

Network Working Group  
Request for Comments: 3810  
Updates: [2710](#)  
Category: Standards Track

R. Vida, Ed.  
L. Costa, Ed.  
LIP6  
June 2004

## Multicast Listener Discovery Version 2 (MLDv2) for IPv6

### Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

### Copyright Notice

Copyright (C) The Internet Society (2004).

### 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.

## Table of Contents

1. Introduction. . . . .	2
2. Protocol Overview . . . . .	3
3. The Service Interface for Requesting IP Multicast Reception .	9
4. Multicast Listening State Maintained by Nodes . . . . .	11
5. Message Formats . . . . .	13
6. Protocol Description for Multicast Address Listeners. . . . .	27
7. Protocol Description for Multicast Routers. . . . .	34
8. Interoperation with MLDv1 . . . . .	48
9. List of Timers, Counters, and their Default Values. . . . .	51
10. Security Considerations . . . . .	55
11. IANA Considerations . . . . .	56
12. References. . . . .	56
13. Acknowledgments . . . . .	57
Appendix A. Design Rationale. . . . .	58
Appendix B. Summary of Changes from MLDv1 . . . . .	59
Editors' Contact Information. . . . .	61
Authors' Addresses. . . . .	61
Full Copyright Statement. . . . .	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.

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]. Due to the lack of italics, emphasis is indicated herein by bracketing a word or phrase in "\*" characters. Furthermore, square brackets are used to denote the value of the enclosed variable, as opposed to the variable itself, written without brackets.

## 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 A2](#) 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 A1](#).

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:

- o 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 A3](#).

- o 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:

- o 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.
- o 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,

- 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 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:

- o Multicast Listener Query (Type = decimal 130)
- o 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:

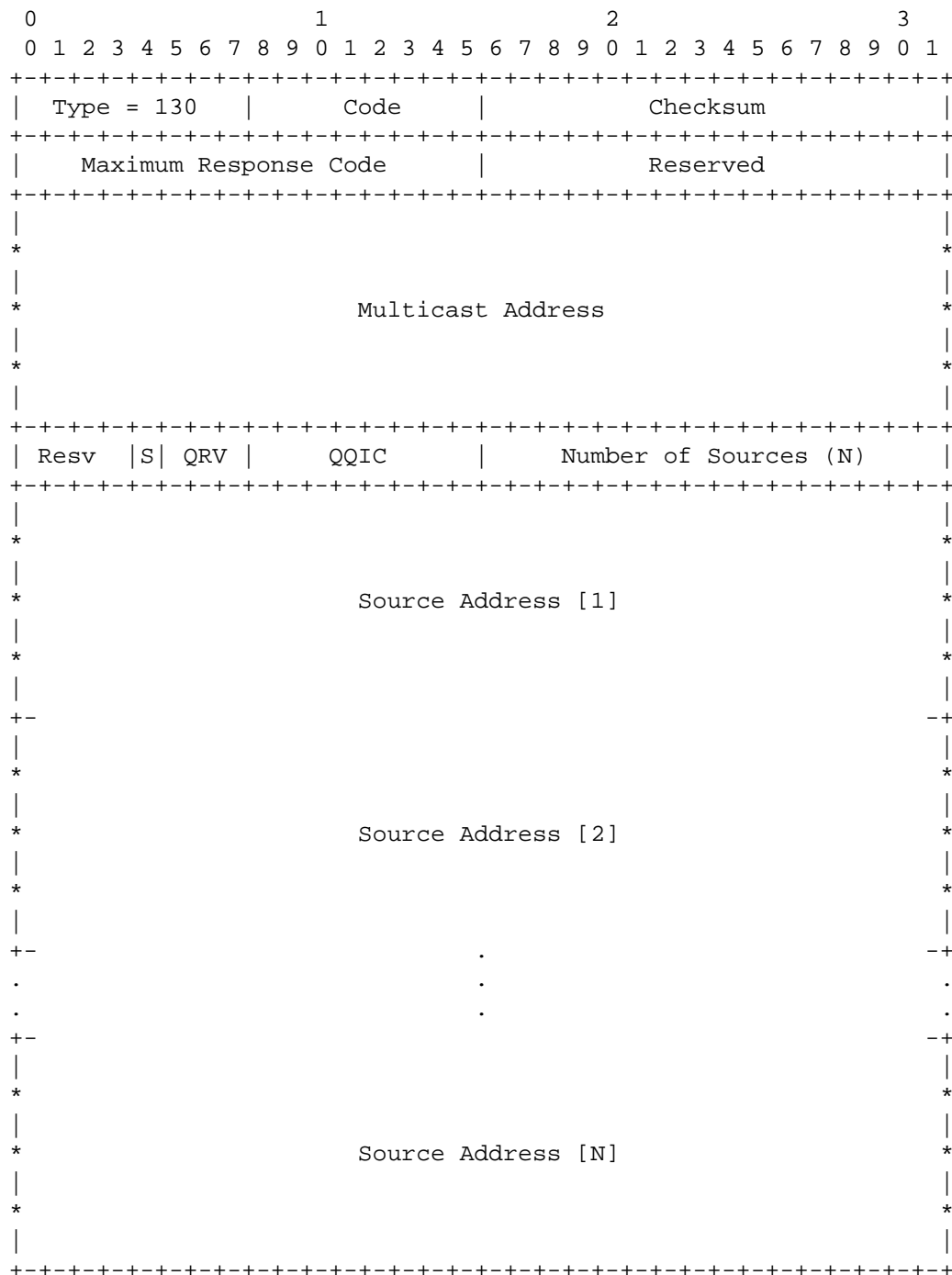
- o Version 1 Multicast Listener Report (Type = decimal 131) [RFC2710]
- o Version 1 Multicast Listener Done (Type = decimal 132) [RFC2710]

