BIER IPv6 Encapsulation (BIERv6) for Multicast VPN in IPv6 networks", draft-xie-bier-ipv6-mvpn-03 (work in progress), October 2020.

[I-D.zhang-bier-bierin6]

Zhang, Z., Zhang, Z., Wijnands, I., Mishra, M., Bidgoli, H., and G. Mishra, "Supporting BIER in IPv6 Networks (BIERin6)", draft-zhang-bier-bierin6-09 (work in progress), February 2021.

[RFC2119]

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

- [RFC8279] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast Using Bit Index Explicit Replication (BIER)", RFC 8279, DOI 10.17487/RFC8279, November 2017, <<https://www.rfc-editor.org/info/rfc8279>>.
- [RFC8296] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Tantsura, J., Aldrin, S., and I. Meilik, "Encapsulation for Bit Index Explicit Replication (BIER) in MPLS and Non-MPLS Networks", RFC 8296, DOI 10.17487/RFC8296, January 2018, <<https://www.rfc-editor.org/info/rfc8296>>.
- [RFC8663] Xu, X., Bryant, S., Farrel, A., Hassan, S., Henderickx, W., and Z. Li, "MPLS Segment Routing over IP", RFC 8663, DOI 10.17487/RFC8663, December 2019, <<https://www.rfc-editor.org/info/rfc8663>>.

Authors' Addresses

Zhenbin Li
Huawei Technologies

Email: lizhenbin@huawei.com

Gyan Mishra
Verizon Inc.

Email: gyan.s.mishra@verizon.com

Zhuangzhuang Qin
China Unicom

Email: qinzhuangzhuang@chinaunicom.cn

Network Working Group
Internet-Draft
Obsoletes: 3376 (if approved)
Intended status: Standards Track
Expires: January 13, 2022

B. Haberman, Ed.
JHU APL
July 12, 2021

Internet Group Management Protocol, Version 3
draft-mcast-pim-3376bis-00

Abstract

This document specifies a revised Version 3 of the Internet Group Management Protocol, IGMPv3. IGMP is the protocol used by IPv4 systems to report their IP multicast group memberships to neighboring multicast routers. Version 3 of IGMP adds support for source filtering, that is, the ability for a system to report interest in receiving packets only from specific source addresses, or from all but specific source addresses, sent to a particular multicast address. That information may be used by multicast routing protocols to avoid delivering multicast packets from specific sources to networks where there are no interested receivers.

This document obsoletes RFC 3376.

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 <https://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 January 13, 2022.

Copyright Notice

Copyright (c) 2021 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 (<https://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 | 4 |
| 2. The Service Interface for Requesting IP Multicast Reception . | 5 |
| 3. Multicast Reception State Maintained by Systems | 6 |
| 3.1. Socket State | 6 |
| 3.2. Interface State | 7 |
| 4. Message Formats | 9 |
| 4.1. Membership Query Message | 10 |
| 4.1.1. Max Resp Code | 11 |
| 4.1.2. Checksum | 12 |
| 4.1.3. Group Address | 12 |
| 4.1.4. Resv (Reserved) | 12 |
| 4.1.5. S Flag (Suppress Router-Side Processing) | 12 |
| 4.1.6. QRV (Querier's Robustness Variable) | 12 |
| 4.1.7. QQIC (Querier's Query Interval Code) | 12 |
| 4.1.8. Number of Sources (N) | 13 |
| 4.1.9. Source Address [i] | 13 |
| 4.1.10. Additional Data | 13 |
| 4.1.11. Query Variants | 14 |
| 4.1.12. IP Destination Addresses for Queries | 14 |
| 4.2. Version 3 Membership Report Message | 14 |
| 4.2.1. Reserved | 16 |
| 4.2.2. Checksum | 16 |
| 4.2.3. Number of Group Records (M) | 16 |
| 4.2.4. Group Record | 16 |
| 4.2.5. Record Type | 17 |
| 4.2.6. Aux Data Len | 17 |
| 4.2.7. Number of Sources (N) | 17 |
| 4.2.8. Multicast Address | 17 |
| 4.2.9. Source Address [i] | 17 |
| 4.2.10. Auxiliary Data | 17 |
| 4.2.11. Additional Data | 17 |
| 4.2.12. Group Record Types | 18 |
| 4.2.13. IP Source Addresses for Reports | 19 |
| 4.2.14. IP Destination Addresses for Reports | 19 |
| 4.2.15. Notation for Group Records | 20 |
| 4.2.16. Membership Report Size | 20 |

| | | |
|---------|--|----|
| 5. | Description of the Protocol for Group Members | 21 |
| 5.1. | Action on Change of Interface State | 22 |
| 5.2. | Action on Reception of a Query | 24 |
| 6. | Description of the Protocol for Multicast Routers | 26 |
| 6.1. | Conditions for IGMP Queries | 27 |
| 6.2. | IGMP State Maintained by Multicast Routers | 28 |
| 6.2.1. | Definition of Router Filter-Mode | 28 |
| 6.2.2. | Definition of Group Timers | 29 |
| 6.2.3. | Definition of Source Timers | 30 |
| 6.3. | IGMPv3 Source-Specific Forwarding Rules | 31 |
| 6.4. | Action on Reception of Reports | 32 |
| 6.4.1. | Reception of Current-State Records | 32 |
| 6.4.2. | Reception of Filter-Mode-Change and Source-List- Change Records | 33 |
| 6.5. | Switching Router Filter-Modes | 35 |
| 6.6. | Action on Reception of Queries | 36 |
| 6.6.1. | Timer Updates | 36 |
| 6.6.2. | Querier Election | 36 |
| 6.6.3. | Building and Sending Specific Queries | 36 |
| 7. | Interoperation With Older Versions of IGMP | 37 |
| 7.1. | Query Version Distinctions | 38 |
| 7.2. | Group Member Behavior | 38 |
| 7.2.1. | In the Presence of Older Version Queriers | 38 |
| 7.2.2. | In the Presence of Older Version Group Members | 39 |
| 7.3. | Multicast Router Behavior | 40 |
| 7.3.1. | In the Presence of Older Version Queriers | 40 |
| 7.3.2. | In the Presence of Older Version Group Members | 40 |
| 8. | List of Timers, Counters and Their Default Values | 42 |
| 8.1. | Robustness Variable | 42 |
| 8.2. | Query Interval | 42 |
| 8.3. | Query Response Interval | 43 |
| 8.4. | Group Membership Interval | 43 |
| 8.5. | Other Querier Present Interval | 43 |
| 8.6. | Startup Query Interval | 43 |
| 8.7. | Startup Query Count | 43 |
| 8.8. | Last Member Query Interval | 43 |
| 8.9. | Last Member Query Count | 44 |
| 8.10. | Last Member Query Time | 44 |
| 8.11. | Unsolicited Report Interval | 44 |
| 8.12. | Older Version Querier Present Interval | 44 |
| 8.13. | Older Host Present Interval | 44 |
| 8.14. | Configuring Timers | 45 |
| 8.14.1. | Robustness Variable | 45 |
| 8.14.2. | Query Interval | 45 |
| 8.14.3. | Max Response Time | 45 |
| 9. | Security Considerations | 46 |
| 9.1. | Query Message | 46 |
| 9.2. | Current-State Report messages | 47 |

| | |
|--|----|
| 9.3. State-Change Report Messages | 48 |
| 9.4. 9.4. IPSEC Usage | 48 |
| 10. IANA Considerations | 49 |
| 11. Contributors | 49 |
| 12. Acknowledgments | 49 |
| 13. References | 49 |
| 13.1. Normative References | 49 |
| 13.2. Informative References | 50 |
| Appendix A. Design Rationale | 50 |
| A.1. The Need for State-Change Messages | 50 |
| A.2. Host Suppression | 51 |
| A.3. Switching Router Filter Modes from EXCLUDE to INCLUDE . . | 51 |
| Appendix B. Summary of Changes from IGMPv2 | 52 |
| Appendix C. Summary of Changes from RFC 3376 | 53 |
| Author's Address | 53 |

1. Introduction

The Internet Group Management Protocol (IGMP) is used by IPv4 systems (hosts and routers) to report their IP multicast group memberships to any neighboring multicast routers. Note that an IP multicast router may itself be a member of one or more multicast groups, in which case it performs both the multicast router part of the protocol (to collect the membership information needed by its multicast routing protocol) and the group member part of the protocol (to inform itself and other, neighboring multicast routers of its memberships).

IGMP is also used for other IP multicast management functions, using message types other than those used for group membership reporting. This document specifies only the group membership reporting functions and messages.

This document specifies Version 3 of IGMP. Version 1, specified in [RFC1112], was the first widely-deployed version and the first version to become an Internet Standard. Version 2, specified in [RFC2236], added support for low leave latency, that is, a reduction in the time it takes for a multicast router to learn that there are no longer any members of a particular group present on an attached network. Version 3 adds support for source filtering, that is, the ability for a system to report interest in receiving packets only from specific source addresses, as required to support Source-Specific Multicast [RFC3569], or from all but specific source addresses, sent to a particular multicast address. Version 3 is designed to be interoperable with Versions 1 and 2.

This document obsoletes [RFC3376].

The capitalized 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].

2. The Service Interface for Requesting IP Multicast Reception

Within an IP system, there is (at least conceptually) a service interface used by upper-layer protocols or application programs to ask the IP layer to enable and disable reception of packets sent to specific IP multicast addresses. In order to take full advantage of the capabilities of IGMPv3, a system's IP service interface must support the following operation:

```
IPMulticastListen ( socket, interface, multicast-address,  
                   filter-mode, source-list )
```

where:

- o "socket" is an implementation-specific parameter used to distinguish among different requesting entities (e.g., programs or processes) within the system; the socket parameter of BSD Unix system calls is a specific example.
- o "interface" is a local identifier of the network interface on which reception of the specified multicast address is to be enabled or disabled. Interfaces may be physical (e.g., an Ethernet interface) or virtual (e.g., the endpoint of a Frame Relay virtual circuit or the endpoint of an IP-in-IP "tunnel"). An implementation may allow a special "unspecified" value to be passed as the interface parameter, in which case the request would apply to the "primary" or "default" interface of the system (perhaps established by system configuration). If reception of the same multicast address is desired on more than one interface, IPMulticastListen is invoked separately for each desired interface.
- o "multicast-address" is the IP multicast address, or group, to which the request pertains. If reception of more than one multicast address on a given interface is desired, IPMulticastListen is invoked separately for each desired multicast address.
- o "filter-mode" may be either INCLUDE or EXCLUDE. In INCLUDE mode, reception of packets sent to the specified multicast address is requested only from those IP source addresses listed in the source-list parameter. In EXCLUDE mode, reception of packets sent

to the given multicast address is requested from all IP source addresses except those listed in the source-list parameter.

- o "source-list" is an unordered list of zero or more IP unicast addresses from which multicast reception is desired or not desired, depending on the filter mode. An implementation MAY impose a limit on the size of source lists, but that limit MUST NOT be less than 64 addresses per list. When an operation causes the source list size limit to be exceeded, the service interface MUST return an error.

For a given combination of socket, interface, and multicast address, only a single filter mode and source list can be in effect at any one time. However, either the filter mode or the source list, or both, may be changed by subsequent `IPMulticastListen` requests that specify the same socket, interface, and multicast address. Each subsequent request completely replaces any earlier request for the given socket, interface and multicast address.

Previous versions of IGMP did not support source filters and had a simpler service interface consisting of Join and Leave operations to enable and disable reception of a given multicast address (from all sources) on a given interface. The equivalent operations in the new service interface follow:

The Join operation is equivalent to:

```
IPMulticastListen ( socket, interface, multicast-address,
                    EXCLUDE, {} )
```

and the Leave operation is equivalent to:

```
IPMulticastListen ( socket, interface, multicast-address,
                    INCLUDE, {} )
```

where {} is an empty source list.

An example of an API providing the capabilities outlined in this service interface is in [RFC3678].

3. Multicast Reception State Maintained by Systems

3.1. Socket State

For each socket on which `IPMulticastListen` has been invoked, the system records the desired multicast reception state for that socket. That state conceptually consists of a set of records of the form:

(interface, multicast-address, filter-mode, source-list)

The socket state evolves in response to each invocation of `IPMulticastListen` on the socket, as follows:

- o If the requested filter mode is `INCLUDE` and the requested source list is empty, then the entry corresponding to the requested interface and multicast address is deleted if present. If no such entry is present, the request is ignored.
- o If the requested filter mode is `EXCLUDE` or the requested source list is non-empty, then the entry corresponding to the requested interface and multicast address, if present, is changed to contain the requested filter mode and source list. If no such entry is present, a new entry is created, using the parameters specified in the request.

3.2. Interface State

In addition to the per-socket multicast reception state, a system must also maintain or compute multicast reception state for each of its interfaces. That state conceptually consists of a set of records of the form:

(multicast-address, filter-mode, source-list)

At most one record per multicast-address exists for a given interface. This per-interface state is derived from the per-socket state, but may differ from the per-socket state when different sockets have differing filter modes and/or source lists for the same multicast address and interface. For example, suppose one application or process invokes the following operation on socket `s1`:

```
IPMulticastListen ( s1, i, m, INCLUDE, {a, b, c} )
```

requesting reception on interface `i` of packets sent to multicast address `m`, only if they come from source `a`, `b`, or `c`. Suppose another application or process invokes the following operation on socket `s2`:

```
IPMulticastListen ( s2, i, m, INCLUDE, {b, c, d} )
```

requesting reception on the same interface `i` of packets sent to the same multicast address `m`, only if they come from sources `b`, `c`, or `d`. In order to satisfy the reception requirements of both sockets, it is necessary for interface `i` to receive packets sent to `m` from any one of the sources `a`, `b`, `c`, or `d`. Thus, in this example, the reception state of interface `i` for multicast address `m` has filter mode `INCLUDE` and source list `{a, b, c, d}`.

After a multicast packet has been accepted from an interface by the IP layer, its subsequent delivery to the application or process listening on a particular socket depends on the multicast reception state of that socket [and possibly also on other conditions, such as what transport-layer port the socket is bound to]. So, in the above example, if a packet arrives on interface *i*, destined to multicast address *m*, with source address *a*, it will be delivered on socket *s1* but not on socket *s2*. Note that IGMP Queries and Reports are not subject to source filtering and must always be processed by hosts and routers.

Filtering of packets based upon a socket's multicast reception state is a new feature of this service interface. The previous service interface [RFC1112] described no filtering based upon multicast join state; rather, a join on a socket simply caused the host to join a group on the given interface, and packets destined for that group could be delivered to all sockets whether they had joined or not.

The general rules for deriving the per-interface state from the per-socket state are as follows: For each distinct (interface, multicast-address) pair that appears in any socket state, a per-interface record is created for that multicast address on that interface. Considering all socket records containing the same (interface, multicast-address) pair,

- o if any such record has a filter mode of EXCLUDE, then the filter mode of the interface record is EXCLUDE, and the source list of the interface record is the intersection of the source lists of all socket records in EXCLUDE mode, minus those source addresses that appear in any socket record in INCLUDE mode. For example, if the socket records for multicast address *m* on interface *i* are:

from socket *s1*: (*i*, *m*, EXCLUDE, {*a*, *b*, *c*, *d*})

from socket *s2*: (*i*, *m*, EXCLUDE, {*b*, *c*, *d*, *e*})

from socket *s3*: (*i*, *m*, INCLUDE, {*d*, *e*, *f*})

then the corresponding interface record on interface *i* is:

(*m*, EXCLUDE, {*b*, *c*})

If a fourth socket is added, such as:

from socket *s4*: (*i*, *m*, EXCLUDE, {})

then the interface record becomes:

(m, EXCLUDE, {})

- o if all such records have a filter mode of INCLUDE, then the filter mode of the interface record is INCLUDE, and the source list of the interface record is the union of the source lists of all the socket records. For example, if the socket records for multicast address m on interface i are:

from socket s1: (i, m, INCLUDE, {a, b, c})

from socket s2: (i, m, INCLUDE, {b, c, d})

from socket s3: (i, m, INCLUDE, {e, f})

then the corresponding interface record on interface i is:

(m, INCLUDE, {a, b, c, d, e, f})

An implementation MUST NOT use an EXCLUDE interface record to represent a group when all sockets for this group are in INCLUDE state. If system resource limits are reached when an interface state source list is calculated, an error MUST be returned to the application which requested the operation.

The above rules for deriving the interface state are (re-)evaluated whenever an IPMulticastListen invocation modifies the socket state by adding, deleting, or modifying a per-socket state record. Note that a change of socket state does not necessarily result in a change of interface state.

4. Message Formats

IGMP messages are encapsulated in IPv4 datagrams, with an IP protocol number of 2. Every IGMP message described in this document is sent with an IP Time-to-Live of 1, IP Precedence of Internetwork Control (e.g., Type of Service 0xc0), and carries an IP Router Alert option [RFC2113] in its IP header. IGMP message types are registered per [RFC3228].

There are two IGMP message types of concern to the IGMPv3 protocol described in this document:

| Type Number (hex) | Message Name |
|-------------------|-----------------------------|
| 0x11 | Membership Query |
| 0x22 | Version 3 Membership Report |

Table 1: New messages introduced by IGMP3

An implementation of IGMPv3 MUST also support the following three message types, for interoperation with previous versions of IGMP (see Section 7):

| Type Number (hex) | Message Name | Reference |
|-------------------|-----------------------------|-----------|
| 0x12 | Version 1 Membership Report | [RFC1112] |
| 0x16 | Version 2 Membership Report | [RFC2236] |
| 0x17 | Version 2 Leave Group | [RFC2236] |

Table 2: Legacy IGMP messages

Unrecognized message types MUST be silently ignored. Other message types may be used by newer versions or extensions of IGMP, by multicast routing protocols, or for other uses.

In this document, unless otherwise qualified, the capitalized words "Query" and "Report" refer to IGMP Membership Queries and IGMP Version 3 Membership Reports, respectively.

4.1. Membership Query Message

Membership Queries are sent by IP multicast routers to query the multicast reception state of neighboring interfaces. Queries have the following format:

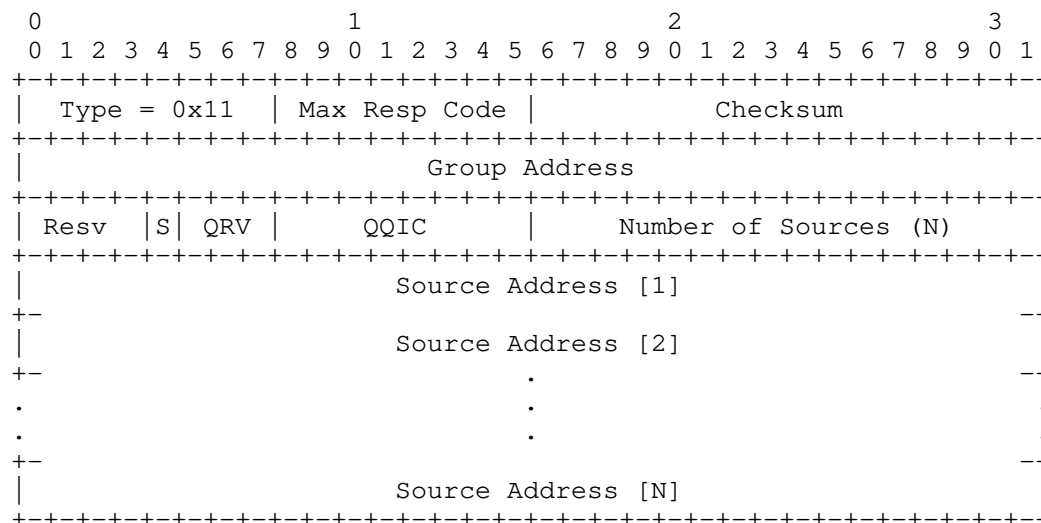


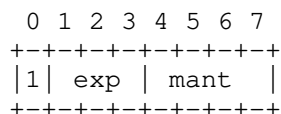
Figure 1: IGMPv3 Query Message

4.1.1. Max Resp Code

The Max Resp Code field specifies the maximum time allowed before sending a responding report. The actual time allowed, called the Max Resp Time, is represented in units of 1/10 second and is derived from the Max Resp Code as follows:

If Max Resp Code < 128, Max Resp Time = Max Resp Code

If Max Resp Code >= 128, Max Resp Code represents a floating-point value as follows:



$$\text{Max Resp Time} = (\text{mant} \mid 0x10) \ll (\text{exp} + 3)$$

Figure 2: Max Resp Code Representation

Small values of Max Resp Time allow IGMPv3 routers to tune the "leave latency" (the time between the moment the last host leaves a group and the moment the routing protocol is notified that there are no more members). Larger values, especially in the exponential range, allow tuning of the burstiness of IGMP traffic on a network.

4.1.2. Checksum

The Checksum is the 16-bit one's complement of the one's complement sum of the whole IGMP message (the entire IP payload). For computing the checksum, the Checksum field is set to zero. When receiving packets, the checksum MUST be verified before processing a packet [RFC1071].

4.1.3. Group Address

The Group Address field is set to zero when sending a General Query, and set to the IP multicast address being queried when sending a Group-Specific Query or Group-and-Source-Specific Query (see Section 4.1.9, below).

4.1.4. Resv (Reserved)

The Resv field is set to zero on transmission, and ignored on reception.

4.1.5. S Flag (Suppress Router-Side Processing)

When set to one, the S Flag indicates to any receiving multicast routers that they are to suppress the normal timer updates they perform upon hearing a Query. It does not, however, suppress the querier election or the normal "host-side" processing of a Query that a router may be required to perform as a consequence of itself being a group member.

4.1.6. QRV (Querier's Robustness Variable)

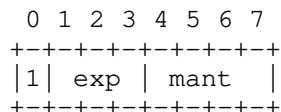
If non-zero, the QRV field contains the [Robustness Variable] value used by the querier, i.e., the sender of the Query. If the querier's [Robustness Variable] exceeds 7, the maximum value of the QRV field, the QRV is set to zero. Routers adopt the QRV value from the most recently received Query as their own [Robustness Variable] value, unless that most recently received QRV was zero, in which case the receivers use the default [Robustness Variable] value specified in section 8.1 or a statically configured value.

4.1.7. QQIC (Querier's Query Interval Code)

The Querier's Query Interval Code field specifies the [Query Interval] used by the querier. The actual interval, called the Querier's Query Interval (QQI), is represented in units of seconds and is derived from the Querier's Query Interval Code as follows:

If $QQIC < 128$, $QQI = QQIC$

If QQIC >= 128, QQIC represents a floating-point value as follows:



$$QQI = (\text{mant} \mid 0x10) \ll (\text{exp} + 3)$$

Figure 3: QQIC Representation

Multicast routers that are not the current querier adopt the QQI value from the most recently received Query as their own [Query Interval] value, unless that most recently received QQI was zero, in which case the receiving routers use the default [Query Interval] value specified in Section 8.2.

4.1.8. Number of Sources (N)

The Number of Sources (N) field specifies how many source addresses are present in the Query. This number is zero in a General Query or a Group-Specific Query, and non-zero in a Group-and-Source-Specific Query. This number is limited by the MTU of the network over which the Query is transmitted. For example, on an Ethernet with an MTU of 1500 octets, the IP header including the Router Alert option consumes 24 octets, and the IGMP fields up to including the Number of Sources (N) field consume 12 octets, leaving 1464 octets for source addresses, which limits the number of source addresses to 366 (1464/4).

4.1.9. Source Address [i]

The Source Address [i] fields are a vector of n IP unicast addresses, where n is the value in the Number of Sources (N) field.

4.1.10. Additional Data

If the Packet Length field in the IP header of a received Query indicates that there are additional octets of data present, beyond the fields described here, IGMPv3 implementations MUST include those octets in the computation to verify the received IGMP Checksum, but MUST otherwise ignore those additional octets. When sending a Query, an IGMPv3 implementation MUST NOT include additional octets beyond the fields described here.

4.1.11. Query Variants

There are three variants of the Query message:

1. A General Query is sent by a multicast router to learn the complete multicast reception state of the neighboring interfaces (that is, the interfaces attached to the network on which the Query is transmitted). In a General Query, both the Group Address field and the Number of Sources (N) field are zero.
2. A Group-Specific Query is sent by a multicast router to learn the reception state, with respect to a single multicast address, of the neighboring interfaces. In a Group-Specific Query, the Group Address field contains the multicast address of interest, and the Number of Sources (N) field contains zero.
3. A Group-and-Source-Specific Query is sent by a multicast router to learn if any neighboring interface desires reception of packets sent to a specified multicast address, from any of a specified list of sources. In a Group-and-Source-Specific Query, the Group Address field contains the multicast address of interest, and the Source Address [i] fields contain the source address(es) of interest.

4.1.12. IP Destination Addresses for Queries

In IGMPv3, General Queries are sent with an IP destination address of 224.0.0.1, the all-systems multicast address. Group-Specific and Group-and-Source-Specific Queries are sent with an IP destination address equal to the multicast address of interest. However, a system MUST accept and process any Query whose IP Destination Address field contains any of the addresses (unicast or multicast) assigned to the interface on which the Query arrives.

