

PIM Working Group  
Internet Draft  
Intended status: Standards Track  
Expires: December 1, 2018

Dave Allan  
Ericsson  
Jeff Tantsura  
Nuage  
Ian Duncan  
Ciena  
June 1, 2018

A Framework for Computed Multicast Applied to SR-MPLS  
draft-allan-pim-sr-mpls-multicast-framework-00

Abstract

This document describes a multicast solution for SR-MPLS. It is consistent with the Segment Routing architecture in that an IGP is augmented to distribute information in addition to the link state. In this solution it is multicast group membership information sufficient to synchronize state in a given network domain. Computation is employed to determine the topology of any loosely specified multicast distribution tree.

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/ietf/lid-abstracts.txt>.

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

This Internet-Draft will expire in December 1st, 2018.

Copyright and License Notice

Copyright (c) 2018 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. Authors.....	3
1.2. Requirements Language.....	3
2. Changes from the last version.....	3
3. Conventions used in this document.....	4
3.1. Terminology.....	4
4. Solution Overview.....	5
4.1. Mapping source specific trees onto the segment routing architecture.....	6
4.2. Role of the Routing System.....	6
4.3. MDT Construction Requirements.....	6
4.4. Simplification and Pruning - theory of operation.....	7
5. Elements of Procedure.....	7
5.1. Triggers for Computation.....	7
5.2. FIB Determination.....	8
5.2.1. Information in the IGP.....	8
5.2.2. Computation of individual segments.....	8
5.3. FIB Generation.....	12
5.4. FIB installation.....	12
6. Related work.....	13
6.1. IGP Extensions.....	13
6.2. BGP Extensions.....	13
7. Observations.....	14
8. Acknowledgements.....	14
9. Security Considerations.....	14
10. IANA Considerations.....	14
11. References.....	14
11.1. Normative References.....	14
11.2. Informative References.....	15
12. Authors' Addresses.....	15

## 1. Introduction

This memo describes a solution for multicast for SR-MPLS in which source specific multicast distribution trees (MDTs) are computed from information distributed via an IGP. Computation uses information in the IGP to determine if a given node in the network has a role as a root, a leaf or replication point in a given MDT. Unicast tunnels are employed to interconnect the nodes determined to have a role.

Therefore multicast topological instructions only need be installed in nodes that have one of these three roles to fully instantiate an MDT.

Although this approach might appear to be computationally intensive, a significant amount of computation can be avoided if and when the computing agent determines that the node it is computing for has no role in a given MDT. If there will be no need to install a multicast topological instruction in that node for the given MDT, the computing agent can abandon computation for the MDT and move on to other tasks, such as converging other MDTs. This permits a computed approach to multicast convergence to be computationally tractable.

This approach is proposed as a solution for networks for which an implementation of an alternative data plane, such as BIER, offers technical or economic challenges.

### 1.1. Authors

David Allan, Jeff Tantsura, Ian Duncan

### 1.2. Requirements Language

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

## 2. Changes from the last version

Clarification in motivation.

Editorial corrections and improvements.

Clarification of the description of upstream pruning in section 5.2.2

Alignment of terminology with current segment routing practice.

### 3. Conventions used in this document

#### 3.1. Terminology

Candidate replication point (CRP) - is a node that potentially needs to install a multicast topological instruction to replicate multicast traffic as determined at an intermediate step in multicast segment computation. It will either resolve to having no role or a role as a replication point once multicast has converged.

Candidate role - refers to any potential combination of roles on a given multicast segment as determined at some intermediate step in MDT computation. For example, a node with a candidate role may be a leaf and may also be a candidate replication point.

Computing agent- refers to the agent that will compute the FIB for the MDTs in a given network on behalf of one node (distributed model) or multiple nodes (SR controller(s) in a centralized model).

Downstream - refers to the direction along the shortest path to one or more leaves for a given multicast distribution tree

Multicast convergence - is when all computation and multicast topological instruction installation to ensure the FIB reflects the multicast information in the IGP is complete.

MDT - multicast distribution tree. Is a tree composed of one or more multicast segments.

Multicast segment - is a portion of the multicast tree where only the root and the leaves have been specified, and computation based upon the current state of the IGP database is employed to determine and install the required topological instructions to implement the segment. For SR-MPLS a multicast segment is implemented as a p2mp LSP. A multicast segment is identified by a multicast SID.

Multicast SID - Is the topological instruction that is used to implement a multicast segment. As per a unicast SR-MPLS segment, the rightmost 20 bits of a multicast SID is encoded as a label. It is drawn from the SRGB for the domain.

Pinned path - Is a unique shortest path extending from a leaf upstream towards the root for a given multicast segment. Therefore, it is a component of the multicast segment that it has been determined must be there. It will not necessarily extend from the leaf all the way to the root during intermediate computation steps. A pinned path can result from pruning operations.

Role - refers specifically to a node that is either a root, a leaf, a replication node, or a pinned waypoint for a given MDT.

Unicast convergence - is when all computation and topological instruction installation to ensure the FIB reflects the unicast information in the IGP is complete.

Upstream - refers to the direction along the shortest path to the root of a given MDT.

#### 4. Solution Overview

This memo describes a multicast architecture in which multicast topological instructions are only installed in those nodes that have roles as a root, a leaf, or a replication point for a given multicast segment. The a-priori established mesh of unicast tunnels (using node-SIDs) are used as interconnect between the nodes that have a role in a given multicast SID. Hence on an outgoing interface where the next node in that path of the MDT is not immediately adjacent, the operation will typically be a CONTINUE of the multicast SID and a PUSH of the node-SID.

A loosely specified MDT is composed of a single multicast segment and the routing of the MDT is delegated entirely to computation driven by information in the IGP database.

Explicitly routed MDTs are expressed as a tree of concatenated multicast segments where both the leaves of each segment and the waypoints coupling a given segment to the upstream and/or downstream segment(s) is specified in information flooded in the IGP by the overall root of the MDT. The segments themselves will be computed as per a loosely specified MDT.

A PE acting as an overall root for a given tree is expected to be configured by the operator as to where to source multicast traffic from, be it an attachment circuit, interworking function for client technology or other. Similarly, a leaf for a given tree is expected to be configured by the operator as to the disposition of received multicast traffic.

A computed segment is guaranteed to be loop free in a stable fault free system. A concatenation of segments to construct an MDT will similarly be loop free as any collision of segments can be disambiguated in the data plane via the SIDs.

This architecture significantly reduces the number of multicast topological instructions that needs to be installed in the data plane

to support multicast. This also means that the impact of many failures in the network on multicast traffic distribution will be recovered by unicast local repair or unicast convergence with subsequent multicast convergence acting in the role of network re-optimization (as opposed to restoration).

#### 4.1. Mapping source specific trees onto the segment routing architecture

A computed source specific tree for a given multicast group corresponds to one or more multicast segments in the SR architecture. Each multicast segment is assigned a SID, typically by management configuration of the node that will be the overall root for the source specific tree. The root node then uses the IGP to advertise this information to all nodes in the IGP area/domain.

A multicast group is implemented as the set of source specific trees from all nodes that have registered transmit interest to all nodes that have registered receive interest in a multicast group.

#### 4.2. Role of the Routing System

The role of the IGP is to communicate topology information, multicast capability and associated algorithm, multicast registrations, unicast to node-SID bindings, multicast to SID bindings and waypoints in multi-segment MDTs. No changes to topology or unicast to node-SID binding advertisements are proposed by this memo.

The multicast registrations/bindings will be in the form of source, group, transmit/receive interest and the SID to use for the source specific multicast tree. Registrations are originated by any node that has send or receive interest in a given multicast group. Nodes will use the combination of topology and multicast registrations to determine the nodes that have a role in each source specific tree and the SID information to then derive the required FIB state.

#### 4.3. MDT Construction Requirements

A multicast segment in an MDT is constructed such that between any pair of nodes that have a role in the segment and are connected by a unicast tunnel, there is not another node on the shortest path between the two with a role in that segment. This ensures that copies of a packet forwarded by a multicast segment will traverse a link only once in a stable system and avoids the potential scenario whereby a packet needs to be replicated twice on a given interface.

Note that this can be satisfied by a minimum cost shortest path tree, but this is not an absolute requirement. The pruning rules specified

in this memo will meet this requirement without necessarily producing an absolute minimum cost multicast segment (or incurring the associated computational cost).

#### 4.4. Simplification and Pruning - theory of operation

The role of nodes in a given multicast segment is determined by first producing an inclusive shortest path tree with all possible paths between the root and leaves, and then applying a set of simplification and pruning rules repeatedly until either an acyclic tree is produced, or no further prunes are possible.

For the majority of multicast segments these rules will authoritatively produce a minimum cost tree. For those segments that are not able to be authoritatively resolved, there is a set of pruning operations applied that are not guaranteed to produce a tree that meets the requirements of 3.3, therefore these trees require auditing and potential correction according to a further set of agreed rules. This avoids the necessity and computational overhead of an exhaustive search of the solution space.

A computing agent during computation of a segment may conclude that none of the nodes that it is computing on behalf of will have a role at any point in the computation process and abandon computation of that segment.

#### 5. Elements of Procedure

##### 5.1. Triggers for Computation

MDT computation is triggered by changes to the IGP database. These are in the form of either changes in registered multicast group interest, addition or removal of a multi-segment MDT descriptor, or topology changes.

A change in registered interest for a group will require re-computation of all MDTs that implement the multicast group.

A topology change will require the computation of some number of multicast segments, the actual number will depend on the implementation of tree computation but at a minimum will be all trees for which there is not an optimal shortest path solution as a result of the topology change.

## 5.2. FIB Determination

### 5.2.1. Information in the IGP

Group membership information for a multicast segment is obtained from the IGP. This is true for single segment MDTs as well as multi-segment MDTs. Included in the multi-segment MDT specification is the waypoint nodes in MDT and the upstream and downstream SIDs. The specified node is expected to cross connect the SIDs to join the segments together acting in the role of leaf for the upstream segment and root for the downstream segment.

When a waypoint in an MDT descriptor does not exist in the IGP, the assumption is that the node identified by the waypoint SID has failed. The response of the other nodes in the system in FIB determination is to add the leaves of the downstream segment to the upstream segment.

An example of this would be consider a node "x", and another node "y". At some point in time, "x" advertises a tree that identifies "y" as a waypoint that cross connects upstream SID "a" to downstream SID "b". At some later point node "y" fails. The other nodes in the network will compute segment "a" as if it included all leaves and waypoints in segment "b". All apriori state installed for segment "b" would be removed as the failure of "y" has required "b" to be subsumed by "a".

### 5.2.2. Computation of individual segments

FIB generation for a multicast segment is the result of computation, ultimately as applied to all source specific trees in the network. All computing agents in a given network computing a tree for a given multicast segment must implement a common algorithm for tree generation, as all MUST agree on the solution.

One algorithm is as follows:

All possible shortest paths to the set of leaves for the MDT is determined. Then simplification and pruning rules are repeatedly applied until no further prunes are possible or the MDT is determined to be resolved.

The distinction between simplification rules and pruning rules is the former will not change the candidate role of a node with respect to the MDT under consideration and therefore can be performed in any order, while the latter will affect candidate node roles and must be



performed in an agreed order between all participating computing agents.

The philosophy of the application of these rules could be expressed as "simplify as much as possible, and prune that which cannot be". The rules are:

- 1) Simplification: Eliminate any links and nodes not on a potential shortest path from the root to the leaves for the MDT under consideration.
- 2) Simplification: Replace any nodes that do not have a potential role in the MDT with links.

This will be nodes that are not a leaf, a root or a candidate replication point. For example:

Root-----A-----B

B is a leaf. A is not but is in a potential shortest path from root to B. However, A will have no role in the MDT that serves B as it provides simple transit therefore is replaced with a direct connection between the root and B.

Root-----B

Note that such simplification also needs to avoid the creation of duplicate parallel links. For example:

```

      /-----A-----\
Root                               B
      \-----C-----/

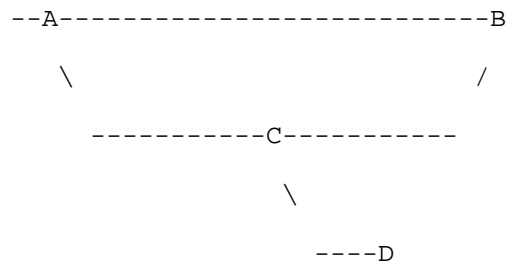
```

Where A and C have no role and the cost root-A-B = cost root-C-B, they can be replaced with a single link from Root to B.

- 3) Simplification: Eliminate of fewer hop paths

When for a given set of leaves, a node has multiple downstream links that converge on a common downstream point, and that set of leaves is only a subset of the leaves reachable on one or more of the links, any link that only serves that subset of leaves can be eliminated.

For example:



Link AB is cost 2, link AC and CB are cost 1 (cost of link CD does not affect the example).

B and D are leaves of a root upstream of A. From A, link AB can reach leaf B. Path AC can reach leaf B and D. In this case path A-B can be eliminated from consideration. The set of leaves reachable via link A-B is a subset of that reachable by A-C, and the paths from A that serves that subset converges at B.

#### 4) Prune: upstream links.

The normal procedure is to determine the best-closest upstream leaf or pinned path and then compare all upstream adjacencies with that metric. Note that the best-closest upstream leaf or pinned path may not be directly connected to the node under consideration. Where there is more than one equally close upstream leaf or pinned path, the highest ranked is selected with the ranking being that a leaf is ranked superior to a pinned path, and the lowest unicast SID is selected when the leaf/pinned path ranking is equal.

Then examine each of the remaining upstream adjacencies:

- a. If the upstream adjacency extends closer to the root than the closest leaf or pinned path, then that adjacency can be pruned.
- b. If the upstream adjacency extends the same distance towards the root as the best-closest adjacency, then it can be eliminated as it has already been ranked lower than the best-closest adjacency. Note that this would include non-leaf and non-pinned path candidate replication points.
- c. If the upstream adjacency is a candidate replication point closer than the best-closest leaf or pinned path, then it is left alone.

When for a given node all possible upstream adjacencies that can be pruned have been identified, each is removed, and any simplifications that can be performed as a result of the prune are performed. This is the equivalent of a localized check for 2 and 3 above and is then performed iteratively in response to changes to the graph as a result of pruning.

The procedure is to implement all simplifications of type 1, 2 and 3 above, then loop on type 4 prunes until such time as the MDT is fully resolved from the point of view of the node under consideration, or no further prunes are possible. Step 4 is required to be performed in a specific order if there is more than one computing agent generating topological instructions for a given multicast segment. This memo suggests that the nodes are processed according to a ranking of nodes from closest to the root to the farthest, and from lowest unicast SID to the highest within a given distance from the root.

At the end of pruning and simplification, either:

- 1) The node whom the computing agent is computing for has no role in the multicast segment under consideration
- 2) A unique shortest path to the root has been determined for all leaves in the multicast segment that are downstream of the node under consideration (also termed as a pinned path from the root to every leaf).
- 3) A unique shortest path to the root has not been determined for all leaves downstream of the node under consideration in the multicast segment.

If 1 or 2 then the multicast segment is considered to be resolved, and for 2, the computation can progress directly to the topological instruction generation step for that segment.

If 3 (not all downstream leaves have a unique shortest path), additional pruning steps are applied. These steps are NOT guaranteed to produce a lowest cost tree, and therefore require an additional audit and possible modification to ensure when forwarding a maximum of one copy of a packet will traverse an interface.

For segments not authoritatively resolved by the above rules, a prune that will not authoritatively result in a minimum cost tree is applied. For the purpose of interoperability, the following rule is applied: A computing agent will select the closest node to the root with a candidate role that does not have a unique shortest path to the root. Where more than one such node exists, the one with the

lowest node-SID is selected. For that node, the best upstream link is selected and all other upstream links pruned. The best upstream link is defined as the link with the closest node with a candidate role that potentially serves the highest number of leaves. Where there is a tie, once again the node with the lowest unicast SID is selected.

Once the links have been pruned, rules 2 through 4 are repeatedly applied until either the tree is fully resolved, or again no further prunes are possible, in which case the next closest remaining unresolved node has the same prune applied.

For all segments not resolved by the initial prune rules, they are audited to ensure all nodes that have a role in the tree do not have a node with a role between them and their upstream node on the tree. If they do, the old upstream adjacency is removed, and the superior one added.

### 5.3. FIB Generation

The topology components that remain at the end of the simplification and pruning operations will reflect all nodes that have a role in a given multicast segment plus the necessary tunnels (as all intervening multi-path scenarios will have been simplified away). From this the topological instructions to put in the FIB can be generated:

All nodes that have a role in a given multicast segment and have nodes upstream in the segment will need to accept the multicast SID for the MDT from at minimum, all upstream interfaces.

All nodes that have a role in a given segment and have nodes immediately downstream in the segment will need to replicate packets simply labelled with the multicast SID onto those interfaces.

All nodes that have a role in a given segment and have nodes reachable via a tunnel downstream set the FIB to push the tunnel unicast SID for the downstream node onto any replicated copies of a received packet, and identify the set of interfaces on the shortest path for the tunnel SID.