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           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Maximum Response Delay = (mant | 0x1000) << (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).

#### 5.1.7. 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 >= 128, QQIC represents a floating-point value as follows:

```

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

```

QQI = (mant | 0x10) << (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 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:

- o 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.

- o 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.
- o 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

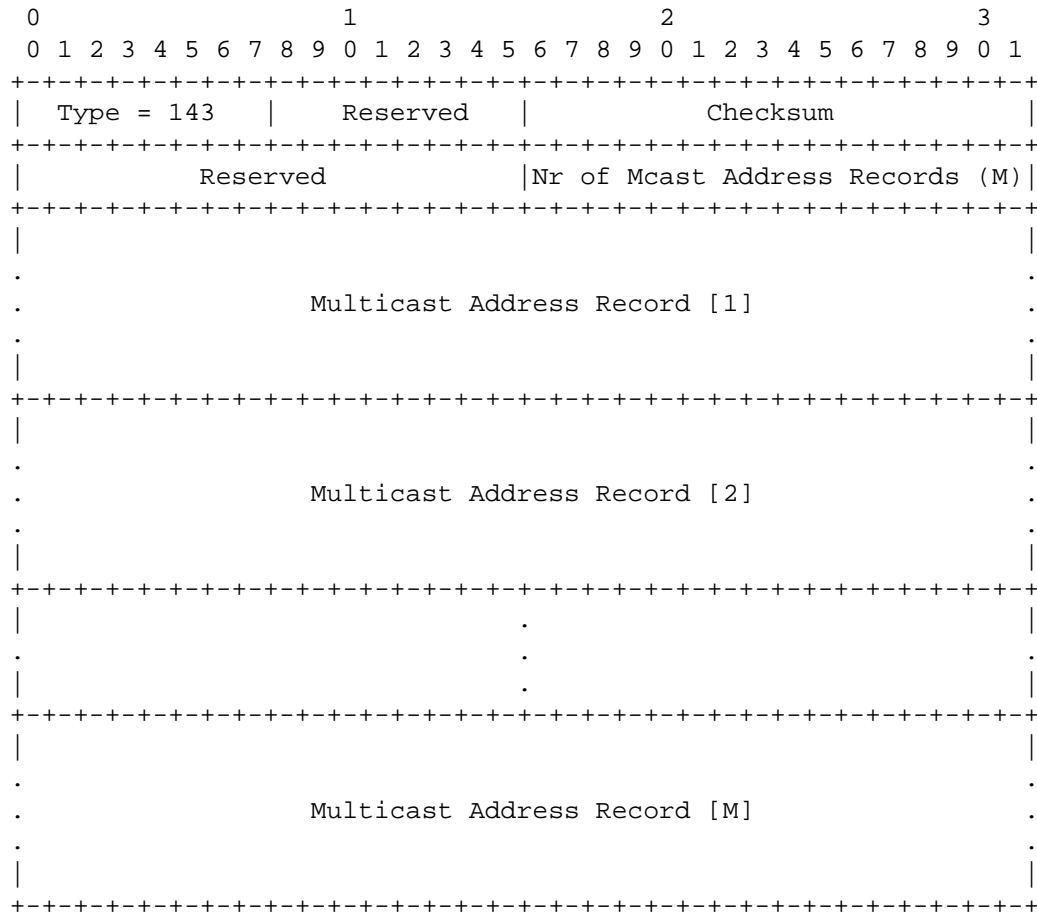
All MLDv2 Queries MUST be sent with a valid IPv6 link-local source address. If a node (router or host) receives a Query message with the IPv6 Source Address set to the unspecified address (::), or any other address that is not a valid IPv6 link-local address, it MUST silently discard the message and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