4.2. Version 3 Membership Report Message

Version 3 Membership Reports are sent by IP systems to report (to neighboring routers) the current multicast reception state, or changes in the multicast reception state, of their interfaces. Reports have the following format:

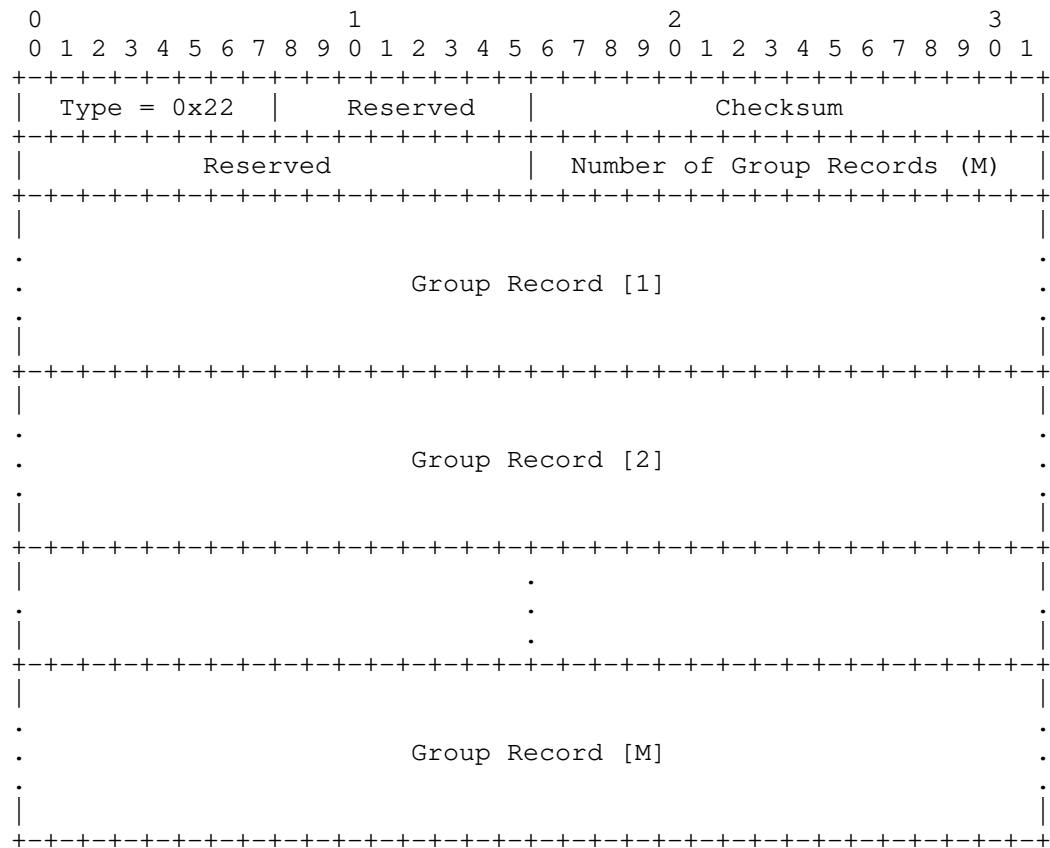


Figure 4: IGMPv3 Report Message

where each Group Record has the following internal format:

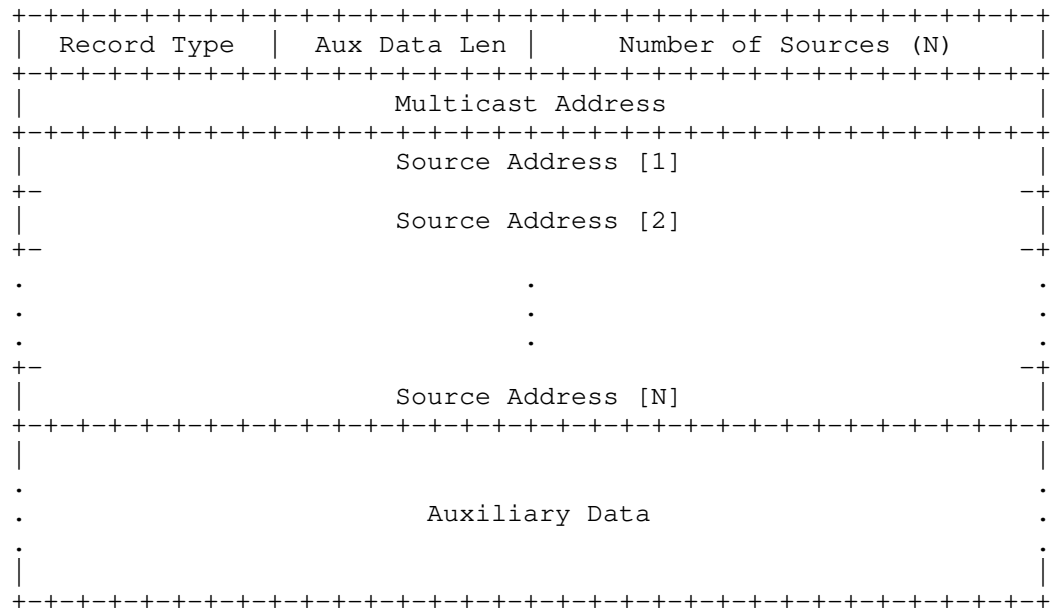


Figure 5: IGMPv3 Report Group Record

4.2.1. Reserved

The Reserved fields are set to zero on transmission, and ignored on reception.

4.2.2. Checksum

The Checksum is the 16-bit one's complement of the one's complement sum of the whole IGMP message (the entire IP payload). For computing the checksum, the Checksum field is set to zero. When receiving packets, the checksum MUST be verified before processing a message.

4.2.3. Number of Group Records (M)

The Number of Group Records (M) field specifies how many Group Records are present in this Report.

4.2.4. Group Record

Each Group Record is a block of fields containing information pertaining to the sender's membership in a single multicast group on the interface from which the Report is sent.

4.2.5. Record Type

See section Section 4.2.12, below.

4.2.6. Aux Data Len

The Aux Data Len field contains the length of the Auxiliary Data field in this Group Record, in units of 32-bit words. It may contain zero, to indicate the absence of any auxiliary data.

4.2.7. Number of Sources (N)

The Number of Sources (N) field specifies how many source addresses are present in this Group Record.

4.2.8. Multicast Address

The Multicast Address field contains the IP multicast address to which this Group Record pertains.

4.2.9. Source Address [i]

The Source Address [i] fields are a vector of n IP unicast addresses, where n is the value in this record's Number of Sources (N) field.

4.2.10. Auxiliary Data

The Auxiliary Data field, if present, contains additional information pertaining to this Group Record. The protocol specified in this document, IGMPv3, does not define any auxiliary data. Therefore, implementations of IGMPv3 MUST NOT include any auxiliary data (i.e., MUST set the Aux Data Len field to zero) in any transmitted Group Record, and MUST ignore any auxiliary data present in any received Group Record. The semantics and internal encoding of the Auxiliary Data field are to be defined by any future version or extension of IGMP that uses this field.

4.2.11. Additional Data

If the Packet Length field in the IP header of a received Report indicates that there are additional octets of data present, beyond the last Group Record, IGMPv3 implementations MUST include those octets in the computation to verify the received IGMP Checksum, but MUST otherwise ignore those additional octets. When sending a Report, an IGMPv3 implementation MUST NOT include additional octets beyond the last Group Record.

4.2.12. Group Record Types

There are a number of different types of Group Records that may be included in a Report message:

- o A Current-State Record is sent by a system in response to a Query received on an interface. It reports the current reception state of that interface, with respect to a single multicast address. The Record Type of a Current-State Record may be one of the following two values:
 - 1 - `MODE_IS_INCLUDE` - indicates that the interface has a filter mode of `INCLUDE` for the specified multicast address. The Source Address [i] fields in this Group Record contain the interface's source list for the specified multicast address, if it is non-empty.
 - 2 - `MODE_IS_EXCLUDE` - indicates that the interface has a filter mode of `EXCLUDE` for the specified multicast address. The Source Address [i] fields in this Group Record contain the interface's source list for the specified multicast address, if it is non-empty.
- o A Filter-Mode-Change Record is sent by a system whenever a local invocation of `IPMulticastListen` causes a change of the filter mode (i.e., a change from `INCLUDE` to `EXCLUDE`, or from `EXCLUDE` to `INCLUDE`), of the interface-level state entry for a particular multicast address. The Record is included in a Report sent from the interface on which the change occurred. The Record Type of a Filter-Mode-Change Record may be one of the following two values:
 - 3 - `CHANGE_TO_INCLUDE_MODE` - indicates that the interface has changed to `INCLUDE` filter mode for the specified multicast address. The Source Address [i] fields in this Group Record contain the interface's new source list for the specified multicast address, if it is non-empty.
 - 4 - `CHANGE_TO_EXCLUDE_MODE` - indicates that the interface has changed to `EXCLUDE` filter mode for the specified multicast address. The Source Address [i] fields in this Group Record contain the interface's new source list for the specified multicast address, if it is non-empty.
- o A Source-List-Change Record is sent by a system whenever a local invocation of `IPMulticastListen` causes a change of source list that is not coincident with a change of filter mode, of the interface-level state entry for a particular multicast address. The Record is included in a Report sent from the interface on

which the change occurred. The Record Type of a Source-List-Change Record may be one of the following two values:

- 5 - ALLOW_NEW_SOURCES - indicates that the Source Address [i] fields in this Group Record contain a list of the additional sources that the system wishes to hear from, for packets sent to the specified multicast address. If the change was to an INCLUDE source list, these are the addresses that were added to the list; if the change was to an EXCLUDE source list, these are the addresses that were deleted from the list.
- 6 - BLOCK_OLD_SOURCES - indicates that the Source Address [i] fields in this Group Record contain a list of the sources that the system no longer wishes to hear from, for packets sent to the specified multicast address. If the change was to an INCLUDE source list, these are the addresses that were deleted from the list; if the change was to an EXCLUDE source list, these are the addresses that were added to the list.

If a change of source list results in both allowing new sources and blocking old sources, then two Group Records are sent for the same multicast address, one of type ALLOW_NEW_SOURCES and one of type BLOCK_OLD_SOURCES.

We use the term State-Change Record to refer to either a Filter-Mode-Change Record or a Source-List-Change Record.

Unrecognized Record Type values MUST be silently ignored.

4.2.13. IP Source Addresses for Reports

An IGMP report is sent with a valid IP source address for the destination subnet. The 0.0.0.0 source address may be used by a system that has not yet acquired an IP address. Note that the 0.0.0.0 source address may simultaneously be used by multiple systems on a LAN. Routers MUST accept a report with a source address of 0.0.0.0.

4.2.14. IP Destination Addresses for Reports

Version 3 Reports are sent with an IP destination address of 224.0.0.22, to which all IGMPv3-capable multicast routers listen. A system that is operating in version 1 or version 2 compatibility modes sends version 1 or version 2 Reports to the multicast group specified in the Group Address field of the Report. In addition, a system MUST accept and process any version 1 or version 2 Report

whose IP Destination Address field contains any of the addresses (unicast or multicast) assigned to the interface on which the Report arrives.

4.2.15. Notation for Group Records

In the rest of this document, we use the following notation to describe the contents of a Group Record pertaining to a particular multicast address:

| | | |
|-------------|---|---|
| IS_IN (x) | - | Type MODE_IS_INCLUDE, source addresses x |
| IS_EX (x) | - | Type MODE_IS_EXCLUDE, source addresses x |
| TO_IN (x) | - | Type CHANGE_TO_INCLUDE_MODE, source addresses x |
| TO_EX (x) | - | Type CHANGE_TO_EXCLUDE_MODE, source addresses x |
| ALLOW (x) | - | Type ALLOW_NEW_SOURCES, source addresses x |
| BLOCK (x) | - | Type BLOCK_OLD_SOURCES, source addresses x |

where x is either:

- o a capital letter (e.g., "A") to represent the set of source addresses, or
- o a set expression (e.g., "A+B"), where "A+B" means the union of sets A and B, "A*B" means the intersection of sets A and B, and "A-B" means the removal of all elements of set B from set A.

4.2.16. Membership Report Size

If the set of Group Records required in a Report does not fit within the size limit of a single Report message (as determined by the MTU of the network on which it will be sent), the Group Records are sent in as many Report messages as needed to report the entire set.

If a single Group Record contains so many source addresses that it does not fit within the size limit of a single Report message, if its Type is not MODE_IS_EXCLUDE or CHANGE_TO_EXCLUDE_MODE, it is split into multiple Group Records, each containing a different subset of the source addresses and each sent in a separate Report message. If its Type is MODE_IS_EXCLUDE or CHANGE_TO_EXCLUDE_MODE, a single Group Record is sent, containing as many source addresses as can fit, and

the remaining source addresses are not reported; though the choice of which sources to report is arbitrary, it is preferable to report the same set of sources in each subsequent report, rather than reporting different sources each time.

5. Description of the Protocol for Group Members

IGMP is an asymmetric protocol, specifying separate behaviors for group members -- that is, hosts or routers that wish to receive multicast packets -- and multicast routers. This section describes the part of IGMPv3 that applies to all group members. (Note that a multicast router that is also a group member performs both parts of IGMPv3, receiving and responding to its own IGMP message transmissions as well as those of its neighbors. The multicast router part of IGMPv3 is described in Section 6.)

A system performs the protocol described in this section over all interfaces on which multicast reception is supported, even if more than one of those interfaces is connected to the same network.

For interoperability with multicast routers running older versions of IGMP, systems maintain a MulticastRouterVersion variable for each interface on which multicast reception is supported. This section describes the behavior of group member systems on interfaces for which MulticastRouterVersion = 3. The algorithm for determining MulticastRouterVersion, and the behavior for versions other than 3, are described in Section 7.

The all-systems multicast address, 224.0.0.1, is handled as a special case. On all systems -- that is all hosts and routers, including multicast routers -- reception of packets destined to the all-systems multicast address, from all sources, is permanently enabled on all interfaces on which multicast reception is supported. No IGMP messages are ever sent regarding the all-systems multicast address.

There are two types of events that trigger IGMPv3 protocol actions on an interface:

- o a change of the interface reception state, caused by a local invocation of IPMulticastListen.
- o reception of a Query.

(Received IGMP messages of types other than Query are silently ignored, except as required for interoperation with earlier versions of IGMP.)

The following subsections describe the actions to be taken for each of these two cases. In those descriptions, timer and counter names appear in square brackets. The default values for those timers and counters are specified in Section 8.

5.1. Action on Change of Interface State

An invocation of `IPMulticastListen` may cause the multicast reception state of an interface to change, according to the rules in Section 3.2. Each such change affects the per-interface entry for a single multicast address.

A change of interface state causes the system to immediately transmit a State-Change Report from that interface. The type and contents of the Group Record(s) in that Report are determined by comparing the filter mode and source list for the affected multicast address before and after the change, according to the table below. If no interface state existed for that multicast address before the change (i.e., the change consisted of creating a new per-interface record), or if no state exists after the change (i.e., the change consisted of deleting a per-interface record), then the "non-existent" state is considered to have a filter mode of `INCLUDE` and an empty source list.

| Old State | New State | State-Change Record Sent |
|--------------------------|--------------------------|---------------------------------------|
| <code>INCLUDE (A)</code> | <code>INCLUDE (B)</code> | <code>ALLOW (B-A), BLOCK (A-B)</code> |
| <code>EXCLUDE (A)</code> | <code>EXCLUDE (B)</code> | <code>ALLOW (A-B), BLOCK (B-A)</code> |
| <code>INCLUDE (A)</code> | <code>EXCLUDE (B)</code> | <code>TO_EX (B)</code> |
| <code>EXCLUDE (A)</code> | <code>INCLUDE (B)</code> | <code>TO_IN (B)</code> |

If the computed source list for either an `ALLOW` or a `BLOCK` State-Change Record is empty, that record is omitted from the Report message.

To cover the possibility of the State-Change Report being missed by one or more multicast routers, it is retransmitted [Robustness Variable] - 1 more times, at intervals chosen at random from the range (0, [Unsolicited Report Interval]).

If more changes to the same interface state entry occur before all the retransmissions of the State-Change Report for the first change have been completed, each such additional change triggers the immediate transmission of a new State-Change Report.

The contents of the new transmitted report are calculated as follows. As was done with the first report, the interface state for the affected group before and after the latest change is compared. The report records expressing the difference are built according to the table above. However these records are not transmitted in a message but instead merged with the contents of the pending report, to create the new State-Change report. The rules for merging the difference

report resulting from the state change and the pending report are described below.

The transmission of the merged State-Change Report terminates retransmissions of the earlier State-Change Reports for the same multicast address, and becomes the first of [Robustness Variable] transmissions of State-Change Reports.

Each time a source is included in the difference report calculated above, retransmission state for that source needs to be maintained until [Robustness Variable] State-Change reports have been sent by the host. This is done in order to ensure that a series of successive state changes do not break the protocol robustness.

If the interface reception-state change that triggers the new report is a filter-mode change, then the next [Robustness Variable] State-Change Reports will include a Filter-Mode-Change record. This applies even if any number of source-list changes occur in that period. The host has to maintain retransmission state for the group until the [Robustness Variable] State-Change reports have been sent. When [Robustness Variable] State-Change reports with Filter-Mode-Change records have been transmitted after the last filter-mode change, and if source-list changes to the interface reception have scheduled additional reports, then the next State-Change report will include Source-List-Change records.

Each time a State-Change Report is transmitted, the contents are determined as follows. If the report should contain a Filter-Mode-Change record, then if the current filter-mode of the interface is INCLUDE, a TO_IN record is included in the report, otherwise a TO_EX record is included. If instead the report should contain Source-List-Change records, an ALLOW and a BLOCK record are included. The contents of these records are built according to the table below.

| Record | Sources Included |
|--------|---|
| TO_IN | All in the current interface state that must be forwarded |
| TO_EX | All in the current interface state that must be blocked |
| ALLOW | All with retransmission state that must be forwarded |
| BLOCK | All with retransmission state that must be blocked |

If the computed source list for either an ALLOW or a BLOCK record is empty, that record is omitted from the State-Change report.

Note: When the first State-Change report is sent, the non-existent pending report to merge with, can be treated as a source-change report with empty ALLOW and BLOCK records (no sources have retransmission state).

5.2. Action on Reception of a Query

When a system receives a Query, it does not respond immediately. Instead, it delays its response by a random amount of time, bounded by the Max Resp Time value derived from the Max Resp Code in the received Query message. A system may receive a variety of Queries on different interfaces and of different kinds (e.g., General Queries, Group-Specific Queries, and Group-and-Source-Specific Queries), each of which may require its own delayed response.

Before scheduling a response to a Query, the system must first consider previously scheduled pending responses and in many cases schedule a combined response. Therefore, the system must be able to maintain the following state:

- o A timer per interface for scheduling responses to General Queries.
- o A per-group and interface timer for scheduling responses to Group-Specific and Group-and-Source-Specific Queries.
- o A per-group and interface list of sources to be reported in the response to a Group-and-Source-Specific Query.

When a new Query with the Router-Alert option arrives on an interface, provided the system has state to report, a delay for a response is randomly selected in the range (0, [Max Resp Time]) where Max Resp Time is derived from Max Resp Code in the received Query message. The following rules are then used to determine if a Report needs to be scheduled and the type of Report to schedule. The rules are considered in order and only the first matching rule is applied.

1. If there is a pending response to a previous General Query scheduled sooner than the selected delay, no additional response needs to be scheduled.
2. If the received Query is a General Query, the interface timer is used to schedule a response to the General Query after the selected delay. Any previously pending response to a General Query is canceled.
3. If the received Query is a Group-Specific Query or a Group-and-Source-Specific Query and there is no pending response to a previous Query for this group, then the group timer is used to

schedule a report. If the received Query is a Group-and-Source-Specific Query, the list of queried sources is recorded to be used when generating a response.

4. If there already is a pending response to a previous Query scheduled for this group, and either the new Query is a Group-Specific Query or the recorded source-list associated with the group is empty, then the group source-list is cleared and a single response is scheduled using the group timer. The new response is scheduled to be sent at the earliest of the remaining time for the pending report and the selected delay.
5. If the received Query is a Group-and-Source-Specific Query and there is a pending response for this group with a non-empty source-list, then the group source list is augmented to contain the list of sources in the new Query and a single response is scheduled using the group timer. The new response is scheduled to be sent at the earliest of the remaining time for the pending report and the selected delay.

When the timer in a pending response record expires, the system transmits, on the associated interface, one or more Report messages carrying one or more Current-State Records (see section Section 4.2.12), as follows:

1. If the expired timer is the interface timer (i.e., it is a pending response to a General Query), then one Current-State Record is sent for each multicast address for which the specified interface has reception state, as described in Section 3.2. The Current-State Record carries the multicast address and its associated filter mode (MODE_IS_INCLUDE or MODE_IS_EXCLUDE) and source list. Multiple Current-State Records are packed into individual Report messages, to the extent possible.

This naive algorithm may result in bursts of packets when a system is a member of a large number of groups. Instead of using a single interface timer, implementations are recommended to spread transmission of such Report messages over the interval (0, [Max Resp Time]). Note that any such implementation MUST avoid the "ack-implosion" problem, i.e., MUST NOT send a Report immediately on reception of a General Query.

2. If the expired timer is a group timer and the list of recorded sources for the that group is empty (i.e., it is a pending response to a Group-Specific Query), then if and only if the interface has reception state for that group address, a single Current-State Record is sent for that address. The Current-State

Record carries the multicast address and its associated filter mode (MODE_IS_INCLUDE or MODE_IS_EXCLUDE) and source list.

3. If the expired timer is a group timer and the list of recorded sources for that group is non-empty (i.e., it is a pending response to a Group-and-Source-Specific Query), then if and only if the interface has reception state for that group address, the contents of the responding Current-State Record is determined from the interface state and the pending response record, as specified in the following table:

| interface state | set of sources in the pending response record | Current-State Record |
|-----------------|---|----------------------|
| INCLUDE (A) | B | IS_IN (A*B) |
| EXCLUDE (A) | B | IS_IN (B-A) |

If the resulting Current-State Record has an empty set of source addresses, then no response is sent.

Finally, after any required Report messages have been generated, the source lists associated with any reported groups are cleared.

6. Description of the Protocol for Multicast Routers

The purpose of IGMP is to enable each multicast router to learn, for each of its directly attached networks, which multicast addresses are of interest to the systems attached to those networks. IGMP version 3 adds the capability for a multicast router to also learn which sources are of interest to neighboring systems, for packets sent to any particular multicast address. The information gathered by IGMP is provided to whichever multicast routing protocol is being used by the router, in order to ensure that multicast packets are delivered to all networks where there are interested receivers.

This section describes the part of IGMPv3 that is performed by multicast routers. Multicast routers may also themselves become members of multicast groups, and therefore also perform the group member part of IGMPv3, described in Section 5.

A multicast router performs the protocol described in this section over each of its directly-attached networks. If a multicast router has more than one interface to the same network, it only needs to operate this protocol over one of those interfaces. On each interface over which this protocol is being run, the router **MUST** enable reception of multicast address 224.0.0.22, from all sources

(and MUST perform the group member part of IGMPv3 for that address on that interface).

Multicast routers need to know only that at least one system on an attached network is interested in packets to a particular multicast address from a particular source; a multicast router is not required to keep track of the interests of each individual neighboring system. (However, see Appendix A.2 point 1 for discussion.)

IGMPv3 is backward compatible with previous versions of the IGMP protocol. In order to remain backward compatible with older IGMP systems, IGMPv3 multicast routers MUST also implement versions 1 and 2 of the protocol (see section Section 7).

6.1. Conditions for IGMP Queries

Multicast routers send General Queries periodically to request group membership information from an attached network. These queries are used to build and refresh the group membership state of systems on attached networks. Systems respond to these queries by reporting their group membership state (and their desired set of sources) with Current-State Group Records in IGMPv3 Membership Reports.

As a member of a multicast group, a system may express interest in receiving or not receiving traffic from particular sources. As the desired reception state of a system changes, it reports these changes using Filter-Mode-Change Records or Source-List-Change Records. These records indicate an explicit state change in a group at a system in either the group record's source list or its filter-mode. When a group membership is terminated at a system or traffic from a particular source is no longer desired, a multicast router must query for other members of the group or listeners of the source before deleting the group (or source) and pruning its traffic.

To enable all systems on a network to respond to changes in group membership, multicast routers send specific queries. A Group-Specific Query is sent to verify there are no systems that desire reception of the specified group or to "rebuild" the desired reception state for a particular group. Group-Specific Queries are sent when a router receives a State-Change record indicating a system is leaving a group.

A Group-and-Source Specific Query is used to verify there are no systems on a network which desire to receive traffic from a set of sources. Group-and-Source Specific Queries list sources for a particular group which have been requested to no longer be forwarded. This query is sent by a multicast router to learn if any systems desire reception of packets to the specified group address from the

specified source addresses. Group-and-Source Specific Queries are only sent in response to State-Change Records and never in response to Current-State Records. Section 4.1.11 describes each query in more detail.

6.2. IGMP State Maintained by Multicast Routers

Multicast routers implementing IGMPv3 keep state per group per attached network. This group state consists of a filter-mode, a list of sources, and various timers. For each attached network running IGMP, a multicast router records the desired reception state for that network. That state conceptually consists of a set of records of the form:

(multicast address, group timer, filter-mode, (source records))

Each source record is of the form:

(source address, source timer)

If all sources within a given group are desired, an empty source record list is kept with filter-mode set to EXCLUDE. This means hosts on this network want all sources for this group to be forwarded. This is the IGMPv3 equivalent to a IGMPv1 or IGMPv2 group join.

6.2.1. Definition of Router Filter-Mode

To reduce internal state, IGMPv3 routers keep a filter-mode per group per attached network. This filter-mode is used to condense the total desired reception state of a group to a minimum set such that all systems' memberships are satisfied. This filter-mode may change in response to the reception of particular types of group records or when certain timer conditions occur. In the following sections, we use the term "router filter-mode" to refer to the filter-mode of a particular group within a router. Section 6.4 describes the changes of a router filter-mode per group record received.

Conceptually, when a group record is received, the router filter-mode for that group is updated to cover all the requested sources using the least amount of state. As a rule, once a group record with a filter-mode of EXCLUDE is received, the router filter-mode for that group will be EXCLUDE.

When a router filter-mode for a group is EXCLUDE, the source record list contains two types of sources. The first type is the set which represents conflicts in the desired reception state; this set must be forwarded by some router on the network. The second type is the set

of sources which hosts have requested to not be forwarded. Appendix A describes the reasons for keeping this second set when in EXCLUDE mode.

When a router filter-mode for a group is INCLUDE, the source record list is the list of sources desired for the group. This is the total desired set of sources for that group. Each source in the source record list must be forwarded by some router on the network.

Because a reported group record with a filter-mode of EXCLUDE will cause a router to transition its filter-mode for that group to EXCLUDE, a mechanism for transitioning a router's filter-mode back to INCLUDE must exist. If all systems with a group record in EXCLUDE filter-mode cease reporting, it is desirable for the router filter-mode for that group to transition back to INCLUDE mode. This transition occurs when the group timer expires and is explained in detail in Section 6.5.