### 5.4. FIB installation

FIB installation needs to acknowledge two aspects of the hybrid tunnel and role model of multicast tree construction. The first is that because of the sparse state model simple tree adds, moves, and changes may require the installation of topological instructions where they did not previously exist, and such changes may impact

existing services. The second is that it is possible to retain the knowledge to prioritize computation of those trees impacted the failure of a node with a role.

To address this in the distributed model, there are three stages of topological instruction installation for multicast convergence:

1) Immediate:

- a. Installation of topological instructions for multicast segments impacted by the failure of a node in the network, and installation of topological instructions for segments in nodes that have not previously had a role in the given segment.
- b. Installation of topological instructions for waypoints in multi-segment MDTs.

2) After T1: Update topological instructions for nodes that both had and have a role in a given multicast segment.

3) After T2: Removal of topological instructions for nodes that transition from having a role to not having a role for a given multicast segment.

T1 and T2 are network wide configurable values.

When an SR-Controller is used, it is only necessary to properly sequence the installation of state. This also suggests that when there is more than one SR-Controller, the division of responsibility should be on the basis of MDT ownership.

## 6. Related work

### 6.1. IGP Extensions

The required IGP changes are documented in [MCAST-ISIS] and [MCAST-OPSF].

### 6.2. BGP Extensions

This memo will require the specification of a new PMSI Tunnel Attribute (SPRING P2MP tunnel, tentatively 0x0c) to order to integrate into the multicast framework documented in RFC 6514

## 7. Observations

This technique is not confined to SR-MPLS:

- with the provision of a global label space (to be employed as per a multicast SID), an MPLS-LDP network would also provide the requisite mesh of unicast tunnels and be capable of implementing this approach to multicast.
- It is also possible to envision an SRv6 implementation but would require the ability to rewrite the SRH at each hop.

This memo focuses on an implementation based upon nodes that are IGP speakers and converge independently so is written in a form that assumes a node, computing agent and IGP speaker are one in the same. It should be observed that the relative frugality of data plane state would suggest that separation of computation from nodes in the data plane combined with management or "software defined networking" based population of the multicast FIB entries may also be useful modes of network operation.

## 8. Acknowledgements

Thanks to Uma Chunduri for his detailed review and suggestions.

## 9. Security Considerations

For a future version of this document.

## 10. IANA Considerations

This document requires the allocation of a PMSI tunnel type to identify a SPRING P2MP tunnel type from the P-Multicast Service Interface Tunnel (PMSI Tunnel) Tunnel Types registry.

## 11. References

### 11.1. Normative References

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

## 11.2. Informative References

- [MCAST-ISIS] Allan et.al., "IS-IS extensions for Computed Multicast applied to MPLS based Segment Routing", IETF work in progress, draft-allan-isis-spring-multicast-00, July 2016
- [MCAST-OSPF] Allan et.al., "OSPF extensions for Computed Multicast applied to MPLS based Segment Routing", IETF work in progress, draft-allan-ospf-spring-multicast-00, July 2016
- [SR-ARCH] Filsfils et.al., "Segment Routing Architecture", IETF work in progress, draft-ietf-spring-segment-routing-15, January 2018
- [RFC6514] Aggarwal et.al., "BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs", IETF RFC 6514, February 2012
- [RFC7385] Andersson & Swallow "IANA Registry for P-Multicast Service Interface (PMSI) Tunnel Type Code Points", IETF RFC 7385, October 2014

## 12. Authors' Addresses

Dave Allan (editor)  
Ericsson  
2455 Augustine Drive  
Santa Clara 95054  
USA  
Email: david.i.allan@ericsson.com

Jeff Tantsura  
Email: jefftant.ietf@gmail.com

Ian Duncan  
Ciena  
iduncan@ciena.com  
5050 Innovation Drive  
Kanata, ON K2K 0J2

Protocol Independent Multicast (pim)  
Internet-Draft  
Intended status: Standards Track  
Expires: September 20, 2018

A. Peter, Ed.  
Individual contributor  
R. Kebler  
V. Nagarajan  
Juniper Networks, Inc.  
T. Eckert  
Huawei USA - Futurewei Technologies Inc.  
S. Venaas  
Cisco Systems, Inc.  
March 19, 2018

Reliable Transport For PIM Register States  
draft-anish-reliable-pim-registers-02

Abstract

This document introduces a hard-state, reliable transport for the existing PIM-SM registers states. This eliminates the needs for periodic NULL-registers and register-stop in response to each data-register or NULL-registers.

This specification uses the existing PIM reliability mechanisms defined by PIM Over Reliable Transport [RFC6559]. This is simply a means to transmit reliable PIM messages and does not require the support for Join/Prune messages over PORT as defined in [RFC6559].

Requirements Language

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

Status of This Memo

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

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

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



This Internet-Draft will expire on September 20, 2018.

#### Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

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

#### Table of Contents

1. Introduction . . . . .	3
2. Reliable Register Overview . . . . .	4
3. Targeted Hellos . . . . .	4
3.1. New Hello Optional TLV's . . . . .	5
3.2. Differences from Link-Level hellos . . . . .	5
3.3. Address in Hello message . . . . .	6
3.4. Timer Values . . . . .	6
3.5. Targeted Neighbor . . . . .	6
4. Reliable Connection setup . . . . .	7
4.1. Active FHR . . . . .	7
4.2. Connection setup between two RP's . . . . .	7
4.3. Hello Generation ID and reconnect . . . . .	7
4.4. Handling Connection or reachability loss . . . . .	7
5. Anycast RP's . . . . .	8
5.1. Targeted Hellos and Neighbors . . . . .	8
5.2. Anycast-RP connection setup . . . . .	8
5.3. Anycast-RP state sync . . . . .	8
5.4. Anycast-RP change . . . . .	9
5.5. Anycast-RP with MSDP . . . . .	9
6. PIM-registers and Interoperation with legacy PIM nodes . . . . .	10
6.1. Initial packet-loss avoidance with PORT . . . . .	10
6.2. First-Hop-Router does not support PORT . . . . .	10
6.3. RP does not support PORT . . . . .	10
6.4. Data-Register free operations . . . . .	10
7. PORT message . . . . .	10
7.1. PORT register message TLV . . . . .	10
7.2. Sending and receiving PORT register messages . . . . .	12
7.3. PORT register-stop message TLV . . . . .	12
7.4. Sending and receiving PORT register stop messages . . . . .	13

7.5. PORT Keep-Alive Message . . . . .	13
8. Management Considerations . . . . .	13
9. IANA Considerations . . . . .	14
9.1. PIM Hello Options TLV . . . . .	14
9.2. PIM PORT Message Type . . . . .	14
10. Security Considerations . . . . .	14
10.1. PIM Register Threats . . . . .	14
10.2. Targeted Hello Threats . . . . .	15
10.3. TCP or SCTP security threats . . . . .	15
11. References . . . . .	15
11.1. Normative References . . . . .	15
11.2. Informative References . . . . .	16
Authors' Addresses . . . . .	16

## 1. Introduction

Protocol Independent Multicast-Sparse Mode Register mechanism serves the following purposes.

- a, With a register, First-Hop-Router (FHR) informs the RP (that way the network) that a particular multicast stream is active
- b, A register helps avoid initial packet loss. (Initial packet loss could happen in an anycast-RP deployment even when packet registers are used.)
- c, Through its periodic refreshes register keeps RP informed about the aliveness of this multicast stream.

As it is defined in [RFC4601] , register mechanisms face limitations, when the number of multicast streams on the network is high, especially when one RP is expected to serve a large number of streams. These problems are mainly due to these factors.

- a, PIM register needs control-plane and data-plane intervention to handle it.
- b, Due to the nature of PIM register, First-Hop-Router and RP now needs to maintain states and timers for each register state entry.
- c, PIM register's requirements for periodic refresh and expiry, is quite aggressive and makes them vulnerable when the PIM speaker could not find cycles to meet these needs

To take for instance a major multicast application the IPTV. With the streaming servers connected to FHR. A restarting, FHR would result in a burst of register messages at line rate. The RP may get

overloaded with packet registers. Which will continue until RP is able to create states and do a register-stop. In the meantime many flows may go unserved due to drops. In addition to affecting multicast streams it may lead to starvation for other processing done by the controlplane application. With Anycast RP, this becomes even more tricky due to the control-plane's job to forward the registers to the rp-set.

In general: PIM registers have limitation in connections across WAN. It has no flow-control mechanisms, making PIM not compatible with IETF transport/congestion control expectations. It is challenging to deploy it over WAN or other bandwidth limited networks. High amount of state: periodic retransmission creates undesirable processing load. Especially with larger mesh-groups (re-send same (S,G) N-times, periodically).

## 2. Reliable Register Overview

Reliable PIM register extends PIM PORT [RFC6559] to have PIM register states to be sent over a reliable transport.

This document introduces 'targeted' hellos between any two PIM peers. This helps in capability negotiation and discovery between two PIM speakers (FHR and RP in the context of this document). Once this discovery happens, First-Hop-Router would setup a reliable transport connection based on the negotiated parameters.

Over this reliable connection, First-Hop-Router would start sending to RP the source and group addresses of the multicast streams active with it. When any of this stream stops, First-Hop-Router would send an update to RP about the streams that have stopped. This way once a reliable connection is setup, First-Hop-Router would update RP with its existing active multicast streams. Subsequently it would send incremental updates about the change to RP.

For a multicast application that may demand initial packets or for bursty sources existing data-registers may be used. For them the RP would now respond with a 'reliable'-register-stop, which could persist until the First-Hop-Router withdraws the register-state.

## 3. Targeted Hellos

PIM hellos defined in PIM-SM [RFC4601] confines them to link level. This document extends these hellos to support 'targeted' hellos.

Targeted hellos are identical to existing hellos messages except that they would have an unicast address as its destination address. It

would traverse multiple hops using the unicast routing to reach the targeted hello neighbor.

### 3.1. New Hello Optional TLV's

Option Type: Targeted hello

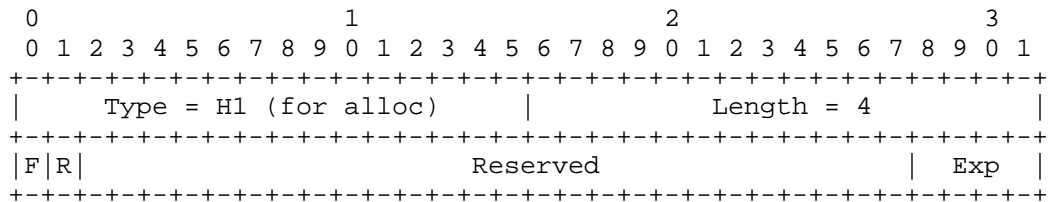


Figure 1: PIM Hello Optional TLV

Assigned Hello Type values can be found in IANA PIM registry.

Type: This is subject to IANA allocation. Its stated as H1 for reference

Length: Length in bytes for the value part of the Type/Length/Value encoding fixed as 4.

F: To be set by a router that wants to be a First-Hop-Router.

R: To be set by a RP that is capable taking the role of an RP as per the current states.

Reserved: Set to zero on transmission and ignored on receipt.

Exp: For experimental use [RFC3692]. One expected use of these bits would be to signal experimental capabilities. For example, if a router supports an experimental feature, it may set a bit to indicate this. The default behavior, unless a router supports a particular experiment, is to keep these bits reset and ignore the bits on receipt.

### 3.2. Differences from Link-Level hellos

The Major differences that Link-Level-Hellos have over Interface hellos are,

1. Destination address would be an unicast address unlike ALL-PIM-ROUTER destination address for link-level hellos

2. TTL value would be the system default TTL
3. Targeted Hellos SHOULD carry Targeted Hello Optional TLV (Defined in this document.)
4. Holdtime SHOULD NOT be set as 0xffff by a targeted hello sender, and such hellos should be discarded up on receive.

### 3.3. Address in Hello message

When sending targeted hellos, the sender SHOULD send with its primary reachable address (may be its loopback address) as the source address for the hellos. The other addresses that are relevant SHOULD be added in the secondary address list.

### 3.4. Timer Values

The timers relevant to this specification are in relation to PIM hello. The recommended timer values are

- 1: PIM Targeted Hello default refresh time : 60s (2 \* Default Link-level hello time)
- 2: PIM Targeted Hello default hold time : 210s (3.5 times targeted hello default refresh time)

### 3.5. Targeted Neighbor

A Targeted PIM neighbor is a neighbor-ship established by virtue of exchanging targeted hello messages.

A First-Hop-Router (The initiator) that learns the RP's address would start sending hellos to the known RP address (could be anycast-address).

The RP (The Responder) when it receives this hello, would add sender as a targeted neighbor and would respond to this targeted neighbor from its primary address. The responder SHOULD also include its anycast address (If available) in the secondary address list. The First-Hop-Router when receiving this hello would form a targeted neighbor with the anycast address.

The RP upon hold-time-out for the neighbor would remove this neighbor and its associated states.

The initiator or responder upon having a need to terminate a targeted neighbor MAY send hello with hold-time as 0.

#### 4. Reliable Connection setup

A reliable connection has to be setup between the First-Hop-Router and RP for reliable registers to happen. Targeted hellos works as the medium for discovery and capability-negotiation between the two peers.

##### 4.1. Active FHR

Once First-Hop-Router and RP discovery each other, First-Hop-Router takes the active role. First-Hop-Router would listen for RP to connect once it forms targeted neighbor-ship with RP. The RP would be expected to use its primary address, which it would have used as the source address in its PIM hellos.

##### 4.2. Connection setup between two RP's

In a network if there happens to be two RP's which are First-Hop-Router's too, then the mechanism could result in two connections getting established. It's desirable to have just one connection instead of two. First-Hop-Router could detect this condition the when it receives hello with targeted hello header identifying that the RP want to be First-Hop-Router too.

In this condition the connection setup could used the procedures stated in PIM over reliable transport [RFC6559]

##### 4.3. Hello Generation ID and reconnect

If RP or First-Hop-Router gets into a situation needing for capability-renegotiation or reconnect, it would change the hello generation ID (gen-ID) to notify it peer to reset all the states and re-init this peering. The trigger for this could be configuration change or local operating parameter change, restart, etc. . .

##### 4.4. Handling Connection or reachability loss

Connection loss or reachability loss could happen for one or more of the following reasons

- 1: PORT Keep-alive time out
- 2: Targeted neighbor loss
- 3: Reliable Connection close

Upon detecting one of these conditions, the connection with the peer SHOULD be closed immediately and the states created by the peer

SHOULD be cleared after a grace-period, long enough for the peer to re-establish connection and re-sync the states.

This interval for re-sync would involve the initial time needed for re-establishing the connection, followed by transmission and reception of the states from FHR to RP over the reliable connection.

The ideal interval for this re-sync period could not be predicted, hence this should be a configurable parameter with default value as 300s.

## 5. Anycast RP's

PIM uses Anycast-RP [RFC4610] as a mechanism for RP redundancy. This section describes how anycast-RP would work with this specification.

### 5.1. Targeted Hellos and Neighbors

An RP that serves an anycast RP address, would know the primary addresses of other RP's serving the anycast address. These anycast-RP's would form a full mesh of targeted hello-neighbor-ships. In its targeted hello options tlv, the R bit MUST be set. The secondary address list in the PIM hello message SHOULD include the anycast-addresses that the sender is servicing.

### 5.2. Anycast-RP connection setup

A full mesh of connection is needed among the anycast-RP's of the same anycast address. Once targeted neighbor-ship is established, it would use the PIM PORT [RFC6559] procedures to setup reliable connection among them.

### 5.3. Anycast-RP state sync

An anycast-RP that gets the register state from a peer who's address is in the RP-set of address for the given group would update the register state and would retain the state. If the peer address is not in the RP-set address for the RP-group range, then the RP would replicate the state to all the other RP's in the RP-set. This procedure and forwarding rules are similar to the existing forwarding rules in Anycast-rp [RFC4610] register specification.

An RP should identify register state as a combinations of (source, group, 'PORT connection'). Where 'PORT connection' is the reliable connection with the PORT peer which had reported this s,g. Following considerations are made for a register-state identity.

- A. Reconnect: Connection between RP and First-Hop-Router could get re-established for various reasons. The register-states would get retransmitted over the new connection. Then it should be possible for RP to identify and timeout register-states on the old connection and retain the right set of states.
- B. DR-change: When DR in the First-Hop-LAN changes, a new First-Hop-Router would be retransmitting the same set of SG's that are already known and the old DR would be withdrawing the states advertised by it.
- C. FHR primary address change: In this case too connection would get re-established and state handling would be similar to case A.
- D. Multi-homed sources (but not on same LAN): In this case different First-Hop-Routers could be sending the same register-states. Then RP should be capable of identifying register-state along with the peer.

#### 5.4. Anycast-RP change

In the event of nearest anycast-RP changing over to a different router, First-Hop-Router would detect that when it starts receiving PIM hellos with a different primary address for the same anycast address. This can also happen if the primary address of present anycast-RP has changed.

Upon detecting this scenario, the First-Hop-Router would establish connection and transmit its states to the new peer. Subsequently older connection would get terminated due to neighbor timeout.

