

PIM Working Group
Internet-Draft
Intended status: Informational
Expires: October 12, 2012

H. Asaeda
Keio University
N. Leymann
Deutsche Telekom AG
April 10, 2012

IGMP/MLD-Based Explicit Membership Tracking Function for Multicast
Routers
draft-ietf-pim-explicit-tracking-01

Abstract

This document describes the IGMP/MLD-based explicit membership tracking function for multicast routers. The explicit tracking function is useful for accounting and contributes to saving network resource and fast leaves (i.e. shortened leave latency).

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on October 12, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

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

described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Explicit Tracking Function	4
2.1. Reducing the Number of Specific Queries	4
2.2. Shortening Leave Latencies	5
2.3. Considerations	5
3. Membership State Information	6
4. Multicast Router Behavior	7
5. Interoperability and Compatibility	8
6. Security Considerations	8
7. Acknowledgements	8
8. References	9
8.1. Normative References	9
8.2. Informative References	9
Authors' Addresses	9

1. Introduction

The Internet Group Management Protocol (IGMP) [1] for IPv4 and the Multicast Listener Discovery Protocol (MLD) [2] for IPv6 are the standard protocols used by listener hosts and multicast routers. When a host starts listening particular multicast channels, it sends IGMP/MLD State-Change Report messages specifying the corresponding channel information as the join/leave request to its upstream router (i.e., an adjacent multicast router or IGMP/MLD proxy [4]). This "unsolicited" Report is sent only once upon reception.

IGMP/MLD are non-reliable protocols; the unsolicited Report messages may be lost or not be reached to upstream routers. To recover the problem, the routers need to update membership information by sending IGMP/MLD General Query messages periodically. Member hosts then reply with "solicited" Report messages whenever they receive the Query messages.

Multicast routers are able to periodically maintain the multicast listener (or membership) state of downstream hosts attached on the same link by getting unsolicited Report messages and synchronize the actual membership state within the General Query timer interval (i.e., [Query Interval] value defined in [1][2].) However, this approach does not guarantee that the membership state is always perfectly synchronized. To minimize the possibility of having the outdated membership information, routers may shorten the periodic General Query timer interval. Unfortunately, this would increase the number of transmitted solicited Report messages and induce network congestion. And the more the network congestion is occurred, the more IGMP/MLD Report messages may be lost and the membership state information may be outdated in the router.

The IGMPv3 [1] and MLDv2 [2] protocols can provide the capability of keeping track of downstream (adjacent) multicast listener state to multicast routers. This document describes the "IGMP/MLD-based explicit member tracking function" for multicast routers and details the way for routers to implement the function. By enabling the explicit tracking function, routers can keep track of the downstream multicast membership state. This function implements the following requirements:

- o Per-host accounting
- o Reducing the number of transmitted Query and Report messages
- o Shortening leave latencies

- o Maintaining multicast channel characteristics (or statistics)

where this document mainly focuses on the above second and third bullets in the following sections.

The explicit tracking function does not change message formats used by the standard IGMPv3 [1] and MLDv2 [2], and their lightweight version protocols [3]. It does not change a multicast data sender's and receiver's behavior as well.

2. Explicit Tracking Function

2.1. Reducing the Number of Specific Queries

The explicit tracking function reduces the number of Group-Specific or Group-and-Source Specific Query messages transmitted from a router, and then the number of Current-State Report messages transmitted from member hosts. As the result, network resources used for IGMP/MLD query-and-reply communications between a router and member hosts can be saved.

According to [1] and [2], whenever a router receives the State-Change Report, it sends the corresponding Group-Specific or Group-and-Source Specific Query messages to confirm whether the Report sender is the last member host or not. After getting these Query messages, all member hosts joining the corresponding channel reply with own Current-State Report messages. This condition requires transmitting a number of Current-State Report messages and consumes network resources especially when many hosts have been joining the same channel.

On the other hand, if a router enables the explicit tracking function, it does not need to always ask Current-State Report message transmission to the member hosts whenever it receives the State-Change Report. This is because the explicit tracking function works with the expectation that the State-Change Report sender is the last remaining member of the channel. Even if this expectation is wrong (i.e., the State-Change Report sender was not the sole member), other members remaining in the same channel will reply with identical Report messages, so the end result is the same and no problem occurs. (Section 3 details the point.)

In addition, the processing of IGMP membership or MLD listener reports consumes CPU resources on the IGMP/MLD querier devices itself. Therefore, the explicit tracking function reduces not only the network load but also the CPU load on the querier devices as well.

snooping switch [5]. If the timer to refresh membership record on snooping switch is shorter than the General Query timer interval (i.e. [Query Interval]),

2.2. Shortening Leave Latencies

The explicit tracking function works with the expectation that the State-Change Report sender is the last remaining member of the channel. Thanks to this functionality, a router can tune timers and values related to decide that the State-Change Report sender was the sole member.

The [Last Member Query Interval] (LMQI) and [Last Listener Query Interval] (LLQI) values specify the maximum time allowed before sending a responding Report. The [Last Member Query Count] (LMQC) and [Last Listener Query Count] (LLQC) are the number of Group-Specific Queries or Group-and-Source Specific Queries sent before the router assumes there are no local members. The [Last Member Query Time] (LMQT) and [Last Listener Query Time] (LLQT) values are the total time the router should wait for a report, after the Querier has sent the first query.

The default values for LMQI/LLQI defined in the standard specifications [1][2] are 1 second. For the router enabling the explicit tracking function, LMQI/LLQI would be set to 1 second or shorter. The LMQC/LLQC may be set to "1" for the router, whereas their default values are the [Robustness Variable] value whose default value is "2". Smaller LMQC/LLQC give smaller LMQT/LLQT; this condition shortens the leave latencies.

2.3. Considerations

As with the basic concepts of IGMP and MLD, the explicit tracking function does not guarantee the membership state is always perfectly synchronized; routers enabling the explicit tracking function still need to send IGMPv3/MLDv2 Query messages and inquire solicited IGMPv3/MLDv2 Report messages from downstream members to maintain downstream membership state.

- o IGMP/MLD messages are non-reliable and may be lost in the transmission, therefore routers need to confirm the membership by sending Query messages.
- o To preserve compatibility with older versions of IGMP/MLD, routers need to support downstream hosts that are not upgraded to the latest versions of IGMP/MLD and run the report suppression mechanism.

- o It is impossible to identify hosts when hosts send IGMP reports with a source address of 0.0.0.0.

Regarding the last bullet, the IGMPv3 specification [1] mentions that an IGMPv3 Report is usually sent with a valid IP source address, although it permits that a host uses the 0.0.0.0 source address (as it happens that the host has not yet acquired an IP address), and routers MUST accept a report with a source address of 0.0.0.0. The MLDv2 specification [2] mentions that an MLDv2 Report MUST be sent with a valid IPv6 link-local source address, although an MLDv2 Report can be sent with the unspecified address (::), if the sending interface has not acquired a valid link-local address yet. [2] also mentions that routers silently discard a message that is not sent with a valid link-local address or sent with the unspecified address, without taking any action, because of the security consideration.

Another concern is that the explicit tracking function requires additional processing capability and a possibly large memory for routers to keep all membership states. Especially when a router needs to maintain a large number of member hosts, this resource requirement may be potentially-impacted. Operators may decide to disable this function when their routers do not have enough memory resources.

3. Membership State Information

The explicit tracking function is implemented with the following membership state information:

(S, G, number of receivers, (receiver records))

where each receiver record is of the form:

(IGMP/MLD Membership/Listener Report sender's address)

This state information must work properly when a receiver (i.e., Report sender) sends the same Report messages multiple times.

In the state information, each "S" and "G" indicates a single IPv4/IPv6 address. "S" is set to "Null" for an Any-Source Multicast (ASM) communication (i.e., (*,G) join reception). In order to simplify the implementation, the explicit tracking function does not keep the state of (S,G) join with EXCLUDE filter mode. If a router receives (S,G) join/leave request with EXCLUDE filter mode from the downstream hosts, it translates the join/leave request to (*,G) join state/leave request and records the state and the receivers' addresses into the maintained membership state information. Note that this membership

state translation does not change the routing protocol behavior; the routing protocol must deal with the original join/leave request and translate the request only for the membership state information.

4. Multicast Router Behavior

The explicit tracking function makes routers expect whether the State-Change Report sender is the last remaining member of the channel. Therefore the router transmits a corresponding Current-State Report message only when the router thinks that the State-Change Report sender is the last remaining member of the channel. This contributes to saving the network resources and also shortening leave latency.

To synchronize the membership state information, when a multicast router receives a Current-State or State-Change Report message, it adds the receiver IP address to or delete from the receiver records or creates the corresponding membership state information. If there are no more receiver records left, the membership state information is deleted from the router.

However, the membership state information may be still outdated in the router. It may be happened especially in a mobile multicast environment that some member hosts have joined to or left from the network without sending State-Change Report messages. Or, some State-Change Report messages are lost due to network congestion. Therefore, the router enabling the explicit tracking function ought to send the periodic General Query regularly.