6.2.2. Definition of Group Timers

The group timer is only used when a group is in EXCLUDE mode and it represents the time for the filter-mode of the group to expire and switch to INCLUDE mode. We define a group timer as a decrementing timer with a lower bound of zero kept per group per attached network. Group timers are updated according to the types of group records received.

A group timer expiring when a router filter-mode for the group is EXCLUDE means there are no listeners on the attached network in EXCLUDE mode. At this point, a router will transition to INCLUDE filter-mode. Section 6.5 describes the actions taken when a group timer expires while in EXCLUDE mode.

The following table summarizes the role of the group timer. Section 6.4 describes the details of setting the group timer per type of group record received.

| Group Filter-Mode | Group Timer Value | Actions/Comments |
|-------------------|-------------------|---|
| INCLUDE | Timer ≥ 0 | All members in INCLUDE mode. |
| EXCLUDE | Timer > 0 | At least one member in EXCLUDE mode. |
| EXCLUDE | Timer $= 0$ | No more listeners to group. If all source timers have expired then delete Group Record. If there are still source record timers running, switch to INCLUDE filter-mode using those source records with running timers as the INCLUDE source record state. |

6.2.3. Definition of Source Timers

A source timer is kept per source record and is a decrementing timer with a lower bound of zero. Source timers are updated according to the type and filter-mode of the group record received. Source timers are always updated (for a particular group) whenever the source is present in a received record for that group. Section 6.4 describes the setting of source timers per type of group records received.

A source record with a running timer with a router filter-mode for the group of INCLUDE means that there is currently one or more systems (in INCLUDE filter-mode) which desire to receive that source. If a source timer expires with a router filter-mode for the group of INCLUDE, the router concludes that traffic from this particular source is no longer desired on the attached network, and deletes the associated source record.

Source timers are treated differently when a router filter-mode for a group is EXCLUDE. If a source record has a running timer with a router filter-mode for the group of EXCLUDE, it means that at least one system desires the source. It should therefore be forwarded by a router on the network. Appendix A describes the reasons for keeping state for sources that have been requested to be forwarded while in EXCLUDE state.

If a source timer expires with a router filter-mode for the group of EXCLUDE, the router informs the routing protocol that there is no longer a receiver on the network interested in traffic from this source.

When a router filter-mode for a group is EXCLUDE, source records are only deleted when the group timer expires. Section 6.3 describes the actions that should be taken dependent upon the value of a source timer.

6.3. IGMPv3 Source-Specific Forwarding Rules

When a multicast router receives a datagram from a source destined to a particular group, a decision has to be made whether to forward the datagram onto an attached network or not. The multicast routing protocol in use is in charge of this decision, and should use the IGMPv3 information to ensure that all sources/groups desired on a subnetwork are forwarded to that subnetwork. IGMPv3 information does not override multicast routing information; for example, if the IGMPv3 filter-mode group for G is EXCLUDE, a router may still forward packets for excluded sources to a transit subnet.

To summarize, the following table describes the forwarding suggestions made by IGMP to the routing protocol for traffic originating from a source destined to a group. It also summarizes the actions taken upon the expiration of a source timer based on the router filter-mode of the group.

| Group Filter-Mode | Group Timer Value | Action |
|-------------------|--------------------|--|
| INCLUDE | TIMER > 0 | Suggest to forward traffic from source |
| INCLUDE | TIMER == 0 | Suggest to stop forwarding traffic from source and remove source record. If there are no more source records for the group, delete group record. |
| INCLUDE | No Source Elements | Suggest to not forward source |
| EXCLUDE | TIMER > 0 | Suggest to forward traffic from source |
| EXCLUDE | TIMER == 0 | Suggest to not forward traffic from source (DO NOT remove record) |
| EXCLUDE | No Source Elements | Suggest to forward traffic from source |

6.4. Action on Reception of Reports

6.4.1. Reception of Current-State Records

When receiving Current-State Records, a router updates both its group and source timers. In some circumstances, the reception of a type of group record will cause the router filter-mode for that group to change. The table below describes the actions, with respect to state and timers that occur to a router's state upon reception of Current-State Records.

The following notation is used to describe the updating of source timers. The notation (A, B) will be used to represent the total number of sources for a particular group, where

A = set of source records whose source timers > 0 (Sources that at least one host has requested to be forwarded)

B = set of source records whose source timers = 0 (Sources that IGMP will suggest to the routing protocol not to forward)

Note that there will only be two sets when a router's filter-mode for a group is EXCLUDE. When a router's filter-mode for a group is INCLUDE, a single set is used to describe the set of sources requested to be forwarded (e.g., simply (A)).

In the following tables, abbreviations are used for several variables (all of which are described in detail in Section 8). The variable GMI is an abbreviation for the Group Membership Interval, which is the time in which group memberships will time out. The variable LMQT is an abbreviation for the Last Member Query Time, which is the total time spent after Last Member Query Count retransmissions. LMQT represents the "leave latency", or the difference between the transmission of a membership change and the change in the information given to the routing protocol.

Within the "Actions" section of the router state tables, we use the notation 'A=J', which means that the set A of source records should have their source timers set to value J. 'Delete A' means that the set A of source records should be deleted. 'Group Timer=J' means that the Group Timer for the group should be set to value J.

| Router State ----- | Report Rec'd ----- | New Router State ----- | Actions ----- |
|-----------------------|-----------------------|---------------------------|--|
| INCLUDE (A) | IS_IN (B) | INCLUDE (A+B) | (B)=GMI |
| INCLUDE (A) | IS_EX (B) | EXCLUDE (A*B,B-A) | (B-A)=0 Delete (A-B) Group Timer=GMI |
| EXCLUDE (X,Y) | IS_IN (A) | EXCLUDE (X+A,Y-A) | (A)=GMI |
| EXCLUDE (X,Y) | IS_EX (A) | EXCLUDE (A-Y,Y*A) | (A-X-Y)=GMI Delete (X-A) Delete (Y-A) Group Timer=GMI |

6.4.2. Reception of Filter-Mode-Change and Source-List-Change Records

When a change in the global state of a group occurs in a system, the system sends either a Source-List-Change Record or a Filter-Mode-Change Record for that group. As with Current-State Records, routers must act upon these records and possibly change their own state to reflect the new desired membership state of the network.

Routers must query sources that are requested to be no longer forwarded to a group. When a router queries or receives a query for a specific set of sources, it lowers its source timers for those sources to a small interval of Last Member Query Time seconds. If group records are received in response to the queries which express interest in receiving traffic from the queried sources, the corresponding timers are updated.

Similarly, when a router queries a specific group, it lowers its group timer for that group to a small interval of Last Member Query Time seconds. If any group records expressing EXCLUDE mode interest in the group are received within the interval, the group timer for the group is updated and the suggestion to the routing protocol to forward the group stands without any interruption.

During a query period (i.e., Last Member Query Time seconds), the IGMP component in the router continues to suggest to the routing protocol that it forwards traffic from the groups or sources that it is querying. It is not until after Last Member Query Time seconds without receiving a record expressing interest in the queried group or sources that the router may prune the group or sources from the network.

The following table describes the changes in group state and the action(s) taken when receiving either Filter-Mode-Change or Source-List-Change Records. This table also describes the queries which are sent by the querier when a particular report is received.

We use the following notation for describing the queries which are sent. We use the notation 'Q(G)' to describe a Group-Specific Query to G. We use the notation 'Q(G,A)' to describe a Group-and-Source Specific Query to G with source-list A. If source-list A is null as a result of the action (e.g., A*B) then no query is sent as a result of the operation.

In order to maintain protocol robustness, queries sent by actions in the table below need to be transmitted [Last Member Query Count] times, once every [Last Member Query Interval].

If while scheduling new queries, there are already pending queries to be retransmitted for the same group, the new and pending queries have to be merged. In addition, received host reports for a group with pending queries may affect the contents of those queries. Section 6.6.3 describes the process of building and maintaining the state of pending queries.

| Router State ----- | Report Rec'd | New Router State ----- | Actions ----- |
|-----------------------|--------------|---------------------------|---|
| INCLUDE (A) | ALLOW (B) | INCLUDE (A+B) | (B)=GMI |
| INCLUDE (A) | BLOCK (B) | INCLUDE (A) | Send Q(G,A*B) |
| INCLUDE (A) | TO_EX (B) | EXCLUDE (A*B,B-A) | (B-A)=0 Delete (A-B) Send Q(G,A*B) Group Timer=GMI |
| INCLUDE (A) | TO_IN (B) | INCLUDE (A+B) | (B)=GMI Send Q(G,A-B) |
| EXCLUDE (X,Y) | ALLOW (A) | EXCLUDE (X+A,Y-A) | (A)=GMI |
| EXCLUDE (X,Y) | BLOCK (A) | EXCLUDE (X+(A-Y),Y) | (A-X-Y)=Group Timer Send Q(G,A-Y) |
| EXCLUDE (X,Y) | TO_EX (A) | EXCLUDE (A-Y,Y*A) | (A-X-Y)=Group Timer Delete (X-A) Delete (Y-A) Send Q(G,A-Y) Group Timer=GMI |
| EXCLUDE (X,Y) | TO_IN (A) | EXCLUDE (X+A,Y-A) | (A)=GMI Send Q(G,X-A) Send Q(G) |

6.5. Switching Router Filter-Modes

The group timer is used as a mechanism for transitioning the router filter-mode from EXCLUDE to INCLUDE.

When a group timer expires with a router filter-mode of EXCLUDE, a router assumes that there are no systems with a filter-mode of EXCLUDE present on the attached network. When a router's filter-mode for a group is EXCLUDE and the group timer expires, the router filter-mode for the group transitions to INCLUDE.

A router uses source records with running source timers as its state for the switch to a filter-mode of INCLUDE. If there are any source records with source timers greater than zero (i.e., requested to be forwarded), a router switches to filter-mode of INCLUDE using those source records. Source records whose timers are zero (from the previous EXCLUDE mode) are deleted.

For example, if a router's state for a group is EXCLUDE(X,Y) and the group timer expires for that group, the router switches to filter-mode of INCLUDE with state INCLUDE(X).

6.6. Action on Reception of Queries

6.6.1. Timer Updates

When a router sends or receives a query with a clear Suppress Router-Side Processing flag, it must update its timers to reflect the correct timeout values for the group or sources being queried. The following table describes the timer actions when sending or receiving a Group-Specific or Group-and-Source Specific Query with the Suppress Router-Side Processing flag not set.

| Query | Action |
|--------|---|
| Q(G,A) | Source Timer for sources in A are lowered to LMQT |
| Q(G) | Group Timer is lowered to LMQT |

When a router sends or receives a query with the Suppress Router-Side Processing flag set, it will not update its timers.

6.6.2. Querier Election

IGMPv3 elects a single querier per subnet using the same querier election mechanism as IGMPv2, namely by IP address. When a router receives a query with a lower IP address, it sets the Other-Querier-Present timer to Other Querier Present Interval and ceases to send queries on the network if it was the previously elected querier. After its Other-Querier Present timer expires, it should begin sending General Queries.

If a router receives an older version query, it MUST use the oldest version of IGMP on the network. For a detailed description of compatibility issues between IGMP versions see section Section 7.

6.6.3. Building and Sending Specific Queries

6.6.3.1. Building and Sending Group Specific Queries

When a table action "Send Q(G)" is encountered, then the group timer must be lowered to LMQT. The router must then immediately send a group specific query as well as schedule [Last Member Query Count - 1] query retransmissions to be sent every [Last Member Query Interval] over [Last Member Query Time].

When transmitting a group specific query, if the group timer is larger than LMQT, the "Suppress Router-Side Processing" bit is set in the query message.

6.6.3.2. Building and Sending Group and Source Specific Queries

When a table action "Send Q(G,X)" is encountered by a querier in the table in Section 6.4.2, the following actions must be performed for each of the sources in X of group G, with source timer larger than LMQT:

- o Set number of retransmissions for each source to [Last Member Query Count].
- o Lower source timer to LMQT.

The router must then immediately send a group and source specific query as well as schedule [Last Member Query Count - 1] query retransmissions to be sent every [Last Member Query Interval] over [Last Member Query Time]. The contents of these queries are calculated as follows.

When building a group and source specific query for a group G, two separate query messages are sent for the group. The first one has the "Suppress Router-Side Processing" bit set and contains all the sources with retransmission state and timers greater than LMQT. The second has the "Suppress Router-Side Processing" bit clear and contains all the sources with retransmission state and timers lower or equal to LMQT. If either of the two calculated messages does not contain any sources, then its transmission is suppressed.

Note: If a group specific query is scheduled to be transmitted at the same time as a group and source specific query for the same group, then transmission of the group and source specific message with the "Suppress Router-Side Processing" bit set may be suppressed.

7. Interoperation With Older Versions of IGMP

IGMP version 3 hosts and routers interoperate with hosts and routers that have not yet been upgraded to IGMPv3. This compatibility is maintained by hosts and routers taking appropriate actions depending on the versions of IGMP operating on hosts and routers within a network.

7.1. Query Version Distinctions

The IGMP version of a Membership Query message is determined as follows:

IGMPv1 Query: length = 8 octets AND Max Resp Code field is zero

IGMPv2 Query: length = 8 octets AND Max Resp Code field is non-zero

IGMPv3 Query: length >= 12 octets

Query messages that do not match any of the above conditions (e.g., a Query of length 10 octets) MUST be silently ignored.

7.2. Group Member Behavior

7.2.1. In the Presence of Older Version Queriers

In order to be compatible with older version routers, IGMPv3 hosts MUST operate in version 1 and version 2 compatibility modes. IGMPv3 hosts MUST keep state per local interface regarding the compatibility mode of each attached network. A host's compatibility mode is determined from the Host Compatibility Mode variable which can be in one of three states: IGMPv1, IGMPv2 or IGMPv3. This variable is kept per interface and is dependent on the version of General Queries heard on that interface as well as the Older Version Querier Present timers for the interface.

In order to switch gracefully between versions of IGMP, hosts keep both an IGMPv1 Querier Present timer and an IGMPv2 Querier Present timer per interface. IGMPv1 Querier Present is set to Older Version Querier Present Timeout seconds whenever an IGMPv1 Membership Query is received. IGMPv2 Querier Present is set to Older Version Querier Present Timeout seconds whenever an IGMPv2 General Query is received.

The Host Compatibility Mode of an interface changes whenever an older version query (than the current compatibility mode) is heard or when certain timer conditions occur. When the IGMPv1 Querier Present timer expires, a host switches to Host Compatibility mode of IGMPv2 if it has a running IGMPv2 Querier Present timer. If it does not have a running IGMPv2 Querier Present timer then it switches to Host Compatibility of IGMPv3. When the IGMPv2 Querier Present timer expires, a host switches to Host Compatibility mode of IGMPv3.

The Host Compatibility Mode variable is based on whether an older version General query was heard in the last Older Version Querier

Present Timeout seconds. The Host Compatibility Mode is set depending on the following:

| Host Compatibility Mode | Timer State |
|-------------------------|---|
| IGMPv3 (default) | IGMPv2 Querier Present not running and IGMPv1 Querier Present not running |
| IGMPv2 | IGMPv2 Querier Present running and IGMPv1 Querier Present not running |
| IGMPv1 | IGMPv1 Querier Present running |

If a host receives a query which causes its Querier Present timers to be updated and correspondingly its compatibility mode, it should switch compatibility modes immediately.

When Host Compatibility Mode is IGMPv3, a host acts using the IGMPv3 protocol on that interface. When Host Compatibility Mode is IGMPv2, a host acts in IGMPv2 compatibility mode, using only the IGMPv2 protocol, on that interface. When Host Compatibility Mode is IGMPv1, a host acts in IGMPv1 compatibility mode, using only the IGMPv1 protocol on that interface.

An IGMPv1 router will send General Queries with the Max Resp Code set to 0. This MUST be interpreted as a value of 100 (10 seconds).

An IGMPv2 router will send General Queries with the Max Resp Code set to the desired Max Resp Time, i.e., the full range of this field is linear and the exponential algorithm described in Section 4.1.1 is not used.

Whenever a host changes its compatibility mode, it cancels all its pending response and retransmission timers.

7.2.2. In the Presence of Older Version Group Members

An IGMPv3 host may be placed on a network where there are hosts that have not yet been upgraded to IGMPv3. A host MAY allow its IGMPv3 Membership Record to be suppressed by either a Version 1 Membership Report, or a Version 2 Membership Report.

7.3. Multicast Router Behavior

7.3.1. In the Presence of Older Version Queriers

IGMPv3 routers may be placed on a network where at least one router on the network has not yet been upgraded to IGMPv3. The following requirements apply:

- o If any older versions of IGMP are present on routers, the querier MUST use the lowest version of IGMP present on the network. This must be administratively assured; routers that desire to be compatible with IGMPv1 and IGMPv2 MUST have a configuration option to act in IGMPv1 or IGMPv2 compatibility modes. When in IGMPv1 mode, routers MUST send Periodic Queries with a Max Resp Code of 0 and truncated at the Group Address field (i.e., 8 bytes long), and MUST ignore Leave Group messages. They SHOULD also warn about receiving an IGMPv2 or IGMPv3 query, although such warnings MUST be rate-limited. When in IGMPv2 mode, routers MUST send Periodic Queries truncated at the Group Address field (i.e., 8 bytes long), and SHOULD also warn about receiving an IGMPv3 query (such warnings MUST be rate-limited). They also MUST fill in the Max Resp Time in the Max Resp Code field, i.e., the exponential algorithm described in Section 4.1.1 is not used.
- o If a router is not explicitly configured to use IGMPv1 or IGMPv2 and hears an IGMPv1 Query or IGMPv2 General Query, it SHOULD log a warning. These warnings MUST be rate-limited.

7.3.2. In the Presence of Older Version Group Members

IGMPv3 routers may be placed on a network where there are hosts that have not yet been upgraded to IGMPv3. In order to be compatible with older version hosts, IGMPv3 routers MUST operate in version 1 and version 2 compatibility modes. IGMPv3 routers keep a compatibility mode per group record. A group's compatibility mode is determined from the Group Compatibility Mode variable which can be in one of three states: IGMPv1, IGMPv2 or IGMPv3. This variable is kept per group record and is dependent on the version of Membership Reports heard for that group as well as the Older Version Host Present timer for the group.

In order to switch gracefully between versions of IGMP, routers keep an IGMPv1 Host Present timer and an IGMPv2 Host Present timer per group record. The IGMPv1 Host Present timer is set to Older Version Host Present Timeout seconds whenever an IGMPv1 Membership Report is received. The IGMPv2 Host Present timer is set to Older Version Host Present Timeout seconds whenever an IGMPv2 Membership Report is received.

The Group Compatibility Mode of a group record changes whenever an older version report (than the current compatibility mode) is heard or when certain timer conditions occur. When the IGMPv1 Host Present timer expires, a router switches to Group Compatibility mode of IGMPv2 if it has a running IGMPv2 Host Present timer. If it does not have a running IGMPv2 Host Present timer then it switches to Group Compatibility of IGMPv3. When the IGMPv2 Host Present timer expires and the IGMPv1 Host Present timer is not running, a router switches to Group Compatibility mode of IGMPv3. Note that when a group switches back to IGMPv3 mode, it takes some time to regain source-specific state information. Source-specific information will be learned during the next General Query, but sources that should be blocked will not be blocked until [Group Membership Interval] after that.

The Group Compatibility Mode variable is based on whether an older version report was heard in the last Older Version Host Present Timeout seconds. The Group Compatibility Mode is set depending on the following:

| Group Compatibility Mode | Timer State |
|--------------------------|---|
| IGMPv3 (default) | IGMPv2 Host Present not running and IGMPv1 Host Present not running |
| IGMPv2 | IGMPv2 Host Present running and IGMPv1 Host Present not running |
| IGMPv1 | IGMPv1 Host Present running |

If a router receives a report which causes its older Host Present timers to be updated and correspondingly its compatibility mode, it SHOULD switch compatibility modes immediately.

When Group Compatibility Mode is IGMPv3, a router acts using the IGMPv3 protocol for that group.

When Group Compatibility Mode is IGMPv2, a router internally translates the following IGMPv2 messages for that group to their IGMPv3 equivalents:

| IGMPv2 Message | IGMPv3 Equivalent |
|----------------|-------------------|
| Report | IS_EX({}) |
| Leave | TO_IN({}) |

IGMPv3 BLOCK messages are ignored, as are source-lists in TO_EX() messages (i.e., any TO_EX() message is treated as TO_EX({})).

When Group Compatibility Mode is IGMPv1, a router internally translates the following IGMPv1 and IGMPv2 messages for that group to their IGMPv3 equivalents:

| IGMPv2 Message | IGMPv3 Equivalent |
|----------------|-------------------|
| v1 Report | IS_EX({}) |
| v2 Report | IS_EX({}) |

In addition to ignoring IGMPv3 BLOCK messages and source-lists in TO_EX() messages as in IGMPv2 Group Compatibility Mode, IGMPv2 Leave messages and IGMPv3 TO_IN() messages are also ignored.

8. List of Timers, Counters and Their Default Values

Most of these timers are configurable. If non-default settings are used, they MUST be consistent among all systems on a single link. Note that parentheses are used to group expressions to make the algebra clear.

8.1. Robustness Variable

The Robustness Variable allows tuning for the expected packet loss on a network. If a network is expected to be lossy, the Robustness Variable may be increased. IGMP is robust to (Robustness Variable - 1) packet losses. The Robustness Variable MUST NOT be zero, and SHOULD NOT be one. Default: 2

8.2. Query Interval

The Query Interval is the interval between General Queries sent by the Querier. Default: 125 seconds.

By varying the [Query Interval], an administrator may tune the number of IGMP messages on the network; larger values cause IGMP Queries to be sent less often.

8.3. Query Response Interval

The Max Response Time used to calculate the Max Resp Code inserted into the periodic General Queries. Default: 100 (10 seconds)

By varying the [Query Response Interval], an administrator may tune the burstiness of IGMP messages on the network; larger values make the traffic less bursty, as host responses are spread out over a larger interval. The number of seconds represented by the [Query Response Interval] must be less than the [Query Interval].

8.4. Group Membership Interval

The Group Membership Interval is the amount of time that must pass before a multicast router decides there are no more members of a group or a particular source on a network.

This value MUST be ((the Robustness Variable) times (the Query Interval)) plus (one Query Response Interval).

8.5. Other Querier Present Interval

The Other Querier Present Interval is the length of time that must pass before a multicast router decides that there is no longer another multicast router which should be the querier. This value MUST be ((the Robustness Variable) times (the Query Interval)) plus (one half of one Query Response Interval).

8.6. Startup Query Interval

The Startup Query Interval is the interval between General Queries sent by a Querier on startup. Default: 1/4 the Query Interval.

8.7. Startup Query Count

The Startup Query Count is the number of Queries sent out on startup, separated by the Startup Query Interval. Default: the Robustness Variable.

8.8. Last Member Query Interval

The Last Member Query Interval is the Max Response Time used to calculate the Max Resp Code inserted into Group-Specific Queries sent in response to Leave Group messages. It is also the Max Response Time used in calculating the Max Resp Code for Group-and-Source-Specific Query messages. Default: 10 (1 second)

Note that for values of LMQUI greater than 12.8 seconds, a limited set of values can be represented, corresponding to sequential values of Max Resp Code. When converting a configured time to a Max Resp Code value, it is recommended to use the exact value if possible, or the next lower value if the requested value is not exactly representable.

This value may be tuned to modify the "leave latency" of the network. A reduced value results in reduced time to detect the loss of the last member of a group or source.

8.9. Last Member Query Count

The Last Member Query Count is the number of Group-Specific Queries sent before the router assumes there are no local members. The Last Member Query Count is also the number of Group-and-Source-Specific Queries sent before the router assumes there are no listeners for a particular source. Default: the Robustness Variable.

8.10. Last Member Query Time

The Last Member Query Time is the time value represented by the Last Member Query Interval, multiplied by the Last Member Query Count. It is not a tunable value, but may be tuned by changing its components.

8.11. Unsolicited Report Interval

The Unsolicited Report Interval is the time between repetitions of a host's initial report of membership in a group. Default: 1 second.

8.12. Older Version Querier Present Interval

The Older Version Querier Present Interval is the timeout for transitioning a host back to IGMPv3 mode once an older version query is heard. When an older version query is received, hosts set their Older Version Querier Present Timer to Older Version Querier Present Interval.

It is RECOMMENDED to use the default values for calculating the interval value as hosts do not know the values configured on the querying routers. This value SHOULD be [Robustness Variable] times [Query Interval] plus (10 times the Max Resp Time in the last received query message).

8.13. Older Host Present Interval

The Older Host Present Interval is the time-out for transitioning a group back to IGMPv3 mode once an older version report is sent for

that group. When an older version report is received, routers set their Older Host Present Timer to Older Host Present Interval.

This value MUST be ((the Robustness Variable) times (the Query Interval)) plus (one Query Response Interval).

8.14. Configuring Timers

This section is meant to provide advice to network administrators on how to tune these settings to their network. Ambitious router implementations might tune these settings dynamically based upon changing characteristics of the network.

8.14.1. Robustness Variable

The Robustness Variable tunes IGMP to expected losses on a link. IGMPv3 is robust to (Robustness Variable - 1) packet losses, e.g., if the Robustness Variable is set to the default value of 2, IGMPv3 is robust to a single packet loss but may operate imperfectly if more losses occur. On lossy subnetworks, the Robustness Variable should be increased to allow for the expected level of packet loss. However, increasing the Robustness Variable increases the leave latency of the subnetwork. (The leave latency is the time between when the last member stops listening to a source or group and when the traffic stops flowing.)

8.14.2. Query Interval

The overall level of periodic IGMP traffic is inversely proportional to the Query Interval. A longer Query Interval results in a lower overall level of IGMP traffic. The Query Interval MUST be equal to or longer than the Max Response Time inserted in General Query messages.

8.14.3. Max Response Time

The burstiness of IGMP traffic is inversely proportional to the Max Response Time. A longer Max Response Time will spread Report messages over a longer interval. However, a longer Max Response Time in Group-Specific and Source-and-Group-Specific Queries extends the leave latency. (The leave latency is the time between when the last member stops listening to a source or group and when the traffic stops flowing.) The expected rate of Report messages can be calculated by dividing the expected number of Reporters by the Max Response Time. The Max Response Time may be dynamically calculated per Query by using the expected number of Reporters for that Query as follows:

| Query Type | Expected number of Reporters |
|---------------------------------|---|
| General Query | All systems on subnetwork |
| Group-Specific Query | All systems that had expressed interest in the group on the subnetwork |
| Source-and-Group-Specific Query | All systems on the subnetwork that had expressed interest in the source and group |

A router is not required to calculate these populations or tune the Max Response Time dynamically; these are simply guidelines.