#### 5.1.15. Destination Addresses for Queries

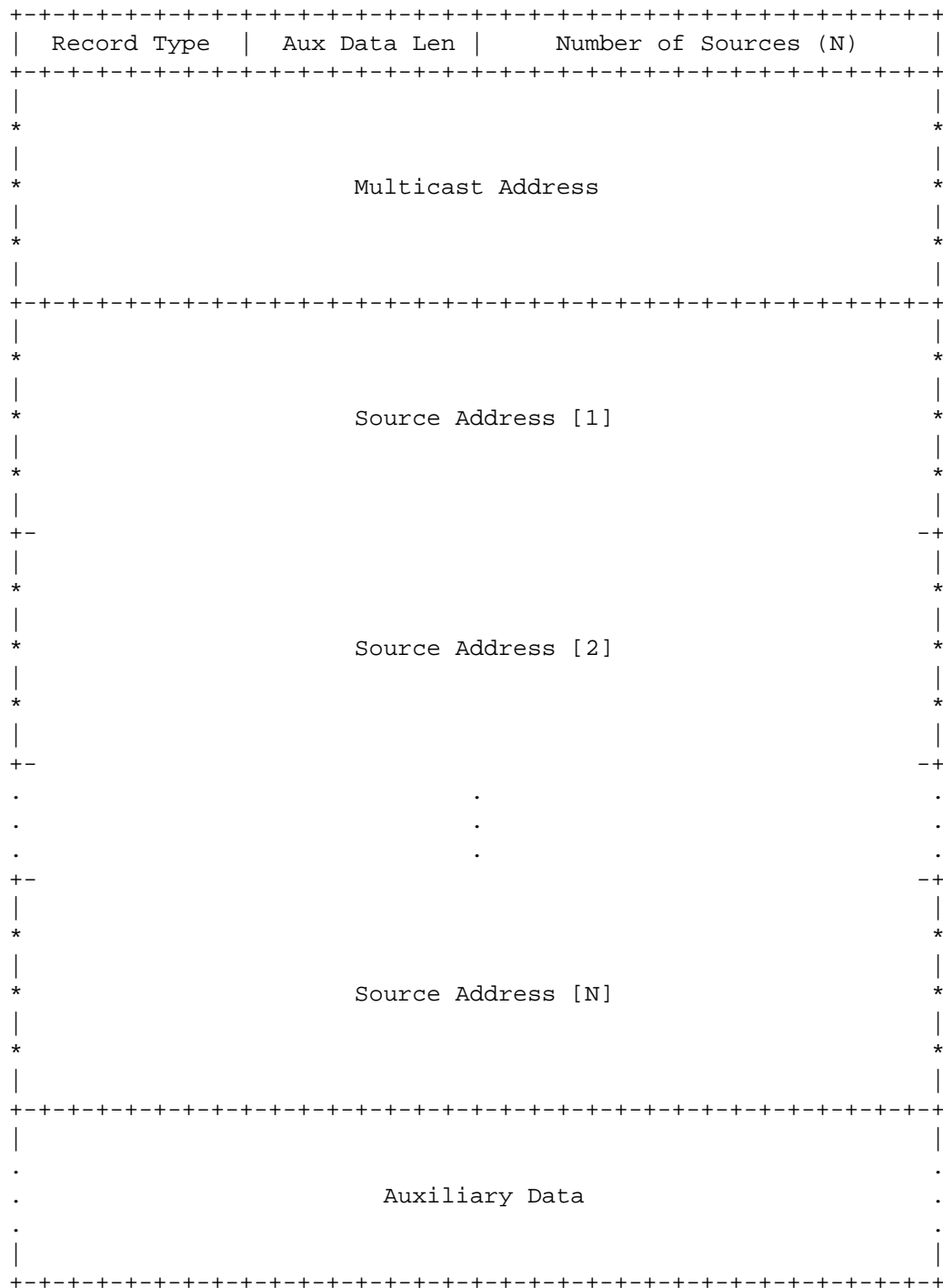
In MLDv2, General Queries are sent to the link-scope all-nodes multicast address (FF02::1). Multicast Address Specific and Multicast Address and Source Specific Queries are sent with an IP destination address equal to the multicast address of interest. \*However\*, a node MUST accept and process any 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:

- o 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:

Value	Name and Meaning
-------	------------------

-----	-----
-------	-------

- |   |  |
|---|--|
| 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.
- o     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.
- o     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:

- 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.

#### 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:

- o 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.
- o 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).

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

- o a change of the per-interface listening state, caused by a local invocation of `IPv6MulticastListen`;
- o the firing of a specific timer;
- o 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:

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

- o an Interface Timer for scheduling responses to General Queries;
- o a Multicast Address Timer for scheduling responses to Multicast Address (and Source) Specific Queries, for each multicast address the node has to report on;
- o 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 A2](#) 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:

- o 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 A3](#).