#### 5.5. Anycast-RP with MSDP

MSDP [RFC3618] is an alternative mechanism for 'active multicast stream state' synchronization between RP's. When MSDP is used, PIM's anycast synchronization need not be used. An anycast-RP network could use MSDP instead of PIM procedures for state synchronization among anycast-RP's. This document does not state any change in MSDP specification and usage

In such deployments, PIM will not have RP-set configured. As RP-set address is not available PIM procedures for Anycast-RP synchronization does not apply.

MSDP being a soft-state oriented protocol, it depends on frequent state refreshes over the reliable TCP transport. The support for mesh-groups in MSDP could be advantageous in some case.



## 6. PIM-registers and Interoperation with legacy PIM nodes

It may not be possible for PIM node to migrate altogether onto a PORT-registers in one go. Also there could be a few nodes in the network, which may not support PORT register states. This section states how both could interoperate.

### 6.1. Initial packet-loss avoidance with PORT

If its found that a few streams in the multicast network has to have initial packets to be delivered to the receiver, the existing PIM register mechanism could be used for them. For these streams a PORT register-stop message would be sent by the RP to First-Hop-Router.

### 6.2. First-Hop-Router does not support PORT

If the First-Hop-Router is not capable of doing PORT-register, then it would not establish targeted hello neighbor-ship with the RP. Hence reliable connection also would not be established. To handle such scenarios RP should accept PIM register messages and should respond to them with register-stop messages.

### 6.3. RP does not support PORT

If the RP is not capable of doing PORT-register, then it would not respond to the targeted hellos from the RP. Hence reliable connection also would not be established. In this case First-Hop-Router could sent existing packet registers to RP.

### 6.4. Data-Register free operations

If initial packet loss is acceptable in a multicast network, then Data-Registers could be avoided altogether in such networks. In such network PORT-Register-state specified in this document alone would be supported.

## 7. PORT message

This document defines new PORT register state message and PORT register-stop messages, to the existing messages in PORT specification.

### 7.1. PORT register message TLV

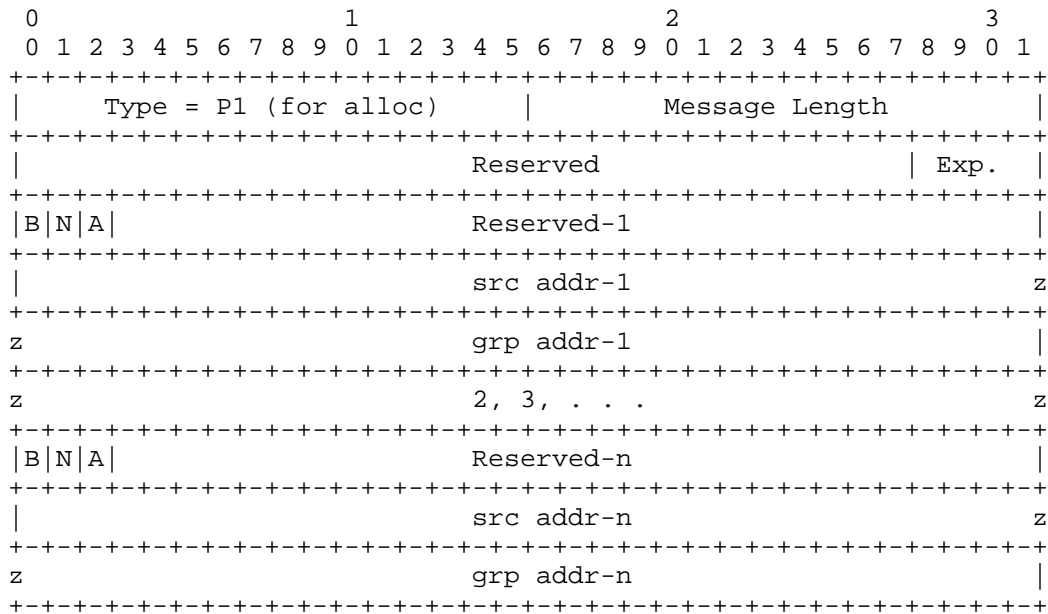


Figure 2: PORT Register State Message

Type: This is subject to IANA allocation. It would be next unallocated value, which is referred until allocation as P1.

Length: Length in bytes for the value part of the Type/Length/Value

B: As specified in [RFC4601] (set as 0 on send and ignore when received)

N: As specified in [RFC4601] (set as 1 on send and ignore when received.)

A: This flag signifies if the SG is to be Added or Deleted. When cleared, it indicates that the First-Hop-Router is withdrawing the SG.

src addr-x : This is the encoded source address of an ipv4/ipv6 stream

grp addr-x : This is the encoded group address of an ipv4/ipv6 stream

Reserved: Set to zero on transmission and ignored on receipt. These reserved bits are for properties that apply to the entire message.

Reserved-n: Set to zero on transmission and ignored on receipt. These reserved bits are for properties that apply to any particular sg.

Exp: : For experimental use.

## 7.2. Sending and receiving PORT register messages

The First-Hop-Router upon learning a new stream would send a register state add message to the corresponding RP. If the reliable connection got setup later, then once the connection becomes established it would send the entire list of streams active with it.

When KAT timer for a multicast stream expires, it would send an update to RP to remove that stream from its list.

An RP would maintain a database of multicast streams (src, grp) active along with the peer from which it had learned it. If the receiver RP is an anycast RP, it SHOULD re-transmit this register state message to each of the other anycast RP. An RP SHOULD not re-transmit a register state message it received from an another anycast-RP.

## 7.3. PORT register-stop message TLV

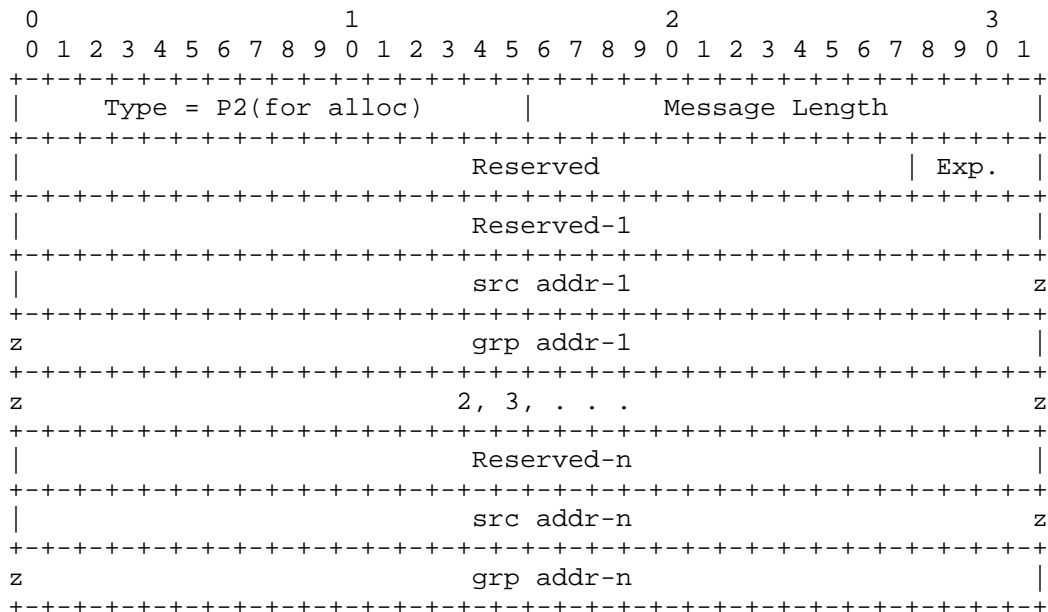


Figure 3: PORT Register Stop Message

Type: This is subject to IANA allocation. It would be next unallocated value, which is referred until allocation as P2.

Length: Length in bytes for the value part of the Type/Length/Value

src addr-x : This is the encoded source address of an ipv4/ipv6 stream

grp addr-x : This is the encoded group address of an ipv4/ipv6 stream

Reserved: Set to zero on transmission and ignored on receipt. These reserved bits are for properties that apply to the entire message.

Reserved-n: Set to zero on transmission and ignored on receipt. These reserved bits are for properties that apply to any particular sg.

Exp: : For experimental use.

#### 7.4. Sending and receiving PORT register stop messages

PORT register-stop messages are send only as a response to receiving packet registers from a PIM peer, with which a reliable connection has been established. If reliable connection is not available, the RP should consider the peer as a legacy node and should respond to this PIM register-stop message as defined in PIM-SM [RFC4601]

The First-Hop-Router up on receiving PORT-Register-Stop message should treat that as an indication from RP that it does not require the packets over the PIM tunnel and should stop sending register messages.

#### 7.5. PORT Keep-Alive Message

The PORT Keep-alive messages as specified in PIM over Reliable Transport [RFC6559] would be used to check the liveliness of the peer and the reliable session

#### 8. Management Considerations

PORT-register is capable of configuration free operations. But its recommended to have it as configuration controlled.

Implementation should provide knobs needed to stop supporting data-registers on a router.

## 9. IANA Considerations

This specification introduces new TLV in PIM hello and in PIM PORT messages. Hence the tlv ids for these needs IANA allocation

### 9.1. PIM Hello Options TLV

The following Hello TLV types needs IANA allocation. Place holder are kept to differentiate the different types.

Value	Length	Name	Reference
H1 (next-available)	4 (Fixed)	Targeted-Hello-Options	This document

Table 1: Place holder values for PIM Hello TLV type until IANA allocation

### 9.2. PIM PORT Message Type

The following PIM PORT message TLV types needs IANA allocation. Place holder are kept to differentiate the different types.

Value	Name	Reference
P1 (Next available)	PORT Register-state	This document
P2 (Next available)	PORT Register-stop	This document

Table 2: Place holder values for PIM PORT TLV type for IANA allocation

## 10. Security Considerations

### 10.1. PIM Register Threats

PIM register is considered as security vulnerability for PIM networks. [RFC4609] The concern arises mainly due to the existing PIM register protocol design where in any remote node could start sending line-rate multicast traffic as PIM registers due to malfunction, mis-configuration or from a malicious remote node.

## 10.2. Targeted Hello Threats

This document introduces targeted hellos. This could be seen as a new security threat. Targeted hellos are part of other IETF protocol implementations, which are widely deployed. In future introduction of a mechanism similar to those stated in RFC 7349 [RFC7349] could be used in PIM.

## 10.3. TCP or SCTP security threats

The security perception for this is stated in [RFC6559].

## 11. References

### 11.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3618] Fenner, B., Ed. and D. Meyer, Ed., "Multicast Source Discovery Protocol (MSDP)", RFC 3618, DOI 10.17487/RFC3618, October 2003, <<https://www.rfc-editor.org/info/rfc3618>>.
- [RFC4601] Fenner, B., Handley, M., Holbrook, H., and I. Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", RFC 4601, DOI 10.17487/RFC4601, August 2006, <<https://www.rfc-editor.org/info/rfc4601>>.
- [RFC4609] Savola, P., Lehtonen, R., and D. Meyer, "Protocol Independent Multicast - Sparse Mode (PIM-SM) Multicast Routing Security Issues and Enhancements", RFC 4609, DOI 10.17487/RFC4609, October 2006, <<https://www.rfc-editor.org/info/rfc4609>>.
- [RFC4610] Farinacci, D. and Y. Cai, "Anycast-RP Using Protocol Independent Multicast (PIM)", RFC 4610, DOI 10.17487/RFC4610, August 2006, <<https://www.rfc-editor.org/info/rfc4610>>.
- [RFC6559] Farinacci, D., Wijnands, IJ., Venaas, S., and M. Napierala, "A Reliable Transport Mechanism for PIM", RFC 6559, DOI 10.17487/RFC6559, March 2012, <<https://www.rfc-editor.org/info/rfc6559>>.

## 11.2. Informative References

[RFC7349] Zheng, L., Chen, M., and M. Bhatia, "LDP Hello Cryptographic Authentication", RFC 7349, DOI 10.17487/RFC7349, August 2014, <<https://www.rfc-editor.org/info/rfc7349>>.

## Authors' Addresses

Anish Peter (editor)  
Individual contributor  
Brunton Road  
Bangalore, KA 560001  
India  
  
Email: [anish.ietf@gmail.com](mailto:anish.ietf@gmail.com)

Robert Kebler  
Juniper Networks, Inc.  
1194 N. Mathilda Ave.  
Sunnyvale, CA 94089  
US  
  
Email: [rkebler@juniper.net](mailto:rkebler@juniper.net)

Vikram Nagarajan  
Juniper Networks, Inc.  
Electra, Exora Business Park  
Bangalore, KA 560103  
India  
  
Email: [vikramna@juniper.net](mailto:vikramna@juniper.net)

Toerless Eckert  
Huawei USA - Futurewei Technologies Inc.  
  
Email: [tte+ietf@cs.fau.de](mailto:tte+ietf@cs.fau.de)

Stig Venaas  
Cisco Systems, Inc.  
  
Email: [stig@cisco.com](mailto:stig@cisco.com)

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: December 21, 2018

Yiqun. Cai  
Heidi. Ou  
Alibaba Group  
Sri. Vallepalli  
Mankamana. Mishra  
Stig. Venaas  
Cisco Systems  
Andy. Green  
British Telecom  
June 19, 2018

PIM Designated Router Load Balancing  
draft-ietf-pim-drlb-08

Abstract

On a multi-access network, one of the PIM routers is elected as a Designated Router (DR). On the last hop LAN, the PIM DR is responsible for tracking local multicast listeners and forwarding traffic to these listeners if the group is operating in PIM-SM. In this document, we propose a modification to the PIM-SM protocol that allows more than one of these last hop routers to be selected so that the forwarding load can be distributed among these routers.

Status of This Memo

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

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

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

This Internet-Draft will expire on December 21, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.



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

## Table of Contents

1. Introduction . . . . .	2
2. Terminology . . . . .	5
3. Applicability . . . . .	5
4. Functional Overview . . . . .	6
4.1. GDR Candidates . . . . .	6
4.2. Hash Mask and Hash Algorithm . . . . .	7
4.3. Modulo Hash Algorithm . . . . .	8
4.4. PIM Hello Options . . . . .	9
5. Hello Option Formats . . . . .	9
5.1. PIM DR Load Balancing Capability (DRLBC) Hello Option . . . . .	9
5.2. PIM DR Load Balancing GDR (DRLBGDR) Hello Option . . . . .	10
6. Protocol Specification . . . . .	11
6.1. PIM DR Operation . . . . .	11
6.2. PIM GDR Candidate Operation . . . . .	12
6.2.1. Router Receives New DRLBGDR . . . . .	13
6.2.2. Router Receives Updated DRLBGDR . . . . .	13
6.3. PIM Assert Modification . . . . .	14
7. Compatibility . . . . .	15
8. Manageability Considerations . . . . .	15
9. IANA Considerations . . . . .	16
10. Security Considerations . . . . .	16
11. Acknowledgement . . . . .	16
12. References . . . . .	16
12.1. Normative References . . . . .	16
12.2. Informative References . . . . .	17
Authors' Addresses . . . . .	17

## 1. Introduction

On a multi-access LAN such as an Ethernet, one of the PIM routers is elected as a DR. The PIM DR has two roles in the PIM-SM protocol. On the first hop network, the PIM DR is responsible for registering an active source with the Rendezvous Point (RP) if the group is operating in PIM-SM. On the last hop LAN, the PIM DR is responsible for tracking local multicast listeners and forwarding to these listeners if the group is operating in PIM-SM.

Consider the following last hop LAN in Figure 1:

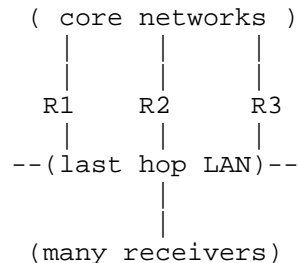


Figure 1: Last Hop LAN

Assume R1 is elected as the Designated Router. According to [RFC4601], R1 will be responsible for forwarding traffic to that LAN on behalf of any local members. In addition to keeping track of IGMP and MLD membership reports, R1 is also responsible for initiating the creation of source and/or shared trees towards the senders or the RPs.

Forcing sole data plane forwarding responsibility on the PIM DR uncovers a limitation in the protocol. In comparison, even though an OSPF DR or an IS-IS DIS handles additional duties while running the OSPF or IS-IS protocols, they are not required to be solely responsible for forwarding packets for the network. On the other hand, on a last hop LAN, only the PIM DR is asked to forward packets while the other routers handle only control traffic (and perhaps drop packets due to RPF failures). Hence the forwarding load of a last hop LAN is concentrated on a single router.

This leads to several issues. One of the issues is that the aggregated bandwidth will be limited to what R1 can handle towards this particular interface. It is very common that the last hop LAN usually consists of switches that run IGMP/MLD or PIM snooping. This allows the forwarding of multicast packets to be restricted only to segments leading to receivers who have indicated their interest in multicast groups using either IGMP or MLD. The emergence of the switched Ethernet allows the aggregated bandwidth to exceed, sometimes by a large number, that of a single link. For example, let us modify Figure 1 and introduce an Ethernet switch in Figure 2.