9. Security Considerations

We consider the ramifications of a forged message of each type, and describe the usage of IPSEC AH to authenticate messages if desired.

9.1. Query Message

A forged Query message from a machine with a lower IP address than the current Querier will cause Querier duties to be assigned to the forger. If the forger then sends no more Query messages, other routers' Other Querier Present timer will time out and one will resume the role of Querier. During this time, if the forger ignores Leave Messages, traffic might flow to groups with no members for up to [Group Membership Interval].

A DoS attack on a host could be staged through forged Group-and-Source-Specific Queries. The attacker can find out about membership of a specific host with a general query. After that it could send a large number of Group-and-Source-Specific queries, each with a large source list and the Maximum Response Time set to a large value. The host will have to store and maintain the sources specified in all of those queries for as long as it takes to send the delayed response. This would consume both memory and CPU cycles in order to augment the recorded sources with the source lists included in the successive queries.

To protect against such a DoS attack, a host stack implementation could restrict the number of Group-and-Source-Specific Queries per group membership within this interval, and/or record only a limited number of sources.

Forged Query messages from the local network can be easily traced. There are three measures necessary to defend against externally forged Queries:

Routers SHOULD NOT forward Queries. This is easier for a router to accomplish if the Query carries the Router-Alert option.

Hosts SHOULD ignore v2 or v3 Queries without the Router-Alert option.

Hosts SHOULD ignore v1, v2 or v3 General Queries sent to a multicast address other than 224.0.0.1, the all-systems address.

9.2. Current-State Report messages

A forged Report message may cause multicast routers to think there are members of a group on a network when there are not. Forged Report messages from the local network are meaningless, since joining a group on a host is generally an unprivileged operation, so a local user may trivially gain the same result without forging any messages. Forged Report messages from external sources are more troublesome; there are two defenses against externally forged Reports:

Ignore the Report if you cannot identify the source address of the packet as belonging to a network assigned to the interface on which the packet was received. This solution means that Reports sent by mobile hosts without addresses on the local network will be ignored. Report messages with a source address of 0.0.0.0 SHOULD be accepted on any interface.

Ignore Report messages without Router Alert options [RFC2113], and require that routers not forward Report messages. (The requirement is not a requirement of generalized filtering in the forwarding path, since the packets already have Router Alert options in them.) This solution breaks backwards compatibility with implementations of IGMPv1 or earlier versions of IGMPv2 which did not require Router Alert.

A forged Version 1 Report Message may put a router into "version 1 members present" state for a particular group, meaning that the router will ignore Leave messages. This can cause traffic to flow to groups with no members for up to [Group Membership Interval]. This can be solved by providing routers with a configuration switch to ignore Version 1 messages completely. This breaks automatic compatibility with Version 1 hosts, so should only be used in situations where "fast leave" is critical.

A forged Version 2 Report Message may put a router into "version 2 members present" state for a particular group, meaning that the router will ignore IGMPv3 source-specific state messages. This can cause traffic to flow from unwanted sources for up to [Group Membership Interval]. This can be solved by providing routers with a configuration switch to ignore Version 2 messages completely. This breaks automatic compatibility with Version 2 hosts, so should only be used in situations where source include and exclude is critical.

9.3. State-Change Report Messages

A forged State-Change Report message will cause the Querier to send out Group-Specific or Source-and-Group-Specific Queries for the group in question. This causes extra processing on each router and on each member of the group, but can not cause loss of desired traffic. There are two defenses against externally forged State-Change Report messages:

Ignore the State-Change Report message if you cannot identify the source address of the packet as belonging to a subnet assigned to the interface on which the packet was received. This solution means that State-Change Report messages sent by mobile hosts without addresses on the local subnet will be ignored. State-Change Report messages with a source address of 0.0.0.0 SHOULD be accepted on any interface.

Ignore State-Change Report messages without Router Alert options [RFC2113], and require that routers not forward State-Change Report messages. (The requirement is not a requirement of generalized filtering in the forwarding path, since the packets already have Router Alert options in them.)

9.4. IPSEC Usage

In addition to these measures, IPSEC in Authentication Header mode [RFC2402] may be used to protect against remote attacks by ensuring that IGMPv3 messages came from a system on the LAN (or, more specifically, a system with the proper key). When using IPSEC, the messages sent to 224.0.0.1 and 224.0.0.22 should be authenticated using AH. When keying, there are two possibilities:

1. Use a symmetric signature algorithm with a single key for the LAN (or a key for each group). This allows validation that a packet was sent by a system with the key. This has the limitation that any system with the key can forge a message; it is not possible to authenticate the individual sender precisely. It also requires disabling IPsec's Replay Protection.

2. When appropriate key management standards have been developed, use an asymmetric signature algorithm. All systems need to know the public key of all routers, and all routers need to know the public key of all systems. This requires a large amount of key management but has the advantage that senders can be authenticated individually so e.g., a host cannot forge a message that only routers should be allowed to send.

This solution only directly applies to Query and Leave messages in IGMPv1 and IGMPv2, since Reports are sent to the group being reported and it is not feasible to agree on a key for host-to-router communication for arbitrary multicast groups.

10. IANA Considerations

All IGMP types described in this document are already assigned in [RFC3228].

11. Contributors

Brad Cain, Steve Deering, Isidor Kouvelas, Bill Fenner, and Ajit Thyagarajan are the authors of RFC 3376, which forms the bulk of the content contained herein.

12. Acknowledgments

We would like to thank Ran Atkinson, Luis Costa, Toerless Eckert, Dino Farinacci, Serge Fdida, Wilbert de Graaf, Sumit Gupta, Mark Handley, Bob Quinn, Michael Speer, Dave Thaler and Rolland Vida for comments and suggestions on RFC 3376.

13. References

13.1. Normative References

- [RFC1112] Deering, S., "Host extensions for IP multicasting", STD 5, RFC 1112, DOI 10.17487/RFC1112, August 1989, <<https://www.rfc-editor.org/info/rfc1112>>.
- [RFC2113] Katz, D., "IP Router Alert Option", RFC 2113, DOI 10.17487/RFC2113, February 1997, <<https://www.rfc-editor.org/info/rfc2113>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

- [RFC2236] Fenner, W., "Internet Group Management Protocol, Version 2", RFC 2236, DOI 10.17487/RFC2236, November 1997, <<https://www.rfc-editor.org/info/rfc2236>>.
- [RFC2402] Kent, S. and R. Atkinson, "IP Authentication Header", RFC 2402, DOI 10.17487/RFC2402, November 1998, <<https://www.rfc-editor.org/info/rfc2402>>.
- [RFC3228] Fenner, B., "IANA Considerations for IPv4 Internet Group Management Protocol (IGMP)", BCP 57, RFC 3228, DOI 10.17487/RFC3228, February 2002, <<https://www.rfc-editor.org/info/rfc3228>>.

13.2. Informative References

- [RFC1071] Braden, R., Borman, D., and C. Partridge, "Computing the Internet checksum", RFC 1071, DOI 10.17487/RFC1071, September 1988, <<https://www.rfc-editor.org/info/rfc1071>>.
- [RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, DOI 10.17487/RFC3376, October 2002, <<https://www.rfc-editor.org/info/rfc3376>>.
- [RFC3569] Bhattacharyya, S., Ed., "An Overview of Source-Specific Multicast (SSM)", RFC 3569, DOI 10.17487/RFC3569, July 2003, <<https://www.rfc-editor.org/info/rfc3569>>.
- [RFC3678] Thaler, D., Fenner, B., and B. Quinn, "Socket Interface Extensions for Multicast Source Filters", RFC 3678, DOI 10.17487/RFC3678, January 2004, <<https://www.rfc-editor.org/info/rfc3678>>.

Appendix A. Design Rationale

A.1. The Need for State-Change Messages

IGMPv3 specifies two types of Membership Reports: Current-State and State Change. This section describes the rationale for the need for both these types of Reports.

Routers need to distinguish Membership Reports that were sent in response to Queries from those that were sent as a result of a change in interface state. Membership reports that are sent in response to Membership Queries are used mainly to refresh the existing state at the router; they typically do not cause transitions in state at the router. Membership Reports that are sent in response to changes in

interface state require the router to take some action in response to the received report (see Section 6.4).

The inability to distinguish between the two types of reports would force a router to treat all Membership Reports as potential changes in state and could result in increased processing at the router as well as an increase in IGMP traffic on the network.

A.2. Host Suppression

In IGMPv1 and IGMPv2, a host would cancel sending a pending membership reports if a similar report was observed from another member on the network. In IGMPv3, this suppression of host membership reports has been removed. The following points explain the reasons behind this decision.

1. Routers may want to track per-host membership status on an interface. This allows routers to implement fast leaves (e.g., for layered multicast congestion control schemes) as well as track membership status for possible accounting purposes.
2. Membership Report suppression does not work well on bridged LANs. Many bridges and Layer2/Layer3 switches that implement IGMP snooping do not forward IGMP messages across LAN segments in order to prevent membership report suppression. Removing membership report suppression eases the job of these IGMP snooping devices.
3. By eliminating membership report suppression, hosts have fewer messages to process; this leads to a simpler state machine implementation.
4. In IGMPv3, a single membership report now bundles multiple multicast group records to decrease the number of packets sent. In comparison, the previous versions of IGMP required that each multicast group be reported in a separate message.

A.3. Switching Router Filter Modes from EXCLUDE to INCLUDE

If there exist hosts in both EXCLUDE and INCLUDE modes for a single multicast group in a network, the router must be in EXCLUDE mode as well (see section 6.2.1). In EXCLUDE mode, a router forwards traffic from all sources unless that source exists in the exclusion source list. If all hosts in EXCLUDE mode cease to exist, it would be desirable for the router to switch back to INCLUDE mode seamlessly without interrupting the flow of traffic to existing receivers.

One of the ways to accomplish this is for routers to keep track of all sources desired by hosts that are in INCLUDE mode even though the router itself is in EXCLUDE mode. If the group timer now expires in EXCLUDE mode, it implies that there are no hosts in EXCLUDE mode on the network (otherwise a membership report from that host would have refreshed the group timer). The router can then switch to INCLUDE mode seamlessly with the list of sources currently being forwarded in its source list.

Appendix B. Summary of Changes from IGMPv2

While the main additional feature of IGMPv3 is the addition of source filtering, the following is a summary of other changes from RFC 2236.

- o State is maintained as Group + List-of-Sources, not simply Group as in IGMPv2.
- o Interoperability with IGMPv1 and IGMPv2 systems is defined as operations on the IGMPv3 state.
- o The IP Service Interface has changed to allow specification of source-lists.
- o The Querier includes its Robustness Variable and Query Interval in Query packets to allow synchronization of these variables on non-Queriers.
- o The Max Response Time in Query messages has an exponential range, changing the maximum from 25.5 seconds to about 53 minutes, for use on links with huge numbers of systems.
- o Hosts retransmit state-change messages for increased robustness.
- o Additional data sections are defined to allow later extensions.
- o Report packets are sent to 224.0.0.22, to assist layer-2 switches in snooping.
- o Report packets can contain multiple group records, to allow reporting of full current state using fewer packets.
- o Hosts no longer perform suppression, to simplify implementations and permit explicit membership tracking.
- o New Suppress Router-Side Processing (S) flag in Query messages fixes robustness issues which were also present in IGMPv2.

Appendix C. Summary of Changes from RFC 3376

The following is a list of changes made since RFC 3376.

- o Modified definition of Older Version Querier Present Interval to address Erratum 4375.

Author's Address

Brian Haberman (editor)
Johns Hopkins University Applied Physics Lab

Email: brian@innovationslab.net

Network Working Group
Internet-Draft
Obsoletes: 3810 (if approved)
Intended status: Standards Track
Expires: January 13, 2022

B. Haberman, Ed.
JHU APL
July 12, 2021

Multicast Listener Discovery Version 2 (MLDv2) for IPv6
draft-mcast-pim-3810bis-00

Abstract

This document updates RFC 2710, and it specifies Version 2 of the Multicast Listener Discovery Protocol (MLDv2). MLD is used by an IPv6 router to discover the presence of multicast listeners on directly attached links, and to discover which multicast addresses are of interest to those neighboring nodes. MLDv2 is designed to be interoperable with MLDv1. MLDv2 adds the ability for a node to report interest in listening to packets with a particular multicast address only from specific source addresses or from all sources except for specific source addresses.

This document obsoletes RFC 3810.

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 <https://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 January 13, 2022.

Copyright Notice

Copyright (c) 2021 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

(<https://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 | 4 |
| 2. Protocol Overview | 5 |
| 2.1. Building Multicast Listening State on Multicast Address Listeners | 6 |
| 2.2. Exchanging Messages between the Querier and the Listening Nodes | 6 |
| 2.3. Building Multicast Address Listener State on Multicast Routers | 9 |
| 3. The Service Interface for Requesting IP Multicast Reception . | 11 |
| 4. Multicast Listening State Maintained by Nodes | 13 |
| 4.1. Per-Socket State | 13 |
| 4.2. Per-Interface State | 13 |
| 5. Message Formats | 15 |
| 5.1. Multicast Listener Query Message | 16 |
| 5.1.1. Code | 18 |
| 5.1.2. Checksum | 18 |
| 5.1.3. Maximum Response Code | 18 |
| 5.1.4. Reserved | 18 |
| 5.1.5. Multicast Address | 18 |
| 5.1.6. Resv | 19 |
| 5.1.7. S Flag (Suppress Router-Side Processing) | 19 |
| 5.1.8. QRV (Querier's Robustness Variable) | 19 |
| 5.1.9. QQIC (Querier's Query Interval Code) | 19 |
| 5.1.10. Number of Sources (N) | 20 |
| 5.1.11. Source Address [i] | 20 |
| 5.1.12. Additional Data | 20 |
| 5.1.13. Query Variants | 20 |
| 5.1.14. Source Addresses for Queries | 21 |
| 5.1.15. Destination Addresses for Queries | 21 |
| 5.2. Version 2 Multicast Listener Report Message | 21 |
| 5.2.1. Reserved | 24 |
| 5.2.2. Checksum | 24 |
| 5.2.3. Nr of Mcast Address Records (M) | 24 |
| 5.2.4. Multicast Address Record | 24 |
| 5.2.5. Record Type | 24 |
| 5.2.6. Aux Data Len | 24 |
| 5.2.7. Number of Sources (N) | 24 |
| 5.2.8. Multicast Address | 24 |

| | | |
|---------|---|----|
| 5.2.9. | Source Address [i] | 25 |
| 5.2.10. | Auxiliary Data | 25 |
| 5.2.11. | Additional Data | 25 |
| 5.2.12. | Multicast Address Record Types | 25 |
| 5.2.13. | Source Addresses for Reports | 27 |
| 5.2.14. | Destination Addresses for Reports | 28 |
| 5.2.15. | Multicast Listener Report Size | 28 |
| 6. | Protocol Description for Multicast Address Listeners | 29 |
| 6.1. | Action on Change of Per-Interface State | 30 |
| 6.2. | Action on Reception of a Query | 32 |
| 6.3. | Action on Timer Expiration | 34 |
| 7. | Description of the Protocol for Multicast Routers | 36 |
| 7.1. | Conditions for MLD Queries | 37 |
| 7.2. | MLD State Maintained by Multicast Routers | 38 |
| 7.2.1. | Definition of Router Filter Mode | 39 |
| 7.2.2. | Definition of Filter Timers | 40 |
| 7.2.3. | Definition of Source Timers | 41 |
| 7.3. | MLDv2 Source Specific Forwarding Rules | 43 |
| 7.4. | Action on Reception of Reports | 44 |
| 7.4.1. | Reception of Current State Records | 44 |
| 7.4.2. | Reception of Filter Mode Change and Source List Change Records | 45 |
| 7.5. | Switching Router Filter Modes | 46 |
| 7.6. | Action on Reception of Queries | 47 |
| 7.6.1. | Timer Updates | 47 |
| 7.6.2. | Querier Election | 47 |
| 7.6.3. | Building and Sending Specific Queries | 48 |
| 8. | Interoperation with MLDv1 | 49 |
| 8.1. | Query Version Distinctions | 49 |
| 8.2. | Multicast Address Listener Behavior | 49 |
| 8.2.1. | In the Presence of MLDv1 Routers | 50 |
| 8.2.2. | In the Presence of MLDv1 Multicast Address Listeners | 50 |
| 8.3. | Multicast Router Behavior | 50 |
| 8.3.1. | In the Presence of MLDv1 Routers | 50 |
| 8.3.2. | In the Presence of MLDv1 Multicast Address Listeners | 51 |
| 9. | List of Timers, Counters, and their Default Values | 52 |
| 9.1. | Robustness Variable | 52 |
| 9.2. | Query Interval | 52 |
| 9.3. | Query Response Interval | 52 |
| 9.4. | Multicast Address Listening Interval | 53 |
| 9.5. | Other Querier Present Timeout | 53 |
| 9.6. | Startup Query Interval | 53 |
| 9.7. | Startup Query Count | 53 |
| 9.8. | Last Listener Query Interval | 53 |
| 9.9. | Last Listener Query Count | 54 |
| 9.10. | Last Listener Query Time | 54 |
| 9.11. | Unsolicited Report Interval | 54 |
| 9.12. | Older Version Querier Present Timeout | 54 |

| | |
|--|----|
| 9.13. Older Version Host Present Timeout | 54 |
| 9.14. Configuring timers | 54 |
| 9.14.1. Robustness Variable | 55 |
| 9.14.2. Query Interval | 55 |
| 9.14.3. Maximum Response Delay | 55 |
| 10. Security Considerations | 56 |
| 10.1. Query Message | 56 |
| 10.2. Current State Report messages | 57 |
| 10.3. State Change Report messages | 57 |
| 11. IANA Considerations | 57 |
| 12. Contributors | 57 |
| 13. Acknowledgments | 57 |
| 14. References | 58 |
| 14.1. Normative References | 58 |
| 14.2. Informative References | 58 |
| Appendix A. Design Rationale | 59 |
| A.1. The Need for State Change Messages | 59 |
| A.2. Host Suppression | 60 |
| A.3. Switching router filter modes from EXCLUDE to INCLUDE . . | 60 |
| Appendix B. Summary of Changes | 61 |
| B.1. MLDv1 | 61 |
| B.2. MLDv2 | 62 |
| Author's Address | 62 |

1. Introduction

The Multicast Listener Discovery Protocol (MLD) is used by IPv6 routers to discover the presence of multicast listeners (i.e., nodes that wish to receive multicast packets) on their directly attached links, and to discover specifically which multicast addresses are of interest to those neighboring nodes. Note that a multicast router may itself be a listener of one or more multicast addresses; in this case it performs both the "multicast router part" and the "multicast address listener part" of the protocol, to collect the multicast listener information needed by its multicast routing protocol on the one hand, and to inform itself and other neighboring multicast routers of its listening state on the other hand.

This document specifies Version 2 of MLD. The previous version of MLD is specified in [RFC2710]. In this document we will refer to it as MLDv1. MLDv2 is a translation of the IGMPv3 protocol [RFC3376] for IPv6 semantics.

The MLDv2 protocol, when compared to MLDv1, adds support for "source filtering", i.e., the ability for a node to report interest in listening to packets only from specific source addresses, as required to support Source-Specific Multicast [RFC3569], or from *all but*

specific source addresses, sent to a particular multicast address. MLDv2 is designed to be interoperable with MLDv1.

This document obsoletes [RFC3810].

The capitalized 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].

2. Protocol Overview

This section gives a brief description of the protocol operation. The following sections present the protocol details.

MLD is an asymmetric protocol; it specifies separate behaviors for multicast address listeners (i.e., hosts or routers that listen to multicast packets) and multicast routers. The purpose of MLD is to enable each multicast router to learn, for each of its directly attached links, which multicast addresses and which sources have interested listeners on that link. The information gathered by MLD is provided to whichever multicast routing protocol is used by the router, in order to ensure that multicast packets are delivered to all links where there are listeners interested in such packets.

Multicast routers only need to know that at least one node on an attached link is listening to packets for a particular multicast address, from a particular source; a multicast router is not required to individually keep track of the interests of each neighboring node. (Nevertheless, see Appendix A.2 item 1 for discussion.)

A multicast router performs the router part of the MLDv2 protocol (described in details in Section 7) on each of its directly attached links. If a multicast router has more than one interface connected to the same link, it only needs to operate the protocol on one of those interfaces. The router behavior depends on whether there are several multicast routers on the same subnet, or not. If that is the case, a querier election mechanism (described in Section 7.6.2) is used to elect a single multicast router to be in Querier state. This router is called the Querier. All multicast routers on the subnet listen to the messages sent by multicast address listeners, and maintain the same multicast listening information state, so that they can take over the querier role, should the present Querier fail. Nevertheless, only the Querier sends periodical or triggered query messages on the subnet, as described in Section 7.1.

A multicast address listener performs the listener part of the MLDv2 protocol (described in details in Section 6) on all interfaces on

which multicast reception is supported, even if more than one of those interfaces are connected to the same link.

2.1. Building Multicast Listening State on Multicast Address Listeners

Upper-layer protocols and applications that run on a multicast address listener node use specific service interface calls (described in Section 3) to ask the IP layer to enable or disable reception of packets sent to specific multicast addresses. The node keeps Multicast Address Listening state for each socket on which the service interface calls have been invoked (Section 4.1). In addition to this per-socket multicast listening state, a node must also maintain or compute multicast listening state for each of its interfaces (Section 4.2). Conceptually, that state consists of a set of records, with each record containing an IPv6 multicast address, a filter mode, and a source list. The filter mode may be either INCLUDE or EXCLUDE. In INCLUDE mode, reception of packets sent to the specified multicast address is enabled only from the source addresses listed in the source list. In EXCLUDE mode, reception of packets sent to the given multicast address is enabled from all source addresses except those listed in the source list.

At most one record per multicast address exists for a given interface. This per-interface state is derived from the per-socket state, but may differ from it when different sockets have differing filter modes and/or source lists for the same multicast address and interface. After a multicast packet has been accepted from an interface by the IP layer, its subsequent delivery to the application connected to a particular socket depends on the multicast listening state of that socket (and possibly also on other conditions, such as what transport-layer port the socket is bound to). Note that MLDv2 messages are not subject to source filtering and must always be processed by hosts and routers.

2.2. Exchanging Messages between the Querier and the Listening Nodes

There are three types of MLDv2 query messages: General Queries, Multicast Address Specific Queries, and Multicast Address and Source Specific Queries. The Querier periodically sends General Queries, to learn multicast address listener information from an attached link. These queries are used to build and refresh the Multicast Address Listener state inside all multicast routers on the link.

Nodes respond to these queries by reporting their per-interface Multicast Address Listening state, through Current State Report messages sent to a specific multicast address all MLDv2 routers on the link listen to. On the other hand, if the listening state of a node changes, the node immediately reports these changes through a

State Change Report message. The State Change Report contains either Filter Mode Change records, Source List Change records, or records of both types. A detailed description of the report messages is presented in Section 5.2.12.

Both router and listener state changes are mainly triggered by the expiration of a specific timer, or the reception of an MLD message (listener state change can be also triggered by the invocation of a service interface call). Therefore, to enhance protocol robustness, in spite of the possible unreliability of message exchanges, messages are retransmitted several times. Furthermore, timers are set so as to take into account the possible message losses, and to wait for retransmissions.

Periodical General Queries and Current State Reports do not apply this rule, in order not to overload the link; it is assumed that in general these messages do not generate state changes, their main purpose being to refresh existing state. Thus, even if one such message is lost, the corresponding state will be refreshed during the next reporting period.

As opposed to Current State Reports, State Change Reports are retransmitted several times, in order to avoid them being missed by one or more multicast routers. The number of retransmissions depends on the so-called Robustness Variable. This variable allows tuning the protocol according to the expected packet loss on a link. If a link is expected to be lossy (e.g., a wireless connection), the value of the Robustness Variable may be increased. MLD is robust to [Robustness Variable]-1 packet losses. This document recommends a default value of 2 for the Robustness Variable (see Section 9.1).

If more changes to the same per-interface state entry occur before all the retransmissions of the State Change Report for the first change have been completed, each additional change triggers the immediate transmission of a new State Change Report. Section 6.1 shows how the content of this new report is computed. Retransmissions of the new State Change Report will be scheduled as well, in order to ensure that each instance of state change is transmitted at least [Robustness Variable] times.

If a node on a link expresses, through a State Change Report, its desire to no longer listen to a particular multicast address (or source), the Querier must query for other listeners of the multicast address (or source) before deleting the multicast address (or source) from its Multicast Address Listener state and stopping the corresponding traffic. Thus, the Querier sends a Multicast Address Specific Query to verify whether there are nodes still listening to a specified multicast address or not. Similarly, the Querier sends a

Multicast Address and Source Specific Query to verify whether, for a specified multicast address, there are nodes still listening to a specific set of sources, or not. Section 5.1.13 describes each query in more detail.

Both Multicast Address Specific Queries and Multicast Address and Source Specific Queries are only sent in response to State Change Reports, never in response to Current State Reports. This distinction between the two types of reports is needed to avoid the router treating all Multicast Listener Reports as potential changes in state. By doing so, the fast leave mechanism of MLDv2, described in more detail in Section 2.2, might not be effective if a State Change Report is lost, and only the following Current State Report is received by the router. Nevertheless, it avoids an increased processing at the router and it reduces the MLD traffic on the link. More details on the necessity of distinguishing between the two report types can be found in Appendix A.1.

Nodes respond to the above queries through Current State Reports, that contain their per-interface Multicast Address Listening state only for the multicast addresses (or sources) being queried.

As stated earlier, in order to ensure protocol robustness, all the queries, except the periodical General Queries, are retransmitted several times within a given time interval. The number of retransmissions depends on the Robustness Variable. If, while scheduling new queries, there are pending queries to be retransmitted for the same multicast address, the new queries and the pending queries have to be merged. In addition, host reports received for a multicast address with pending queries may affect the contents of those queries. The process of building and maintaining the state of pending queries is presented in Section 7.6.3.