Regarding the leave latency, as specified in Section 2.2, the explicit tracking function contributes to the fast leave by setting LMQI/LLQI to "1" second or shorter and LMQC/LLQC to "1". However, if LMQC/LLQC is configured "2" or bigger value, then the router's behavior may be changed from the standard specification. According to [1] and [2], a router sends a Group- (and-Source) Specific Query [LMQC - 1] or [LLQC - 1] times when it receives State Change Report message (e.g. leave request) from a member host, in order to confirm whether or not the host is the only remaining member. However, this document RECOMMENDS that if the router enabling the explicit tracking function receives the corresponding Current State Report before the Specific Query retransmission, it cancels sending the same Specific Query for other [LMQC - 1] or [LLQC - 1] times.

Note that there is some risk that a router misses or loses Report messages sent from remaining members if the router adopts small LMQC/LLQC; however the wrong expectation would be lower happened for the router enabling the explicit tracking function. And to avoid the

problem, a router can start sending a Group- (and-Source) Specific Query message when it expects the number of the remaining members is small, such as 5, but not 0.

5. Interoperability and Compatibility

The explicit tracking function does not work with the older versions of IGMP or MLD, IGMPv1 [6], IGMPv2 [7] or MLDv1 [8], because a member host using these protocols adopts a report suppression mechanism by which a host would cancel sending a pending membership Reports if a similar Report was observed from another member on the network.

If a multicast router enabling the explicit tracking function changes its compatibility mode to the older versions of IGMP or MLD, the router should turn off the explicit tracking function but should not flush the maintained membership state information (i.e., keep the current membership state information as is). When the router changes back to IGMPv3 or MLDv2 mode, it would resume the function with the kept membership state information, even if the state information is outdated. This manner would give "smooth state transition" that does not initiate the membership state from scratch and synchronizes the actual membership state smoothly.

There are several points TBD in the further discussions regarding the interoperability and compatibility issues. At first, it is necessary whether a multicast router enabling the explicit tracking function needs to detect adjacent routers that do not support the explicit tracking function on the link or not. After the clarification, this document will describe the method how to detect them. It would be done by a new signaling message, but the new message leads compatibility problems for older routers or other routing protocols such as PIM-DM. All of these discussions are TBD.

6. Security Considerations

There is no additional security considerations.

7. Acknowledgements

Toerless Eckert, Stig Venaas, and others provided many constructive and insightful comments.

8. References

8.1. Normative References

- [1] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, October 2002.
- [2] Vida, R. and L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, June 2004.
- [3] Liu, H., Cao, W., and H. Asaeda, "Lightweight Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Version 2 (MLDv2) Protocols", RFC 5790, February 2010.

8.2. Informative References

- [4] Fenner, B., He, H., Haberman, B., and H. Sandick, "Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD)-Based Multicast Forwarding ("IGMP/MLD Proxying")", RFC 4605, August 2006.
- [5] Christensen, M., Kimball, K., and F. Solensky, "Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches", RFC 4541, May 2006.
- [6] Deering, S., "Host Extensions for IP Multicasting", RFC 1112, August 1989.
- [7] Fenner, W., "Internet Group Management Protocol, Version 2", RFC 2373, July 1997.
- [8] Deering, S., Fenner, W., and B. Haberman, "Multicast Listener Discovery (MLD) for IPv6", RFC 2710, October 1999.

Authors' Addresses

Hitoshi Asaeda
Keio University
Graduate School of Media and Governance
5322 Endo
Fujisawa, Kanagawa 252-0882
Japan

Email: asaeda@wide.ad.jp
URI: <http://web.sfc.wide.ad.jp/~asaeda/>

Nicolai Leymann
Deutsche Telekom AG
Winterfeldtstrasse 21-27
Berlin 10781
Germany

Email: n.leymann@telekom.de

PIM Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 14, 2013

H. Liu
Huawei Technologies
T. Tsou
Huawei Technologies (USA)
July 13, 2012

PIM MTU Hello Option for PIM Message Encapsulation
draft-lts-pim-hello-mtu-01

Abstract

This memo introduces a new PIM Hello MTU Option which is carried in PIM Hello messages. The MTU option enables interface MTU information to be exchanged among PIM neighbors, and PIM messages to be encapsulated in an efficient and consistent way.

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 14, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	3
2. Terminology	4
3. MTU Option and its Operation Rule	4
4. Option Format	5
5. IANA Considerations	5
6. Security Considerations	5
7. Acknowledgements	6
8. Normative References	6
Authors' Addresses	6

1. Introduction

A PIM router often needs to preserve a great many (*,G) or (S,G) multicast forwarding states to enable traffic forwarding for large scale of multicast channels. These states are usually set up and kept alive by each downstream router periodically sending Join Messages carrying its own forwarding states to its upstream neighbor. For each round of assembling these states into a PIM message, multiple segments of packets might be generated due to the MTU limitation on the sending PIM interface.

Current implementation uses merely sending link MTU to calculate maximum PIM packet length without considering the receiving MTU of the neighbor(s). It has some drawbacks because if the MTU of the sending interface is larger than that of the receiving one, PIM protocol packets encapsulated according to the sending MTU will most possibly be discarded by the receiving router and the forwarding states cannot be properly established as a result. There are already faults being reported caused by inconsistent MTU configuration among PIM neighbors.

Even though the problem could be resolved by requiring each PIM downstream interface to take less or equal MTU value than its upstream interface, it is inflexible for operation and does not scale because the interface or link conditions across the network might be diverse in practice. As a remedy, this memo recommends exchanging link MTU information among PIM neighbors by using a new Hello MTU Option. The option is carried in periodical PIM Hello messages for a router to inform its receiving link MTU parameter on an interface to the connected neighbor(s), so that the MTU information could be referenced by the neighbor(s) when they are sending PIM protocol messages on this link.

PIM MTU Option can be applied to all variants of PIM protocols, i.e., PIM-SM, PIM-SSM, PIM-DM, and BIDIR-PIM, on both IPv4 and IPv6 networks. There is an exception for the processing of PIM-SM Register/Register-Stop Message, which should reference the MTU information on the entire path between source DR and RP, as described in 4.4.1 of [RFC4601].

It should be noted that PIM MTU Option extension is different from multicast PMTU discovery mentioned in [RFC1981]. Section 5.2 of RFC1981 describes that an implementation could maintain a single PMTU learned across the whole multicast distribution tree. This might result in using smaller packets than necessary for a lot of paths. And because the end to end paths can be very dynamic it could make the effort too complex. This PMTU is used in encapsulating a 'multicast data packet' to avoid fragmentation in multicast data

plane as the packet travels on all paths of the tree. Whereas PIM MTU option works in control plane and has a per-hop nature - it only functions between adjacent one-hop PIM neighbors to guide the sending of a 'PIM protocol message'.

The maintenance of MTU in control plane (by PIM Hello MTU Option) and data plane (by PMTU) are for different purposes and are run independently - the control plane makes sure that forwarding paths are setup even there exists asymmetric MTUs on different links, while the data plane is to make multicast delivery efficient by avoiding fragmenting/reassembling operation, which could be done by means of acquiring minimal MTU on all paths, and of applying it in generating a data packet on first-hop or head-end. Control plane cannot preclude fragmentation, but it is the premise of normal data forwarding - even if some data packets exceeding limitation of some points of the paths cannot be processed properly, other packets meeting the PMTU requirements will be normally forwarded and delivered.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. MTU Option and its Operation Rule

To record the minimum usable sending MTU value on an interface, a new General Purpose non-group-specific state - Sending MTU state is introduced in PIM protocols (for General Purpose State referring to 4.1.1 of [RFC4601] and [RFC3973], and 3.1.1 of [RFC5015]). It is 32-bit long and is unique on an interface whether the link connected is point-to-point or multi-accessed. The initial value of the Sending MTU state should be set to the outbound MTU of the interface, taking either the configured MTU or the default MTU value (referring to 7.1 of [RFC1191] for common MTU for different link types).

When an MTU Hello Option is received from a neighbor, a PIM router parses the MTU value in the option and decides whether or not it should accept the value and store it in the Sending MTU field. A router should not accept too small a value to prevent extreme fragmentation from deteriorating the router's performance. If the MTU value is valid from a legal neighbor, it compares the value with the MTU value currently stored in the Sending MTU field, and makes the replacement if the former is less than the latter.

Unlike other PIM Hello option, MTU Option is not required being supported simultaneously by all PIM neighbors connecting to a network. An MTU-capable router only considers the MTU of a trusty neighbor from which a valid MTU option is received. An MTU-capable PIM router should use MTU option in its Hello message, and should keep the Sending MTU state to the initial value if no neighbor reports a valid MTU Option. Finally, an MTU-incapable router should ignore an MTU option on reception.

The Sending MTU state should be checked before sending a multicast PIM message, to ensure the length of the message does not exceed the MTU limit of both the sending and receiving links. It should be noted that as a convention, the length calculation starts from the beginning of an IP header.