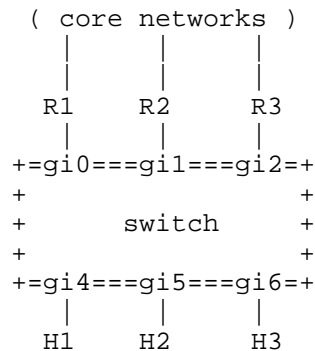


Figure 2: Last Hop Network with Ethernet Switch

Let us assume that each individual link is a Gigabit Ethernet. Each router, R1, R2 and R3, and the switch have enough forwarding capacity to handle hundreds of Gigabits of data.

Let us further assume that each of the hosts requests 500 Mbps of unique multicast data. This totals to 1.5 Gbps of data, which is less than what each switch or the combined uplink bandwidth across the routers can handle, even under failure of a single router.

On the other hand, the link between R1 and switch, via port gi0, can only handle a throughput of 1Gbps. And if R1 is the only DR (the PIM DR elected using the procedure defined by [RFC4601]) at least 500 Mbps worth of data will be lost because the only link that can be used to draw the traffic from the routers to the switch is via gi0. In other words, the entire network's throughput is limited by the single connection between the PIM DR and the switch (or the last hop LAN as in Figure 1).

The problem may also manifest itself in a different way. For example, R1 happens to forward 500 Mbps worth of unicast data to H1, and at the same time, H2 and H3 each request 300 Mbps of different multicast data. R1 experiences packet drop once again. while, in the meantime, there is sufficient forwarding capacity left on R2 and R3 and unused link capacity between the switch and R2/R3.

Another important issue is related to failover. If R1 is the only forwarder on the last hop router for shared LAN, when R1 goes out of service, multicast forwarding for the entire LAN has to be rebuilt by the newly elected PIM DR. However, if there was a way that allowed

multiple routers to forward to the LAN for different groups, failure of one of the routers would only lead to disruption to a subset of the flows, therefore improving the overall resilience of the network.

There is limitation in the hash algorithm used in this document, but this draft provides the option to have different and more consistent hash algorithms in the future.

In this document, we propose a modification to the PIM-SM protocol that allows more than one of these routers, called Group Designated Routers (GDR) to be selected so that the forwarding load can be distributed among a number of routers.

## 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].

With respect to PIM, this document follows the terminology that has been defined in [RFC4601].

This document also introduces the following new acronyms:

- o GDR: GDR stands for "Group Designated Router". For each multicast flow, either a (\*,G) for ASM, or an (S,G) for SSM, a hash algorithm (described below) is used to select one of the routers as a GDR. The GDR is responsible for initiating the forwarding tree building process for the corresponding multicast flow.
- o GDR Candidate: a last hop router that has the potential to become a GDR. A GDR Candidate must have the same DR priority and must run the same GDR election hash algorithm as the DR router. It must send and process new PIM Hello Options as defined in this document. There might be more than one GDR Candidate on a LAN, but only one can become GDR for a specific multicast flow.

## 3. Applicability

The proposed change described in this specification applies to PIM-SM last hop routers only.

It does not alter the behavior of a PIM DR on the first hop network. This is because the source tree is built using the IP address of the sender, not the IP address of the PIM DR that sends the registers towards the RP. The load balancing between first hop routers can be achieved naturally if an IGP provides equal cost multiple paths (which it usually does in practice). Also distributing the load to

do registering does not justify the additional complexity required to support it.

#### 4. Functional Overview

In the existing PIM DR election, when multiple last hop routers are connected to a multi-access LAN (for example, an Ethernet), one of them is selected to act as PIM DR. The PIM DR is responsible for sending local Join/Prune messages towards the RP or source. In order to elect the PIM DR, each PIM router on the LAN examines the received PIM Hello messages and compares its DR priority and IP address with those of its neighbors. The router with the highest DR priority is the PIM DR. If there are multiple such routers, their IP addresses are used as the tie-breaker, as described in [RFC4601].

In order to share forwarding load among last hop routers, besides the normal PIM DR election, the GDR is also elected on the last hop multi-access LAN. There is only one PIM DR on the multi-access LAN, but there might be multiple GDR Candidates.

For each multicast flow, that is, (\*,G) for ASM and (S,G) for SSM, a hash algorithm is used to select one of the routers to be the GDR. A new DR Load Balancing Capability (DRLBC) PIM Hello Option, which contains hash algorithm type, is announced by routers on interfaces where this specification is enabled. Last hop routers with the new DRLBC Option advertised in its Hello, and using the same GDR election hash algorithm and the same DR priority as the PIM DR, are considered as GDR Candidates.

Hash Masks are defined for Source, Group and RP separately, in order to handle PIM ASM/SSM. The masks, as well as a sorted list of GDR Candidates' Addresses, are announced by DR in a new DR Load Balancing GDR (DRLBGDR) PIM Hello Option.

A hash algorithm based on the announced Source, Group, or RP masks allows one GDR to be assigned to a corresponding multicast state. And that GDR is responsible for initiating the creation of the multicast forwarding tree for multicast traffic.

##### 4.1. GDR Candidates

GDR is the new concept introduced by this specification. GDR Candidates are routers eligible for GDR election on the LAN. To become a GDR Candidate, a router MUST support this specification, have the same DR priority and run the same GDR election hash algorithm as the DR on the LAN.

For example, assume there are 4 routers on the LAN: R1, R2, R3 and R4, which all support this specification. R1, R2 and R3 have the same DR priority while R4's DR priority is less preferred. In this example, R4 will not be eligible for GDR election, because R4 will not become a PIM DR unless all of R1, R2 and R3 go out of service.

Furthermore, assume router R1 wins the PIM DR election, R1 and R2 run the same hash algorithm for GDR election, while R3 runs a different one. In this case, only R1 and R2 will be eligible for GDR election, while R3 will not.

As a DR, R1 will include its own Load Balancing Hash Masks and the identity of R1 and R2 (the GDR Candidates) in its DRLBGDR Hello Option.

#### 4.2. Hash Mask and Hash Algorithm

A Hash Mask is used to extract a number of bits from the corresponding IP address field (32 for v4, 128 for v6) and calculate a hash value. A hash value is used to select a GDR from GDR Candidates advertised by PIM DR. For example, 0.0.255.0 defines a Hash Mask for an IPv4 address that masks the first, the second, and the fourth octets.

There are three Hash Masks defined,

- o RP Hash Mask
- o Source Hash Mask
- o Group Hash Mask

The hash masks need to be configured on the PIM routers that can potentially become a PIM DR, unless the implementation provides default Hash Mask. An implementation SHOULD provide masks with default values 255.255.255.255 (IPv4) and FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF (IPv6).

- o If the group is ASM and the RP Hash Mask announced by the PIM DR is not 0, calculate the value of hashvalue\_RP [Section 4.3] to determine GDR.
- o If the group is ASM and the RP Hash Mask announced by the PIM DR is 0, obtain the value of hashvalue\_Group [Section 4.3] to determine GDR.
- o If the group is SSM, use hashvalue\_SG [Section 4.3] to determine GDR.

A simple Modulo hash algorithm will be discussed in this document. However, to allow another hash algorithms to be used, a 4-bytes "Hash Algorithm Type" field is included in DRLBC Hello Option to specify the hash algorithm used by a last hop router.

If different hash algorithm types are advertised among last hop routers, only last hop routers running the same hash algorithm as the DR (and having the same DR priority as the DR) are eligible for GDR election.

#### 4.3. Modulo Hash Algorithm

Modulo hash algorithm is discussed here with a detailed description on `hashvalue_RP`. The same algorithm is described in brief for `hashvalue_Group` using the group address instead of the RP address for an ASM group with `RP_hashmask==0`, and also with `hashvalue_SG` for a the source address of an (S,G), instead of the RP address,

- o For ASM groups, with a non-zero `RP_Hash Mask`, hash value is calculated as:

$$\text{hashvalue\_RP} = (((\text{RP\_address} \& \text{RP\_hashmask}) \gg N) \& 0xFFFF) \% M$$

`RP_address` is the address of the RP defined for the group. `N` is the number of zeros, counted from the least significant bit of the `RP_hashmask`. `M` is the number of GDR Candidates.

For example, Router X with IPv4 address 203.0.113.1 receives a DRLBGDR Hello Option from the DR, which announces RP Hash Mask 0.0.255.0 and a list of GDR Candidates, sorted by IP addresses from high to low: 203.0.113.3, 203.0.113.2 and 203.0.113.1. The ordinal number assigned to those addresses would be:

0 for 203.0.113.3; 1 for 203.0.113.2; 2 for 203.0.113.1 (Router X)

Assume there are 2 RPs: RP1 192.0.2.1 for Group1 and RP2 198.51.100.2 for Group2. Following the modulo hash algorithm:

`N` is 8 for 0.0.255.0, and `M` is 3 for the total number of GDR Candidates. The `hashvalue_RP` for RP1 192.0.2.1 is:

$$(((192.0.2.1 \& 0.0.255.0) \gg 8) \& 0xFFFF \% 3) = 2 \% 3 = 2$$

matches the ordinal number assigned to Router X. Router X will be the GDR for Group1, which uses 192.0.2.1 as the RP.

The `hashvalue_RP` for RP2 198.51.100.2 is:

$((198.51.100.2 \& 0.0.255.0) \gg 8) \& 0xFFFF \% 3 = 100 \% 3 = 1$

which is different from Router X's ordinal number(2) hence, Router X will not be GDR for Group2.

- o If RP\_hashmask is 0, a hash value for ASM group is calculated using the group Hash Mask:

$\text{hashvalue\_Group} = (((\text{Group\_address} \& \text{Group\_hashmask}) \gg N) \& 0xFFFF) \% M$

Compare hashvalue\_Group with Ordinal number assigned to Router X, to decide if Router X is the GDR.

- o For SSM groups, a hash value is calculated using both the source and group Hash Mask:

$\text{hashvalue\_SG} = (((\text{Source\_address} \& \text{Source\_hashmask}) \gg N_S) \& 0xFFFF) \wedge (((\text{Group\_address} \& \text{Group\_hashmask}) \gg N_G) \& 0xFFFF) \% M$

#### 4.4. PIM Hello Options

When a last hop PIM router sends a PIM Hello from an interface with this specification enabled, it includes a new option, called "Load Balancing Capability (DRLBC)".

Besides this DRLBC Hello Option, the elected PIM DR also includes a new "DR Load Balancing GDR (DRLBGDR) Hello Option". The DRLBGDR Hello Option consists of three Hash Masks as defined above and also the sorted list of all GDR Candidates' Address on the last hop LAN.

The elected PIM DR uses DRLBC Hello Option advertised by all routers on the last hop LAN to compose its DRLBGDR. The GDR Candidates use DRLBGDR Hello Option advertised by PIM DR to calculate hash value.

### 5. Hello Option Formats

#### 5.1. PIM DR Load Balancing Capability (DRLBC) Hello Option



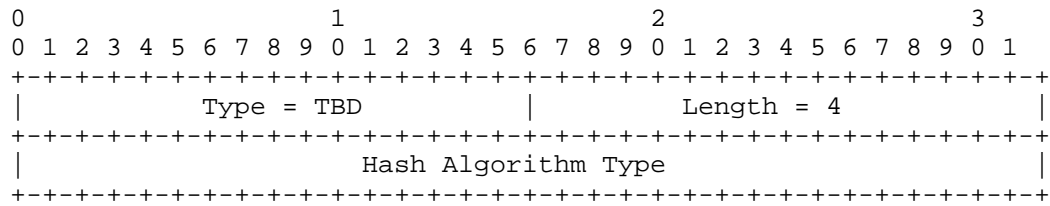


Figure 3: Capability Hello Option

Type: TBD.

Length: 4 octets

Hash Algorithm Type: 0 for Modulo hash algorithm

This DRLBC Hello Option SHOULD be advertised by last hop routers from interfaces with this specification enabled.

## 5.2. PIM DR Load Balancing GDR (DRLBGDR) Hello Option

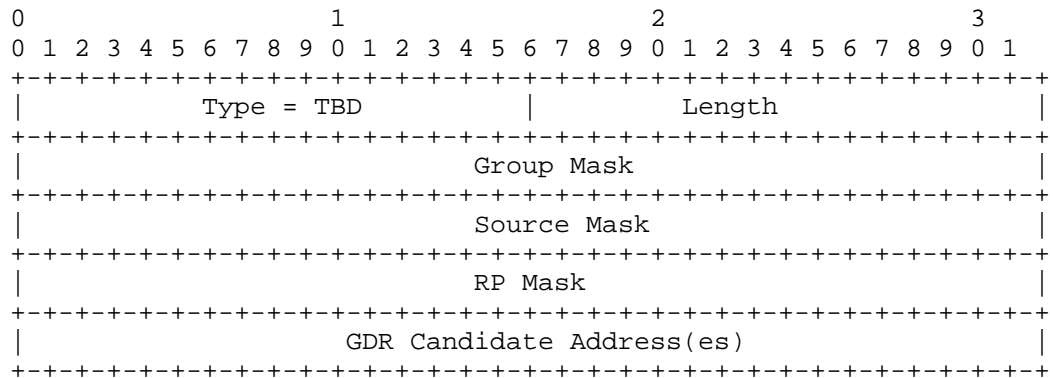


Figure 4: GDR Hello Option

Type: TBD

Length: 3 x (4 byte or 16 byte) + n x (4 byte or 16 byte) where n is the number of GDR candidates.

Group Mask (32/128 bits): Mask

Source Mask (32/128 bits): Mask

RP Mask (32/128 bits): Mask

All masks MUST be in the same address family as the Hello IP header.

GDR Address (32/128 bits): Address(es) of GDR Candidate(s)

All addresses must be in the same address family as the Hello IP header. The addresses are sorted in descending order. The order is converted to the ordinal number associated with each GDR candidate in hash value calculation. For example, addresses advertised are R3, R2, R1, the ordinal number assigned to R3 is 0, to R2 is 1 and to R1 is 2.

If "Interface ID" option, as described in [RFC6395], presents in a GDR Candidate's PIM Hello message, and the "Router ID" portion is non-zero,

- + For IPv4, the "GDR Candidate Address" will be set directly to "Router ID".
- + For IPv6, the "GDR Candidate Address" will be set to the IPv4-IPv6 translated address of "Router ID", as described in [RFC4291], that is the "Router-ID" is appended to the prefix of 96-bits zeros.

If the "Interface ID" option is not present in a GDR Candidate's PIM Hello message, or if the "Interface ID" option is present but the "Router ID" field is zero, the "GDR Candidate Address" will be the IPv4 or IPv6 source address from PIM Hello message.

This DRLBGDR Hello Option MUST only be advertised by the elected PIM DR.

## 6. Protocol Specification

### 6.1. PIM DR Operation

The DR election process is still the same as defined in [RFC4601]. A DR that has this specification enabled on the interface advertises the new DRLBGDR Hello Option, which contains value of masks from user configuration, followed by a sorted list of all GDR Candidates' Addresses, from the highest value to the lowest value. Moreover,

same as non-DR routers, DR also advertises DRLBC Hello Option to indicate its capability of supporting this specification and the type of its GDR election hash algorithm.

If a PIM DR receives a PIM Hello with DRLBGDR Option, the PIM DR SHOULD ignore the TLV.

If a PIM DR receives a neighbor DRLBC Hello Option, which contains the same hash algorithm type as the DR, and the neighbor has the same DR priority as the DR, PIM DR SHOULD consider the neighbor as a GDR Candidate and insert the GDR Candidate's Address into the sorted list of DRLBGDR Option.

## 6.2. PIM GDR Candidate Operation

When an IGMP/MLD join is received, without this specification, only PIM DR will handle the join and potentially run into the issues described earlier. Using this specification, a hash algorithm is used on GDR Candidate to determine which router is going to be responsible for building forwarding trees on behalf of the host.

If a router supports this specification then each of the interfaces where multicast protocol is enabled, it MUST advertise DRLBC Hello Option in its PIM Hello. Though DRLBC option in PIM hello does not guarantee that this router would be considered as a GDR candidate. For example, this router may have lower priority configured on shared LAN compare to other PIM routers. Once DR election is done, DRLBGDR Hello option would be received from the current PIM DR on the link which would contain list of GDR.

A GDR Candidate may receive a DRLBGDR Hello Option from PIM DR with different Hash Masks from those configured on it. The GDR Candidate must use the Hash Masks advertised by the PIM DR to calculate the hash value.

A GDR Candidate may receive a DRLBGDR Hello Option from a PIM router which is not DR. The GDR Candidate MUST ignore such DRLBGDR Hello Option.

A GDR Candidate may receive a Hello from the elected PIM DR, and the PIM DR does not support this specification. The GDR election described by this specification will not take place, that is only the PIM DR joins the multicast tree.