Protocol robustness is also enhanced through the use of the S flag (Suppress Router-Side Processing). As described above, when a Multicast Address Specific or a Multicast Address and Source Specific Query is sent by the Querier, a number of retransmissions of the query are scheduled. In the original (first) query the S flag is clear. When the Querier sends this query, it lowers the timers for the concerned multicast address (or source) to a given value; similarly, any non-querier multicast router that receives the query lowers its timers in the same way. Nevertheless, while waiting for the next scheduled queries to be sent, the Querier may receive a report that updates the timers. The scheduled queries still have to be sent, in order to ensure that a non-querier router keeps its state synchronized with the current Querier (the non-querier router might have missed the first query). Nevertheless, the timers should not be

lowered again, as a valid answer was already received. Therefore, in subsequent queries the Querier sets the S flag.

2.3. Building Multicast Address Listener State on Multicast Routers

Multicast routers that implement MLDv2 (whether they are in Querier state or not) keep state per multicast address per attached link. This multicast address listener state consists of a Filter Mode, a Filter Timer, and a Source List, with a timer associated to each source from the list. The Filter Mode is used to summarize the total listening state of a multicast address to a minimum set, such that all nodes' listening states are respected. The Filter Mode may change in response to the reception of particular types of report messages, or when certain timer conditions occur.

A router is in INCLUDE mode for a specific multicast address on a given interface if all the listeners on the link interested in that address are in INCLUDE mode. The router state is represented through the notation INCLUDE (A), where A is a list of sources, called the "Include List". The Include List is the set of sources that one or more listeners on the link have requested to receive. All the sources from the Include List will be forwarded by the router. Any other source that is not in the Include List will be blocked by the router.

A source can be added to the current Include List if a listener in INCLUDE mode sends a Current State or a State Change Report that includes that source. Each source from the Include List is associated with a source timer that is updated whenever a listener in INCLUDE mode sends a report that confirms its interest in that specific source. If the timer of a source from the Include List expires, the source is deleted from the Include List.

Besides this "soft leave" mechanism, there is also a "fast leave" scheme in MLDv2; it is also based on the use of source timers. When a node in INCLUDE mode expresses its desire to stop listening to a specific source, all the multicast routers on the link lower their timers for that source to a given value. The Querier then sends a Multicast Address and Source Specific Query, to verify whether there are other listeners for that source on the link, or not. If a report that includes this source is received before the timer expiration, all the multicast routers on the link update the source timer. If not, the source is deleted from the Include List. The handling of the Include List, according to the received reports, is detailed in Tables 7.4.1 and 7.4.2.

A router is in EXCLUDE mode for a specific multicast address on a given interface if there is at least one listener in EXCLUDE mode for

that address on the link. When the first report is received from such a listener, the router sets the Filter Timer that corresponds to that address. This timer is reset each time an EXCLUDE mode listener confirms its listening state through a Current State Report. The timer is also updated when a listener, formerly in INCLUDE mode, announces its filter mode change through a State Change Report message. If the Filter Timer expires, it means that there are no more listeners in EXCLUDE mode on the link. In this case, the router switches back to INCLUDE mode for that multicast address.

When the router is in EXCLUDE mode, the router state is represented by the notation EXCLUDE (X,Y), where X is called the "Requested List" and Y is called the "Exclude List". All sources, except those from the Exclude List, will be forwarded by the router. The Requested List has no effect on forwarding. Nevertheless, the router has to maintain the Requested List for two reasons:

To keep track of sources that listeners in INCLUDE mode listen to. This is necessary to assure a seamless transition of the router to INCLUDE mode, when there is no listener in EXCLUDE mode left. This transition should not interrupt the flow of traffic to listeners in INCLUDE mode for that multicast address. Therefore, at the time of the transition, the Requested List should contain the set of sources that nodes in INCLUDE mode have explicitly requested.

When the router switches to INCLUDE mode, the sources in the Requested List are moved to the Include List, and the Exclude List is deleted. Before switching, the Requested List can contain an inexact guess of the sources listeners in INCLUDE mode listen to - might be too large or too small. These inexactitudes are due to the fact that the Requested List is also used for fast blocking purposes, as described below. If such a fast blocking is required, some sources may be deleted from the Requested List (as shown in Tables 7.4.1 and 7.4.2) in order to reduce router state. Nevertheless, in each such case the Filter Timer is updated as well. Therefore, listeners in INCLUDE mode will have enough time, before an eventual switching, to reconfirm their interest in the eliminated source(s), and rebuild the Requested List accordingly. The protocol ensures that when a switch to INCLUDE mode occurs, the Requested List will be accurate. Details about the transition of the router to INCLUDE mode are presented in Appendix A.3.

To allow the fast blocking of previously unblocked sources. If the router receives a report that contains such a request, the concerned sources are added to the Requested List. Their timers are set to a given small value, and a Multicast Address and Source Specific Query is sent by the Querier, to check whether there are

nodes on the link still interested in those sources, or not. If no node announces its interest in receiving those specific source, the timers of those sources expire. Then, the sources are moved from the Requested List to the Exclude List. From then on, the sources will be blocked by the router.

The handling of the EXCLUDE mode router state, according to the received reports, is detailed in Tables 7.4.1 and 7.4.2.

Both the MLDv2 router and listener behaviors described in this document were defined to ensure backward interoperability with MLDv1 hosts and routers. Interoperability issues are detailed in Section 8.

3. The Service Interface for Requesting IP Multicast Reception

Within an IP system, there is (at least conceptually) a service interface used by upper-layer protocols or application programs to ask the IP layer to enable or disable reception of packets sent to specific IP multicast addresses. In order to take full advantage of the capabilities of MLDv2, a node's IP service interface must support the following operation:

```
IPv6MulticastListen ( socket, interface, IPv6 multicast-address,  
                      filter-mode, source-list )
```

where:

- o "socket" is an implementation-specific parameter used to distinguish among different requesting entities (e.g., programs, processes) within the node; the socket parameter of BSD Unix system calls is a specific example.
- o "interface" is a local identifier of the network interface on which reception of the specified multicast address is to be enabled or disabled. Interfaces may be physical (e.g., an Ethernet interface) or virtual (e.g., the endpoint of a Frame Relay virtual circuit or an IP-in-IP "tunnel"). An implementation may allow a special "unspecified" value to be passed as the interface parameter, in which case the request would apply to the "primary" or "default" interface of the node (perhaps established by system configuration). If reception of the same multicast address is desired on more than one interface, IPv6MulticastListen is invoked separately for each desired interface.
- o "IPv6 multicast address" is the multicast address to which the request pertains. If reception of more than one multicast address

on a given interface is desired, `IPv6MulticastListen` is invoked separately for each desired address.

- o "filter mode" may be either `INCLUDE` or `EXCLUDE`. In `INCLUDE` mode, reception of packets sent to the specified multicast address is requested only from the source addresses listed in the source list parameter. In `EXCLUDE` mode, reception of packets sent to the given multicast address is requested from all source addresses except those listed in the source list parameter.
- o "source list" is an unordered list of zero or more unicast addresses from which multicast reception is desired or not desired, depending on the filter mode. An implementation MAY impose a limit on the size of source lists. When an operation causes the source list size limit to be exceeded, the service interface SHOULD return an error.

For a given combination of socket, interface, and IPv6 multicast address, only a single filter mode and source list can be in effect at any one time. Nevertheless, either the filter mode or the source list, or both, may be changed by subsequent `IPv6MulticastListen` requests that specify the same socket, interface, and IPv6 multicast address. Each subsequent request completely replaces any earlier request for the given socket, interface, and multicast address.

The MLDv1 protocol did not support source filters, and had a simpler service interface; it consisted of Start Listening and Stop Listening operations to enable and disable listening to a given multicast address (from all sources) on a given interface. The equivalent operations in the new service interface are as follows:

The Start Listening operation is equivalent to:

```
IPv6MulticastListen ( socket, interface, IPv6 multicast address,  
                     EXCLUDE, {} )
```

and the Stop Listening operation is equivalent to:

```
IPv6MulticastListen ( socket, interface, IPv6 multicast address,  
                     INCLUDE, {} )
```

where `{}` is an empty source list.

An example of an API that provides the capabilities outlined in this service interface is given in [RFC3678].

4. Multicast Listening State Maintained by Nodes

4.1. Per-Socket State

For each socket on which `IPv6MulticastListen` has been invoked, the node records the desired multicast listening state for that socket. That state conceptually consists of a set of records of the form:

(interface, IPv6 multicast address, filter mode, source list)

The per-socket state evolves in response to each invocation of `IPv6MulticastListen` on the socket, as follows:

If the requested filter mode is `INCLUDE` and the requested source list is empty, then the entry that corresponds to the requested interface and multicast address is deleted, if present. If no such entry is present, the request has no effect.

If the requested filter mode is `EXCLUDE` or the requested source list is non-empty, then the entry that corresponds to the requested interface and multicast address, if present, is changed to contain the requested filter mode and source list. If no such entry is present, a new entry is created, using the parameters specified in the request.

4.2. Per-Interface State

In addition to the per-socket multicast listening state, a node must also maintain or compute multicast listening state for each of its interfaces. That state conceptually consists of a set of records of the form:

(IPv6 multicast address, filter mode, source list)

At most one record per multicast address exists for a given interface. This per-interface state is derived from the per-socket state, but may differ from it when different sockets have differing filter modes and/or source lists for the same multicast address and interface. For example, suppose one application or process invokes the following operation on socket `s1`:

```
IPv6MulticastListen ( s1, i, m, INCLUDE, {a, b, c} )
```

requesting reception on interface `i` of packets sent to multicast address `m`, only if they come from the sources `a`, `b`, or `c`. Suppose another application or process invokes the following operation on socket `s2`:

```
IPv6MulticastListen ( s2, i, m, INCLUDE, {b, c, d} )
```

requesting reception on the same interface *i* of packets sent to the same multicast address *m*, only if they come from sources *b*, *c*, or *d*. In order to satisfy the reception requirements of both sockets, it is necessary for interface *i* to receive packets sent to *m* from any one of the sources *a*, *b*, *c*, or *d*. Thus, in this example, the listening state of interface *i* for multicast address *m* has filter mode INCLUDE and source list {*a*, *b*, *c*, *d*}.

After a multicast packet has been accepted from an interface by the IP layer, its subsequent delivery to the application or process that listens on a particular socket depends on the multicast listening state of that socket (and possibly also on other conditions, such as what transport-layer port the socket is bound to). So, in the above example, if a packet arrives on interface *i*, destined to multicast address *m*, with source address *a*, it may be delivered on socket *s1* but not on socket *s2*. Note that MLDv2 messages are not subject to source filtering and must always be processed by hosts and routers.

Requiring the filtering of packets based upon a socket's multicast reception state is a new feature of this service interface. The previous service interface described no filtering based upon multicast listening state; rather, a Start Listening operation on a socket simply caused the node to start to listen to a multicast address on the given interface; packets sent to that multicast address could be delivered to all sockets, whether they had started to listen or not.

The general rules for deriving the per-interface state from the per-socket state are as follows: for each distinct (interface, IPv6 multicast address) pair that appears in any per-socket state, a per-interface record is created for that multicast address on that interface. Considering all socket records that contain the same (interface, IPv6 multicast address) pair,

if any such record has a filter mode of EXCLUDE, then the filter mode of the interface record is EXCLUDE, and the source list of the interface record is the intersection of the source lists of all socket records in EXCLUDE mode, minus those source addresses that appear in any socket record in INCLUDE mode. For example, if the socket records for multicast address *m* on interface *i* are:

```
from socket s1: ( i, m, EXCLUDE, {a, b, c, d} )
```

```
from socket s2: ( i, m, EXCLUDE, {b, c, d, e} )
```

```
from socket s3: ( i, m, INCLUDE, {d, e, f} )
```

then the corresponding interface record on interface *i* is:

```
( m, EXCLUDE, {b, c} )
```

If a fourth socket is added, such as:

```
From socket s4: ( i, m, EXCLUDE, {} )
```

then the interface record becomes:

```
( m, EXCLUDE, {} )
```

if all such records have a filter mode of INCLUDE, then the filter mode of the interface record is INCLUDE, and the source list of the interface record is the union of the source lists of all the socket records. For example, if the socket records for multicast address *m* on interface *i* are:

```
from socket s1: ( i, m, INCLUDE, {a, b, c} )
```

```
from socket s2: ( i, m, INCLUDE, {b, c, d} )
```

```
from socket s3: ( i, m, INCLUDE, {e, f} )
```

then the corresponding interface record on interface *i* is:

```
( m, INCLUDE, {a, b, c, d, e, f} )
```

An implementation MUST NOT use an EXCLUDE interface record for a multicast address if all sockets for this multicast address are in INCLUDE state. If system resource limits are reached when a per-interface state source list is calculated, an error MUST be returned to the application which requested the operation.

The above rules for deriving the per-interface state are (re)evaluated whenever an IPv6MulticastListen invocation modifies the per-socket state by adding, deleting, or modifying a per-socket state record. Note that a change of the per-socket state does not necessarily result in a change of the per-interface state.

5. Message Formats

MLDv2 is a sub-protocol of ICMPv6, that is, MLDv2 message types are a subset of ICMPv6 messages, and MLDv2 messages are identified in IPv6 packets by a preceding Next Header value of 58. All MLDv2 messages described in this document MUST be sent with a link-local IPv6 Source Address, an IPv6 Hop Limit of 1, and an IPv6 Router Alert option [RFC2711] in a Hop-by-Hop Options header. (The Router Alert option

is necessary to cause routers to examine MLDv2 messages sent to IPv6 multicast addresses in which the routers themselves have no interest.) MLDv2 Reports can be sent with the source address set to the unspecified address [RFC3513], if a valid link-local IPv6 source address has not been acquired yet for the sending interface. (See Section 5.2.13. for details.)

There are two MLD message types of concern to the MLDv2 protocol described in this document:

Multicast Listener Query (Type = decimal 130)

Version 2 Multicast Listener Report (Type = decimal 143). See Section 11 for IANA considerations.

To assure the interoperability with nodes that implement MLDv1 (see Section 8), an implementation of MLDv2 must also support the following two message types:

Version 1 Multicast Listener Report (Type = decimal 131) [RFC2710]

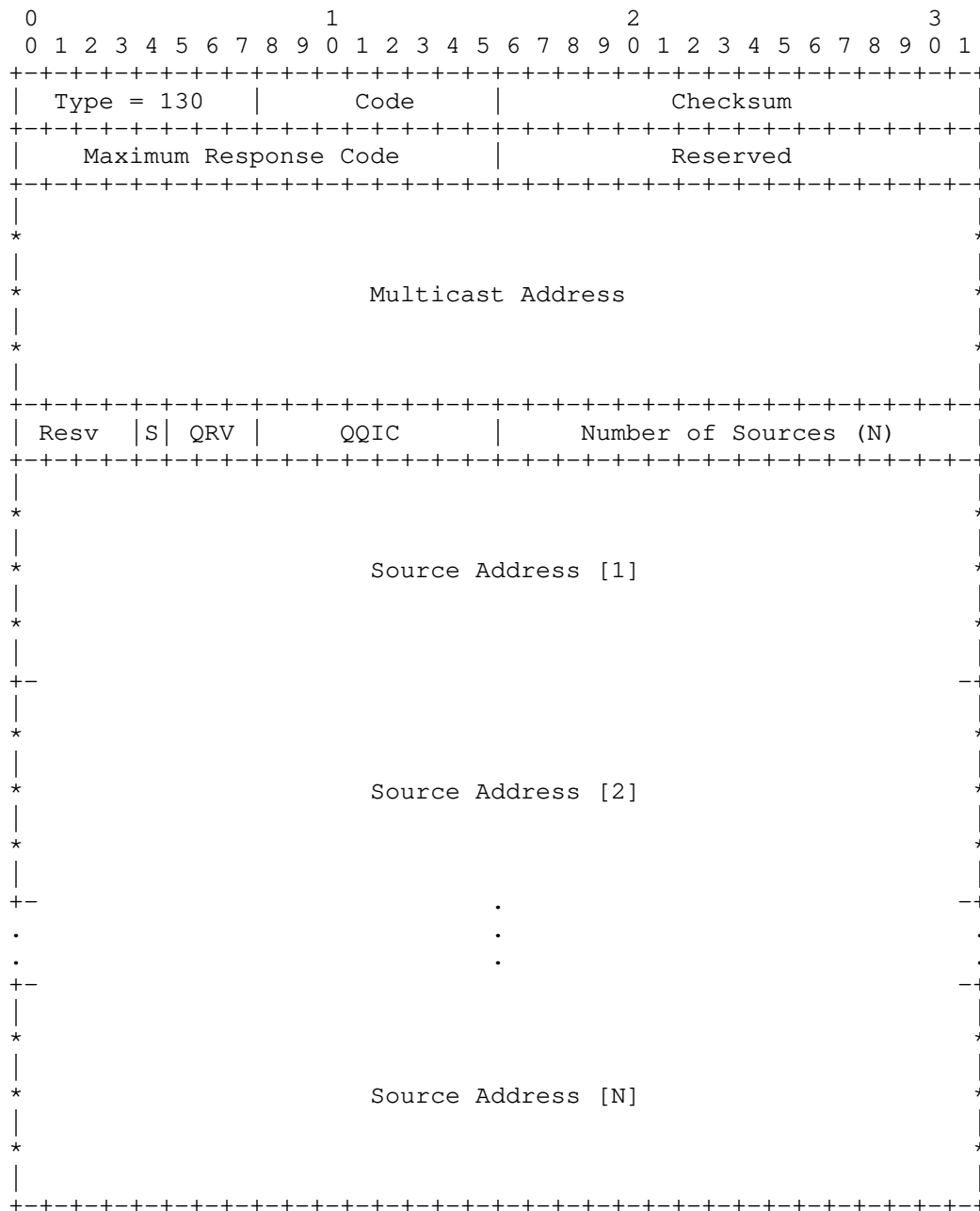
Version 1 Multicast Listener Done (Type = decimal 132) [RFC2710]

Unrecognized message types MUST be silently ignored. Other message types may be used by newer versions or extensions of MLD, by multicast routing protocols, or for other uses.

In this document, unless otherwise qualified, the capitalized words "Query" and "Report" refer to MLD Multicast Listener Queries and MLD Version 2 Multicast Listener Reports, respectively.

5.1. Multicast Listener Query Message

Multicast Listener Queries are sent by multicast routers in Querier State to query the multicast listening state of neighboring interfaces. Queries have the following format:



5.1.1. Code

Initialized to zero by the sender; ignored by receivers.

5.1.2. Checksum

The standard ICMPv6 checksum; it covers the entire MLDv2 message, plus a "pseudo-header" of IPv6 header fields [RFC2463]. For computing the checksum, the Checksum field is set to zero. When a packet is received, the checksum MUST be verified before processing it.

5.1.3. Maximum Response Code

The Maximum Response Code field specifies the maximum time allowed before sending a responding Report. The actual time allowed, called the Maximum Response Delay, is represented in units of milliseconds, and is derived from the Maximum Response Code as follows:

If Maximum Response Code < 32768, Maximum Response Delay = Maximum Response Code

If Maximum Response Code >=32768, Maximum Response Code represents a floating-point value as follows:

```

  0 1 2 3 4 5 6 7 8 9 A B C D E F
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| 1 | exp |               mant                |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

$$\text{Maximum Response Delay} = (\text{mant} \mid 0x1000) \ll (\text{exp}+3)$$

Small values of Maximum Response Delay allow MLDv2 routers to tune the "leave latency" (the time between the moment the last node on a link ceases to listen to a specific multicast address and the moment the routing protocol is notified that there are no more listeners for that address). Larger values, especially in the exponential range, allow the tuning of the burstiness of MLD traffic on a link.

5.1.4. Reserved

Initialized to zero by the sender; ignored by receivers.

5.1.5. Multicast Address

For a General Query, the Multicast Address field is set to zero. For a Multicast Address Specific Query or Multicast Address and Source

Specific Query, it is set to the multicast address being queried (see Section 5.1.10, below).

5.1.6. Resv

Initialized to zero by the sender; ignored by receivers.

5.1.7. S Flag (Suppress Router-Side Processing)

When set to one, the S Flag indicates to any receiving multicast routers that they have to suppress the normal timer updates they perform upon hearing a Query. Nevertheless, it does not suppress the querier election or the normal "host-side" processing of a Query that a router may be required to perform as a consequence of itself being a multicast listener.

5.1.8. QRV (Querier's Robustness Variable)

If non-zero, the QRV field contains the [Robustness Variable] value used by the Querier. If the Querier's [Robustness Variable] exceeds 7 (the maximum value of the QRV field), the QRV field is set to zero.

Routers adopt the QRV value from the most recently received Query as their own [Robustness Variable] value, unless that most recently received QRV was zero, in which case they use the default [Robustness Variable] value specified in Section 9.1, or a statically configured value.

5.1.9. QQIC (Querier's Query Interval Code)

The Querier's Query Interval Code field specifies the [Query Interval] used by the Querier. The actual interval, called the Querier's Query Interval (QQI), is represented in units of seconds, and is derived from the Querier's Query Interval Code as follows:

If $QQIC < 128$, $QQI = QQIC$

If $QQIC \geq 128$, QQIC represents a floating-point value as follows:

```

  0 1 2 3 4 5 6 7
+---+---+---+---+
| 1 | exp | mant  |
+---+---+---+---+
```

$$QQI = (\text{mant} \mid 0x10) \ll (\text{exp} + 3)$$

Multicast routers that are not the current Querier adopt the QQI value from the most recently received Query as their own [Query

Interval] value, unless that most recently received QQI was zero, in which case the receiving routers use the default [Query Interval] value specified in Section 9.2.

5.1.10. Number of Sources (N)

The Number of Sources (N) field specifies how many source addresses are present in the Query. This number is zero in a General Query or a Multicast Address Specific Query, and non-zero in a Multicast Address and Source Specific Query. This number is limited by the MTU of the link over which the Query is transmitted. For example, on an Ethernet link with an MTU of 1500 octets, the IPv6 header (40 octets) together with the Hop-By-Hop Extension Header (8 octets) that includes the Router Alert MLD option consume 48 octets; the MLD fields up to the Number of Sources (N) field consume 28 octets; thus, there are 1424 octets left for source addresses, which limits the number of source addresses to 89 (1424/16).

5.1.11. Source Address [i]

The Source Address [i] fields are a vector of n unicast addresses, where n is the value in the Number of Sources (N) field.

5.1.12. Additional Data

If the Payload Length field in the IPv6 header of a received Query indicates that there are additional octets of data present, beyond the fields described here, MLDv2 implementations MUST include those octets in the computation to verify the received MLD Checksum, but MUST otherwise ignore those additional octets. When sending a Query, an MLDv2 implementation MUST NOT include additional octets beyond the fields described above.

5.1.13. Query Variants

There are three variants of the Query message:

A "General Query" is sent by the Querier to learn which multicast addresses have listeners on an attached link. In a General Query, both the Multicast Address field and the Number of Sources (N) field are zero.

A "Multicast Address Specific Query" is sent by the Querier to learn if a particular multicast address has any listeners on an attached link. In a Multicast Address Specific Query, the Multicast Address field contains the multicast address of interest, while the Number of Sources (N) field is set to zero.

A "Multicast Address and Source Specific Query" is sent by the Querier to learn if any of the sources from the specified list for the particular multicast address has any listeners on an attached link or not. In a Multicast Address and Source Specific Query the Multicast Address field contains the multicast address of interest, while the Source Address [i] field(s) contain(s) the source address(es) of interest.

5.1.14. Source Addresses for Queries

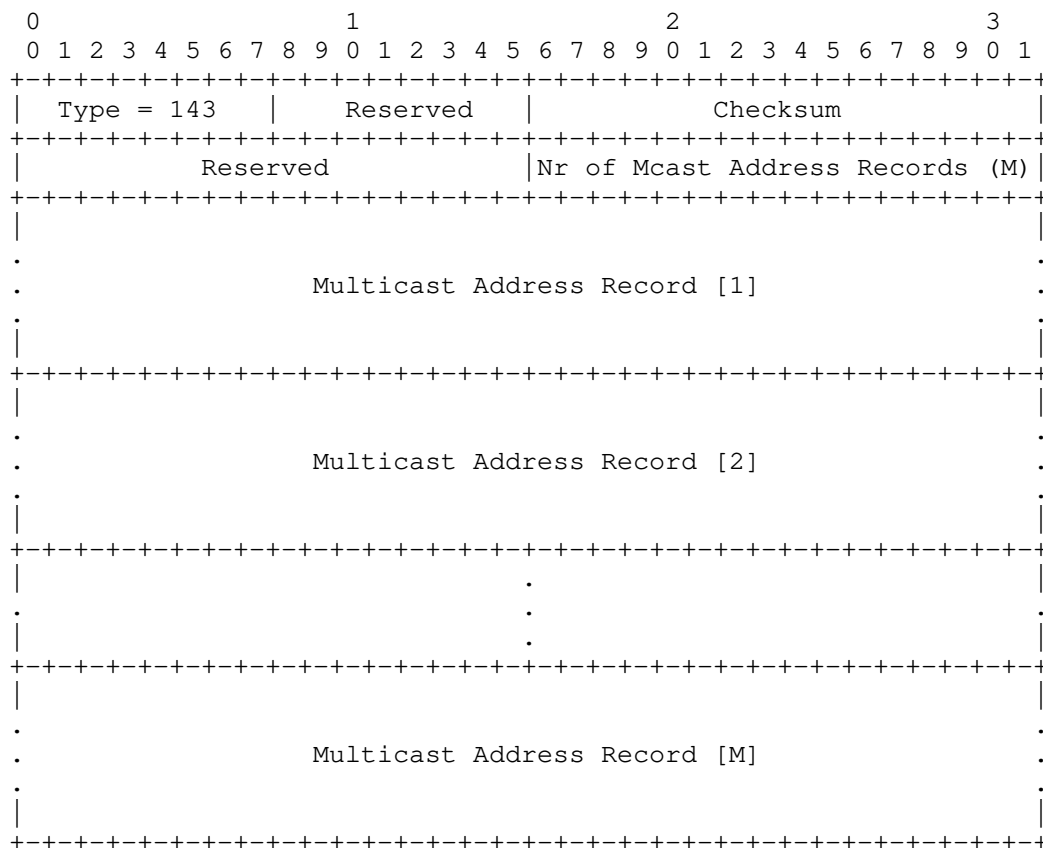
All MLDv2 Queries MUST be sent with a valid IPv6 link-local source address. If a node (router or host) receives a Query message with the IPv6 Source Address set to the unspecified address (::), or any other address that is not a valid IPv6 link-local address, it MUST silently discard the message and SHOULD log a warning.

5.1.15. Destination Addresses for Queries

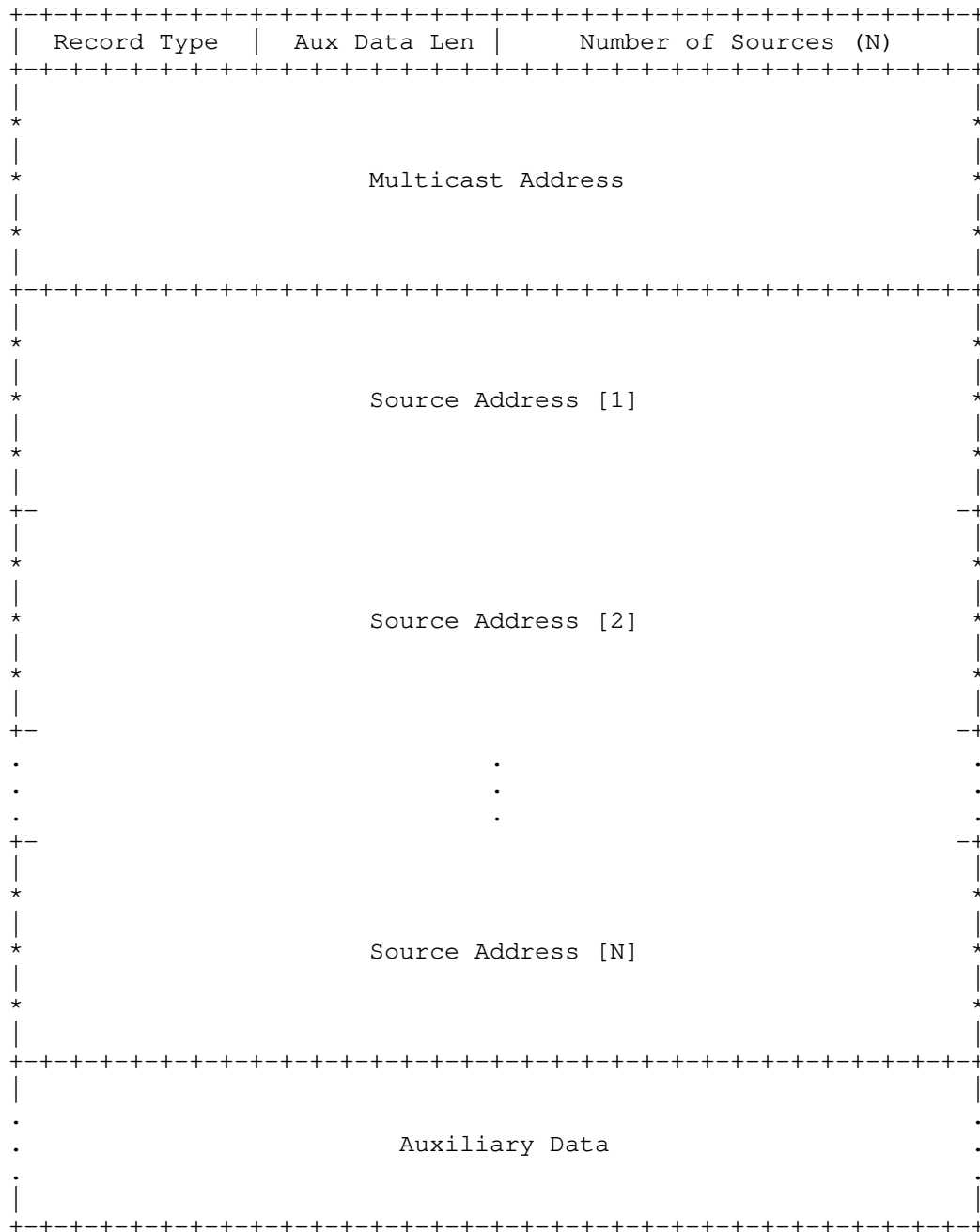
In MLDv2, General Queries are sent to the link-scope all-nodes multicast address (FF02::1). Multicast Address Specific and Multicast Address and Source Specific Queries are sent with an IP destination address equal to the multicast address of interest. However, a node MUST accept and process any Query whose IP Destination Address field contains any of the addresses (unicast or multicast) assigned to the interface on which the Query arrives. This might be useful, e.g., for debugging purposes.

5.2. Version 2 Multicast Listener Report Message

Version 2 Multicast Listener Reports are sent by IP nodes to report (to neighboring routers) the current multicast listening state, or changes in the multicast listening state, of their interfaces. Reports have the following format:



Each Multicast Address Record has the following internal format:



5.2.1. Reserved

The Reserved fields are set to zero on transmission, and ignored on reception.

5.2.2. Checksum

The standard ICMPv6 checksum; it covers the entire MLDv2 message, plus a "pseudo-header" of IPv6 header fields [RFC2460][RFC2463]. In order to compute the checksum, the Checksum field is set to zero. When a packet is received, the checksum MUST be verified before processing it.

5.2.3. Nr of Mcast Address Records (M)

The Nr of Mcast Address Records (M) field specifies how many Multicast Address Records are present in this Report.

5.2.4. Multicast Address Record

Each Multicast Address Record is a block of fields that contain information on the sender listening to a single multicast address on the interface from which the Report is sent.

5.2.5. Record Type

It specifies the type of the Multicast Address Record. See Section 5.2.12 for a detailed description of the different possible Record Types.

5.2.6. Aux Data Len

The Aux Data Len field contains the length of the Auxiliary Data Field in this Multicast Address Record, in units of 32-bit words. It may contain zero, to indicate the absence of any auxiliary data.

5.2.7. Number of Sources (N)

The Number of Sources (N) field specifies how many source addresses are present in this Multicast Address Record.

5.2.8. Multicast Address

The Multicast Address field contains the multicast address to which this Multicast Address Record pertains.

5.2.9. Source Address [i]

The Source Address [i] fields are a vector of n unicast addresses, where n is the value in this record's Number of Sources (N) field.

5.2.10. Auxiliary Data

The Auxiliary Data field, if present, contains additional information that pertain to this Multicast Address Record. The protocol specified in this document, MLDv2, does not define any auxiliary data. Therefore, implementations of MLDv2 MUST NOT include any auxiliary data (i.e., MUST set the Aux Data Len field to zero) in any transmitted Multicast Address Record, and MUST ignore any such data present in any received Multicast Address Record. The semantics and the internal encoding of the Auxiliary Data field are to be defined by any future version or extension of MLD that uses this field.

5.2.11. Additional Data

If the Payload Length field in the IPv6 header of a received Report indicates that there are additional octets of data present, beyond the last Multicast Address Record, MLDv2 implementations MUST include those octets in the computation to verify the received MLD Checksum, but MUST otherwise ignore those additional octets. When sending a Report, an MLDv2 implementation MUST NOT include additional octets beyond the last Multicast Address Record.

5.2.12. Multicast Address Record Types

There are a number of different types of Multicast Address Records that may be included in a Report message:

A "Current State Record" is sent by a node in response to a Query received on an interface. It reports the current listening state of that interface, with respect to a single multicast address. The Record Type of a Current State Record may be one of the following two values:

- 1 - MODE_IS_INCLUDE - indicates that the interface has a filter mode of INCLUDE for the specified multicast address. The Source Address [i] fields in this Multicast Address Record contain the interface's source list for the specified multicast address. A MODE_IS_INCLUDE Record is never sent with an empty source list.
- 2 - MODE_IS_EXCLUDE - indicates that the interface has a filter mode of EXCLUDE for the specified multicast address. The Source Address [i] fields in this Multicast Address Record

contain the interface's source list for the specified multicast address, if it is non-empty.

A "Filter Mode Change Record" is sent by a node whenever a local invocation of IPv6MulticastListen causes a change of the filter mode (i.e., a change from INCLUDE to EXCLUDE, or from EXCLUDE to INCLUDE) of the interface-level state entry for a particular multicast address, whether the source list changes at the same time or not. The Record is included in a Report sent from the interface on which the change occurred. The Record Type of a Filter Mode Change Record may be one of the following two values:

- 3 - CHANGE_TO_INCLUDE_MODE - indicates that the interface has changed to INCLUDE filter mode for the specified multicast address. The Source Address [i] fields in this Multicast Address Record contain the interface's new source list for the specified multicast address, if it is non-empty.
- 4 - CHANGE_TO_EXCLUDE_MODE - indicates that the interface has changed to EXCLUDE filter mode for the specified multicast address. The Source Address [i] fields in this Multicast Address Record contain the interface's new source list for the specified multicast address, if it is non-empty.

A "Source List Change Record" is sent by a node whenever a local invocation of IPv6MulticastListen causes a change of source list that is not coincident with a change of filter mode, of the interface-level state entry for a particular multicast address. The Record is included in a Report sent from the interface on which the change occurred. The Record Type of a Source List Change Record may be one of the following two values:

- 5 - ALLOW_NEW_SOURCES - indicates that the Source Address [i] fields in this Multicast Address Record contain a list of the additional sources that the node wishes to listen to, for packets sent to the specified multicast address. If the change was to an INCLUDE source list, these are the addresses that were added to the list; if the change was to an EXCLUDE source list, these are the addresses that were deleted from the list.
- 6 - BLOCK_OLD_SOURCES - indicates that the Source Address [i] fields in this Multicast Address Record contain a list of the sources that the node no longer wishes to listen to, for packets sent to the specified multicast address. If the change was to an INCLUDE source list, these are the addresses that were deleted from the list; if the change was

to an EXCLUDE source list, these are the addresses that were added to the list.

If a change of source list results in both allowing new sources and blocking old sources, then two Multicast Address Records are sent for the same multicast address, one of type ALLOW_NEW_SOURCES and one of type BLOCK_OLD_SOURCES.

We use the term "State Change Record" to refer to either a Filter Mode Change Record or a Source List Change Record.

Multicast Address Records with an unrecognized Record Type value MUST be silently ignored, with the rest of the report being processed.

In the rest of this document, we use the following notation to describe the contents of a Multicast Address Record that pertains to a particular multicast address:

- IS_IN (x) - Type MODE_IS_INCLUDE, source addresses x
- IS_EX (x) - Type MODE_IS_EXCLUDE, source addresses x
- TO_IN (x) - Type CHANGE_TO_INCLUDE_MODE, source addresses x
- TO_EX (x) - Type CHANGE_TO_EXCLUDE_MODE, source addresses x
- ALLOW (x) - Type ALLOW_NEW_SOURCES, source addresses x
- BLOCK (x) - Type BLOCK_OLD_SOURCES, source addresses x

where x is either:

a capital letter (e.g., "A") to represent the set of source addresses, or

a set expression (e.g., "A+B"), where "A+B" means the union of sets A and B, "A*B" means the intersection of sets A and B, and "A-B" means the removal of all elements of set B from set A.

5.2.13. Source Addresses for Reports

An MLDv2 Report MUST be sent with a valid IPv6 link-local source address, or the unspecified address (::), if the sending interface has not acquired a valid link-local address yet. Sending reports with the unspecified address is allowed to support the use of IP multicast in the Neighbor Discovery Protocol [RFC2461]. For stateless autoconfiguration, as defined in [RFC2462], a node is required to join several IPv6 multicast groups, in order to perform Duplicate Address Detection (DAD). Prior to DAD, the only address the reporting node has for the sending interface is a tentative one, which cannot be used for communication. Thus, the unspecified address must be used.

On the other hand, routers MUST silently discard a message that is not sent with a valid link-local address, without taking any action on the contents of the packet. Thus, a Report is discarded if the router cannot identify the source address of the packet as belonging to a link connected to the interface on which the packet was received. A Report sent with the unspecified address is also discarded by the router. This enhances security, as unidentified reporting nodes cannot influence the state of the MLDv2 router(s). Nevertheless, the reporting node has modified its listening state for multicast addresses that are contained in the Multicast Address Records of the Report message. From now on, it will treat packets sent to those multicast addresses according to this new listening state. Once a valid link-local address is available, a node SHOULD generate new MLDv2 Report messages for all multicast addresses joined on the interface.

5.2.14. Destination Addresses for Reports

Version 2 Multicast Listener Reports are sent with an IP destination address of FF02:0:0:0:0:0:0:16, to which all MLDv2-capable multicast routers listen (see Section 11 for IANA considerations related to this special destination address). A node that operates in version 1 compatibility mode (see details in Section 8) sends version 1 Reports to the multicast address specified in the Multicast Address field of the Report. In addition, a node MUST accept and process any version 1 Report whose IP Destination Address field contains any of the IPv6 addresses (unicast or multicast) assigned to the interface on which the Report arrives. This might be useful, e.g., for debugging purposes.

5.2.15. Multicast Listener Report Size

If the set of Multicast Address Records required in a Report does not fit within the size limit of a single Report message (as determined by the MTU of the link on which it will be sent), the Multicast Address Records are sent in as many Report messages as needed to report the entire set.

If a single Multicast Address Record contains so many source addresses that it does not fit within the size limit of a single Report message, then:

- if its Type is not IS_EX or TO_EX, it is split into multiple Multicast Address Records; each such record contains a different subset of the source addresses, and is sent in a separate Report.

- if its Type is IS_EX or TO_EX, a single Multicast Address Record is sent, with as many source addresses as can fit; the remaining

source addresses are not reported. Although the choice of which sources to report is arbitrary, it is preferable to report the same set of sources in each subsequent report, rather than reporting different sources each time.

6. Protocol Description for Multicast Address Listeners

MLD is an asymmetric protocol, as it specifies separate behaviors for multicast address listeners -- that is, hosts or routers that listen to multicast packets -- and multicast routers. This section describes the part of MLDv2 that applies to all multicast address listeners. (Note that a multicast router that is also a multicast address listener performs both parts of MLDv2; it receives and it responds to its own MLD messages, as well as to those of its neighbors.) The multicast router part of MLDv2 is described in Section 7.

A node performs the protocol described in this section over all interfaces on which multicast reception is supported, even if more than one of those interfaces are connected to the same link.

For interoperability with multicast routers that run the MLDv1 protocol, nodes maintain a Host Compatibility Mode variable for each interface on which multicast reception is supported. This section describes the behavior of multicast address listener nodes on interfaces for which Host Compatibility Mode = MLDv2. The algorithm for determining Host Compatibility Mode, and the behavior if its value is set to MLDv1, are described in Section 8.

The link-scope all-nodes multicast address, (FF02::1), is handled as a special case. On all nodes -- that is all hosts and routers, including multicast routers -- listening to packets destined to the all-nodes multicast address, from all sources, is permanently enabled on all interfaces on which multicast listening is supported. No MLD messages are ever sent regarding neither the link-scope all-nodes multicast address, nor any multicast address of scope 0 (reserved) or 1 (node-local). Multicast listeners MUST send MLD messages for all multicast addresses except for the link-scope all-nodes multicast address and any multicast addresses of scope less than 2.

There are three types of events that trigger MLDv2 protocol actions on an interface:

- a change of the per-interface listening state, caused by a local invocation of IPv6MulticastListen;

- the firing of a specific timer;

the reception of a Query.

(Received MLD messages of types other than Query are silently ignored, except as required for interoperation with nodes that implement MLDv1.)

The following subsections describe the actions to be taken for each case. Timer and counter names appear in square brackets. Default values for those timers and counters are specified in Section 9.

6.1. Action on Change of Per-Interface State

An invocation of IPv6MulticastListen may cause the multicast listening state of an interface to change, according to the rules in Section 4.2. Each such change affects the per-interface entry for a single multicast address.

A change of per-interface state causes the node to immediately transmit a State Change Report from that interface. The type and contents of the Multicast Address Record(s) in that Report are determined by comparing the filter mode and source list for the affected multicast address before and after the change, according to the table below. If no per-interface state existed for that multicast address before the change (i.e., the change consisted of creating a new per-interface record), or if no state exists after the change (i.e., the change consisted of deleting a per-interface record), then the "non-existent" state is considered to have an INCLUDE filter mode and an empty source list.

| Old State | New State | State Change Record Sent |
|-------------|-------------|--------------------------|
| INCLUDE (A) | INCLUDE (B) | ALLOW (B-A), BLOCK (A-B) |
| EXCLUDE (A) | EXCLUDE (B) | ALLOW (A-B), BLOCK (B-A) |
| INCLUDE (A) | EXCLUDE (B) | TO_EX (B) |
| EXCLUDE (A) | INCLUDE (B) | TO_IN (B) |

If the computed source list for either an ALLOW or a BLOCK State Change Record is empty, that record is omitted from the Report.

To cover the possibility of the State Change Report being missed by one or more multicast routers, [Robustness Variable] - 1 retransmissions are scheduled, through a Retransmission Timer, at intervals chosen at random from the range (0, [Unsolicited Report Interval]).

If more changes to the same per-interface state entry occur before all the retransmissions of the State Change Report for the first change have been completed, each such additional change triggers the immediate transmission of a new State Change Report.

The contents of the new Report are calculated as follows:

As for the first Report, the per-interface state for the affected multicast address before and after the latest change is compared.

The records that express the difference are built according to the table above. Nevertheless, these records are not transmitted in a separate message, but they are instead merged with the contents of the pending report, to create the new State Change Report. The rules for calculating this merged report are described below.

The transmission of the merged State Change Report terminates retransmissions of the earlier State Change Reports for the same multicast address, and becomes the first of [Robustness Variable] transmissions of the new State Change Reports. These transmissions are necessary in order to ensure that each instance of state change is transmitted at least [Robustness Variable] times.

Each time a source is included in the difference report calculated above, retransmission state for that source needs to be maintained until [Robustness Variable] State Change Reports have been sent by the node. This is done in order to ensure that a series of successive state changes do not break the protocol robustness. Sources in retransmission state can be kept in a per multicast address Retransmission List, with a Source Retransmission Counter associated to each source in the list. When a source is included in the list, its counter is set to [Robustness Variable]. Each time a State Change Report is sent the counter is decreased by one unit. When the counter reaches zero, the source is deleted from the Retransmission List for that multicast address.

If the per-interface listening change that triggers the new report is a filter mode change, then the next [Robustness Variable] State Change Reports will include a Filter Mode Change Record. This applies even if any number of source list changes occur in that period. The node has to maintain retransmission state for the multicast address until the [Robustness Variable] State Change Reports have been sent. This can be done through a per multicast address Filter Mode Retransmission Counter. When the filter mode changes, the counter is set to [Robustness Variable]. Each time a State Change Report is sent the counter is decreased by one unit. When the counter reaches zero, i.e., [Robustness Variable] State Change Reports with Filter Mode Change Records have been transmitted

after the last filter mode change, and if source list changes have resulted in additional reports being scheduled, then the next State Change Report will include Source List Change Records.

Each time a per-interface listening state change triggers the Immediate transmission of a new State Change Report, its contents are determined as follows. If the report should contain a Filter Mode Change Record, i.e., the Filter Mode Retransmission Counter for that multicast address has a value higher than zero, then, if the current filter mode of the interface is INCLUDE, a TO_IN record is included in the report; otherwise a TO_EX record is included. If instead the report should contain Source List Change Records, i.e., the Filter Mode Retransmission Counter for that multicast address is zero, an ALLOW and a BLOCK record is included. The contents of these records are built according to the table below.

| Record | Sources Included |
|--------|---|
| TO_IN | All in the current per-interface state that must be forwarded |
| TO_EX | All in the current per-interface state that must be blocked |
| ALLOW | All with retransmission state (i.e., all sources from the Retransmission List) that must be forwarded |
| BLOCK | All with retransmission state that must be blocked |

If the computed source list for either an ALLOW or a BLOCK record is empty, that record is omitted from the State Change Report.

Note: When the first State Change Report is sent, the non-existent pending report to merge with can be treated as a Source Change Report with empty ALLOW and BLOCK records (no sources have retransmission state).

The building of a scheduled State Change Report, triggered by the firing of a Retransmission Timer, instead of a per-interface listening state change, is described in Section 6.3.

6.2. Action on Reception of a Query

Upon reception of an MLD message that contains a Query, the node checks if the source address of the message is a valid link-local address, if the Hop Limit is set to 1, and if the Router Alert option is present in the Hop-By-Hop Options header of the IPv6 packet. If any of these checks fails, the packet is dropped.

If the validity of the MLD message is verified, the node starts to process the Query. Instead of responding immediately, the node delays its response by a random amount of time, bounded by the Maximum Response Delay value derived from the Maximum Response Code in the received Query message. A node may receive a variety of Queries on different interfaces and of different kinds (e.g., General Queries, Multicast Address Specific Queries, and Multicast Address and Source Specific Queries), each of which may require its own delayed response.

Before scheduling a response to a Query, the node must first consider previously scheduled pending responses and, in many cases, schedule a combined response. Therefore, for each of its interfaces on which it operates the listener part of the MLDv2 protocol, the node must be able to maintain the following state:

- an Interface Timer for scheduling responses to General Queries;

- a Multicast Address Timer for scheduling responses to Multicast Address (and Source) Specific Queries, for each multicast address the node has to report on;

- a per-multicast-address list of sources to be reported in response to a Multicast Address and Source Specific Query.

When a new valid General Query arrives on an interface, the node checks whether it has any per-interface listening state record to report on, or not. Similarly, when a new valid Multicast Address (and Source) Specific Query arrives on an interface, the node checks whether it has a per-interface listening state record that corresponds to the queried multicast address (and source), or not. If it does, a delay for a response is randomly selected in the range (0, [Maximum Response Delay]), where Maximum Response Delay is derived from the Maximum Response Code inserted in the received Query message. The following rules are then used to determine if a Report needs to be scheduled or not, and the type of Report to schedule. (The rules are considered in order and only the first matching rule is applied.)

1. If there is a pending response to a previous General Query scheduled sooner than the selected delay, no additional response needs to be scheduled.
2. If the received Query is a General Query, the Interface Timer is used to schedule a response to the General Query after the selected delay. Any previously pending response to a General Query is canceled.

3. If the received Query is a Multicast Address Specific Query or a Multicast Address and Source Specific Query and there is no pending response to a previous Query for this multicast address, then the Multicast Address Timer is used to schedule a report. If the received Query is a Multicast Address and Source Specific Query, the list of queried sources is recorded to be used when generating a response.
4. If there is already a pending response to a previous Query scheduled for this multicast address, and either the new Query is a Multicast Address Specific Query or the recorded source list associated with the multicast address is empty, then the multicast address source list is cleared and a single response is scheduled, using the Multicast Address Timer. The new response is scheduled to be sent at the earliest of the remaining time for the pending report and the selected delay.
5. If the received Query is a Multicast Address and Source Specific Query and there is a pending response for this multicast address with a non-empty source list, then the multicast address source list is augmented to contain the list of sources in the new Query, and a single response is scheduled using the Multicast Address Timer. The new response is scheduled to be sent at the earliest of the remaining time for the pending report and the selected delay.

6.3. Action on Timer Expiration

There are several timers that, upon expiration, trigger protocol actions on an MLDv2 Multicast Address Listener node. All these actions are related to pending reports scheduled by the node.

1. If the expired timer is the Interface Timer (i.e., there is a pending response to a General Query), then one Current State Record is sent for each multicast address for which the specified interface has listening state, as described in Section 4.2. The Current State Record carries the multicast address and its associated filter mode (MODE_IS_INCLUDE or MODE_IS_EXCLUDE) and Source list. Multiple Current State Records are packed into individual Report messages, to the extent possible.

This naive algorithm may result in bursts of packets when a node listens to a large number of multicast addresses. Instead of using a single Interface Timer, implementations are recommended to spread transmission of such Report messages over the interval (0, [Maximum Response Delay]). Note that any such implementation MUST avoid the "ack-implosion" problem, i.e., MUST NOT send a Report immediately upon reception of a General Query.

2. If the expired timer is a Multicast Address Timer and the list of recorded sources for that multicast address is empty (i.e., there is a pending response to a Multicast Address Specific Query), then if, and only if, the interface has listening state for that multicast address, a single Current State Record is sent for that address. The Current State Record carries the multicast address and its associated filter mode (MODE_IS_INCLUDE or MODE_IS_EXCLUDE) and source list, if any.
3. If the expired timer is a Multicast Address Timer and the list of recorded sources for that multicast address is non-empty (i.e., there is a pending response to a Multicast Address and Source Specific Query), then if, and only if, the interface has listening state for that multicast address, the contents of the corresponding Current State Record are determined from the per-interface state and the pending response record, as specified in the following table:

| per-interface state | set of sources in the pending response record | Current State Record |
|---------------------|---|----------------------|
| INCLUDE (A) | B | IS_IN (A*B) |
| EXCLUDE (A) | B | IS_IN (B-A) |

If the resulting Current State Record has an empty set of source addresses, then no response is sent. After the required Report messages have been generated, the source lists associated with any reported multicast addresses are cleared.

4. If the expired timer is a Retransmission Timer for a multicast address (i.e., there is a pending State Change Report for that multicast address), the contents of the report are determined as follows. If the report should contain a Filter Mode Change Record, i.e., the Filter Mode Retransmission Counter for that multicast address has a value higher than zero, then, if the current filter mode of the interface is INCLUDE, a TO_IN record is included in the report; otherwise a TO_EX record is included. In both cases, the Filter Mode Retransmission Counter for that multicast address is decremented by one unit after the transmission of the report.

If instead the report should contain Source List Change Records, i.e., the Filter Mode Retransmission Counter for that multicast address is zero, an ALLOW and a BLOCK record is included. The contents of these records are built according to the table below:

| Record | Sources included |
|--------|---|
| TO_IN | All in the current per-interface state that must be forwarded |
| TO_EX | All in the current per-interface state that must be blocked |
| ALLOW | All with retransmission state (i.e., all sources from the Retransmission List) that must be forwarded. For each included source, its Source Retransmission Counter is decreased with one unit after the transmission of the report. If the counter reaches zero, the source is deleted from the Retransmission List for that multicast address. |
| BLOCK | All with retransmission state (i.e., all sources from the Retransmission List) that must be blocked. For each included source, its Source Retransmission Counter is decreased with one unit after the transmission of the report. If the counter reaches zero, the source is deleted from the Retransmission List for that multicast address. |

If the computed source list for either an ALLOW or a BLOCK record is empty, that record is omitted from the State Change Report.

7. Description of the Protocol for Multicast Routers

The purpose of MLD is to enable each multicast router to learn, for each of its directly attached links, which multicast addresses have listeners on that link. MLD version 2 adds the capability for a multicast router to also learn which sources have listeners among the neighboring nodes, for packets sent to any particular multicast address. The information gathered by MLD is provided to whichever multicast routing protocol is used by the router, in order to ensure that multicast packets are delivered to all links where there are interested listeners.

This section describes the part of MLDv2 that is performed by multicast routers. Multicast routers may themselves become multicast address listeners, and therefore also perform the multicast listener part of MLDv2, described in Section 6.

A multicast router performs the protocol described in this section over each of its directly attached links. If a multicast router has

more than one interface to the same link, it only needs to operate this protocol over one of those interfaces.

For each interface over which the router operates the MLD protocol, the router must configure that interface to listen to all link-layer multicast addresses that can be generated by IPv6 multicasts. For example, an Ethernet-attached router must set its Ethernet address reception filter to accept all Ethernet multicast addresses that start with the hexadecimal value 3333 [RFC2464]; in the case of an Ethernet interface that does not support the filtering of such a multicast address range, it must be configured to accept ALL Ethernet multicast addresses, in order to meet the requirements of MLD.

On each interface over which this protocol is being run, the router MUST enable reception of the link-scope "all MLDv2-capable routers" multicast address from all sources, and MUST perform the multicast address listener part of MLDv2 for that address on that interface.

Multicast routers only need to know that at least one node on an attached link listens to packets for a particular multicast address from a particular source; a multicast router is not required to individually keep track of the interests of each neighboring node. (Nevertheless, see Appendix A.2 item 1 for discussion.)

MLDv2 is backward compatible with the MLDv1 protocol. For a detailed description of compatibility issues see Section 8.

7.1. Conditions for MLD Queries

The behavior of a router that implements the MLDv2 protocol depends on whether there are several multicast routers on the same subnet, or not. If it is the case, a querier election mechanism (described in Section 7.6.2) is used to elect a single multicast router to be in Querier state. All the multicast routers on the subnet listen to the messages sent by multicast address listeners, and maintain the same multicast listening information state, so that they can quickly and correctly take over the querier functionality, should the present Querier fail. Nevertheless, it is only the Querier that sends periodical or triggered query messages on the subnet.

The Querier periodically sends General Queries to request Multicast Address Listener information from an attached link. These queries are used to build and refresh the Multicast Address Listener state of routers on attached links.

Nodes respond to these queries by reporting their Multicast Address Listening state (and set of sources they listen to) with Current State Multicast Address Records in MLDv2 Multicast Listener Reports.

As a listener of a multicast address, a node may express interest in listening or not listening to traffic from particular sources. As the desired listening state of a node changes, it reports these changes using Filter Mode Change Records or Source List Change Records. These records indicate an explicit state change in a multicast address at a node in either the Multicast Address Record's source list or its filter mode. When Multicast Address Listening is terminated at a node or traffic from a particular source is no longer desired, the Querier must query for other listeners of the multicast address or of the source before deleting the multicast address (or source) from its Multicast Address Listener state and pruning its traffic.

To enable all nodes on a link to respond to changes in multicast address listening, the Querier sends specific queries. A Multicast Address Specific Query is sent to verify that there are no nodes that listen to the specified multicast address or to "rebuild" the listening state for a particular multicast address. Multicast Address Specific Queries are sent when the Querier receives a State Change Record indicating that a node ceases to listen to a multicast address. They are also sent in order to enable a fast transition of a router from EXCLUDE to INCLUDE mode, in case a received State Change Record motivates this action.

A Multicast Address and Source Specific Query is used to verify that there are no nodes on a link which listen to traffic from a specific set of sources. Multicast Address and Source Specific Queries list sources for a particular multicast address which have been requested to no longer be forwarded. This query is sent by the Querier in order to learn if any node listens to packets sent to the specified multicast address, from the specified source addresses. Multicast Address and Source Specific Queries are only sent in response to State Change Records and never in response to Current State Records. Section 5.1.13 describes each query in more detail.

7.2. MLD State Maintained by Multicast Routers

Multicast routers that implement the MLDv2 protocol keep state per multicast address per attached link. This multicast address state consists of a filter mode, a list of sources, and various timers. For each attached link on which MLD runs, a multicast router records the listening state for that link. That state conceptually consists of a set of records of the form:

```
(IPv6 multicast address, Filter Timer,  
 Router Filter Mode, (source records) )
```

Each source record is of the form:

(IPv6 source address, source timer)

If all sources for a multicast address are listened to, an empty source record list is kept with the Router Filter Mode set to EXCLUDE. This means that nodes on this link want all sources for this multicast address to be forwarded. This is the MLDv2 equivalent of an MLDv1 listening state.

7.2.1. Definition of Router Filter Mode

To reduce internal state, MLDv2 routers keep a filter mode per multicast address per attached link. This filter mode is used to summarize the total listening state of a multicast address to a minimum set such that all nodes' listening states are respected. The filter mode may change in response to the reception of particular types of Multicast Address Records or when certain timer conditions occur. In the following sections, we use the term "Router Filter Mode" to refer to the filter mode of a particular multicast address within a router. Section 7.4 describes the changes of the Router Filter Mode per Multicast Address Record received.

A router is in INCLUDE mode for a specific multicast address on a given interface if all the listeners on the link interested in that address are in INCLUDE mode. The router state is represented through the notation INCLUDE (A), where A is called the "Include List". The Include List is the set of sources that one or more listeners on the link have requested to receive. All the sources from the Include List will be forwarded by the router. Any other source that is not in the Include List will be blocked by the router.

A router is in EXCLUDE mode for a specific multicast address on a given interface if there is at least one listener in EXCLUDE mode interested in that address on the link. Conceptually, when a Multicast Address Record is received, the Router Filter Mode for that multicast address is updated to cover all the requested sources using the least amount of state. As a rule, once a Multicast Address Record with a filter mode of EXCLUDE is received, the Router Filter Mode for that multicast address will be set to EXCLUDE. Nevertheless, if all nodes with a multicast address record having filter mode set to EXCLUDE cease reporting, it is desirable for the Router Filter Mode for that multicast address to transition back to INCLUDE mode. This transition occurs when the Filter Timer expires, and is explained in detail in Section 7.5.

When the router is in EXCLUDE mode, the router state is represented through the notation EXCLUDE (X,Y), where X is called the "Requested List" and Y is called the "Exclude List". All sources, except those from the Exclude List, will be forwarded by the router. The

Requested List has no effect on forwarding. Nevertheless, it has to be maintained for several reasons, as explained in Section 7.2.3.

The exact handling of both the INCLUDE and EXCLUDE mode router state, according to the received reports, is presented in details in Tables 7.4.1 and 7.4.2.

7.2.2. Definition of Filter Timers

The Filter Timer is only used when the router is in EXCLUDE mode for a specific multicast address, and it represents the time for the Router Filter Mode of the multicast address to expire and switch to INCLUDE mode. A Filter Timer is a decrementing timer with a lower bound of zero. One Filter Timer exists per multicast address record. Filter Timers are updated according to the types of Multicast Address Records received.

If a Filter Timer expires, with the Router Filter Mode for that multicast address being EXCLUDE, it means that there are no more listeners in EXCLUDE mode on the attached link. At this point, the router transitions to INCLUDE filter mode. Section 7.5 describes the actions taken when a Filter Timer expires while in EXCLUDE mode.

The following table summarizes the role of the Filter Timer. Section 7.4 describes the details of setting the Filter Timer per type of Multicast Address Record received.

| Router Filter Mode | Filter Timer Value | Actions/Comments |
|--------------------|--------------------|--|
| INCLUDE | Not Used | All listeners in INCLUDE mode. |
| EXCLUDE | Timer > 0 | At least one listener in EXCLUDE mode. |
| EXCLUDE | Timer == 0 | No more listeners in EXCLUDE mode for the multicast address. If the Requested List is empty, delete Multicast Address Record. If not, switch to INCLUDE filter mode; the sources in the Requested List are moved to the Include List, and the Exclude List is deleted. |

7.2.3. Definition of Source Timers

A Source Timer is a decrementing timer with a lower bound of zero. One Source Timer is kept per source record. Source timers are updated according to the type and filter mode of the Multicast Address Record received. Section 7.4 describes the setting of source timers per type of Multicast Address Records received.

In the following, abbreviations are used for several variables (all of which are described in detail in Section 9). The variable MALI stands for the Multicast Address Listening Interval, which is the time in which multicast address listening state will time out. The variable LLQT is the Last Listener Query Time, which is the total time the router should wait for a report, after the Querier has sent the first query. During this time, the Querier should send [Last Member Query Count]-1 retransmissions of the query. LLQT represents the "leave latency", or the difference between the transmission of a listener state change and the modification of the information passed to the routing protocol.

If the router is in INCLUDE filter mode, a source can be added to the current Include List if a listener in INCLUDE mode sends a Current State or a State Change Report which includes that source. Each source from the Include List is associated with a source timer that is updated whenever a listener in INCLUDE mode sends a report that confirms its interest in that specific source. If the timer of a source from the Include List expires, the source is deleted from the Include List. If there are no more source records left, the multicast address record is deleted from the router.

Besides this "soft leave" mechanism, there is also a "fast leave" scheme in MLDv2; it is also based on the use of source timers. When a node in INCLUDE mode expresses its desire to stop listening to a specific source, all the multicast routers on the link lower their timer for that source to a small interval of LLQT milliseconds. The Querier then sends then a Multicast Address and Source Specific Query, to verify whether there are other listeners for that source on the link, or not. If a corresponding report is received before the timer expires, all the multicast routers on the link update their source timer. If not, the source is deleted from the Include List. The handling of the Include List, according to the received reports, is detailed in Tables 7.4.1 and 7.4.2.

Source timers are treated differently when the Router Filter Mode for a multicast address is EXCLUDE. For sources from the Requested List the source timers have running values; these sources are forwarded by the router. For sources from the Exclude List the source timers are set to zero; these sources are blocked by the router. If the timer

of a source from the Requested List expires, the source is moved to the Exclude List. The router informs then the routing protocol that there is no longer a listener on the link interested in traffic from this source.

The router has to maintain the Requested List for two reasons:

To keep track of sources that listeners in INCLUDE mode listen to. This is necessary in order to assure a seamless transition of the router to INCLUDE mode, when there will be no listener in EXCLUDE mode left. This transition should not interrupt the flow of traffic to the listeners in INCLUDE mode still interested in that multicast address. Therefore, at the moment of the transition, the Requested List should represent the set of sources that nodes in INCLUDE mode have explicitly requested.

When the router switches to INCLUDE mode, the sources in the Requested List are moved to the Include List, and the Exclude List is deleted. Before the switch, the Requested List can contain an inexact guess at the sources that listeners in INCLUDE mode listen to - might be too large or too small. These inexactitudes are due to the fact that the Requested List is also used for fast blocking purposes, as described below. If such a fast blocking is required, some sources may be deleted from the Requested List (as shown in Tables 7.4.1 and 7.4.2) in order to reduce router state. Nevertheless, in each such case the Filter Timer is updated as well. Therefore, listeners in INCLUDE mode will have enough time, before an eventual switching, to reconfirm their interest in the eliminated source(s), and rebuild the Requested List accordingly. The protocol ensures that when a switch to INCLUDE mode occurs, the Requested List will be accurate. Details about the transition of the router to INCLUDE mode are presented in Appendix A.3.

To allow a fast blocking of previously unblocked sources. If the router receives a report that contains such a request, the concerned sources are added to the Requested List. Their timers are set to a small interval of LLQT milliseconds, and a Multicast Address and Source Specific Query is sent by the Querier, to check whether there are nodes on the link still interested in those sources, or not. If no node confirms its interest in receiving a specific source, the timer of that source expires. Then, the source is moved from the Requested List to the Exclude List. From then on, the source will be blocked by the router.

The handling of the EXCLUDE mode router state, according to the received reports, is detailed in Tables 7.4.1 and 7.4.2.

When the Router Filter Mode for a multicast address is EXCLUDE, source records are only deleted when the Filter Timer expires, or when newly received Multicast Address Records modify the source record list of the router.

7.3. MLDv2 Source Specific Forwarding Rules

When a multicast router receives a datagram from a source destined to a particular multicast address, a decision has to be made whether to forward the datagram on an attached link or not. The multicast routing protocol in use is in charge of this decision, and should use the MLDv2 information to ensure that all sources/multicast addresses that have listeners on a link are forwarded to that link. MLDv2 information does not override multicast routing information; for example, if the MLDv2 filter mode for a multicast address is EXCLUDE, a router may still forward packets for excluded sources to a transit link.

To summarize, the following table describes the forwarding suggestions made by MLDv2 to the routing protocol for traffic originating from a source destined to a multicast address. It also summarizes the actions taken upon the expiration of a source timer based on the Router Filter Mode of the multicast address.

| Router Filter Mode | Source Timer Value | Action |
|--------------------|--------------------|---|
| INCLUDE | TIMER > 0 | Suggest to forward traffic from source |
| INCLUDE | TIMER == 0 | Suggest to stop forwarding traffic from source and remove source record. If there are no more source records, delete multicast address record |
| EXCLUDE | TIMER > 0 | Suggest to forward traffic from source |
| EXCLUDE | TIMER == 0 | Suggest to not forward traffic from source. Move the source from the Requested List to the Exclude List (DO NOT remove source record) |
| EXCLUDE | No Source Element | Suggest to forward traffic from all sources |

7.4. Action on Reception of Reports

Upon reception of an MLD message that contains a Report, the router checks if the source address of the message is a valid link-local address, if the Hop Limit is set to 1, and if the Router Alert option is present in the Hop-By-Hop Options header of the IPv6 packet. If any of these checks fails, the packet is dropped. If the validity of the MLD message is verified, the router starts to process the Report.

7.4.1. Reception of Current State Records

When receiving Current State Records, a router updates both its Filter Timer and its source timers. In some circumstances, the reception of a type of multicast address record will cause the Router Filter Mode for that multicast address to change. The table below describes the actions, with respect to state and timers, that occur to a router's state upon reception of Current State Records.

If the router is in INCLUDE filter mode for a multicast address, we will use the notation INCLUDE (A), where A denotes the associated Include List. If the router is in EXCLUDE filter mode for a multicast address, we will use the notation EXCLUDE (X,Y), where X and Y denote the associated Requested List and Exclude List respectively.

Within the "Actions" section of the router state tables, we use the notation '(A)=J', which means that the set A of source records should have their source timers set to value J. 'Delete (A)' means that the set A of source records should be deleted. 'Filter Timer = J' means that the Filter Timer for the multicast address should be set to value J.

| Router State | Report Received | New Router State | Actions |
|---------------|-----------------|--------------------|---|
| INCLUDE (A) | IS_IN (B) | INCLUDE (A+B) | (B)=MALI |
| INCLUDE (A) | IS_EX (B) | EXCLUDE (A*B, B-A) | (B-A)=0 Delete (A-B) Filter Timer=MALI |
| EXCLUDE (X,Y) | IS_IN (A) | EXCLUDE (X+A, Y-A) | (A)=MALI |
| EXCLUDE (X,Y) | IS_EX (A) | EXCLUDE (A-Y, Y*A) | (A-X-Y)=MALI Delete (X-A) Delete (Y-A) Filter Timer=MALI |

7.4.2. Reception of Filter Mode Change and Source List Change Records

When a change in the global state of a multicast address occurs in a node, the node sends either a Source List Change Record or a Filter Mode Change Record for that multicast address. As with Current State Records, routers must act upon these records and possibly change their own state to reflect the new listening state of the link.

The Querier must query sources or multicast addresses that are requested to be no longer forwarded. When a router queries or receives a query for a specific set of sources, it lowers its source timers for those sources to a small interval of Last Listener Query Time milliseconds. If multicast address records are received in response to the queries which express interest in listening the queried sources, the corresponding timers are updated.

Multicast Address Specific queries can also be used in order to enable a fast transition of a router from EXCLUDE to INCLUDE mode, in case a received Multicast Address Record motivates this action. The Filter Timer for that multicast address is lowered to a small interval of Last Listener Query Time milliseconds. If any multicast address records that express EXCLUDE mode interest in the multicast address are received within this interval, the Filter Timer is updated and the suggestion to the routing protocol to forward the multicast address stands without any interruption. If not, the router will switch to INCLUDE filter mode for that multicast address.

During the query period (i.e., Last Listener Query Time milliseconds) the MLD component in the router continues to suggest to the routing protocol to forward traffic from the multicast addresses or sources that are queried. It is not until after Last Listener Query Time milliseconds without receiving a record that expresses interest in the queried multicast address or sources that the router may prune the multicast address or sources from the link.

The following table describes the changes in multicast address state and the action(s) taken when receiving either Filter Mode Change or Source List Change Records. This table also describes the queries which are sent by the Querier when a particular report is received.

We use the following notation for describing the queries that are sent. We use the notation 'Q(MA)' to describe a Multicast Address Specific Query to the MA multicast address. We use the notation 'Q(MA,A)' to describe a Multicast Address and Source Specific Query to the MA multicast address with source list A. If source list A is null as a result of the action (e.g. A*B), then no query is sent as a result of the operation.

In order to maintain protocol robustness, queries defined in the Actions column of the table below need to be transmitted [Last Listener Query Count] times, once every [Last Listener Query Interval] period.

If while scheduling new queries, there are already pending queries to be retransmitted for the same multicast address, the new and pending queries have to be merged. In addition, received host reports for a multicast address with pending queries may affect the contents of those queries. Section 7.6.3 describes the process of building and maintaining the state of pending queries.

| Router State | Report Received | New Router State | Actions |
|----------------|-----------------|----------------------|---|
| INCLUDE (A) | ALLOW (B) | INCLUDE (A+B) | (B)=MALI |
| INCLUDE (A) | BLOCK (B) | INCLUDE (A) | Send Q (MA, A*B) |
| INCLUDE (A) | TO_EX (B) | EXCLUDE (A*B, B-A) | (B-A)=0 Delete (A-B) Send Q (MA, A*B) Filter Timer=MALI |
| INCLUDE (A) | TO_IN (B) | INCLUDE (A+B) | (B)=MALI Send Q (MA, A-B) |
| EXCLUDE (X, Y) | ALLOW (A) | EXCLUDE (X+A, Y-A) | (A)=MALI |
| EXCLUDE (X, Y) | BLOCK (A) | EXCLUDE (X+(A-Y), Y) | (A-X-Y)=Filter Timer Send Q (MA, A-Y) |
| EXCLUDE (X, Y) | TO_EX (A) | EXCLUDE (A-Y, Y*A) | (A-X-Y)=Filter Timer Delete (X-A) Delete (Y-A) Send Q (MA, A-Y) Filter Timer=MALI |
| EXCLUDE (X, Y) | TO_IN (A) | EXCLUDE (X+A, Y-A) | (A)=MALI Send Q (MA, X-A) Send Q (MA) |

7.5. Switching Router Filter Modes

The Filter Timer is used as a mechanism for transitioning the Router Filter Mode from EXCLUDE to INCLUDE.

When a Filter Timer expires with a Router Filter Mode of EXCLUDE, a router assumes that there are no nodes with a filter mode of EXCLUDE

present on the attached link. Thus, the router transitions to INCLUDE filter mode for the multicast address.

A router uses the sources from the Requested List as its state for the switch to a filter mode of INCLUDE. Sources from the Requested List are moved in the Include List, while sources from the Exclude List are deleted. For example, if a router's state for a multicast address is EXCLUDE(X,Y) and the Filter Timer expires for that multicast address, the router switches to filter mode of INCLUDE with state INCLUDE(X). If at the moment of the switch the Requested List (X) is empty, the multicast address record is deleted from the router.

7.6. Action on Reception of Queries

Upon reception of an MLD message that contains a Query, the router checks if the source address of the message is a valid link-local address, if the Hop Limit is set to 1, and if the Router Alert option is present in the Hop-By-Hop Options header of the IPv6 packet. If any of these checks fails, the packet is dropped.

If the validity of the MLD message is verified, the router starts to process the Query.

7.6.1. Timer Updates

MLDv2 uses the Suppress Router-Side Processing flag to ensure robustness, as explained in Section 2.1. When a router sends or receives a query with a clear Suppress Router-Side Processing flag, it must update its timers to reflect the correct timeout values for the multicast address or sources being queried. The following table describes the timer actions when sending or receiving a Multicast Address Specific or Multicast Address and Source Specific Query with the Suppress Router-Side Processing flag not set.

| Query | Action |
|---------|--|
| ----- | ----- |
| Q(MA,A) | Source Timers for sources in A are lowered to LLQT |
| Q(MA) | Filter Timer is lowered to LLQT |

When a router sends or receives a query with the Suppress Router-Side Processing flag set, it will not update its timers.

7.6.2. Querier Election

MLDv2 elects a single router per subnet to be in Querier state; all the other routers on the subnet should be in Non-Querier state. MLDv2 uses the same querier election mechanism as MLDv1, namely the

IPv6 address. When a router starts operating on a subnet, by default it considers itself as being the Querier. Thus, it sends several General Queries separated by a small time interval (see Section 9.6 and Section 9.7 for details).

When a router receives a query with a lower IPv6 address than its own, it sets the Other Querier Present timer to Other Querier Present Timeout; if it was previously in Querier state, it switches to Non-Querier state and ceases to send queries on the link. After the Other Querier Present timer expires, it should re-enter the Querier state and begin sending General Queries.

All MLDv2 queries MUST be sent with the FE80::/64 link-local source address prefix. Therefore, for the purpose of MLDv2 querier election, an IPv6 address A is considered to be lower than an IPv6 address B if the interface ID represented by the last 64 bits of address A, in big-endian bit order, is lower than the interface ID represented by the last 64 bits of address B.

7.6.3. Building and Sending Specific Queries

7.6.3.1. Building and Sending Multicast Address Specific Queries

When a table action "Send Q(MA)" is encountered, the Filter Timer must be lowered to LLQT. The Querier must then immediately send a Multicast Address Specific query as well as schedule [Last Listener Query Count - 1] query retransmissions to be sent every [Last Listener Query Interval], over [Last Listener Query Time].

When transmitting a Multicast Address Specific Query, if the Filter Timer is larger than LLQT, the "Suppress Router-Side Processing" bit is set in the query message.

7.6.3.2. Building and Sending Multicast Address and Source Specific Queries

When a table action "Send Q(MA,X)" is encountered by the Querier in the table in Section 7.4.2, the following actions must be performed for each of the sources in X that send to multicast address MA, with source timer larger than LLQT:

- Lower source timer to LLQT;

- Add the sources to the Retransmission List;

- Set the Source Retransmission Counter for each source to [Last Listener Query Count].

The Querier must then immediately send a Multicast Address and Source Specific Query as well as schedule [Last Listener Query Count -1] query retransmissions to be sent every [Last Listener Query Interval], over [Last Listener Query Time]. The contents of these queries are calculated as follows.

When building a Multicast Address and Source Specific Query for a multicast address MA, two separate query messages are sent for the multicast address. The first one has the "Suppress Router-Side Processing" bit set and contains all the sources with retransmission state (i.e., sources from the Retransmission List of that multicast address), and timers greater than LLQT. The second has the "Suppress Router-Side Processing" bit clear and contains all the sources with retransmission state and timers lower or equal to LLQT. If either of the two calculated messages does not contain any sources, then its transmission is suppressed.

Note: If a Multicast Address Specific query is scheduled to be transmitted at the same time as a Multicast Address and Source specific query for the same multicast address, then transmission of the Multicast Address and Source Specific message with the "Suppress Router-Side Processing" bit set may be suppressed.

8. Interoperation with MLDv1

MLD version 2 hosts and routers interoperate with hosts and routers that have not yet been upgraded to MLDv2. This compatibility is maintained by hosts and routers taking appropriate actions depending on the versions of MLD operating on hosts and routers within a network.

8.1. Query Version Distinctions

The MLD version of a Multicast Listener Query message is determined as follows:

MLDv1 Query: length = 24 octets

MLDv2 Query: length >= 28 octets

Query messages that do not match any of the above conditions (e.g., a Query of length 26 octets) MUST be silently ignored.

8.2. Multicast Address Listener Behavior

8.2.1. In the Presence of MLDv1 Routers

In order to be compatible with MLDv1 routers, MLDv2 hosts MUST operate in version 1 compatibility mode. MLDv2 hosts MUST keep state per local interface regarding the compatibility mode of each attached link. A host's compatibility mode is determined from the Host Compatibility Mode variable which can be in one of the two states: MLDv1 or MLDv2.

The Host Compatibility Mode of an interface is set to MLDv1 whenever an MLDv1 Multicast Address Listener Query is received on that interface. At the same time, the Older Version Querier Present timer for the interface is set to Older Version Querier Present Timeout seconds. The timer is re-set whenever a new MLDv1 Query is received on that interface. If the Older Version Querier Present timer expires, the host switches back to Host Compatibility Mode of MLDv2.

When Host Compatibility Mode is MLDv2, a host acts using the MLDv2 protocol on that interface. When Host Compatibility Mode is MLDv1, a host acts in MLDv1 compatibility mode, using only the MLDv1 protocol, on that interface.

An MLDv1 Querier will send General Queries with the Maximum Response Code set to the desired Maximum Response Delay, i.e., the full range of this field is linear and the exponential algorithm described in Section 5.1.3. is not used.

Whenever a host changes its compatibility mode, it cancels all its pending responses and retransmission timers.

8.2.2. In the Presence of MLDv1 Multicast Address Listeners

An MLDv2 host may be placed on a link where there are MLDv1 hosts. A host MAY allow its MLDv2 Multicast Listener Report to be suppressed by a Version 1 Multicast Listener Report.

8.3. Multicast Router Behavior

8.3.1. In the Presence of MLDv1 Routers

MLDv2 routers may be placed on a network where there is at least one MLDv1 router. The following requirements apply:

If an MLDv1 router is present on the link, the Querier MUST use the lowest version of MLD present on the network. This must be administratively assured. Routers that desire to be compatible with MLDv1 MUST have a configuration option to act in MLDv1 mode; if an MLDv1 router is present on the link, the system

administrator must explicitly configure all MLDv2 routers to act in MLDv1 mode. When in MLDv1 mode, the Querier MUST send periodic General Queries truncated at the Multicast Address field (i.e., 24 bytes long), and SHOULD also warn about receiving an MLDv2 Query (such warnings must be rate-limited). The Querier MUST also fill in the Maximum Response Delay in the Maximum Response Code field, i.e., the exponential algorithm described in Section 5.1.3. is not used.

If a router is not explicitly configured to use MLDv1 and receives an MLDv1 General Query, it SHOULD log a warning. These warnings MUST be rate-limited.

8.3.2. In the Presence of MLDv1 Multicast Address Listeners

MLDv2 routers may be placed on a network where there are hosts that have not yet been upgraded to MLDv2. In order to be compatible with MLDv1 hosts, MLDv2 routers MUST operate in version 1 compatibility mode. MLDv2 routers keep a compatibility mode per multicast address record. The compatibility mode of a multicast address is determined from the Multicast Address Compatibility Mode variable, which can be in one of the two following states: MLDv1 or MLDv2.

The Multicast Address Compatibility Mode of a multicast address record is set to MLDv1 whenever an MLDv1 Multicast Listener Report is received for that multicast address. At the same time, the Older Version Host Present timer for the multicast address is set to Older Version Host Present Timeout seconds. The timer is re-set whenever a new MLDv1 Report is received for that multicast address. If the Older Version Host Present timer expires, the router switches back to Multicast Address Compatibility Mode of MLDv2 for that multicast address.

Note that when a router switches back to MLDv2 Multicast Address Compatibility Mode for a multicast address, it takes some time to regain source-specific state information. Source-specific information will be learned during the next General Query, but sources that should be blocked will not be blocked until [Multicast Address Listening Interval] after that.