4. Option Format

A Hello MTU Option has the following format:

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               |                               |
|      Type = TBD              |      Length = 4               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               |
|      Value = inbound MTU of this interface                    |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Type: to be assigned by IANA if this option is accepted. The field is 16-bit long.

Length: the length of the Value field. The field is 16-bit long.

Value: inbound MTU value for this interface. The field is 32-bit long.

5. IANA Considerations

The Type field should be allocated by IANA if MTU option is accepted.

6. Security Considerations

The potential security threat for MTU option should be the denial-of-service attack of extremely fragmenting PIM messages, by advertising much smaller MTU value than necessary. A remedy is to require a PIM router to check the validity of a neighbor's MTU value

before accepting it.

7. Acknowledgements

The authors would like to acknowledge Hou Yunlong, Mach Chen, Liu Yisong, Stig Venaas, Bill Fenner, Dino Farinacci, and Chiranjeevi Ramana Rao for their valuable comments and discussions on the work.

8. Normative References

- [RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, November 1990.
- [RFC1981] McCann, J., Deering, S., and J. Mogul, "Path MTU Discovery for IP version 6", RFC 1981, August 1996.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3973] Adams, A., Nicholas, J., and W. Siadak, "Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised)", RFC 3973, January 2005.
- [RFC4601] Fenner, B., Handley, M., Holbrook, H., and I. Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", RFC 4601, August 2006.
- [RFC5015] Handley, M., Kouvelas, I., Speakman, T., and L. Vicisano, "Bidirectional Protocol Independent Multicast (BIDIR-PIM)", RFC 5015, October 2007.

Authors' Addresses

Liu Hui
Huawei Technologies
Building Q14, No.156, Beiqing Rd.
Beijing 100095
China

Phone: 8610-60610012
Email: helen.liu@huawei.com

Tina Tsou
Huawei Technologies (USA)
2330 Central Expressway
Santa Clara CA 95050
USA

Phone: +1 408 330 4424
Email: Tina.Tsou.Zouting@huawei.com

INTERNET-DRAFT
Intended Status: Standards Track
Expires: Expires January 10, 2013

B. Tao, Ed.
Huawei Technologies
Others
July 9, 2012

MPLS PIM Inter-working
draft-cao-mpls-pim-interworking-00

Abstract

This document describes a framework for the inter-working between Protocol Independent Multicast [PIM] and a leaf-driven P2MP tunnel signaling protocol such as [mRSVP-TE] or [mLDP] so that multiple PIM sites around an MPLS network can form a single PIM domain without compromising PIM's features, scalability, and performance.

In this document, PIM modes PIM-SM, PIM-SSM, and PIM-BIDIR are considered.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

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

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

Copyright and License Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1	Introduction	4
1.1	Background	4
1.2	Purpose of This Work	5
1.3	Terminology	6
2.	An Overview: PIM MPLS Inter-working	6
2.1	PIM-SM, PIM-SSM and PIM-BiDir States	6
2.2	PIM-MPLS Border Router(mPMBR)	7
2.3	A Reference Model for PIM-MPLS Inter-working(PMIW)	9
2.3.1	QPI, MPLS Tunnel and M-Flow Spec Binding	10
2.3.2	PIM Support for QPI and M-Flow Spec Binding	10
2.3.3	M-Flow Spec Binding Policies	11
2.3.4	IP Multicast Packet Forwarding at an mPMBR	11
2.4	Impacted PIM messages and procedures	11
2.4.1	PIM Hello and Adjacency Over MPLS Backbone	12
2.4.2	PIM Assert and Message	12
2.4.3	PIM hop-by-hop Bootstrapping and Message	12
2.4.4	PIM Unicast Messages and C-RP Advertisement	12
2.4.5	PIM RP Register and RegisterStop	13
2.4.6	PIM Join/Prune States and In-Band Signaling in MPLS	13
2.4.6.4	Aggregating Two Tunnels to Use A Single Tunnel	14
3	Algorithms and Procedures	14
3.1	Dynamic P2MP Tunnel Creation and Bind An M-Flow Spec To It	14
3.1.1.	In-Band Tunnel Signaling at Leaf LER	16
3.1.2.	In-Band Tunnel Signaling at Transit LSR	17
3.1.3.	In-Band Tunnel Signaling at Root LER	18
3.1.4.	Considerations for Load Balance and Traffic Engineering(TE)	18
3.2.	Operational Procedures for PIM RPT to SPT switch	19
4.	OAM & P	19
5	Security Considerations	20
6	IANA Considerations	20

7	References	20
7.1	Normative References	20
7.2	Informative References	20
	Authors' Addresses	21
	Acknowledgement	21

1 Introduction

1.1 Background

IP multicast data sources and their receivers can be located at multiple separated sites which are around an MPLS backbone. PIM, the most popular IP multicast protocol, runs in each of these sites. A requirement therefore is to use the MPLS backbone tunnels to "connect" these multiple PIM sites as if they were in a single multicast domain under PIM.

Currently there are a few standards to achieve this, most of them for multicast VPN(mVPN) cases:

- a) [RFC6513] and [RFC6514] use a third protocol, the extended BGP, to discover multicast routes and other states from each PIM site, propagate to other sites, and establish P2MP tunnels in the MPLS backbone to support VPN multicast within MPLS backbone.
- b) [I-D.lin-mrsvp-te-mvpn] uses an mRSVP-TE [mRSVP-TE] tunnel within MPLS to transparently multicast both PIM's control and data traffic to other VPN sites, and if necessary, establishes a separated P2MP tunnel for each individual multicast flow. This is an out-of-band method to signal VPN PIM states over MPLS backbone. In this way, separated PIM sites of a VPN form a single PIM domain in the VPN, and PIM adjacencies each pair of MPLS edge routers are maintained across the MPLS backbone.
- c) [I-D.Wijnands-mldp-inband] uses in-band signaling to build mLDP P2MP tunnels to support (S, G) and (*, G) multicast states. Currently, the draft only specifies the encoding and decoding for the above two types of multicast states. It relies on a third protocol to signal PIM ASM related states.

These methods, however, either support a limited portion of PIM features, or, with a concentration in mVPNs, have remaining issues for both network operators and equipment vendors.

[RFC6513] extends BGP to support PIM features cross over an MPLS backbone for multicast VPN cases, however, the support is made critically dependent on the extensions to the third protocol. Some of PIM features such as BSR bootstrapping are not supported yet. To fully support PIM and its future extensions, more extensions to BGP must be made as well, besides those for PIM and the MPLS protocols. This dependency causes additional requirements and complexity for

both network operators and equipment vendors.

The extra protocol involvement also introduces more interactions among protocols and thus causes additional overheads to BGP. In addition, [RFC6513] uses a BGP discovery phase for a PIM state before a tunnel can be signaled in the MPLS backbone. This adds a delay to the dynamic tunnel signaling.

The out-of-bound method limits itself as a solution for particular cases because: a) It applies to a particular MPLS signaling protocol [mRSVP-TE]; b) Fully meshed PIM adjacencies over MPLS are costly and can have scalability concern (See [RFC6517]); c) The default tunnel may not be optimally routed within MPLS and its usage prevents flexible load balancing and traffic engineering at tunnel level; only limited aggregation can be done on (S, G) data tunnels; d) PIM RPT to SPT switch semantic is changed and new procedures and messages are used to discover a (S, G) and propagate it to other sites.

In the third solution, the supported PIM features are limited. Besides, it applies only to [mLDP] as the MPLS signaling protocol.

See [RFC6517] for more information.

1.2 Purpose of This Work

In this document, we introduce a PIM-MPLS inter-working framework which provides the following to resolve the current issues:

- a) Complete the full support of PIM features with only PIM and a point-to-multipoint (P2MP) tunneling protocol involved;
- b) Neutrally support various tunnel signaling protocols, including [mRSVP-TE] and [mLDP]; PIM states are "in-band" signaled by the tunnel signaling protocol to another PIM site while the signaling protocol itself uses the data to set up the tunnel with optimal routing and traffic engineering;
- c) Minimize the changes to the two(2) involved protocols and the introduction of new procedures;
- d) Provide flexible tunnel aggregation and load balancing for scalability and shared resource usage in backbone
- e) Minimize overheads in the backbone and on the PIM-MPLS border routers without introducing performance and scalability bottlenecks. There is no PIM adjacencies cross over the backbone network

It is important to point out that some existing solutions can be made

to work with this framework to have complete PIM support.

[mRSVP-TE] and [mLDP] are protocols to signal point-to-multipoint (P2MP) and multipoint to multipoint (MP2MP) tunnels in an MPLS network, starting from the leaves of these tunnels. The leaf-driven signaling is in the same direction as PIM builds its multicast forwarding information base, i.e., from the multicast data listeners to the senders. This framework will take advantage of this characteristics to set up the forwarding states in an MPLS backbone with less messages and delays.