A router only acts as GDR if it is included in the GDR list of DRLBGDR Hello Option

### 6.2.1. Router Receives New DRLBGDR

When a router receives a new DRLBGDR from the current PIM DR, it need to process and check if router is in list of of GDR

1. If a router is not listed as a GDR candidate in DRLBGDR, no action is needed.
2. If a router is listed as a GDR candidate in DRLBGDR, then it need to process each of the groups in the IGMP/MLD reports. The masks are announced in the PIM Hello by DR as DRLBGDR Hello option. For each of groups in the reports it (PIM Router) needs to run hash algorithm (described in section 4.3) based on the announced Source, Group or RP masks to determine if it is GDR for specified group. If the hash result is to be the GDR for the multicast flow, it does build the multicast forwarding tree. If it is not the GDR for the multicast flow, no action is needed.

### 6.2.2. Router Receives Updated DRLBGDR

If a router (GDR or non GDR) receives an unchanged DRLBGDR from the current PIM DR, no action is needed.

If a router (GDR or non GDR) receives a new or modified DRLBGDR from the current PIM DR. It requires processing as described below:

1. If it was GDR and still included in current GDR list: it needs to process each of the groups and run the hash algorithm to check if it is still the GDR for the given group.

If it was the GDR for group G and the new hash result chose it as the GDR, then no processing is required.

If it was the GDR for a group earlier and now it is no longer the GDR, then it sets its assert metric for the multicast flow to be (PIM\_ASSERT\_INFINITY - 1), as explained in Sec 6.3

If it was not the GDR for a group earlier, than even the new hash does not make it GDR. For the multicast group no processing is required.

If it was not the GDR for an earlier group and now becomes the GDR, it starts building multicast forwarding tree for this flow.

2. If it was not the GDR , and updated DRLBGDR from current PIM DR contains this router as one of the GDR. In this case this router

being new GDR candidate MUST run hash algorithm for each of the groups (multicast flows) and for given group,

If it is not the GDR, no processing is required.

If it is hashed as the GDR, it needs to build multicast forwarding tree.

### 6.3. PIM Assert Modification

It is possible that the identity of the GDR might change in the middle of an active flow. Examples this could happen include:

When a new PIM router comes up

When a GDR restarts

When the GDR changes, existing traffic might be disrupted. Duplicates or packet losses might be observed. To illustrate the case, consider the following scenario where there are two streams G1 and G2. R1 is the GDR for G1, and R2 is the GDR for G2. When R3 comes up online, it is possible that R3 becomes GDR for both G1 and G2, hence R3 starts to build the forwarding tree for G1 and G2. If R1 and R2 stop forwarding before R3 completes the process, packet loss might occur. On the other hand, if R1 and R2 continue forwarding while R3 is building the forwarding trees, duplicates might occur.

This is not a typical deployment scenario but might still happen. Here we describe a mechanism to minimize the impact. We essentially want to minimize packet loss. Therefore, we would allow a small amount of duplicates and depend on PIM Assert to minimize the duplication.

When the role of GDR changes as above, instead of immediately stopping forwarding, R1 and R2 continue forwarding to G1 and G2 respectively, while, at the same time, R3 build forwarding trees for G1 and G2. This will lead to PIM Asserts.

With the introduction of GDR, the following modification to the Assert packet MUST be done: if a router enables this specification on its downstream interface, but it is not a GDR (before network event it was GDR), it would adjust its Assert metric to (PIM\_ASSERT\_INFINITY - 1).

Using the above example, for G1, assume R1 and R3 agree on the new GDR, which is R3. R1 will set its Assert metric as

(PIM\_ASSERT\_INFINITY - 1). That will make R3, which has normal metric in its Assert as the Assert winner.

For G2, assume it takes a slightly longer time for R2 to find out that R3 is the new GDR and still considers itself being the GDR while R3 already has assumed the role of GDR. Since both R2 and R3 think they are GDRs, they further compare the metric and IP address. If R3 has the better routing metric, or the same metric but a better tie-breaker, the result will be consistent during GDR selection. If unfortunately, R2 has the better metric or the same metric but a better tie-breaker, R2 will become the Assert winner and continues to forward traffic. This will continue until:

The next PIM Hello option from DR selects R3 as the GDR. R3 will then build the forwarding tree and send an Assert.

The process continues until R2 agrees to the selection of R3 as the GDR, and set its own Assert metric to (PIM\_ASSERT\_INFINITY - 1), which will make R3 the Assert winner. During the process, we will see intermittent duplication of traffic but packet loss will be minimized. In the unlikely case that R2 never relinquishes its role as GDR (while every other router thinks otherwise), the proposed mechanism also helps to keep the duplication to a minimum until manual intervention takes place to remedy the situation.

## 7. Compatibility

In case of the hybrid Ethernet shared LAN ( where some PIM router enables specification defined in this draft and some do not enable)

- o If a router which does not support specification defined in this draft becomes DR on link, it MUST be only DR on link as [RFC4601] and there would be no router which would act as GDR.
- o If a router which does not support specification defined in this draft becomes non DR on link, then it should act as non-DR defined in [RFC4601].

## 8. Manageability Considerations

- o All of the routers in LAN that support this specification MUST use identical Hash Algorithm Type (described in section 5.1). In the case of a hybrid Hash Algorithm Type, one MUST go backward to use DR election method defined in PIM-SM [RFC4601]. Migration between different algorithm type is out of the scope of this document.

## 9. IANA Considerations

IANA has temporarily assigned type 34 for the PIM DR Load Balancing Capability (DRLBC) Hello Option, and type 35 for the PIM DR Load Balancing GDR (DRLBGDR) Hello Option. IANA is requested to make these assignments permanent when this document is published as an RFC. The string TBD should be replaced by the assigned values accordingly.

## 10. Security Considerations

Security of the new DR Load Balancing PIM Hello Options is only guaranteed by the security of PIM Hello message, so the security considerations for PIM Hello messages as described in PIM-SM [RFC4601] apply here.

## 11. Acknowledgement

The authors would like to thank Steve Simlo, Taki Millonis for helping with the original idea, Bill Atwood, Bharat Joshi for review comments, Toerless Eckert and Rishabh Parekh for helpful conversation on the document.

Special thanks to Anish Kachinthaya, Anvitha Kachinthaya and Jake Holland for reviewing the document and providing comments.

## 12. References

### 12.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <<https://www.rfc-editor.org/info/rfc4291>>.
- [RFC4601] Fenner, B., Handley, M., Holbrook, H., and I. Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", RFC 4601, DOI 10.17487/RFC4601, August 2006, <<https://www.rfc-editor.org/info/rfc4601>>.
- [RFC6395] Gulrajani, S. and S. Venaas, "An Interface Identifier (ID) Hello Option for PIM", RFC 6395, DOI 10.17487/RFC6395, October 2011, <<https://www.rfc-editor.org/info/rfc6395>>.

## 12.2. Informative References

[HELLO-OPT]

IANA, "PIM Hello Options", IANA PIM-HELLO-OPTIONS, March 2007.

## Authors' Addresses

Yiqun Cai  
Alibaba Group

Email: [yiqun.cai@alibaba-inc.com](mailto:yiqun.cai@alibaba-inc.com)

Heidi Ou  
Alibaba Group

Sri Vallepalli  
Cisco Systems  
3625 Cisco Way,  
San Jose, CALIFORNIA 95134  
UNITED STATES

Email: [svallepa@cisco.com](mailto:svallepa@cisco.com)

Mankamana Mishra  
Cisco Systems  
821 Alder Drive,  
MILPITAS, CALIFORNIA 95035  
UNITED STATES

Email: [mankamis@cisco.com](mailto:mankamis@cisco.com)

Stig Venaas  
Cisco Systems  
821 Alder Drive,  
MILPITAS, CALIFORNIA 95035  
UNITED STATES

Email: [stig@cisco.com](mailto:stig@cisco.com)



Andy Green  
British Telecom  
Adastral Park  
Ipswich IP5 2RE  
United Kingdom

Email: [andy.da.green@bt.com](mailto:andy.da.green@bt.com)

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: July 6, 2020

Y. Cai  
H. Ou  
Alibaba Group  
S. Vallepalli  
M. Mishra  
S. Venaas  
Cisco Systems, Inc.  
A. Green  
British Telecom  
January 3, 2020

PIM Designated Router Load Balancing  
draft-ietf-pim-drlb-15

Abstract

On a multi-access network, one of the PIM-SM (PIM Sparse Mode) routers is elected as a Designated Router. One of the responsibilities of the Designated Router is to track local multicast listeners and forward data to these listeners if the group is operating in PIM-SM. This document specifies a modification to the PIM-SM protocol that allows more than one of the PIM-SM routers to take on this responsibility so that the forwarding load can be distributed among multiple routers.

Status of This Memo

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

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

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

This Internet-Draft will expire on July 6, 2020.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

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

## Table of Contents

1. Introduction . . . . .	2
2. Terminology . . . . .	5
3. Applicability . . . . .	5
4. Functional Overview . . . . .	5
4.1. GDR Candidates . . . . .	6
5. Protocol Specification . . . . .	7
5.1. Hash Mask and Hash Algorithm . . . . .	7
5.2. Modulo Hash Algorithm . . . . .	8
5.2.1. Modulo Hash Algorithm Examples . . . . .	9
5.2.2. Limitations . . . . .	10
5.3. PIM Hello Options . . . . .	11
5.3.1. PIM DR Load Balancing Capability (DRLB-Cap) Hello Option . . . . .	11
5.3.2. PIM DR Load Balancing List (DRLB-List) Hello Option . . . . .	12
5.4. PIM DR Operation . . . . .	13
5.5. PIM GDR Candidate Operation . . . . .	14
5.6. DRLB-List Hello Option Processing . . . . .	14
5.7. PIM Assert Modification . . . . .	15
5.8. Backward Compatibility . . . . .	16
6. Operational Considerations . . . . .	16
7. IANA Considerations . . . . .	17
7.1. Initial registry . . . . .	17
7.2. Assignment of new Hash Algorithms . . . . .	17
8. Security Considerations . . . . .	17
9. Acknowledgement . . . . .	18
10. References . . . . .	18
10.1. Normative References . . . . .	18
10.2. Informative References . . . . .	19
Authors' Addresses . . . . .	19

## 1. Introduction

On a multi-access LAN, such as an Ethernet, with one or more PIM-SM (PIM Sparse Mode) [RFC7761] routers, one of the PIM-SM routers is elected as a Designated Router (DR). The PIM DR has two responsibilities in the PIM-SM protocol. For any active sources on a

LAN, the PIM DR is responsible for registering with the Rendezvous Point (RP) if the group is operating in PIM-SM. Also, the PIM DR is responsible for tracking local multicast listeners and forwarding to these listeners if the group is operating in PIM-SM.

Consider the following LAN in Figure 1:

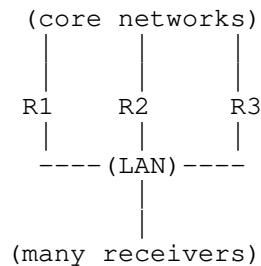


Figure 1: LAN with receivers

Assume R1 is elected as the DR. According to the PIM-SM protocol, R1 will be responsible for forwarding traffic to that LAN on behalf of all local members. In addition to keeping track of membership reports, R1 is also responsible for initiating the creation of source and/or shared trees towards the senders or the RPs. The membership reports would be IGMP or MLD messages. This applies to any versions of the IGMP and MLD protocols. The most recent versions are IGMPv3 [RFC3376] and MLDv2 [RFC3810].

Having a single router acting as DR and being responsible for data plane forwarding leads to several issues. One of the issues is that the aggregated bandwidth will be limited to what R1 can handle with regards to capacity of incoming links, the interface on the LAN, and total forwarding capacity. It is very common that a LAN consists of switches that run IGMP/MLD or PIM snooping [RFC4541]. This allows the forwarding of multicast packets to be restricted only to segments leading to receivers that have indicated their interest in multicast groups using either IGMP or MLD. The emergence of the switched Ethernet allows the aggregated bandwidth to exceed, sometimes by a large number, that of a single link. For example, let us modify Figure 1 and introduce an Ethernet switch in Figure 2.

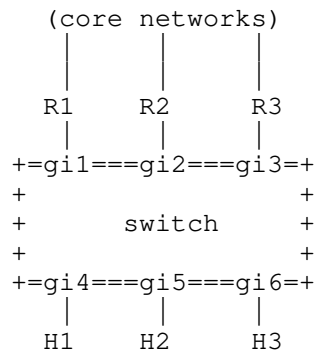


Figure 2: LAN with Ethernet Switch

Let us assume that each individual link is a Gigabit Ethernet. Each router, R1, R2 and R3, and the switch have enough forwarding capacity to handle hundreds of Gigabits of data.

Let us further assume that each of the hosts requests 500 Mbps of unique multicast data. This totals to 1.5 Gbps of data, which is less than what each switch or the combined uplink bandwidth across the routers can handle, even under failure of a single router.

On the other hand, the link between R1 and switch, via port gi1, can only handle a throughput of 1Gbps. And if R1 is the only DR (the PIM DR elected using the procedure defined by [RFC7761]) at least 500 Mbps worth of data will be lost because the only link that can be used to draw the traffic from the routers to the switch is via gi1. In other words, the entire network's throughput is limited by the single connection between the PIM DR and the switch (or LAN as in Figure 1).

Another important issue is related to failover. If R1 is the only forwarder on a shared LAN, when R1 goes out of service, multicast forwarding for the entire LAN has to be rebuilt by the newly elected PIM DR. However, if there were a way that allowed multiple routers to forward to the LAN for different groups, failure of one of the routers would only lead to disruption to a subset of the flows, therefore improving the overall resilience of the network.

This document specifies a modification to the PIM-SM protocol that allows more than one of these routers, called Group Designated

Routers (GDR) to be selected so that the forwarding load can be distributed among a number of routers.

## 2. Terminology

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

With respect to PIM-SM, this document follows the terminology that has been defined in [RFC7761].

This document also introduces the following new acronyms:

- o GDR: Group Designated Router. For each multicast flow, either a (\*,G) for Any-Source Multicast (ASM), or an (S,G) for Source-Specific Multicast (SSM) [RFC4607], a Hash Algorithm (described below) is used to select one of the routers as a GDR. The GDR is responsible for initiating the forwarding tree building process for the corresponding multicast flow.
- o GDR Candidate: a router that has the potential to become a GDR. There might be multiple GDR Candidates on a LAN, but only one can become the GDR for a specific multicast flow.

## 3. Applicability

The extension specified in this document applies to PIM-SM routers acting as last hop routers (there are directly connected receivers). It does not alter the behavior of a PIM DR, or any other routers, on the first hop network (directly connected sources). This is because the source tree is built using the IP address of the sender, not the IP address of the PIM DR that sends PIM registers towards the RP. The load balancing between first hop routers can be achieved naturally if an IGP provides equal cost multiple paths (which it usually does in practice). Also distributing the load to do source registration does not justify the additional complexity required to support it.

## 4. Functional Overview

In the PIM DR election as defined in [RFC7761], when multiple routers are connected to a multi-access LAN (for example, an Ethernet), one of them is elected to act as PIM DR. The PIM DR is responsible for sending local Join/Prune messages towards the RP or source. In order to elect the PIM DR, each PIM router on the LAN examines the received

PIM Hello messages and compares its own DR priority and IP address with those of its neighbors. The router with the highest DR priority is the PIM DR. If there are multiple such routers, their IP addresses are used as the tie-breaker, as described in [RFC7761].

In order to share forwarding load among last hop routers, besides the normal PIM DR election, one or more GDRs are elected on the multi-access LAN. There is only one PIM DR on the multi-access LAN, but there might be multiple GDR Candidates.

For each multicast flow, that is, (\*,G) for ASM and (S,G) for SSM, a Hash Algorithm [Section 5.1] is used to select one of the routers to be the GDR. The new DR Load Balancing Capability (DRLB-Cap) PIM Hello Option is used to announce the Capability as well as the Hash Algorithm type. Routers with the new DRLB-Cap Option advertised in their PIM Hello, using the same GDR election Hash Algorithm and the same DR priority as the PIM DR, are considered as GDR Candidates.

Hash Masks are defined for Source, Group and RP separately, in order to handle PIM ASM/SSM. The masks, as well as a sorted list of GDR Candidate Addresses, are announced by the DR in a new DR Load Balancing List (DRLB-List) PIM Hello Option.

A Hash Algorithm based on the announced Source, Group, or RP masks allows one GDR to be assigned to a corresponding multicast state. That GDR is responsible for initiating the creation of the multicast forwarding tree for multicast traffic.

#### 4.1. GDR Candidates

GDR is the new concept introduced by this specification. GDR Candidates are routers eligible for GDR election on the LAN. To become a GDR Candidate, a router must have the same DR priority and run the same GDR election Hash Algorithm as the DR on the LAN.

For example, assume there are 4 routers on the LAN: R1, R2, R3 and R4, each announcing a DRLB-Cap option. R1, R2 and R3 have the same DR priority while R4's DR priority is less preferred. In this example, R4 will not be eligible for GDR election, because R4 will not become a PIM DR unless all of R1, R2 and R3 go out of service.

Furthermore, assume router R1 wins the PIM DR election, R1 and R2 advertise the same Hash Algorithm for GDR election, while R3 advertises a different one. In this case, only R1 and R2 will be eligible for GDR election, while R3 will not.