- o 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 [sections 9.6](#) and [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:

- o Lower source timer to LLQT;
- o Add the sources to the Retransmission List;
- o 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:

- o 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.

- o 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  $[\text{Robustness Variable}] - 1$  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  $[\text{Query Interval}]$ , 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  $[\text{Query Response Interval}]$ , 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  $[\text{Query Response Interval}]$  must be less than the  $[\text{Query Interval}]$ .

### 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  $([\text{Robustness Variable}] \times [\text{Query Interval}])$  plus  $[\text{Query Response Interval}]$ .

### 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 [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:

Query Type -----	Expected number of Reporters -----
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 FF02:0:0:0:0:0:0:16, 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. References

### 12.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", [RFC 2460](#), December 1998.



- [RFC2463] Conta, A. and S. Deering, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", [RFC 2463](#), December 1998.
- [RFC2464] Crawford, M., "Transmission of IPv6 Packets over Ethernet Networks", [RFC 2464](#), December 1998.
- [RFC2710] Deering, S., Fenner, W. and B. Haberman, "Multicast Listener Discovery (MLD) for IPv6", [RFC 2710](#), October 1999.
- [RFC2711] Partridge, C. and A. Jackson, "IPv6 Router Alert Option," [RFC 2711](#), October 1999.
- [RFC3513] Hinden, R. and S. Deering, "Internet Protocol Version 6 (IPv6) Addressing Architecture, [RFC 3513](#), April 2003.

## 12.2. Informative References

- [RFC2461] Narten, T., Nordmark, E. and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", [RFC 2461](#), December 1998.
- [RFC2462] Thomson, S. and T. Narten, "IPv6 Stateless Address Autoconfiguration", [RFC 2462](#), December 1998.
- [RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B. and A. Thyagarajan, "Internet Group Management Protocol, Version 3", [RFC 3376](#), October 2002.
- [RFC3569] Bhattacharyya, S., Ed., "An Overview of Source-Specific Multicast (SSM)", [RFC 3569](#), July 2003.
- [RFC3678] Thaler, D., Fenner, B. and B. Quinn, "Socket Interface Extensions for Multicast Source Filters", [RFC 3678](#), January 2004.

## 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.

## 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 from MLDv1

The following is a summary of changes from MLDv1, specified in [RFC 2710](#).

- o MLDv2 introduces source filtering.
- o The IP service interface of MLDv2 nodes is modified accordingly. It enables the specification of a filter mode and a source list.
- o 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.
- o 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.

- o Queries include additional fields ([section 5.1](#)).
- o The S flag (Suppress Router-Side Processing) is included in queries in order to fix robustness issues.
- o 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.
- o A new Query type (Multicast Address and Source Specific Query) is introduced.
- o 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.
- o Reports include Multicast Address Records. Information on the listening state for several different multicast addresses can be included in the same Report message.
- o 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.
- o There is no "host suppression", as in MLDv1. All nodes send Report messages.
- o Unsolicited Reports, announcing changes in receiver listening state, are sent [Robustness Variable] times. [RFC 2710](#) is less explicit.
- o There are no Done messages.
- o Interoperability with MLDv1 systems is achieved by MLDv2 state operations.
- o 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.

## Editors' Contact Information

Rolland Vida  
LIP6, Universite Pierre et Marie Curie  
8, rue du Capitaine Scott  
75015 Paris, France

Phone: +33-1.44.27.30.58  
EMail: Rolland.Vida@lip6.fr

Luis Henrique Maciel Kosmalski Costa  
LIP6, Universite Pierre et Marie Curie  
8, rue du Capitaine Scott  
75015 Paris, France

Phone: +33-1.44.27.30.58  
EMail: Luis.Costa@lip6.fr

## Authors' Addresses

This document was written by:

Rolland Vida, LIP6  
EMail: Rolland.Vida@lip6.fr

Luis Henrique Maciel Kosmalski Costa, LIP6  
EMail: Luis.Costa@lip6.fr

Serge Fdida, LIP6  
EMail: Serge.Fdida@lip6.fr

Steve Deering, Cisco Systems, Inc.  
EMail: deering@cisco.com

Bill Fenner, AT&T Labs - Research  
EMail: fenner@research.att.com

Isidor Kouvelas, Cisco Systems, Inc.  
EMail: kouvelas@cisco.com

Brian Haberman, Caspian Networks  
EMail: brian@innovationslab.net

This document is the translation of [[RFC3376](#)] for IPv6 semantics. It was elaborated based on the translation of ([RFC 2236](#)) into [[RFC2710](#)].

## Full Copyright Statement

Copyright (C) The Internet Society (2004). This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

## Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

## Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.