1.3 Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. An Overview: PIM MPLS Inter-working

2.1 PIM-SM, PIM-SSM and PIM-BiDir States

[PIM] defines four(4) types of Multicast Routing Information Base(MRIB) entries:

1. (*, *, RP)
2. (*, G)
3. (S, G)
4. (S, G, rpt)

For each MRIB entry, the framework identifies the following PIM forwarding states for our purpose in this document:

Downstream Per-interface Join/Prune state:

One of {"NoInfo" (NI), "Join" (J)}

Upstream non-interface specific Join/Prune state:

One of {"NotJoined", "Joined"}

For each $(*, G)$, there is a sub-set of states called (S, G, RPT) , one for each (S, G) pair. In this draft, we are only concerned with their following states:

Downstream Per-interface Join/Prune state:

One of {"NoInfo", "Pruned"}

Upstream non-interface specific Join/Prune State:

One of {"RPTNotJoined(G)", "NotPruned(S,G,rpt)",
"Pruned(S,G,rpt)"}

For definitions of these states, readers are referred to the [PIM] document.

This framework makes these PIM states as part of signaling data for a leaf-driven MPLS protocol to signal its P2MP tunnels to achieve optimal multicast routing within the MPLS network and in highly scalable fashion. On the other hand, the remote PIM site at upstream obtains these states from the MPLS signaling data to set up proper PIM forwarding states for the upstream site. These PIM states are therefore "In-Band" signaled to the remote PIM site by MPLS, while MPLS itself sets up optimized multicast LSPs for the traffic to go through the MPLS backbone.

We hereafter call them M-Flow Specs, and for in-band signaling purpose, assign a value to each of the types as the following:

M-Flow Spec Type-1(value 1) for $(*, *, RP)$;
M-Flow Spec Type-2(value 2) for $(*, G)$;
M-Flow Spec Type-3(value 3) for (S, G) ;
M-Flow Spec Type-4(value 4) for (S, G, RPT) ;

2.2 PIM-MPLS Border Router(mPMBR)

An mPMBR is a border router where PIM meets the MPLS backbone and inter-works with an MPLS signaling protocol to set up the tunnels, and where IP multicast packets exit an upstream PIM site to enter the MPLS backbone or exit the MPLS backbone to enter a downstream PIM site. An mPMBR can run a multicast member discovery protocol such as

IGMP or MLD, besides PIM. Therefore the mPMBR can have local listeners.

An mPMBR is called local to a PIM site if it is where the PIM site connects to the MPLS backbone.

For convenience in the following discussions, we define the following macro to determine where a P2MP tunnel ingress, or root, will be, for each M-Flow spec F defined previously:

```
mPMBR_lkup(F) = {  
    RP's local mPMBR address, if F is a (*, *, RP) spec; or  
    RP(G)'s local mPMBR address if F is a (*, G) spec; or  
    S's local mPMBR address if F is a (S, G) spec; or  
    RP(G)'s local mPMBR if F is a (S, G, RPT)  
}
```

An mPMBR needs to specify its router ID and advertise it to unicast routing protocol(s) to make the address reachable from all other mPMBRs. It also specifies its PIM interfaces that face a PIM site, as well as one or more MPLS interfaces over which P2MP tunnels can be signaled to support the inter-working functions as defined in this work. In this framework, P2MP tunnels and their sub-LSPs are dynamically created and removed when M-Flow specs are created or removed on mPMBRs.

When a tunnel is created for an M-Flow spec, a logic interface is also created at the tunnel's ingress and egress endpoints, respectively. This logic interface will act as if it were a PIM interface but it does not actually run PIM on it. At an P2MP egress mPMBR, PIM builds non-interface upstream states on it using the M-Flow spec created by PIM; at the ingress, or P2MP root mPMBR, PIM builds per-interface downstream states using the M-Flow spec data signaled by the tunnel signaling protocol.

An M-Flow spec is said to be "bound" to such a logic interface once the interface is created as above to pass the traffic which will be forwarded per the M-Flow spec by the IP multicast forwarding plane. At a P2MP tunnel's ingress mPMBR, IP multicast packets of the bound M-Flow spec enters this logic interface, which "leads" the packets into the MPLS tunnel. At an egress mPMBR, the packets exit the tunnel and were treated as if they were received from the logic interface as an upstream interface. At this point the packets will be forwarded using the native IP multicast forwarding rules.

We call each of these logic interfaces a Quasi-PIM interface(QPI).

2.3 A Reference Model for PIM-MPLS Inter-working (PMIW)

Figure 1 illustrates the PIM-MPLS inter-working model on an mPMBR. In this model, a QPI is created for an MPLS tunnel at the ingress and egress LSRs, and this QPI is "advertised" to the mPMBR's PIM, so that PIM uses it as if it were a PIM interface except that PIM protocol does not actually run on it, and therefore PIM does not send or receive any PIM protocol control messages over it (i.e. there is no PIM adjacency on a QPI), but can still send and receive IP multicast data packets.

The PIM on each mPMBR uses native PIM procedures to work with its PIM site router(s) and build proper PIM control and multicast packet forwarding states over the PIM interfaces as well QPIs.

In order for the mPMBR PIM to build proper control and forwarding states for a QPI, PIM procedures must be modified to

- i) Extend any validation checks to include the QPI to accept IP multicast packets from the backbone;
- ii) PIM RFP_Interface() macro can return a QPI if the RPF next-hop goes over an IP interface that has MPLS enabled to support the inter-working.

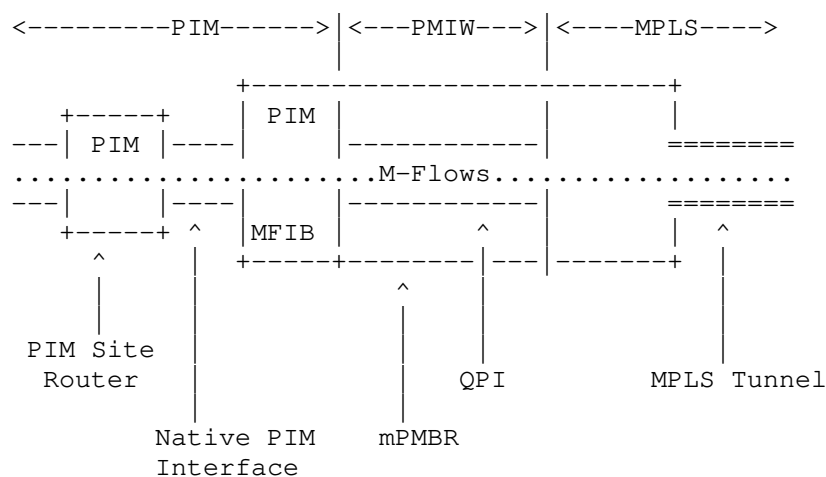


Figure 1: mPMBR PIM-MPLS Inter-working(PMIW) Reference Model

In the Figure 1, an IP multicast packet in mPMBR's PIM segment is forwarded over the PIM interface(s) as well as QPIs using PIM's native forwarding rules; when an IP multicast packet enters a QPI, the packet will be encapsulated with MPLS and enters the corresponding tunnel. When a packet exits the tunnel at the remote mPMBR, the QPI on this receiving mPMBR becomes the incoming interface for the packet, and the packet enters PIM's segment where it will be forwarded under PIM's native rules.

2.3.1 QPI, MPLS Tunnel and M-Flow Spec Binding

When a P2MP tunnel is created in the MPLS backbone for an M-Flow spec, a QPI is also created as an IP multicast logical interface at each of the egress and ingress mPMBRs, and the M-Flow spec is bound to the QPIs at each of the mPMBRs respectively. At the ingress mPMBR, the QPI is where an IP multicast packet enters the P2MP tunnel with the MPLS header, and is subsequently forwarded along the tunnel. At each egress mPMBR, the QPI associated with the tunnel is where the MPLS packet is de-capsulated, and the resulted IP multicast packet continues to be forwarded using PIM's native forwarding rules.

The QPI operational status is the same as the P2MP tunnel operational state.

At an ingress mPMBR, an M-Flow spec F is said to be "bound" to a QPI (therefore the tunnel as well) when the ingress mPMBR sets the IP multicast forwarding rules of F to use the QPI as a downstream interface in order to carry the traffic of F to other PIM sites, via the backbone network.

At an egress mPMBR, an M-Flow spec F is said to be "bound" to a QPI (therefore the tunnel as well) when the egress mPMBR sets the IP multicast forwarding rules of F to use the QPI as an upstream interface in order to receive the traffic of F from another PIM site, via the backbone network.

2.3.2 PIM Support for QPI and M-Flow Spec Binding

In this framework, QPIs are made to PIM as if they were a PIM interface, except that PIM control messages do not go over the QPIs. The implementation needs to establish proper PIM per-interface downstream states and upstream states for the QPI and the data signaled by MPLS, which is originated from other PIM interface states and the MPLS signaled data from the remote PIM sites.

The actual implementation of the binding on each mPMBR is beyond the scope of this work.

The detailed PIM protocol support for the QPI will be completed in a later version.