When Multicast Address Compatibility Mode is MLDv2, a router acts using the MLDv2 protocol for that multicast address. When Multicast Address Compatibility Mode is MLDv1, a router internally translates the following MLDv1 messages for that multicast address to their MLDv2 equivalents:

| MLDv1 Message | MLDv2 Equivalent |
|---------------|------------------|
| Report | IS_EX({}) |
| Done | TO_IN({}) |

MLDv2 BLOCK messages are ignored, as are source-lists in TO_EX() messages (i.e., any TO_EX() message is treated as TO_EX({})). On the other hand, the Querier continues to send MLDv2 queries, regardless of its Multicast Address Compatibility Mode.

9. List of Timers, Counters, and their Default Values

Most of these timers are configurable. If non-default settings are used, they MUST be consistent among all nodes on a single link. Note that parentheses are used to group expressions to make the algebra clear.

9.1. Robustness Variable

The Robustness Variable allows tuning for the expected packet loss on a link. If a link is expected to be lossy, the value of the Robustness Variable may be increased. MLD is robust to [Robustness Variable] - 1 packet losses. The value of the Robustness Variable MUST NOT be zero, and SHOULD NOT be one. Default value: 2.

9.2. Query Interval

The Query Interval variable denotes the interval between General Queries sent by the Querier. Default value: 125 seconds.

By varying the [Query Interval], an administrator may tune the number of MLD messages on the link; larger values cause MLD Queries to be sent less often.

9.3. Query Response Interval

The Maximum Response Delay used to calculate the Maximum Response Code inserted into the periodic General Queries. Default value: 10000 (10 seconds)

By varying the [Query Response Interval], an administrator may tune the burstiness of MLD messages on the link; larger values make the traffic less bursty, as host responses are spread out over a larger interval. The number of seconds represented by the [Query Response Interval] must be less than the [Query Interval].

9.4. Multicast Address Listening Interval

The Multicast Address Listening Interval (MALI) is the amount of time that must pass before a multicast router decides there are no more listeners of a multicast address or a particular source on a link. This value MUST be ([Robustness Variable] times [Query Interval]) plus [Query Response Interval].

9.5. Other Querier Present Timeout

The Other Querier Present Timeout is the length of time that must pass before a multicast router decides that there is no longer another multicast router which should be the Querier. This value MUST be ([Robustness Variable] times ([Query Interval]) plus (one half of [Query Response Interval])).

9.6. Startup Query Interval

The Startup Query Interval is the interval between General Queries sent by a Querier on startup. Default value: 1/4 the [Query Interval].

9.7. Startup Query Count

The Startup Query Count is the number of Queries sent out on startup, separated by the Startup Query Interval. Default value: [Robustness Variable].

9.8. Last Listener Query Interval

The Last Listener Query Interval is the Maximum Response Delay used to calculate the Maximum Response Code inserted into Multicast Address Specific Queries sent in response to Version 1 Multicast Listener Done messages. It is also the Maximum Response Delay used to calculate the Maximum Response Code inserted into Multicast Address and Source Specific Query messages. Default value: 1000 (1 second).

Note that for values of LLQI greater than 32.768 seconds, a limited set of values can be represented, corresponding to sequential values of Maximum Response Code. When converting a configured time to a Maximum Response Code value, it is recommended to use the exact value if possible, or the next lower value if the requested value is not exactly representable.

This value may be tuned to modify the "leave latency" of the link. A reduced value results in reduced time to detect the departure of the last listener for a multicast address or source.

9.9. Last Listener Query Count

The Last Listener Query Count is the number of Multicast Address Specific Queries sent before the router assumes there are no local listeners. The Last Listener Query Count is also the number of Multicast Address and Source Specific Queries sent before the router assumes there are no listeners for a particular source. Default value: [Robustness Variable].

9.10. Last Listener Query Time

The Last Listener Query Time is the time value represented by the Last Listener Query Interval, multiplied by [Last Listener Query Count]. It is not a tunable value, but may be tuned by changing its components.

9.11. Unsolicited Report Interval

The Unsolicited Report Interval is the time between repetitions of a node's initial report of interest in a multicast address. Default value: 1 second.

9.12. Older Version Querier Present Timeout

The Older Version Querier Present Timeout is the time-out for transitioning a host back to MLDv2 Host Compatibility Mode. When an MLDv1 query is received, MLDv2 hosts set their Older Version Querier Present Timer to [Older Version Querier Present Timeout].

This value MUST be ([Robustness Variable] times (the [Query Interval] in the last Query received)) plus ([Query Response Interval]).

9.13. Older Version Host Present Timeout

The Older Version Host Present Timeout is the time-out for transitioning a router back to MLDv2 Multicast Address Compatibility Mode for a specific multicast address. When an MLDv1 report is received for that multicast address, routers set their Older Version Host Present Timer to [Older Version Host Present Timeout].

This value MUST be ([Robustness Variable] times [Query Interval]) plus ([Query Response Interval]).

9.14. Configuring timers

This section is meant to provide advice to network administrators on how to tune these settings to their network. Ambitious router

implementations might tune these settings dynamically based upon changing characteristics of the network.

9.14.1. Robustness Variable

The Robustness Variable tunes MLD to expected losses on a link. MLDv2 is robust to $[\text{Robustness Variable}] - 1$ packet losses, e.g., if the Robustness Variable is set to the default value of 2, MLDv2 is robust to a single packet loss but may operate imperfectly if more losses occur. On lossy links, the value of the Robustness Variable should be increased to allow for the expected level of packet loss. However, increasing the value of the Robustness Variable increases the leave latency of the link (the time between when the last listener stops listening to a source or multicast address and when the traffic stops flowing).

9.14.2. Query Interval

The overall level of periodic MLD traffic is inversely proportional to the Query Interval. A longer Query Interval results in a lower overall level of MLD traffic. The value of the Query Interval MUST be equal to or greater than the Maximum Response Delay used to calculate the Maximum Response Code inserted in General Query messages.

9.14.3. Maximum Response Delay

The burstiness of MLD traffic is inversely proportional to the Maximum Response Delay. A longer Maximum Response Delay will spread Report messages over a longer interval. However, a longer Maximum Response Delay in Multicast Address Specific and Multicast Address And Source Specific Queries extends the leave latency (the time between when the last listener stops listening to a source or multicast address and when the traffic stops flowing.) The expected rate of Report messages can be calculated by dividing the expected number of Reporters by the Maximum Response Delay. The Maximum Response Delay may be dynamically calculated per Query by using the expected number of Reporters for that Query as follows:

| | |
|---|---|
| General Query | All nodes on link |
| Multicast Address Specific Query | All nodes on the link that had expressed interest in the multicast address |
| Multicast Address and Source Specific Query | All nodes on the link that had expressed interest in the source and multicast address |

A router is not required to calculate these populations or tune the Maximum Response Delay dynamically; these are simply guidelines.

10. Security Considerations

We consider the ramifications of a forged message of each type. Note that before processing an MLD message, nodes verify if the source address of the message is a valid link-local address (or the unspecified address), if the Hop Limit is set to 1, and if the Router Alert option is present in the Hop-By-Hop Options header of the IPv6 packet. If any of these checks fails, the packet is dropped. This defends the MLDv2 nodes from acting on forged MLD messages originated off-link. Therefore, in the following we discuss only the effects of on-link forgery.

10.1. Query Message

A forged Query message from a machine with a lower IPv6 address than the current Querier will cause Querier duties to be assigned to the forger. If the forger then sends no more Query messages, other routers' Other Querier Present timer will time out and one will resume the role of Querier. During this time, if the forger ignores Multicast Listener Done Messages, traffic might flow to multicast addresses with no listeners for up to [Multicast Address Listener Interval].

A forged Version 1 Query message will put MLDv2 listeners on that link in MLDv1 Host Compatibility Mode. This scenario can be avoided by providing MLDv2 hosts with a configuration option to ignore Version 1 messages completely.

A DoS attack on a node could be staged through forged Multicast Address and Source Specific Queries. The attacker can find out about the listening state of a specific node with a general query. After that it could send a large number of Multicast Address and Source Specific Queries, each with a large source list and/or long Maximum Response Delay. The node will have to store and maintain the sources specified in all of those queries for as long as it takes to send the delayed response. This would consume both memory and CPU cycles in order to augment the recorded sources with the source lists included in the successive queries.

To protect against such a DoS attack, a node stack implementation could restrict the number of Multicast Address and Source Specific Queries per multicast address within this interval, and/or record only a limited number of sources.

10.2. Current State Report messages

A forged Report message may cause multicast routers to think there are listeners of a multicast address on a link when there are not. Nevertheless, since listening to a multicast address on a host is generally an unprivileged operation, a local user may trivially gain the same result without forging any messages.

A forged Version 1 Report Message may put a router into MLDv1 Multicast Address Compatibility Mode for a particular multicast address, meaning that the router will ignore MLDv2 source specific state messages. This can cause traffic to flow from unwanted sources for up to [Multicast Address Listener Interval]. This can be solved by providing routers with a configuration switch to ignore Version 1 messages completely. This breaks automatic compatibility with Version 1 hosts, so it should only be used in situations where source filtering is critical.

10.3. State Change Report messages

A forged State Change Report message will cause the Querier to send out Multicast Address Specific or Multicast Address and Source Specific Queries for the multicast address in question. This causes extra processing on each router and on each listener of the multicast address, but cannot cause loss of desired traffic.

11. IANA Considerations

IANA has assigned the IPv6 link-local multicast address FF02:0:0:0:0:0:16, called "all MLDv2-capable routers", as described in Section 5.2.14. Version 2 Multicast Listener Reports will be sent to this special address.

In addition, IANA has assigned the ICMPv6 message type value of 143 for Version 2 Multicast Listener Report messages, as specified in Section 4.

12. Contributors

Roland Vida, Luis Henrique Maciel Kosmowski Costa, Serge Fdida, Steve Deering, Bill Fenner, and Isidor Kouvelas are the authors of RFC 3810, which makes up the majority of the content in this document.

13. Acknowledgments

We would like to thank Hitoshi Asaeda, Randy Bush, Francis Dupont, Ted Hardie, Russ Housley, Konstantin Kabassanov, Erik Nordmark,

Shinsuke Suzuki, Margaret Wasserman, Bert Wijnen, and Remi Zara for their valuable comments and suggestions on this document.

14. References

14.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <<https://www.rfc-editor.org/info/rfc2460>>.
- [RFC2463] Conta, A. and S. Deering, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 2463, DOI 10.17487/RFC2463, December 1998, <<https://www.rfc-editor.org/info/rfc2463>>.
- [RFC2464] Crawford, M., "Transmission of IPv6 Packets over Ethernet Networks", RFC 2464, DOI 10.17487/RFC2464, December 1998, <<https://www.rfc-editor.org/info/rfc2464>>.
- [RFC2710] Deering, S., Fenner, W., and B. Haberman, "Multicast Listener Discovery (MLD) for IPv6", RFC 2710, DOI 10.17487/RFC2710, October 1999, <<https://www.rfc-editor.org/info/rfc2710>>.
- [RFC2711] Partridge, C. and A. Jackson, "IPv6 Router Alert Option", RFC 2711, DOI 10.17487/RFC2711, October 1999, <<https://www.rfc-editor.org/info/rfc2711>>.
- [RFC3513] Hinden, R. and S. Deering, "Internet Protocol Version 6 (IPv6) Addressing Architecture", RFC 3513, DOI 10.17487/RFC3513, April 2003, <<https://www.rfc-editor.org/info/rfc3513>>.

14.2. Informative References

- [RFC2461] Narten, T., Nordmark, E., and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", RFC 2461, DOI 10.17487/RFC2461, December 1998, <<https://www.rfc-editor.org/info/rfc2461>>.

- [RFC2462] Thomson, S. and T. Narten, "IPv6 Stateless Address Autoconfiguration", RFC 2462, DOI 10.17487/RFC2462, December 1998, <<https://www.rfc-editor.org/info/rfc2462>>.
- [RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, DOI 10.17487/RFC3376, October 2002, <<https://www.rfc-editor.org/info/rfc3376>>.
- [RFC3569] Bhattacharyya, S., Ed., "An Overview of Source-Specific Multicast (SSM)", RFC 3569, DOI 10.17487/RFC3569, July 2003, <<https://www.rfc-editor.org/info/rfc3569>>.
- [RFC3678] Thaler, D., Fenner, B., and B. Quinn, "Socket Interface Extensions for Multicast Source Filters", RFC 3678, DOI 10.17487/RFC3678, January 2004, <<https://www.rfc-editor.org/info/rfc3678>>.
- [RFC3810] Vida, R., Ed. and L. Costa, Ed., "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, DOI 10.17487/RFC3810, June 2004, <<https://www.rfc-editor.org/info/rfc3810>>.

Appendix A. Design Rationale

A.1. The Need for State Change Messages

MLDv2 specifies two types of Multicast Listener Reports: Current State and State Change. This section describes the rationale for the need for both these types of Reports.

Routers need to distinguish Multicast Listener Reports that were sent in response to Queries from those that were sent as a result of a change in the per-interface state. Multicast Listener Reports that are sent in response to Multicast Address Listener Queries are used mainly to refresh the existing state at the router; they typically do not cause transitions in state at the router. Multicast Listener Reports that are sent in response to changes in the per-interface state require the router to take some action in response to the received report (see Section 7.4).

The inability to distinguish between the two types of reports would force a router to treat all Multicast Listener Reports as potential changes in state and could result in increased processing at the router as well as an increase in MLD traffic on the link.

A.2. Host Suppression

In MLDv1, a host would not send a pending multicast listener report if a similar report was sent by another listener on the link. In MLDv2, the suppression of multicast listener reports has been removed. The following points explain this decision.

1. Routers may want to track per-host multicast listener status on an interface. This would allow routers to implement fast leaves (e.g., for layered multicast congestion control schemes), as well as track listener status for possible security or accounting purposes. The present specification does not require routers to implement per-host tracking. Nevertheless, the lack of host suppression in MLDv2 makes possible to implement either proprietary or future standard behavior of multicast routers that would support per-host tracking, while being fully interoperable with MLDv2 listeners and routers that implement the exact behavior described in this specification.
2. Multicast Listener Report suppression does not work well on bridged LANs. Many bridges and Layer2/Layer3 switches that implement MLD snooping do not forward MLD messages across LAN segments in order to prevent multicast listener report suppression.
3. By eliminating multicast listener report suppression, hosts have fewer messages to process; this leads to a simpler state machine implementation.
4. In MLDv2, a single multicast listener report now bundles multiple multicast address records to decrease the number of packets sent. In comparison, the previous version of MLD required that each multicast address be reported in a separate message.

A.3. Switching router filter modes from EXCLUDE to INCLUDE

If on a link there are nodes in both EXCLUDE and INCLUDE modes for a single multicast address, the router must be in EXCLUDE mode as well (see section 7.2.1). In EXCLUDE mode, a router forwards traffic from all sources except those in the Exclude List. If all nodes in EXCLUDE mode cease to exist or to listen, it would be desirable for the router to switch back to INCLUDE mode seamlessly, without interrupting the flow of traffic to existing listeners.

One of the ways to accomplish this is for routers to keep track of all sources that nodes that are in INCLUDE mode listen to, even though the router itself is in EXCLUDE mode. If the Filter Timer for a multicast address expires, it implies that there are no nodes in

EXCLUDE mode on the link (otherwise a multicast listener report from that node would have refreshed the Filter Timer). The router can then switch to INCLUDE mode seamlessly; sources from the Requested List are moved to the Include List, while sources from the Exclude List are deleted.

Appendix B. Summary of Changes

B.1. MLDv1

The following is a summary of changes from MLDv1, specified in [RFC2710].

MLDv2 introduces source filtering.

The IP service interface of MLDv2 nodes is modified accordingly. It enables the specification of a filter mode and a source list.

An MLDv2 node keeps per-socket and per-interface multicast listening states that include a filter mode and a source list for each multicast address. This enables packet filtering based on a socket's multicast reception state.

MLDv2 state kept on routers includes a filter mode and a list of sources and source timers for each multicast address that has listeners on the link. MLDv1 routers kept only the list of multicast addresses.

Queries include additional fields (Section 5.1).

The S flag (Suppress Router-Side Processing) is included in queries in order to fix robustness issues.

The Querier's Robustness Variable and Query Interval Code are included in Queries in order to synchronize all MLDv2 routers connected to the same link.

A new Query type (Multicast Address and Source Specific Query) is introduced.

The Maximum Response Delay is not directly included in the Query anymore. Instead, an exponential algorithm is used to calculate its value, based on the Maximum Response Code included in the Query. The maximum value is increased from 65535 milliseconds to about 140 minutes.

Reports include Multicast Address Records. Information on the listening state for several different multicast addresses can be included in the same Report message.

Reports are sent to the "all MLDv2-capable multicast routers" address, instead of the multicast address the host listens to, as in MLDv1. This facilitates the operation of layer-2 snooping switches.

There is no "host suppression", as in MLDv1. All nodes send Report messages.

Unsolicited Reports, announcing changes in receiver listening state, are sent [Robustness Variable] times. RFC 2710 is less explicit.

There are no Done messages.

Interoperability with MLDv1 systems is achieved by MLDv2 state operations.

In order to ensure interoperability, hosts maintain a Host Compatibility Mode variable and an Older Version Querier Present timer per interface. Routers maintain a Multicast Address Compatibility Mode variable and an Older Version Host Present timer per multicast address.

B.2. MLDv2

The following summarizes the changes made since RFC 3810.

Added definition of Resv to address Erratum 4773.

Added clarifying text on which multicast addresses require the sending of MLD messages to address Erratum 5977.

Author's Address

Brian Haberman (editor)
Johns Hopkins University Applied Physics Lab

Email: brian@innovationslab.net

Internet Engineering Task Force
Internet-Draft
Intended status: Experimental
Expires: 6 December 2021

V. Govindan
S. Venaas
Cisco
4 June 2021

PIM Join/ Prune Attributes for LISP Environments using Underlay
Multicast
draft-vgovindan-pim-jp-extensions-lisp-01

Abstract

This document specifies an extension to PIM Receiver RLOC Join/ Prune attribute that supports the construction of multicast distribution trees where the root and receivers are located in different Locator/ ID Separation Protocol (LISP) sites and are connected using underlay IP Multicast. This attribute allows the receiver site to signal the underlay multicast group to the control plane of the root ITR (Ingress Tunnel Router).

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 <https://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 6 December 2021.

Copyright Notice

Copyright (c) 2021 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 (<https://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 |
| 1.1. Requirements Language | 3 |
| 2. The case for extending the Received ETR RLOC Attribute of RFC 8059 | 3 |
| 3. Acknowledgements | 4 |
| 4. Contributors | 4 |
| 5. IANA Considerations | 4 |
| 6. Security Considerations | 4 |
| 7. Normative References | 5 |
| Authors' Addresses | 5 |

1. Introduction

The construction of multicast distribution trees where the root and receivers are located in different LISP sites [RFC6830] is defined in [RFC6831].

[RFC6831] specifies that (root-EID, G) data packets are to be LISP-encapsulated into (root-RLOC, G) multicast packets. [RFC8059] defines PIM J/P attribute extensions to construct multicast distribution trees. This document extends the Receiver ETR RLOC PIM J/P attribute [RFC8059] to facilitate the construction of underlay multicast trees for (root-RLOC, G).

Specifically, the assignment of the underlay multicast group needs to be done in consonance with the downstream xTR nodes and avoid unnecessary replication or traffic hairpinning.

Since the Receiver RLOC Attribute defined in [RFC8059] only addresses the Ingress Replication case, an extension of the scope of that PIM J/P attribute is defined by this draft to include scenarios where the underlay uses Multicast transport. The scope extension proposed here complies with the base specification [RFC5384].

This document uses terminology defined in [RFC6830], such as EID, RLOC, ITR, and ETR.

1.1. 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].

2. The case for extending the Received ETR RLOC Attribute of RFC 8059

When LISP based Multicast trees can be built using IP Multicast in the underlay, the mapping between the overlay group address and the underlay group address becomes a very crucial engineering decision:

Flexible mapping of overlay to underlay group ranges:

Three different types of overlay to underlay group mappings are possible: Many to one mapping: Many (root-EID, G) flows originating from a RLOC can be mapped to the same underlay (root-RLOC, G-u) flow. One to many mapping: Conversely the same overlay flow can be mapped to two or more flows e.g. (root-RLOC, G-u1) and (root-RLOC, G-u2) to cater to the requirements of downstream xTR nodes. One to one mapping: Every (root-EID, G) flow is mapped to a different (root-RLOC, G-u) flow. The overlay can use ASM while the underlay can use SSM ranges.

Multicast Address Range constraints:

It is possible that under certain circumstances, different subsets of xTRs subscribing to the same overlay multicast stream would be constrained to use different underlay multicast mapping ranges. This definitely involves a trade-off between replication and the flexibility in assigning address ranges and could be required in certain situations further below.

Inter-site PxTR:

When multiple LISP sites are connected through a LISP based transit, the site border node interconnects the site-facing interfaces and the external LISP based core. Under such circumstances, there could be different ranges of multicast group addresses used for building the (S-RLOC, G) trees inside the LISP site and the external LISP based core. This is desired for various reasons:

Hardware resource restrictions:

Platform limitations could force engineering decisions to be made on restricting multicast address ranges in the underlay.

Other Use-cases:

TBD

Editorial Note: Comments from Stig: There should be some text indicating that the group address used should ideally only be used for LISP encapsulation (if ASM), and perhaps that it is preferable to use an SSM group. Also, that the group obviously must be a group that the underlay supports/allows. I think it is also worth noting that ideally, different ETRs should request the same group.

3. Acknowledgements

The authors would like to thank Dino Farinacci and Victor Moreno for their valuable comments.

4. Contributors

Sankaralingam
Cisco

Email: sankt@cisco.com

Amit Kumar
Cisco

Email: kumaram3@cisco.com

5. IANA Considerations

No new requests to IANA

6. Security Considerations

There is perhaps a new attack vector where an attacker can send a bunch of joins with different group addresses. It may interfere with other multicast traffic if those group addresses overlap. Also, it may take up a lot of resources if replication for thousands of groups are requested. However PIM authentication (?) should come to the rescue here. TBD Since explicit tracking would be done, perhaps it is worth enforcing that for each ETR RLOC (the RLOC used as the source of the overlay join), there could be a configurable number of maximum permissible group(s). TBD

Ed Note: To be addressed - Comments from Stig: Regarding security considerations and PIM authentication. The only solution we have here is to use IP-Sec to sign the J/P messages. I don't know if anyone has tried to use IPSec between LISP RLOCs. Are there any LISP security mechanisms that would help here for authenticating LISP encapsulated messages between xTRs?

7. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC5384] Boers, A., Wijnands, I., and E. Rosen, "The Protocol Independent Multicast (PIM) Join Attribute Format", RFC 5384, DOI 10.17487/RFC5384, November 2008, <<https://www.rfc-editor.org/info/rfc5384>>.
- [RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, DOI 10.17487/RFC6830, January 2013, <<https://www.rfc-editor.org/info/rfc6830>>.
- [RFC6831] Farinacci, D., Meyer, D., Zwiebel, J., and S. Venaas, "The Locator/ID Separation Protocol (LISP) for Multicast Environments", RFC 6831, DOI 10.17487/RFC6831, January 2013, <<https://www.rfc-editor.org/info/rfc6831>>.
- [RFC7761] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I., Parekh, R., Zhang, Z., and L. Zheng, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", STD 83, RFC 7761, DOI 10.17487/RFC7761, March 2016, <<https://www.rfc-editor.org/info/rfc7761>>.
- [RFC8059] Arango, J., Venaas, S., Kouvelas, I., and D. Farinacci, "PIM Join Attributes for Locator/ID Separation Protocol (LISP) Environments", RFC 8059, DOI 10.17487/RFC8059, January 2017, <<https://www.rfc-editor.org/info/rfc8059>>.

Authors' Addresses

Vengada Prasad Govindan
Cisco

Email: venggovi@cisco.com

Stig Venaas
Cisco

Email: svenaas@cisco.com