As a DR, R1 will include its own Load Balancing Hash Masks and the identity of R1 and R2 (the GDR Candidates) in its DRLB-List Hello Option.

## 5. Protocol Specification

### 5.1. Hash Mask and Hash Algorithm

A Hash Mask is used to extract a number of bits from the corresponding IP address field (32 for IPv4, 128 for IPv6) and calculate a hash value. A hash value is used to select a GDR from GDR Candidates advertised by the PIM DR. Hash masks allow for certain flows to always be forwarded by the same GDR, by ignoring certain bits in the hash value calculation, so that the hash values are the same. For example, 0.0.255.0 defines a Hash Mask for an IPv4 address that masks the first, the second, and the fourth octets, which means that only the third octet will influence the hash value computed. Note that the masks need not be a contiguous set of bits. E.g, for IPv4, 15.15.15.15 would be a valid mask.

In the text below, a hash mask is in some places said to be zero. A hash mask is zero if no bits are set. That is, 0.0.0.0 for IPv4 and :: for IPv6. Also, a hash mask is said to be an all-bits-set mask if it is 255.255.255.255 for IPv4 or ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff for IPv6.

There are three Hash Masks defined:

- o RP Hash Mask
- o Source Hash Mask
- o Group Hash Mask

The hash masks need to be configured on the PIM routers that can potentially become a PIM DR, unless the implementation provides default hash mask values. An implementation SHOULD have default hash mask values as follows. The default RP Hash Mask SHOULD be zero (no bits set). The default Source and Group Hash Masks SHOULD both be all-bits-set masks. These default values are likely acceptable for most deployments, and simplify configuration. There is only a need to use other masks if one needs to ensure that certain flows are forwarded by the same GDR.

The DRLB-List Hello Option contains a list of GDR Candidates. The first one listed has ordinal number 0, the second listed ordinal number 1, and the last one has ordinal number N - 1 if there are N candidates listed. The hash value computed will be the ordinal



number of the GDR Candidate that is acting as GDR for the flow in question.

The input to be hashed is determined as follows:

- o If the group is in ASM mode and the RP Hash Mask announced by the PIM DR is not zero (at least one bit is set), calculate the value of `hashvalue_RP` [Section 5.2] to determine the GDR.
- o If the group is in ASM mode and the RP Hash Mask announced by the PIM DR is zero (no bits are set), obtain the value of `hashvalue_Group` [Section 5.2] to determine the GDR.
- o If the group is in SSM mode, use `hashvalue_SG` [Section 5.2] to determine the GDR.

A simple Modulo Hash Algorithm is defined in this document. However, to allow another Hash Algorithms to be used, a 1-octet "Hash Algorithm" field is included in the DRLB-Cap Hello Option to specify the Hash Algorithm used by the router.

If different Hash Algorithms are advertised among the routers on a LAN, only the routers advertising the same Hash Algorithm as the DR (as well as having the same DR priority as the DR) are eligible for GDR election.

## 5.2. Modulo Hash Algorithm

As part of computing the hash, the notation `LSZC(hash_mask)` is used to denote the number of zeroes counted from the least significant bit of a Hash Mask `hash_mask`. As an example, `LSZC(255.255.128)` is 7 and also `LSZC(ffff:8000::)` is 111. If all bits are set, `LSZC` will be 0. If the mask is zero, then `LSZC` will be 32 for IPv4, and 128 for IPv6.

The number of GDR Candidates is denoted as `GDRC`.

The idea behind the Modulo Hash Algorithm is in simple terms that the corresponding mask is applied to a value, then the result is shifted right `LSZC(mask)` bits so that the least significant bits that were masked out are not considered. Then this result is masked by `0xffffffff`, keeping only the last 32 bits of the result (this only makes a difference for IPv6). Finally, the hash value is this result modulo the number of GDR Candidates (`GDRC`).

The Modulo Hash Algorithm for computing the values `hashvalue_RP`, `hashvalue_Group` and `hashvalue_SG` is defined as follows.

`hashvalue_RP` is calculated as:

$$(((RP\_address \& RP\_mask) \gg LSZC(RP\_mask)) \& 0xffffffff) \% GDRC$$

RP\_address is the address of the RP defined for the group and  
RP\_mask is the RP Hash Mask.

hashvalue\_Group is calculated as:

$$(((Group\_address \& Group\_mask) \gg LSZC(Group\_mask)) \& 0xffffffff) \% GDRC$$

Group\_address is the group address and Group\_mask is the Group Hash Mask.

hashvalue\_SG is calculated as:

$$((((Source\_address \& Source\_mask) \gg LSZC(Source\_mask)) \& 0xffffffff) \wedge (((Group\_address \& Group\_mask) \gg LSZC(Group\_mask)) \& 0xffffffff)) \% GDRC$$

Group\_address is the group address and Group\_mask is the Group Hash Mask.

#### 5.2.1. Modulo Hash Algorithm Examples

To help illustrate the algorithm, consider this example. Router X with IPv4 address 203.0.113.1 receives a DRLB-List Hello Option from the DR, which announces RP Hash Mask 0.0.255.0 and a list of GDR Candidates, sorted by IP addresses from high to low: 203.0.113.3, 203.0.113.2 and 203.0.113.1. The ordinal number assigned to those addresses would be:

0 for 203.0.113.3; 1 for 203.0.113.2; 2 for 203.0.113.1 (Router X).

Assume there are 2 RPs: RP1 192.0.2.1 for Group1 and RP2 198.51.100.2 for Group2. Following the modulo Hash Algorithm:

LSZC(0.0.255.0) is 8 and GDRC is 3. The hashvalue\_RP for Group1 with RP RP1 is:

$$(((192.0.2.1 \& 0.0.255.0) \gg 8) \& 0xffffffff \% 3) = 2 \% 3 = 2$$

which matches the ordinal number assigned to Router X. Router X will be the GDR for Group1.

The hashvalue\_RP for Group2 with RP RP2 is:

$$(((198.51.100.2 \& 0.0.255.0) \gg 8) \& 0xffffffff \% 3) = 100 \% 3 = 1$$

which is different from the ordinal number of Router X (2). Hence, Router X will not be GDR for Group2.

For IPv6 consider this example, similar to the above. Router X with IPv6 address fe80::1 receives a DRLB-List Hello Option from the DR, which announces RP Hash Mask ::ffff:ffff:ffff:0 and a list of GDR Candidates, sorted by IP addresses from high to low: fe80::3, fe80::2 and fe80::1. The ordinal number assigned to those addresses would be:

0 for fe80::3; 1 for fe80::2; 2 for fe80::1 (Router X).

Assume there are 2 RPs: RP1 2001:db8::1:0:5678:1 for Group1 and RP2 2001:db8::1:0:1234:2 for Group2. Following the modulo Hash Algorithm:

LSZC(::ffff:ffff:ffff:0) is 16 and GDRC is 3. The hashvalue\_RP for Group1 with RP RP1 is:

$$(((2001:db8::1:0:5678:1 \& ::ffff:ffff:ffff:0) \gg 16) \& 0xffffffff \% 3) = ((::1:0:5678:0 \gg 16) \& 0xffffffff \% 3) = (::1:0:5678 \& 0xffffffff \% 3) = ::5678 \% 3 = 2$$

which matches the ordinal number assigned to Router X. Router X will be the GDR for Group1.

The hashvalue\_RP for Group2 with RP RP2 is:

$$(((2001:db8::1:0:1234:1 \& ::ffff:ffff:ffff:0) \gg 16) \& 0xffffffff \% 3) = ((::1:0:1234:0 \gg 16) \& 0xffffffff \% 3) = (::1:0:1234 \& 0xffffffff \% 3) = ::1234 \% 3 = 1$$

which is different from the ordinal number of Router X (2). Hence, Router X will not be GDR for Group2.

#### 5.2.2. Limitations

The Modulo Hash Algorithm has poor failover characteristics when a shared LAN has more than two GDRs. In the case of more than two GDRs on a LAN, when one GDR fails, all of the groups may be reassigned to a different GDR, even if they were not assigned to the failed GDR. However, many deployments use only two routers on a shared LAN for redundancy purposes. Future work may define new Hash Algorithms where only groups assigned to the failed GDR get reassigned.

The Modulo Hash Algorithm will use at most 32 consecutive bits of the input addresses for its computation. Exactly which bits are used of the source, group or RP addresses, depend on the respective masks.

This limitation may be an issue for IPv6 deployments, since not all bits of the IPv6 addresses are considered. If this causes operational issues, a new hash algorithm would need to be defined.

### 5.3. PIM Hello Options

PIM routers include a new option, called "Load Balancing Capability (DRLB-Cap)" in their PIM Hello messages.

Besides this DRLB-Cap Hello Option, the elected PIM DR also includes a new "DR Load Balancing List (DRLB-List) Hello Option". The DRLB-List Hello Option consists of three Hash Masks as defined above and also a list of GDR Candidate addresses on the LAN. It is recommended that the GDR Candidate addresses are sorted in descending order. This ensures that when using algorithms such as the Modulo algorithm in this document, that it is predictable which GDR is responsible for which groups, regardless of the order the DR learned about the candidates.

#### 5.3.1. PIM DR Load Balancing Capability (DRLB-Cap) Hello Option

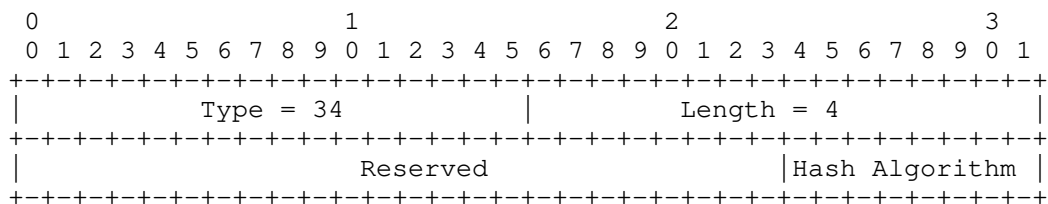


Figure 3: PIM DR Load Balancing Capability Hello Option

Type: 34

Length: 4

Reserved: Transmitted as zero, ignored on receipt.

Hash Algorithm: Hash Algorithm type. A value listed in the IANA Designated Router Load Balancing Hash Algorithms registry. 0 is used for the Modulo algorithm defined in this document.

This DRLB-Cap Hello Option MUST be advertised by routers on all interfaces where DR Load Balancing is enabled. Note that the option is included at most once.

## 5.3.2. PIM DR Load Balancing List (DRLB-List) Hello Option

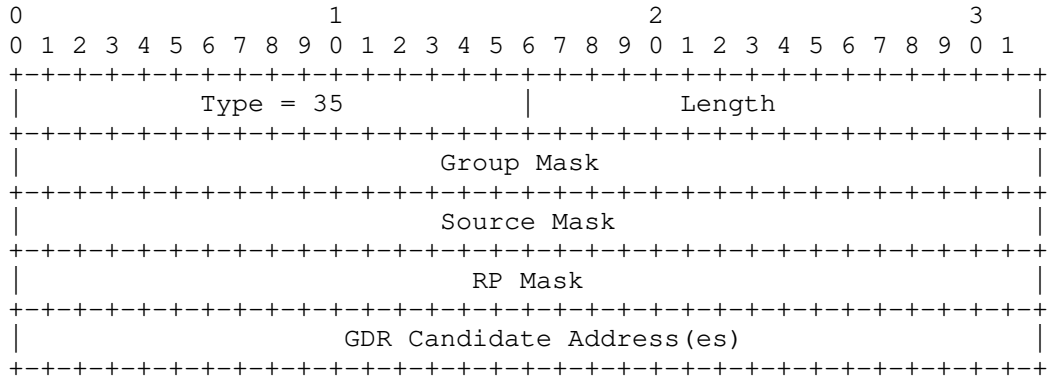


Figure 4: PIM DR Load Balancing List Hello Option

Type: 35

Length:  $(3 + n) \times (4 \text{ or } 16)$  bytes, where  $n$  is the number of GDR candidates.

Group Mask (32/128 bits): Mask applied to group addresses as part of hash computation.

Source Mask (32/128 bits): Mask applied to source addresses as part of hash computation.

RP Mask (32/128 bits): Mask applied to RP addresses as part of hash computation.

All masks MUST have the same number of bits as the IP source address in the PIM Hello IP header.

GDR Candidate Address(es) (32/128 bits): List of GDR Candidate(s)

All addresses MUST be in the same address family as the PIM Hello IP header. It is recommended that the addresses are sorted in descending order.

If the "Interface ID" option, as specified in [RFC6395], is present in a GDR Candidate's PIM Hello message, and the "Router Identifier" portion is non-zero:

- + For IPv4, the "GDR Candidate Address" will be set directly to the "Router Identifier".
- + For IPv6, the "GDR Candidate Address" will be 96 bits of zeroes followed by the 32 bit Router Identifier.

If the "Interface ID" option is not present in a GDR Candidate' PIM Hello message, or if the "Interface ID" option is present but the "Router Identifier" field is zero, the "GDR Candidate Address" will be the IPv4 or IPv6 source address of the PIM Hello message.

This DRLB-List Hello Option MUST only be advertised by the elected PIM DR. It MUST be ignored if received from a non-DR. The option MUST also be ignored if the hash masks are not the correct number of bits, or GDR Candidate addresses are in the wrong address family.

#### 5.4. PIM DR Operation

The DR election process is still the same as defined in [RFC7761]. The DR advertises the new DRLB-List Hello Option, which contains mask values from user configuration (or default values), followed by a list of GDR Candidate Addresses. Note that if a router included the "Interface ID" option in the hello message, and the Router ID is non-zero, the Router ID will be used to form the GDR Candidate address of the router, as discussed in the previous section. It is recommended that the list be sorted, from the highest value to the lowest value. The reason for sorting the list is to make the behavior deterministic, regardless of the order in which the DR learns of new candidates. Note that, as for non-DR routers, the DR also advertises the DRLB-Cap Hello Option to indicate its ability to support the new functionality and the type of GDR election Hash Algorithm it uses.

If a PIM DR receives a neighbor DRLB-Cap Hello Option, which contains the same Hash Algorithm as the DR, and the neighbor has the same DR priority as the DR, PIM DR SHOULD consider the neighbor as a GDR Candidate and insert the GDR Candidate' Address into the list of the DRLB-List Option. However, the DR may have policies limiting which GDR Candidates, or the number of GDR Candidates to include. Likewise, the DR SHOULD include itself in the list of GDR Candidates, but it is permissible not to do so, if for instance there is some policy restricting the candidate set.

If a PIM neighbor included in the list expires, stops announcing the DRLB-Cap Hello Option, changes DR priority, changes Hash Algorithm or otherwise becomes ineligible as a candidate, the DR SHOULD

immediately send a triggered hello with a new list in the DRLB-List option, excluding the neighbor.

If a new router becomes eligible as a candidate, there is no urgency in sending out an updated list. An updated list SHOULD be included in the next hello.

#### 5.5. PIM GDR Candidate Operation

When an IGMP/MLD report is received, a Hash Algorithm is used by the GDR Candidates to determine which router is going to be responsible for building forwarding trees on behalf of the host.

The router MUST include the DRLB-Cap Hello Option in all PIM Hello messages sent on the interface. Note that the presence of the DRLB-Cap Option in the PIM Hello does not guarantee that the router will be considered as a GDR candidate. Once the DR election is done, the DRLB-List Hello Option is received from the current PIM DR containing a list of the selected GDRs Candidates.

A router only acts as a GDR Candidate if it is included in the GDR Candidate list of the DRLB-List Hello Option. See next section for details.

#### 5.6. DRLB-List Hello Option Processing

This section discusses processing of the DRLB-List Hello Option, including the case where it was received in the previous hello, but not in the current hello. All routers MUST ignore the DRLB-List Hello Option if it is received from a PIM router which is not the DR. The option MUST only be processed by routers that are announcing the DRLB-Cap Option, and only if the Hash Algorithm announced by the DR is the same as the local announcement. All GDR Candidates MUST use the Hash Masks advertised in the Option, even if they differ from those the candidate was configured with. The DR MUST also process its own DRLB-List Hello Option.

A router stores the latest option contents that was announced, if any, and deletes the previous contents. The router MUST also compare the new contents with any previous contents, and if there are any changes, continue processing as below. Note that if the option does not pass the above checks, the below processing MUST be done as if the option was not announced.

If the contents of the DRLB-List Option, the masks or the candidate list, differs from the previously saved copy, it is received for the first time, or it is no longer being received or accepted, the option MUST be processed as below.

1. If the local router is included in the GDR Candidate Address(es) field (it will look for its own address, or its Router ID if it announces a non-zero Router ID), for each of the groups, or source and group pairs if the group is in SSM mode, with local receiver interest, the router MUST run the Hash Algorithm to determine which of them it is the GDR for.

If there is no change in the GDR status, then no further action is required.

If the router becomes the new GDR, then a multicast forwarding tree MUST be built [RFC7761].

If the router is no longer the GDR, then it uses an Assert as explained in [Section 5.7].