2.3.3 M-Flow Spec Binding Policies

This framework introduces two sets of policies to restrict the binding of an M-Flow to a Tunnel.

1. Default Policy: AGG_POLICY_0 (value: 0)
 - a. One tunnel per (S, G) M-Flow spec; and
 - b. One tunnel per RP for (*, *, RP) M-Flow spec; and
 - c. One tunnel for all M-Flow specs of (*, G) such that RP(G) gives the same RP

The default policy is used by a LSR when no other policy is explicitly specified. It is the finest grained, and MUST be provided by an implementation.

2. AGG_POLICY_1 (value: 1):
 - a. A separate P-Tunnel is used to aggregate all (S, G) M-Flow specs such that mPMBR_lkup((S,G)) gives the same mPMBR; and
 - b. A separate P-Tunnel is used to aggregate all (*, *, RP) M-Flow specs such that mPMBR_lkup((* ,*,RP)) gives the same mPMBR; and
 - c. A separate P-Tunnel is used to aggregate all (*, G) M-Flow specs such that mPMBR_lkup((* ,G)) gives the same mPMBR.

AGG_POLICY_1 is the most coarse grained and it, besides others, can be optionally provided by an implementation.

2.3.4 IP Multicast Packet Forwarding at an mPMBR

If an IP multicast packet arrives at an mPMBR from a PIM interface, it is forwarded using native IP multicast rules. If an oif is a QPI, the QPI makes the packet to be forwarded into the corresponding MPLS P2MP tunnel.

If an IP multicast packet arrives at an egress mPMBR from a P2MP tunnel, it is handled by the bound QPI, which acts as an iif for IP multicast flow. If there is no bound QPI, the packet is dropped.

2.4 Impacted PIM messages and procedures

2.4.1 PIM Hello and Adjacency Over MPLS Backbone

An mPMBR does not build any PIM adjacency with any other mPMBR over the MPLS backbone. Instead, relevant PIM states are mapped to and from the MPLS signaling data. The details will be covered in later sections when actual procedures are provided. The PIM downstream per-interface states on a QPI is directly mapped from the corresponding MPLS P2MP tunnel's in-band signaled M-Flow spec data. The PIM upstream states, on the other hand, will be in-band signed to other PIM sites by a P2MP tunneling protocol, and at the same time the signaling protocol sets up optimally routed P2MP tunnel within the MPLS for the multicast flows.

There are no impacts to other routers within MPLS backbone or in PIM sites.

2.4.2 PIM Assert and Message An implementation following this framework will not need to make any changes to for PIM asserts, as they will be only applicable inside a PIM site locally, and the framework does not let PIM go cross the backbone.

There are no impacts to any router in MPLS backbone or in PIM sites.

2.4.3 PIM hop-by-hop Bootstrapping and Message

A designated multicast channel within MPLS backbone, called Multicast Bootstrap Tunnel (MBT), is used to carry PIM's bootstrap messages among all mPMBRs. This tunnel can be either an MP2MP or a bi-directional P2MP tree. The root is designated by the MPLS network operator and leaves are all mPMBRs. An MPLS packet entered into the MBT from any mPMBR will reach all other mPMBRs.

At each mPMBR, PIM sends and receives bootstrap messages to each of the mPMBR's PIM interfaces as well as the MBT. Each mPMBRs must implement the functions to send and receive PIM bootstrap messages over the MBT.

There are no other impacts to an mPMBR PIM's native bootstrap procedures, and there are no impacts to other routers other than the mPMBRs.

2.4.4 PIM Unicast Messages and C-RP Advertisement

PIM unicast messages, including CRP advertisement, Register and RegisterStop, are sent and received in raw IP, with PIM protocol number.

Each mPMBR must be able to receive a raw IP PIM packet arrived at a non-PIM interface that is MPLS enabled to support PIM-MPLS inter-working.

There are no other impacts to PIM's native procedures for unicast messages on an mPMBR, and there are no impacts to other routers other than mPMBRs.

2.4.5 PIM RP Register and RegisterStop

Except for the changes common to PIM unicast messages as in the previous subsection, there are no other impacts for PIM RP register and registerStop.

For information purpose, a RP may choose to send a (S, G) join toward the source S after an (S, G) register packet is received by the RP, and the resulted tree may go over the MPLS network. In this case, the mechanism and procedures for (S, G) SPT support apply. The details will be in later subsections.

2.4.6 PIM Join/Prune States and In-Band Signaling in MPLS

An mPMBR, say mpmb, can have four(4) types of M-Flow specs corresponding to PIM's upstream states. Except for the M-Flow spec Type-4, each of the rest, say, F at the mpmb can bind to an MPLS P2MP tunnel in the MPLS network with mpmb as one of its leaf(egress) LSRs, and the root set to mPMBR_lookup(F). A signaling protocol which is to work under this framework will support the binding of the Type-1, Type-2, and Type-3 M-Flow specs with its tunnels. A Type-4 M-Flow spec is associated with a Type-2 M-Flow spec, as an (S, G, RPT) state is embedded into a (*, G) state. Therefore there is no separate binding for a Type-4 M-Flow spec.

Before an M-Flow spec F is bound to a tunnel, the to-be bound tunnel may have some undetermined information such as a tunnel identifier, but the tunnel signaling procedure uses F and other known tunnel identification data during the tunnel signaling. After the M-Flow spec is bound to a tunnel, tunnel's identification data will be completed.

The binding can happen at the leaf, at a transit LSR, or at the root, according to the binding procedure in "Algorithms and Procedures".

2.4.6.4 Aggregating Two Tunnels to Use A Single Tunnel

Two tunnels may be aggregated into a single one, if their M-Flow specs can be merged to form a super set of M-Flow specs, without violating each original tunnel's aggregation policy. The policy tests are performed using the algorithms and procedures as defined in Section 3.

The merged tunnel can be set up using Make-Before-Break(MBB). They must have the same root in order to be aggregated. The procedure of P2MP MBB is beyond the scope of this draft.

3 Algorithms and Procedures

3.1 Dynamic P2MP Tunnel Creation and Bind An M-Flow Spec To It

This section describes the procedure to bind an M-Flow spec to a tunnel.

First, each M-Flow spec F has an aggregation policy to restrict aggregating the M-Flow spec into a tunnel. This policy can be configured explicitly by an operator, or is the default policy if it is not configured.

An M-Flow spec F has the following attributes:

F.agg_policy:	An policy to restrict binding and aggregating F into a tunnel. This policy can be configured explicitly by an operator, or is the default policy if it is not configured.
F.spec_type:	One of the spec types.
F.bound_tnl:	The MPLS tunnel used by the IP multicasting data which enters and exits the tunnel per F's corresponding PIM forwarding rules.

For simplicity purpose, and without implying actual implementations, the framework uses the following macros and functions on each LER or LSR:

```
TS = {all tunnels on a node that supports PIM-MPLS
      inter-working};
mflow_specs(T) = {set of M-Flow specs that have bound to
                  tunnel T}
mflow_spec_type(T) = The type of the M-Flow specs that are
                     bound to tunnel T
```

```
agg_policy_compatible(F, T) = TRUE if and only if
    T.agg_policy derives F.agg_policy

agg_policy_compatible(T1, T2) = TRUE if and only if
    T1.agg_policy derives T2.agg_policy

bind_candidate(F) {
    foreach(T in TS) {
        if ((T' = F.bound_tnl) != NULL && T' signaling is
            completed)
        {
            return T';
        }
        if (F.spec_type == mflow_spec_type(T) &&
            agg_policy_compatible(F, T))
        {
            return T;
        }
    }
    return NULL;
}

mflow_spec_bind(F, LSR) {
    if ((T = bind_candidate(F)) != NULL)
    {
        mflow_specs_merge(T);
        F.bound_tnl = T;
    }
    else if (I_am_leaf(LSR))
    {
        T = initiate_signal_p2mp_tunnel(F);
        mflow_specs(T) = {F};
        F.bound_tnl = T;
    }
    else if (I_am_root(LSR))
    {
        T = continue_signal_p2mp_tunnel(F);
        mflow_specs(T) = {F};
        F.bound_tnl = T;
    }
    else //I am transit LSR
    {
        T = continue_signal_p2mp_tunnel(F);
        mflow_specs(T) = {F};
    }
}
```

init_signal_p2mp_tunnel(F) triggers a P2MP tunnel creation using the local LER as the leaf, and mPMBR_lkup(F) as the root.

continue_signal_p2mp_tunnel(F) continues the signaling procedure for the P2MP tunnel that was initiated for F.

The following defines M-Flow spec merge operation:

```
mflow_specs_merge(T, F)
{
    switch(F.spec_type)
    {
        case Type-1:
            /* mflow_specs(T) is a set of group ranges each with
               a list of (S, G, RPT) entries; F must be a group
               range with a list of its (S, G, RPT) entries */
            mflow_specs(T) = mflow_specs(T) union {F};
            break;
        case Type-2:
            mflow_specs(T).sg_rpt_joins =
                mflow_specs(T).sg_rpt_joins union F.sg_rpt_joins;
            mflow_specs(T).sg_rpt_prunes =
                mflow_specs(T).sg_rpt_prunes intersect
                F.sg_rpt_prunes
            /* sg_rpt_joins and sg_rpt_prunes are the lists of G's
               joined and pruned(S, G, RPT) entries */
            break;
        case Type-3:
            mflow_specs(T) = mflow_specs(T) union {F};
            /* mflow_specs(T) is a set of (S, G) entries */
            break;
        case Type-4:
            break; /* (S, G, RPT) entries go with wildcards */
    }
}
```

The actual procedures for initiate_signal_p2mp_tunnel(F) and continue_signal_p2mp_tunnel(F) are MPLS signaling protocol specific and will be out of scope for this document.

3.1.1. In-Band Tunnel Signaling at Leaf LER

An M-Flow spec F is instantiated when an mPMBR PIM creates a J/P upstream state. There are three cases under which F may be bound to a

P2MP tunnel at the leaf LER:

Case 1: F is an M-Flow spec already bound to a tunnel egress. Therefore the corresponding QPI of the tunnel is used for F.

Case 2: F is not bound to an existing tunnel yet but F's binding policy is compatible with an M-Flow spec which has been bound to a P2MP tunnel. Therefore, F is then bound to the same tunnel, and its QPI is used for F. The leaf mPMBR does not initiate a new tunnel.

Case 3: None of Case 1 and Case 2 is true. Then a tunnel with some unknown identifier is signaled using an extended MPLS protocol, and F as additional data for in-band signaling. Binding does not happen at this time.

After the unknown tunnel signaling is completed, an upstream node, either a transit or the root should have bound F to the tunnel. In addition, it should have also determined if the tunnel is a new one or an existing one which this M-Flow spec is bound with.

In the former case, a new QPI is created for the new tunnel and bound to F. In the latter case, F is bound to the QPI of the existing tunnel.

3.1.2. In-Band Tunnel Signaling at Transit LSR

As a transit LSR receives a P2MP tunnel signaling message for M-Flow spec F, it does the following:

Case 1: bind_candidate(F) gives a candidate tunnel T, and T is then bound to F. Therefore the LSR becomes a branching LSR. It does the following:

- a. Merge F into T.mflow_specs with
mflow_specs_merge(T, F)
- b. The branching LSR MPLS signaling procedure of specific signaling protocol is then performed

Case 2: Otherwise, it is still an unknown tunnel. It does:

- a. Merge F to T.mflow_specs using

mflow_specs_merge(T, F);

- b. The non-branching LSR MPLS signaling procedure of the specific protocol is then performed.

3.1.3. In-Band Tunnel Signaling at Root LER

Assume an M-Flow spec F is received at the root mPMBR from MPLS backbone.

Case 1: F is an M-Flow spec already bound to a tunnel ingress. Therefore the corresponding QPI of the tunnel is used for F.

Case 2: F is not bound to an existing tunnel yet but F's binding policy is compatible with an M-Flow spec which has been bound to a P2MP tunnel. Therefore, F can then be bound to the same tunnel, and its QPI is used for F.

Case 3: Neither Case 1 or Case 2 can bind the M-Flow spec.

- a. Determine if a new tunnel or an existing tunnel is used for the M-Flow spec, based on binding policy and other requirements; if it is a new tunnel, complete the rest of tunnel signaling using the signaling protocol;
- b. Create a QPI for the tunnel and bound it to F;
- c. The new QPI is added to root mPMBR PIM as a quasi-PIM oif, and F is mapped to PIM's downstream state;
- d. Complete the rest signaling at root with specific tunnel signaling procedures

3.1.4. Considerations for Load Balance and Traffic Engineering(TE)

Each LSR including any transit node in the MPLS backbone uses the combination of aggregation and load-balance policies, as well as traffic engineering requirements to decide if an existing tunnel should be shared for an M-Flow spec, and how to route it if the M-Flow spec is to use a separate tunnel.

For example, a transit LSR may decide to merge a new M-Flow spec into an existing P2MP tunnel to avoid allocating new network resources it; or decides to reserve resources initially to be ready for a new tunnel, but once an upstream LSR decides to use an existing tunnel for the M-Flow spec, it will release the resources it has reserved

earlier. But if eventually a new tunnel is created, the reserved resources will be used by the new tunnel.

3.2. Operational Procedures for PIM RPT to SPT switch

This framework uses mPMBR PIM's native RPT to SPT switch to initiate the corresponding traffic switch from the RPT's P2MP tunnel to the SPT's P2MP tunnel. At the downstream egress mPMBR, when a (S, G, RPT) state is created for the bound QPI, the mPMBR's PIM-MPLS inter-working module adds the corresponding M-Flow spec F of Type-4 to the tunnel's M-Flow specs, using `mflow_specs_merge(T, F)`. The new M-Flow spec is then sent to the root mPMBR.

When F is received by the root mPMBR, the native PIM procedure will be performed at the mPMBR to complete the switch. This procedure determines if (S, G) traffic should be stopped on the RPT as traffic now is being received from the SPT.

4. OAM & P

This section is to be completed in future.

5 Security Considerations

There is no additional security requirement for this work.

6 IANA Considerations

There is no IANA impact from the framework.

7 References

7.1 Normative References

[PIM] B. Fenner, M. Handley, H. Holbrook, I. Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", RFC 4601, August 2006.

[mRSVP-TE] R., Li, Q. Zhao, and C. Jacquenet, "Receiver-Driven Multicast Traffic Engineered Label Switched Paths", draft-lzj-mpls-receiver-driven-multicast-rsvp-te-00 (work in progress), March 2012.

[mLDP] Wijnands, IJ., Minei, I., Kompella, K., and B. Thomas, "Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths", RFC 6388, November 2011.

[RFC4875] R. Aggarwal, D. Papadimitriou, S. Yasukawa, "Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs)", RFC 4875, May 2007

7.2 Informative References

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

[RFC6513] E. Rosen, R. Aggarwal, "Multicast in MPLS/BGP IP VPNs", RFC 6513, Feb. 2012.

[RFC6514] R. Aggarwal, E. Rosen, T. Morin, Y. Rekhter, "BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs",

RFC 6514, Feb. 2012.

[RFC6517] T. Morin, B. Niven-Jenkins, Y. Kamite, R. Zhang, N. Leymann, N. Bitar, "Mandatory Features in a Layer 3 Multicast BGP/MPLS VPN Solution", RFC 6517, Feb., 2012

[RFC6037] E. Rosen, Y. Cai, and IJ. Wijnands, "Cisco Systems' Solution for Multicast in BGP/MPLS IP VPNs", RFC 6037, October 2010.

[I-D.Wijnands-mldp-inband] IJ. Wijnands, Ed., T. Eckert, N. Leymann, M. Napierala, "Multipoint LDP in-band signaling for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths", draft-ietf-mpls-mldp-in-band-signaling-06, December 1, 2011

[I-D.rekhter-pim-sm-over-mldp] Rekhter, Y., Aggarwal, R., and N. Leymann, "Carrying PIM-SM in ASM mode Trees over P2MP mLDP LSPs", draft-rekhter-pim-sm-over-mldp-04, August 2010.

[I-D.lin-mrsvp-te-mvpn] L. Han, "Multicast VPN Support by Receiver-Driven Multicast Extensions to RSVP-TE", draft-hlj-l3vpn-mvpn-mrsvp-te-00, July 2012

Authors' Addresses

Bisong Tao
2330 Central Expressway
Santa Clara, CA 95050
EMail: roberttao@huawei.com

Others (TBD)

Acknowledgement

The author(s) would like to thank the members of Huawei USA IP/MPLS Team for their helpful review comments during the work of this draft.

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: January 18, 2013

T. Taylor
C. Zhou
Huawei Technologies
July 17, 2012

Operation of a Dual-Stack Multicast Router With Address Translation
draft-taylor-pim-v4v6-translation-02

Abstract

This document describes the procedures required for correct operation of a multicast router in a mixed IPv4 and IPv6 environment, where some or all of the multicast group and unicast source addresses can be translated between IPv4 and IPv6, and where incoming multicast data may need to be forwarded into both IPv4 and IPv6 distribution trees. The router is assumed to support Protocol Independent Multicast - Sparse Mode (PIM-SM, RFC 4601) in an environment consisting of a mixture of IPv4 and IPv6 local hosts and PIM peers. The work is easily generalized to other versions of PIM.

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 18, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

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

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

Table of Contents

1. Introduction	3
1.1. Terminology	4
2. Model of Multicast Router Operation	4
3. Principles of Operation	7
4. Modifications To RFC 4601	9
5. Processing of PIM Messages and Multicast Data Packets	9
5.1. Hello Messages	9
5.2. Register and Register Stop Messages	10
5.3. Join/Prune Messages	10
5.4. Assert Messages	10
5.5. Multicast Data Packets	11
6. Acknowledgements	11
7. IANA Considerations	11
8. Security Considerations	11
9. References	11
9.1. Normative References	11
9.2. Informative References	12
Authors' Addresses	12

1. Introduction

During the transition from IPv4 to IPv6, operators need to maintain their services, including multicast services. Depending on how the operator evolves its networks, the situation may arise where sources and receivers support different IP versions, or where some part of the network path between the source and receiver supports one IP version, and a succeeding portion supports the other. [ID.v4v6-multicast-ps] describes a number of potential use cases that can occur.