2. If the local router is not included in the GDR Candidate Address(es) field, or if the DRLB-List Hello Option is no longer included in the DR's Hello, or if the DR's Neighbor Liveness Timer expires [RFC7761], for each of the groups, or source and group pairs if the group is in SSM mode, with local receiver interest, for which the router is the GDR, it uses an Assert as explained in [Section 5.7].

#### 5.7. PIM Assert Modification

GDR changes may occur due to configuration change, due to GDR candidates going down, and also new routers coming up and becoming GDR candidates. This may occur while flows are being forwarded. If the GDR for an active flow changes, there is likely to be some disruption, such as packet loss or duplicates. By using asserts, packet loss is minimized, while allowing a small amount of duplicates.

When a router stops acting as the GDR for a group, or source and group pair if SSM, it MUST set the Assert metric preference to maximum (0x7fffffff) and the Assert metric to one less than maximum (0xffffffff). That is, whenever it sends or receives an Assert for the group, it must use these values as the metric preference and metric rather than the values provided by the unicast routing protocol.

The rest of this section is just for illustration purposes and not part of the protocol definition.

To illustrate the behavior when there is a GDR change, consider the following scenario where there are two flows G1 and G2. R1 is the GDR for G1, and R2 is the GDR for G2. When R3 comes up, it is



possible that R3 becomes GDR for both G1 and G2, hence R3 starts to build the forwarding tree for G1 and G2. If R1 and R2 stop forwarding before R3 completes the process, packet loss might occur. On the other hand, if R1 and R2 continue forwarding while R3 is building the forwarding trees, duplicates might occur.

When the role of GDR changes as above, instead of immediately stopping forwarding, R1 and R2 continue forwarding to G1 and G2 respectively, while, at the same time, R3 build forwarding trees for G1 and G2. This will lead to PIM Asserts.

For G1, using the functionality described in this document, R1 and R3 determine the new GDR, which is R3. With the modified Assert behavior, R1 sets its Assert metric to the near maximum value discussed above. That will make R3, which has normal metric in its Assert as the Assert winner.

#### 5.8. Backward Compatibility

In the case of a hybrid Ethernet shared LAN (where some PIM routers support the functionality defined in this document, and some do not);

- o If the DR does not support the new functionality, then there will be no load-balancing.
- o If non-DR routers do not support the new functionality, they will not be considered as Candidate GDRs and it will not take part in load-balancing. Load-balancing may still happen on the link.

#### 6. Operational Considerations

An administrator needs to consider what the total bandwidth requirements are and find a set of routers that together has enough available capacity, while making sure that each of the routers can handle its part, assuming that the traffic is distributed roughly equally among the routers. Ideally, one should also have enough bandwidth to handle the case where at least one router fails. All routers should have reachability to the sources, and RPs if applicable, that is not via the LAN.

Care must be taken when choosing what hash masks to configure. One would typically configure the same masks on all the routers, so that they are the same, regardless of which router is elected as DR. The default masks are likely suitable for most deployment. The RP Hash Mask must be configured (the default is no bits set) if one wishes to hash based on the RP address rather than the group address for ASM. The default masks will use the entire group addresses, and source addresses if SSM, as part of the hash. An administrator may set

other masks that masks out part of the addresses to ensure that certain flows always get hashed to the same router. How this is achieved depends on how the group addresses are allocated.

Only the routers announcing the same Hash Algorithm as the DR would be considered as GDR candidates. Network administrators need to make sure that the desired set of routers announce the same algorithm. Migration between different algorithms is not considered in this document.

## 7. IANA Considerations

IANA has temporarily assigned type 34 for the PIM DR Load Balancing Capability (DRLB-Cap) Hello Option, and type 35 for the PIM DR Load Balancing List (DRLB-List) Hello Option in the PIM-Hello Options registry. IANA is requested to make these assignments permanent when this document is published as an RFC. Note that the option names have changed slightly since the temporary assignments were made. Also, the length of option 34 is always 4, the registry currently says it is variable.

This document requests IANA to create a registry called "Designated Router Load Balancing Hash Algorithms" in the "Protocol Independent Multicast (PIM)" branch of the registry tree. The registry lists Hash Algorithms for use by PIM Designated Router Load Balancing.

### 7.1. Initial registry

The initial content of the registry should be as follows.

Type	Name	Reference
0	Modulo	This document
1-255	Unassigned	

### 7.2. Assignment of new Hash Algorithms

Assignment of new Hash Algorithms is done according to the "IETF Review" model, see [RFC8126].

## 8. Security Considerations

Security of the new DR Load Balancing PIM Hello Options is only guaranteed by the security of PIM Hello messages, so the security

considerations for PIM Hello messages as described in PIM-SM [RFC7761] apply here.

If the DR is subverted it could omit or add certain GDRs or announce an unsupported algorithm. If another router is subverted, it could be made DR and cause similar issues. While these issues are specific to this specification, they are not that different from existing attacks such as subverting a DR and lowering the DR priority, causing a different router to become the DR.

If for any reason, the DR includes a GDR in the announced list which announces a different algorithm from what the DR announces, the GDR is required to ignore the announcement, and there will be no router acting as the DR for the flows that hash to that GDR.

If a GDR is subverted, it could potentially be made to stop forwarding all the traffic it is expected to forward. This is also similar today to if a DR is subverted.

An administrator may be able to achieve the desired load-balancing of known flows, but an attacker may send a single high rate flow which is served by a single GDR, or send multiple flows that are expected to be hashed to the same GDR.

## 9. Acknowledgement

The authors would like to thank Steve Simlo and Taki Millonis for helping with the original idea; Alia Atlas, Bill Atwood, Joe Clarke, Alissa Cooper, Jake Holland, Bharat Joshi, Anish Kachinthaya, Anvitha Kachinthaya, Benjamin Kaduk, Mirja Kuhlewind, Barry Leiba, Ben Niven-Jenkins, Alvaro Retana, Adam Roach, Michael Scharf, Eric Vyncke and Carl Wallace for reviews and comments; and Toerless Eckert and Rishabh Parekh for helpful conversation on the document.

## 10. References

### 10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC6395] Gulrajani, S. and S. Venaas, "An Interface Identifier (ID) Hello Option for PIM", RFC 6395, DOI 10.17487/RFC6395, October 2011, <<https://www.rfc-editor.org/info/rfc6395>>.

- [RFC7761] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I., Parekh, R., Zhang, Z., and L. Zheng, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", STD 83, RFC 7761, DOI 10.17487/RFC7761, March 2016, <<https://www.rfc-editor.org/info/rfc7761>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

## 10.2. Informative References

- [RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, DOI 10.17487/RFC3376, October 2002, <<https://www.rfc-editor.org/info/rfc3376>>.
- [RFC3810] Vida, R., Ed. and L. Costa, Ed., "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, DOI 10.17487/RFC3810, June 2004, <<https://www.rfc-editor.org/info/rfc3810>>.
- [RFC4541] Christensen, M., Kimball, K., and F. Solensky, "Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches", RFC 4541, DOI 10.17487/RFC4541, May 2006, <<https://www.rfc-editor.org/info/rfc4541>>.
- [RFC4607] Holbrook, H. and B. Cain, "Source-Specific Multicast for IP", RFC 4607, DOI 10.17487/RFC4607, August 2006, <<https://www.rfc-editor.org/info/rfc4607>>.

## Authors' Addresses

Yiqun Cai  
Alibaba Group

Email: [yiqun.cai@alibaba-inc.com](mailto:yiqun.cai@alibaba-inc.com)

Heidi Ou  
Alibaba Group

Email: heidi.ou@alibaba-inc.com

Sri Vallepalli  
Cisco Systems, Inc.  
3625 Cisco Way  
San Jose CA 95134  
USA

Email: svallepa@cisco.com

Mankamana Mishra  
Cisco Systems, Inc.  
821 Alder Drive,  
Milpitas CA 95035  
USA

Email: mankamis@cisco.com

Stig Venaas  
Cisco Systems, Inc.  
Tasman Drive  
San Jose CA 95134  
USA

Email: stig@cisco.com

Andy Green  
British Telecom  
Adastral Park  
Ipswich IP5 2RE  
United Kingdom

Email: andy.da.green@bt.com

PIM Working Group  
Internet Draft  
Intended status: Standards Track  
Expires: November 28, 2018

H. Zhao  
Ericsson  
X. Liu  
Jabil  
Y. Liu  
Huawei  
M. Sivakumar  
Cisco  
A. Peter  
Individual

May 29, 2018

A Yang Data Model for IGMP and MLD Snooping  
draft-ietf-pim-igmp-mld-snooping-yang-03.txt

## Abstract

This document defines a YANG data model that can be used to configure and manage Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping devices.

## Status of this Memo

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

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

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

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

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

This Internet-Draft will expire on November 28, 2018.

## Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

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

## Table of Contents

1. Introduction.....	3
1.1. Terminology.....	3
1.2. Tree Diagrams.....	3
2. Design of Data Model.....	4
2.1. Overview.....	4
2.2. IGMP Snooping Instances.....	4
2.3. MLD Snooping Instances.....	7
2.4. IGMP and MLD Snooping References.....	9
2.5. Augment /if:interfaces/if:interface.....	10
2.6. IGMP and MLD Snooping RPC.....	12
3. IGMP and MLD Snooping YANG Module.....	12
4. Security Considerations.....	42
5. IANA Considerations.....	43
6. Normative References.....	44
Appendix A. Data Tree Example.....	45
Authors' Addresses.....	49

## 1. Introduction

This document defines a YANG [RFC6020] data model for the management of Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping devices.

The YANG model in this document conforms to the Network Management Datastore Architecture defined in [I-D.ietf-netmod-revised-datastores]. The "Network Management Datastore Architecture" (NMDA) adds the ability to inspect the current operational values for configuration, allowing clients to use identical paths for retrieving the configured values and the operational values.

### 1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119].

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

### 1.2. Tree Diagrams

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

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



## 2. Design of Data Model

The model covers Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches [RFC4541].

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

### 2.1. Overview

The IGMP and MLD Snooping YANG module defined in this document has all the common building blocks for the IGMP and MLD Snooping protocol.

The YANG module includes IGMP and MLD Snooping instance definition, instance reference in the scenario of BRIDGE, L2VPN. The module also includes the RPC methods for clearing IGMP and MLD Snooping group tables.

This YANG model follows the Guidelines for YANG Module Authors (NMDA) [draft-dsdt-nmda-guidelines-01]. This NMDA ("Network Management Datastore Architecture") architecture provides an architectural framework for datastores as they are used by network management protocols such as NETCONF [RFC6241], RESTCONF [RFC8040] and the YANG [RFC7950] data modeling language.

### 2.2. IGMP Snooping Instances

The YANG module defines igmp-snooping-instance which could be referenced in the BRIDGE or L2VPN scenario to enable IGMP Snooping.

All the IGMP Snooping related attributes have been defined in the igmp-snooping-instance. The read-write attribute is configurable data. The read-only attribute shows state data. The key attribute of the igmp-snooping-instance is name.

The value of type in igmp-snooping-instance is bridge or l2vpn. When it is bridge, the igmp-snooping-instance will be referenced in the BRIDGE scenario. When it is l2vpn, the igmp-snooping-instance will be referenced in the L2VPN scenario.

The value of bridge-mrouter-interface, l2vpn-mrouter-interface-ac, l2vpn-mrouter-interface-pw are filled by routing system dynamically. They are different from static-bridge-mrouter-interface, static-l2vpn-

mrouter-interface-ac, and static-l2vpn-mrouter-interface-pw which are configured statically.

```

module: ietf-igmp-ml-d-snooping

  +--rw igmp-snooping-instances
  |   +--rw igmp-snooping-instance* [name]
  |       +--rw name                               string
  |       +--rw type?                             enumeration
  |       +--rw enable?                           boolean {admin-enable}?
  |       +--rw forwarding-mode?                  enumeration
  |       +--rw explicit-tracking?                boolean {explicit-tracki
ng}?
  |       +--rw exclude-lite?                     boolean {exclude-lite}?
  |       +--rw send-query?                       boolean
  |       +--rw immediate-leave?                  empty {immediate-leave}?
  |       +--rw last-member-query-interval?       uint16
  |       +--rw query-interval?                   uint16
  |       +--rw query-max-response-time?         uint16
  |       +--rw require-router-alert?            boolean {require-router-
alert}?
  |       +--rw robustness-variable?             uint8
  |       +--rw version?                         uint8
  |       +--rw static-bridge-mrouter-interface* if:interface-ref {static
-
mrouter-interface}?
  |       +--rw static-l2vpn-mrouter-interface-ac* if:interface-ref {static
-
mrouter-interface}?
  |       +--rw static-l2vpn-mrouter-interface-pw* l2vpn-instance-pw-ref
{static-mrouter-interface}?
  |       +--rw querier-source?                  inet:ipv4-address
  |       +--rw static-l2-multicast-group* [group source-addr] {static-l2-
multicast-group}?

```

```

|      |  +---rw group                                inet:ipv4-address
|      |  +---rw source-addr                          source-ipv4-addr-type
|      |  +---rw bridge-outgoing-interface*          if:interface-ref
|      |  +---rw l2vpn-outgoing-ac*                  l2vpn-instance-ac-ref
|      |  +---rw l2vpn-outgoing-pw*                  l2vpn-instance-pw-ref
|  +---ro entries-count?                             uint32
|  +---ro bridge-mrouter-interface*                  if:interface-ref
|  +---ro l2vpn-mrouter-interface-ac*                if:interface-ref
|  +---ro l2vpn-mrouter-interface-pw*                l2vpn-instance-pw-ref
|  +---ro group* [address]
|      +---ro address                                inet:ipv4-address
|      +---ro mac-address?                          yang:phys-address
|      +---ro expire?                               uint32
|      +---ro up-time?                              uint32
|      +---ro last-reporter?                        inet:ipv4-address
|      +---ro source* [address]
|          +---ro address                            inet:ipv4-address
|          +---ro bridge-outgoing-interface*        if:interface-ref
|          +---ro l2vpn-outgoing-ac*                l2vpn-instance-ac-ref
|          +---ro l2vpn-outgoing-pw*                l2vpn-instance-pw-ref
|          +---ro up-time?                          uint32
|          +---ro expire?                          uint32
|          +---ro host-count?                       uint32 {explicit-tracking}
?
|          +---ro last-reporter?                    inet:ipv4-address
|          +---ro host* [host-address] {explicit-tracking}?
|              +---ro host-address                  inet:ipv4-address

```

```
|          +--ro host-filter-mode?  enumeration
```

### 2.3. MLD Snooping Instances

The YANG module defines mld-snooping-instance which could be referenced in the BRIDGE or L2VPN scenario to enable MLD Snooping.

The mld-snooping-instance is the same as IGMP snooping except changing IPV4 addresses to IPV6 addresses.

```
module: ietf-igmp-mld-snooping
  +--rw mld-snooping-instances
  |   +--rw mld-snooping-instance* [name]
  |   |   +--rw name                                string
  |   |   +--rw type?                              enumeration
  |   |   +--rw enable?                            boolean {admin-enable
  }?
  |   |   +--rw forwarding-mode?                  enumeration
  |   |   +--rw explicit-tracking?                boolean {explicit-
tracking}?
  |   |   +--rw exclude-lite?                    boolean {exclude-lite
  }?
  |   |   +--rw send-query?                      boolean
  |   |   +--rw immediate-leave?                 empty {immediate-leav
e}?
  |   |   +--rw last-member-query-interval?      uint16
  |   |   +--rw query-interval?                 uint16
  |   |   +--rw query-max-response-time?        uint16
  |   |   +--rw require-router-alert?          boolean {require-rout
er-
alert}?
  |   |   +--rw robustness-variable?            uint8
  |   |   +--rw version?                        uint8
  |   |   +--rw static-bridge-mrouter-interface* if:interface-ref {sta
tic-
mrouter-interface}?
  |   |
```

```

    |      +--rw static-l2vpn-mrouter-interface-ac*   if:interface-ref {sta
tic-
    mrouter-interface}?

    |      +--rw static-l2vpn-mrouter-interface-pw*   l2vpn-instance-pw-ref
    {static-mrouter-interface}?

    |      +--rw querier-source?                      inet:ipv6-address

    |      +--rw static-l2-multicast-group* [group source-addr] {static-l2-
multicast-group}?

    |      |      +--rw group                          inet:ipv6-address
    |      |      +--rw source-addr                    source-ipv6-addr-type
    |      |      +--rw bridge-outgoing-interface*    if:interface-ref
    |      |      +--rw l2vpn-outgoing-ac*            l2vpn-instance-ac-ref
    |      |      +--rw l2vpn-outgoing-pw*            l2vpn-instance-pw-ref
    |      +--ro entries-count?                        uint32
    |      +--ro bridge-mrouter-interface*            if:interface-ref
    |      +--ro l2vpn-mrouter-interface-ac*          if:interface-ref
    |      +--ro l2vpn-mrouter-interface-pw*          l2vpn-instance-pw-ref
    |      +--ro group* [address]
    |      |      +--ro address                        inet:ipv6-address
    |      |      +--ro mac-address?                  yang:phys-address
    |      |      +--ro expire?                        uint32
    |      |      +--ro up-time?                       uint32
    |      |      +--ro last-reporter?                inet:ipv6-address
    |      |      +--ro source* [address]
    |      |      |      +--ro address                inet:ipv6-address
    |      |      |      +--ro bridge-outgoing-interface* if:interface-ref
    |      |      |      +--ro l2vpn-outgoing-ac*        l2vpn-instance-ac-ref
    |      |      |      +--ro l2vpn-outgoing-pw*        l2vpn-instance-pw-ref

```

ng}?		+--ro up-time?	uint32
		+--ro expire?	uint32
		+--ro host-count?	uint32 {explicit-tracki
		+--ro last-reporter?	inet:ipv6-address
		+--ro host* [host-address] {explicit-tracking}?	
		+--ro host-address	inet:ipv6-address
		+--ro host-filter-mode?	enumeration

#### 2.4. IGMP and MLD Snooping References

The `igmp-snooping-instance` could be referenced in the scenario of `BRIDGE` or `L2VPN` to configure the IGMP Snooping. The name of the instance is the key attribute.