If IPv4 sources needed to be received only by IPv4 receivers, IPv6 sources needed to be received only by IPv6 receivers, and the network were dual stack, a multicast router could simply run parallel IPv4 and IPv6 PIM state machines with no interaction between them. In the transitional circumstances described above, however, it is necessary to be able to map between IPv4 and IPv6 source and group addresses. Such a mapping introduces interactions between the two PIM state machines within the multicast router. The situation becomes more complicated if, for administrative reasons, no translation is possible/permitted for some addresses. As will be seen below, the changes to PIM operation under these circumstances will not be large, but do require some care.

A number of authors have already worked on multicast translation. One of the earliest works on the topic was [ID.venaas-v4v6mcastgw], which was aimed at giving IPv6 receivers access to IPv4 sources. The document defines a 1-1 mapping of IPv4 multicast group addresses onto a subset of IPv6 addresses defined by a /96 IPv6 multicast prefix. The multicast router is assumed to sit on the boundary between an IPv4 and IPv6 network, and serves as a rendezvous point for all groups within the /96 prefix so that it can keep track of all IPv6 sources and receive all of their data. It appears to the IPv4 network as an IPv4 multicast host.

Succeeding documents have built on the foundations established by [ID.venaas-v4v6mcastgw], taking advantage of advances in translation standardized by the IETF BEHAVE Working Group. Implementations of the gateway concept have also appeared, with [Kiviniemi] as a notable example. [Kiviniemi] is useful for its summary of related developments up to 2009 and for its extensive discussion of the challenges of translation of multicast data packets.

The present document assumes a more general mode of operation. PIM messages are exchanged with IPv4 as well as IPv6 peers. The multicast router is not necessarily the rendezvous point for translated multicast groups. Instead, reliance is placed on the underlying routing tables to ensure that reverse paths pass through

dual stack multicast routers like the one described in this document when translation between IPv4 and IPv6 is required.

Also unlike previous work, the present document takes the details of translation more or less for granted, with the expectation that they are documented elsewhere. (References are given where available.) Its focus is squarely on changes in behaviour required for correct functioning of the multicast router in the assumed environment.

The discussion which follows is framed in terms of a model of multicast router operation. Needless to say, this model is a descriptive device, not a recommendation for implementation. Following the main discussion, Section 5 summarizes the processing required for each PIM message type.

This document assumes the use of Protocol Independent Multicast - Sparse Mode (PIM-SM) [RFC4601]. Its recommendations can easily be generalized to other flavours of PIM.

1.1. Terminology

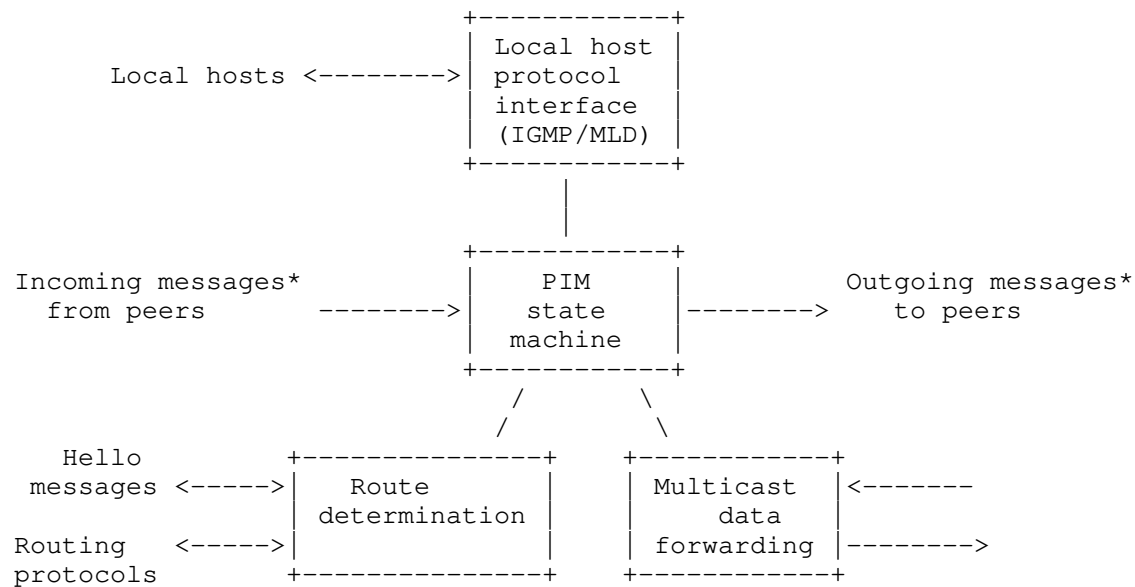
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

This document uses the following terms from Section 2.1 of [RFC4601]:

- o Rendezvous Point (RP);
- o Multicast Routing Information Base (MRIB);
- o Tree Information Base (TIB);
- o RPF Neighbour;
- o upstream;
- o downstream.

2. Model of Multicast Router Operation

Figure 1 provides a model of multicast operation in the absence of address translation. This model consists of four main components: the local host protocol interface, the PIM state machine, route determination (including the MRIB), and the multicast data forwarder.



* PIM Join/Prune, Assert, Register, and Register Stop messages

Figure 1: Model of Multicast Router Operation In the Absence of Address Translation

The local host protocol interface provides the router portion of the host protocol: Internet Group Management Protocol (IGMP) [RFC3376] for IPv4 or Multicast Listener Discovery (MLD) [RFC3810] for IPv6. It passes listener state changes for individual groups and sources to the PIM state machine.

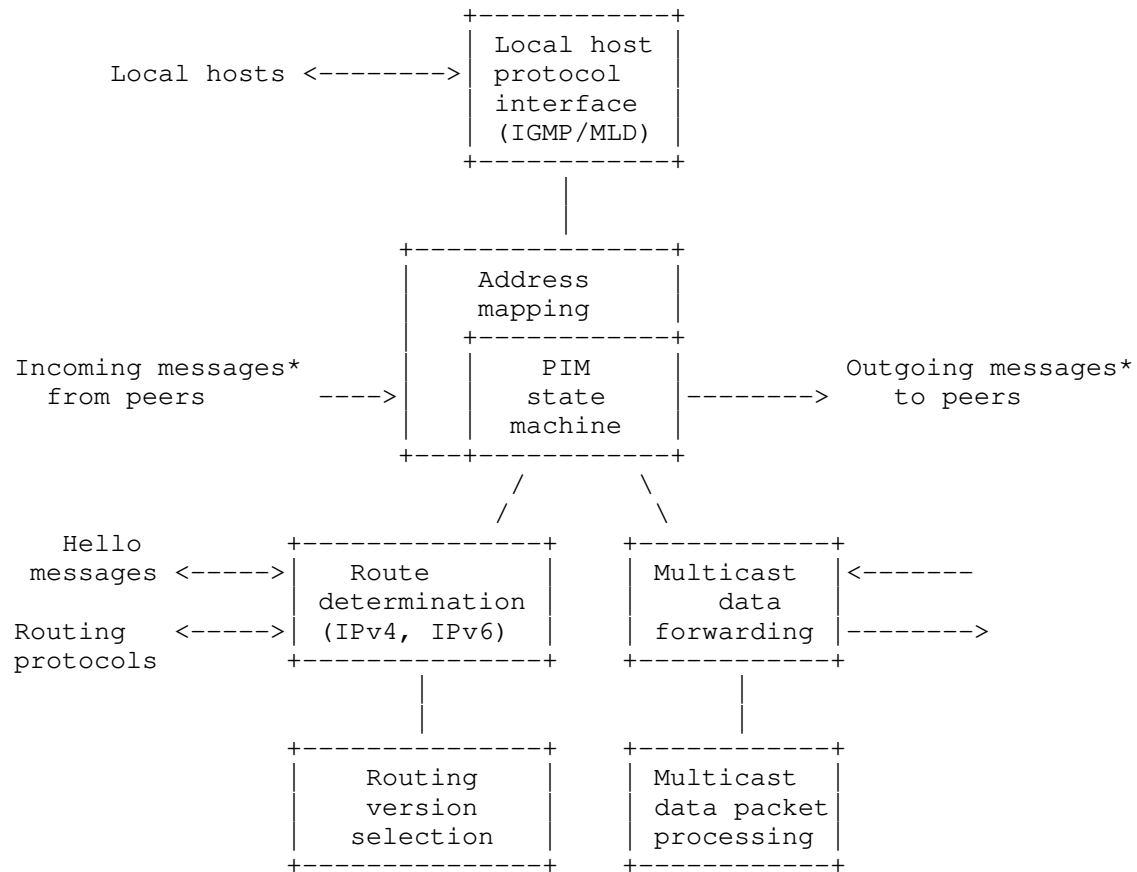
Route determination is built on the Multicast Routing Information Base (MRIB), as well as the secondary address information provided by Hello messages received from neighbouring peers. Upon request from the PIM state machine, it provides the address of the next hop neighbour (RPF neighbour) toward a given rendezvous point (RP) or source, and identifies the interface leading to that neighbour. It also provides metrics in support of Assert processing.

Multicast data forwarding receives multicast data packets, passes the source and destination (multicast group) addresses to the PIM state machine, and is passed in return the identifiers of the interfaces out of which they must be forwarded.

Finally, the PIM state machine executes the protocol procedures described in [RFC4601] and thereby creates and maintains the Tree

Information Base (TIB). As well as the interactions with the other components described above, it exchanges Join/Prune, Assert, Register, and Register Stop messages with its peers.

Figure 2 shows the changes to the model when address translation is introduced. Three new components are added to the picture: address mapping, routing version selection, and multicast data packet processing. Very briefly, address mapping maps source, group, and rendezvous point (RP) addresses between IPv4 and IPv6 in either direction, or returns an indication that no mapping exists. Routing version selection decides whether the next hop toward a rendezvous point or source should be IPv4 or IPv6. Multicast data packet processing accepts a packet of one version and outputs a packet of the other version. This may be accomplished through translation or, possibly, through encapsulation. Further details for all of these components and the changes needed in the PIM state machine will emerge from the discussion that follows.



* PIM Join/Prune, Assert, Register, and Register Stop messages

Figure 2: Model of Multicast Router Operation When Address Translation Is Possible

3. Principles of Operation

This section justifies the changes to the model shown in Figure 2 and provides further details of its operation.

The basic consideration behind the proposed model is that downstream listeners have to be served in the IP version they support, but it is desirable to receive individual streams from upstream in only one version to avoid unnecessary duplication. Address mapping is

unavoidable in this situation. It is actually required for three reasons:

- o internally to the PIM state machine, so state and state-modifying events involving the same source, group, or RP can be collected together regardless of the IP version used to represent its address;
- o externally, to send messages in the appropriate IP version to PIM peers and local hosts; and
- o to support multicast data packet processing when the packet must be forwarded in a different IP version from that in which it was received.

Address mapping can be done at various points in the process. The representation in Figure 2 assumes the following:

- o Source and group addresses in incoming Join/Prune, Assert, and, depending on the role of the router, Register or Register Stop messages are passed to the address mapper. The mapper returns either a corresponding address in the other version or an indication that no mapping is available.
- o Similarly, source and group addresses in the messages received by the local host protocol interface are mapped before the messages are passed to the PIM state machine.
- o The PIM state machine accepts the mapped addresses and indications along with the original messages, and stores them in the state information it maintains.

[To add: references pertaining to the "how" of address mapping.]

As a result, when a message arrives that updates downstream listener state, the PIM state machine is able to relate that state change to the state it already holds for the source or group concerned, even if that earlier state was established by a request in the other IP version. This is because it can match on the mapped counterpart to the address in the earlier request.

Consider now the case that, as a result of downstream events, the PIM state machine decides to send a Join/Prune message upstream. When only one IP version was involved, that was the only IP version that had to be considered when choosing the next hop toward the RP or source. In the situation presented here, however, it is possible that both IPv4 and IPv6 next-hop neighbours are available. A new function, routing version selection, is needed to make the decision.

At this point, no general rule can be given for how routing version selection makes its decision. The obvious initial step is to determine whether the RP or source is reachable in both IP versions. It is assumed for the sake of this discussion that separate IPv4 and IPv6 MRIBs are maintained by the routing determination component. If the target source or RP proves to be reachable by both IPv4 and IPv6, the associated routing metrics can be made available to routing version selection. However, it may well be that these metrics are not comparable. Routing version selection may make use of heuristics, but most likely will be based on local policy. That could take the form of a simple table mapping from target address to preferred next-hop address family.

Once the IP version of the outgoing Join/Prune message has been determined, the PIM state machine can populate the source and group addresses in the message with the same IP version. In the present narrative, this does not require another call to address translation because the necessary mappings have been retained as part of stored state. Other implementations are obviously possible.

Assert logic becomes more complicated in the dual-stack scenario assumed here. One open issue is how to compare metrics if this router will acquire the multicast flow concerned using a different IP version from the version used by its rival peer. As noted below, Assert messages have to be sent out in both versions, because they have to be understood by downstream as well as upstream entities.

[That topic may need more detailed discussion. Also to do: add references and detail the challenges of multicast data packet processing.]

4. Modifications To RFC 4601

[This section should provide detailed changes needed to the specifications in RFC 4601, starting with the state data maintained.]

5. Processing of PIM Messages and Multicast Data Packets

5.1. Hello Messages

Hello messages are not translated. Rather, the differences between the IPv4 and IPv6 versions are as follows:

- o In the packet header, the source address varies between the IPv4 primary address and the IPv6 link-local address on that interface. The destination address is the IPv4 or IPv6 ALL_PIM_ROUTERS

multicast address as applicable.

- o The Address List option varies between the list of secondary IPv4 addresses on that interface and the list of secondary IPv6 addresses on that interface

5.2. Register and Register Stop Messages

The Register and Register Stop messages are routed as unicast messages.

Section 4.9.3 of [RFC4601] requires the header of the multicast data packet encapsulated within a Register message to have the same address family as the packet header of the Register message itself. This may require translation of the enclosed packet header to match the outer header. The procedures described in [RFC6145] MUST be applied to the header as a whole. Translation of the source and group addresses (the packet source and destination addresses) is done as described in Section 2.

The Register Stop message takes its contents from the received Register message, and needs no translation.

5.3. Join/Prune Messages

Join/Prune messages MUST be sent in the IP version indicated by the MRIB when it identifies an RPF neighbour.

Care must be taken when switching from the Rendezvous Point Tree to the shortest-path tree for a given source. The Prune for the Rendezvous Point Tree MUST be sent in the IP version of the RPF neighbour for that tree. This implies that in the (*,G) state described in Section 4.1.3 of [RFC4601], the address family of the last RPF neighbour used MUST be retained, and the address itself MUST NOT be translated.

Multicast group addresses and all joined and pruned source addresses contained in the message are translated as described in Section 3.

5.4. Assert Messages

Assert messages need to reach both upstream and downstream neighbours on a LAN. Hence, if the subject router PIM has received Hello messages in both IP versions on an interface to which an Assert is to be forwarded, it MUST send the Assert message in both IP versions.

The multicast group address and source address contained in the message are translated as described in Section 3.

5.5. Multicast Data Packets

This section applies to multicast data packets being forwarded directly rather than being encapsulated in Register messages. The procedures described in [RFC6145] MUST be applied to the header as a whole. Translation of the source and group addresses (the packet source and destination addresses) is done as described in Section 3.

6. Acknowledgements

Thanks to Simon Perrault for comments on the first version of this document.

7. IANA Considerations

This memo includes no request to IANA.

8. Security Considerations

TBD

9. References

9.1. Normative References

- [I-D.mboned-64-multicast-address-format]
Boucadair, M., Qin, J., Lee, Y., Venaas, S., Li, X., and M. Xu, "IPv4-Embedded IPv6 Multicast Address Format (Work in progress)", February 2012.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC4601] Fenner, B., Handley, M., Holbrook, H., and I. Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", RFC 4601, August 2006.
- [RFC6052] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", RFC 6052, October 2010.
- [RFC6145] Li, X., Bao, C., and F. Baker, "IP/ICMP Translation Algorithm", RFC 6145, April 2011.

9.2. Informative References

- [ID.v4v6-multicast-ps]
Jacquenet, C., Boucadair, M., Lee, Y., Qin, J., Tsou, T.,
and Q. Sun, "IPv4-IPv6 Multicast: Problem Statement and
Use Cases (Work in Progress)", May 2012.
- [ID.venaas-v4v6mcastgw]
Venaas, S., "An IPv4 - IPv6 multicast gateway (Expired
Internet Draft)", February 2003.
- [Kiviniemi]
Kiviniemi, T., "Implementation of an IPv4 to IPv6
Multicast Translator (Master's Thesis)", October 2009.

Author's summary: <<http://iki.fi/teemuki/translator/>>
- [RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A.
Thyagarajan, "Internet Group Management Protocol, Version
3", RFC 3376, October 2002.
- [RFC3810] Vida, R. and L. Costa, "Multicast Listener Discovery
Version 2 (MLDv2) for IPv6", RFC 3810, June 2004.

Authors' Addresses

Tom Taylor
Huawei Technologies
Ottawa,
Canada

Phone:
Email: tom.taylor.stds@gmail.com

Cathy Zhou
Huawei Technologies
Bantian, Longgang District
Shenzhen 518129
P.R. China

Phone:
Email: cathy.zhou@huawei.com