When the `igmp-snooping-instance` is referenced under the bridge view, it means IGMP Snooping is enabled in the whole bridge. When the `igmp-snooping-instance` is referenced under the VLAN view, it means IGMP Snooping is enabled in the certain VLAN of the bridge.

The `mld-snooping-instance` could be referenced in concurrence with `igmp-snooping-instance` to configure the MLD Snooping.

```

+--rw bridges
|   +--rw bridge* [name]
|       +--rw name                name-type
|       +--rw igmp-snooping-instance?  igmp-snooping-instance-ref
|       +--rw mld-snooping-instance?  mld-snooping-instance-ref
|       +--rw component*[name]
|           +--rw name            string
|           +--rw bridge-vlan
|               +--rw vlan* [vid]
|                   +--rw vid                vlan-index-type
|                   +--rw igmp-snooping-instance?  igmp-snooping-instance-ref

```

```

|           +---rw mld-snooping-instance?      mld-snooping-instance-ref
+---rw l2vpn-instances
    +---rw l2vpn-instance* [name]
        +---rw name                               string
        +---rw igmp-snooping-instance?          igmp-snooping-instance-ref
        +---rw mld-snooping-instance?          mld-snooping-instance-ref

```

## 2.5. Augment /if:interfaces/if:interface

This model augment /if:interfaces/if:interface and then add the IGMP and MLD Snooping related attributes under it. The attributes include enable, version, etc.

The static-mrouter-interface and static-l2-multicast-group could be configured statically under the /if:interfaces/if:interface/ims:igmp-mld-snooping view. Meanwhile, you can configure them under the IGMP and MLD Snooping instance view.

The attributes under the statistics are read-only. They show the statistics of IGMP and MLD Snooping related packets.

```
augment /if:interfaces/if:interface:
```

```

+---rw igmp-mld-snooping
    +---rw enable?                               boolean {admin-enable}?
    +---rw version?                             uint8
    +---rw type?                                enumeration
    +---rw static-mrouter-interface
        |  +---rw (static-mrouter-interface)?
        |  |  +---:(bridge)
        |  |  |  +---rw bridge-name?            string
        |  |  |  +---rw vlan-id*                uint32
        |  |  +---:(l2vpn)
        |  |  +---rw l2vpn-instance-name?      string

```

```

+--rw static-l2-multicast-group
|
|  +--rw (static-l2-multicast-group)?
|  |
|  |  +--:(bridge)
|  |  |
|  |  |  +--rw bridge-name?          string
|  |  |
|  |  |  +--rw bridge-group-v4* [group source-addr]
|  |  |  |
|  |  |  |  +--rw group              inet:ipv4-address
|  |  |  |  +--rw source-addr       source-ipv4-addr-type
|  |  |  |  +--rw vlan-id*         uint32
|  |  |  +--rw bridge-group-v6* [group source-addr]
|  |  |  |
|  |  |  |  +--rw group              inet:ipv6-address
|  |  |  |  +--rw source-addr       source-ipv6-addr-type
|  |  |  |  +--rw vlan-id*         uint32
|  |  +--:(l2vpn)
|  |  |
|  |  |  +--rw l2vpn-group-v4* [group source-addr]
|  |  |  |
|  |  |  |  +--rw group              inet:ipv4-address
|  |  |  |  +--rw source-addr       source-ipv4-addr-type
|  |  |  |  +--rw l2vpn-instance-name? string
|  |  |  +--rw l2vpn-group-v6* [group source-addr]
|  |  |  |
|  |  |  |  +--rw group              inet:ipv6-address
|  |  |  |  +--rw source-addr       source-ipv6-addr-type
|  |  |  |  +--rw l2vpn-instance-name? string
|
+--ro statistics
|
|  +--ro received
|  |
|  |  +--ro query?                  yang:counter64
|  |
|  |  +--ro membership-report-v1?  yang:counter64
|  |
|  |  +--ro membership-report-v2?  yang:counter64

```



```

    | +---ro membership-report-v3?  yang:counter64
    | +---ro leave?                  yang:counter64
    | +---ro non-member-leave?      yang:counter64
    | +---ro pim?                   yang:counter64
+---ro sent
    +---ro query?                   yang:counter64
    +---ro membership-report-v1?    yang:counter64
    +---ro membership-report-v2?    yang:counter64
    +---ro membership-report-v3?    yang:counter64
    +---ro leave?                   yang:counter64
    +---ro non-member-leave?        yang:counter64
    +---ro pim?                     yang:counter64

```

## 2.6. IGMP and MLD Snooping RPC

IGMP and MLD Snooping RPC clears the specified IGMP and MLD Snooping group tables.

```

rpcs:
  +---x clear-igmp-snooping-groups {rpc-clear-groups}?
  |   +---w input
  |   |   +---w name?      string
  |   |   +---w group?     inet:ipv4-address
  |   |   +---w source?    inet:ipv4-address
  +---x clear-mld-snooping-groups {rpc-clear-groups}?
  |   +---w input
  |   |   +---w name?      string
  |   |   +---w group?     inet:ipv6-address
  |   |   +---w source?    inet:ipv6-address

```

## 3. IGMP and MLD Snooping YANG Module

```

<CODE BEGINS> file ietf-igmp-mld-snooping@2018-05-03.yang
module ietf-igmp-mld-snooping {
    namespace "urn:ietf:params:xml:ns:yang:ietf-igmp-mld-snooping";

```

```
// replace with IANA namespace when assigned
prefix ims;

import ietf-inet-types {
  prefix inet;
}

import ietf-yang-types {
  prefix "yang";
}

import ietf-interfaces {
  prefix "if";
}

import ietf-l2vpn {
  prefix "l2vpn";
}

import ietf-network-instance {
  prefix "ni";
}

organization
  "IETF PIM Working Group";

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

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

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

               Yisong Liu
               <mailto:liuyisong@huawei.com>

               Anish Peter
               <mailto:anish.ietf@gmail.com>

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

  ";
```

description

"The module defines a collection of YANG definitions common for all Internet Group Management Protocol (IGMP) and Multicast

Listener Discovery (MLD) Snooping devices.

Copyright (c) 2018 IETF Trust and the persons identified as authors of the code. All rights reserved.

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

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

```
revision 2018-05-03 {
  description
    "Initial revision.";
  reference
    "RFC XXXX: A YANG Data Model for IGMP and MLD Snooping";
}

/*
 * Features
 */

feature admin-enable {
  description
    "Support configuration to enable or disable IGMP and MLD
Snooping.";
}

feature immediate-leave {
  description
    "Support configuration of immediate-leave.";
}

feature join-group {
  description
    "Support configuration of join-group.";
}

feature require-router-alert {
  description
    "Support configuration of require-router-alert.";
}

feature static-l2-multicast-group {
  description
    "Support configuration of L2 multicast static-group.";
```

```
    }

    feature static-mrouter-interface {
        description
            "Support configuration of mrouter interface.";
    }

    feature per-instance-config {
        description
            "Support configuration of each VLAN or l2vpn instance or EVPN
instance.";
    }

    feature rpc-clear-groups {
        description
            "Support to clear statistics by RPC for IGMP and MLD
Snooping.";
    }

    feature explicit-tracking {
        description
            "Support configuration of per instance explicit-tracking
hosts.";
    }

    feature exclude-lite {
        description
            "Support configuration of per instance exclude-lite.";
    }

    /*
     * Typedefs
     */
    typedef name-type {
        type string {
            length "0..32";
        }
        description
            "A text string of up to 32 characters, of locally determined
            significance.";
    }
    typedef vlan-index-type {
        type uint32 {
            range "1..4094 | 4096..4294967295";
        }
        description
            "A value used to index per-VLAN tables. Values of 0 and 4095
            are not permitted. The range of valid VLAN indices. If the
            value is greater than 4095, then it represents a VLAN with
            scope local to the particular agent, i.e., one without a
            global VLAN-ID assigned to it. Such VLANs are outside the
```

```
    scope of IEEE 802.1Q, but it is convenient to be able to
    manage them in the same way using this YANG module.";
  reference
    "IEEE Std 802.1Q-2014: Virtual Bridged Local Area Networks.";
}

typedef igmp-snooping-instance-ref {
  type leafref {
    path "/igmp-snooping-instances/igmp-snooping-instance/name";
  }
  description
    "This type is used by data models that need to reference igmp
snooping instance.";
}

typedef mld-snooping-instance-ref {
  type leafref {
    path "/mld-snooping-instances/mld-snooping-instance/name";
  }
  description
    "This type is used by data models that need to reference mld
snooping instance.";
}

typedef l2vpn-instance-ac-ref {
  type leafref {
    path "/ni:network-instances/ni:network-instance/l2vpn:endpoint/l2vpn
:name";
  }
  description "l2vpn-instance-ac-ref";
}

typedef l2vpn-instance-pw-ref {
  type leafref {
    path "/ni:network-instances/ni:network-instance/l2vpn:endpoint/l2vpn:na
me";
  }
  description "l2vpn-instance-pw-ref";
}

typedef source-ipv4-addr-type {
  type union {
    type enumeration {
      enum '*' {
        description
          "Any source address.";
      }
    }
    type inet:ipv4-address;
  }
  description
```

```
    "Multicast source IPV4 address type.";
} // source-ipv4-addr-type

typedef source-ipv6-addr-type {
    type union {
        type enumeration {
            enum '*' {
                description
                "Any source address.";
            }
        }
        type inet:ipv6-address;
    }
    description
    "Multicast source IPV6 address type.";
} // source-ipv6-addr-type

/*
 * Identities
 */

/*
 * Groupings
 */

grouping general-state-attributes {
    description "General State attributes";

    container received {
        config false;
        description "Statistics of received IGMP and MLD Snooping
related packets.";
        uses general-statistics-sent-received;
    }
    container sent {
        config false;
        description "Statistics of sent IGMP and MLD Snooping related
packets.";
        uses general-statistics-sent-received;
    }
} // general-state-attributes

grouping instance-config-attributes-igmp-snooping {
```

```
description "IGMP snooping configuration for each VLAN or l2vpn
instance or EVPN instance.";
```

```
uses instance-config-attributes-igmp-ml-d-snooping;
```

```
leaf querier-source {
  type inet:ipv4-address;
  description "Use the IGMP snooping querier to support IGMP
snooping in a VLAN where PIM and IGMP are not configured.
The IPV4 address is used as the source address in
messages.";
}
```

```
list static-l2-multicast-group {
  if-feature static-l2-multicast-group;
  key "group source-addr";
  description
    "A static multicast route, (*,G) or (S,G).";
}
```

```
leaf group {
  type inet:ipv4-address;
  description
    "Multicast group IPV4 address";
}
```

```
leaf source-addr {
  type source-ipv4-addr-type;
  description
    "Multicast source IPV4 address.";
}
```

```
leaf-list bridge-outgoing-interface {
  when "../..//type = 'bridge'";
  type if:interface-ref;
  description "Outgoing interface in bridge forwarding";
}
```

```
leaf-list l2vpn-outgoing-ac {
  when "../..//type = 'l2vpn'";
  type l2vpn-instance-ac-ref;
  description "Outgoing AC in L2VPN forwarding";
}
```

```
leaf-list l2vpn-outgoing-pw {
  when "../..//type = 'l2vpn'";
  type l2vpn-instance-pw-ref;
  description "Outgoing PW in L2VPN forwarding";
}
```

```

    } // static-l2-multicast-group

    } // instance-config-attributes-igmp-snooping

    grouping instance-config-attributes-igmp-mlD-snooping {
        description
            "IGMP and MLD Snooping configuration of each VLAN.";

        leaf enable {
            if-feature admin-enable;
            type boolean;
            default false;
            description
                "Set the value to true to enable IGMP and MLD Snooping in
the VLAN instance.";
        }

        leaf forwarding-mode {
            type enumeration {
                enum "mac" {
                    description
                        "MAC-based lookup mode";
                }
                enum "ip" {
                    description
                        "IP-based lookup mode";
                }
            }
            default "ip";
            description "The default forwarding mode for IGMP and MLD
Snooping is ip.

                cisco command is as below
                Router(config-vlan-config)# multicast snooping lookup
{ ip | mac }  ";
        }

        leaf explicit-tracking {
            if-feature explicit-tracking;
            type boolean;
            default false;
            description "Tracks IGMP & MLD Snooping v3 membership reports
from individual hosts.
                It contributes to saving network resources and
shortening leave latency.";
        }

        leaf exclude-lite {
            if-feature exclude-lite;

```



```
    type boolean;
    default false;
    description
        "lightweight IGMPv3 and MLDv2 protocols, which simplify the
        standard versions of IGMPv3 and MLDv2.";
    reference "RFC5790";
}

leaf send-query {
    type boolean;
    default true;
    description "Enable quick response for topo changes.
        To support IGMP snooping in a VLAN where PIM and IGMP are
not configured.
        It cooperates with param querier-source. ";
}

/**
leaf mrouter-aging-time {
    type uint16 ;
    default 180;
    description "Aging time for mrouter interface";
}
**/

leaf immediate-leave {
    if-feature immediate-leave;
    type empty;
    description
        "When fast leave is enabled, the IGMP software assumes that
no more than one host is present on each VLAN port.";
}

leaf last-member-query-interval {
    type uint16 {
        range "1..65535";
    }
    units seconds;
    default 1;
    description
        "Last Member Query Interval, which may be tuned to modify
the
        leave latency of the network.";
    reference "RFC3376. Sec. 8.8.";
}

leaf query-interval {

    type uint16;
    units seconds;
    default 125;
```

```

        description
        "The Query Interval is the interval between General
Queries
        sent by the Querier.";
        reference "RFC3376. Sec. 4.1.7, 8.2, 8.14.2.";
    }

    leaf query-max-response-time {

        type uint16;
        units seconds;
        default 10;
        description
        "Query maximum response time specifies the maximum time
        allowed before sending a responding report.";
        reference "RFC3376. Sec. 4.1.1, 8.3, 8.14.3.";

    }

    leaf require-router-alert {
        if-feature require-router-alert;
        type boolean;
        default false;
        description
        "When the value is true, router alert should exist in the IP
head of IGMP or MLD packet.";
    }

    leaf robustness-variable {
        type uint8 {
            range "1..7";
        }
        default 2;
        description
        "Querier's Robustness Variable allows tuning for the
expected
        packet loss on a network.";
        reference "RFC3376. Sec. 4.1.6, 8.1, 8.14.1.";
    }

    leaf version {
        type uint8 {
            range "1..3";
        }
        description "IGMP and MLD Snooping version.";
    }

    leaf-list static-bridge-mrouter-interface {

        when "../type = 'bridge'";

```

```
    if-feature static-mrouter-interface;
    type if:interface-ref;
    description "static mrouter interface in bridge forwarding";
  }

  leaf-list static-l2vpn-mrouter-interface-ac {

    when "../type = 'l2vpn'";
    if-feature static-mrouter-interface;
    type if:interface-ref;
    description "static mrouter interface whose type is interface
in l2vpn forwarding";
  }

  leaf-list static-l2vpn-mrouter-interface-pw {

    when "../type = 'l2vpn'";
    if-feature static-mrouter-interface;
    type l2vpn-instance-pw-ref;
    description "static mrouter interface whose type is pw in l2vpn
forwarding";
  }

} // instance-config-attributes-igmp-ml-d-snooping

grouping instance-config-attributes-ml-d-snooping {
  description "MLD snooping configuration of each VLAN.";

  uses instance-config-attributes-igmp-ml-d-snooping;

  leaf querier-source {
    type inet:ipv6-address;
    description
      "Use the MLD snooping querier to support MLD snooping where PIM
and MLD are not configured.
      The IPV6 address is used as the source address in messages.";
  }

  list static-l2-multicast-group {
    if-feature static-l2-multicast-group;
    key "group source-addr";
    description
      "A static multicast route, (*,G) or (S,G).";

    leaf group {
      type inet:ipv6-address;
      description
```

```
    "Multicast group IPV6 address";
  }

  leaf source-addr {
    type source-ipv6-addr-type;
    description
      "Multicast source IPV6 address.";
  }

  leaf-list bridge-outgoing-interface {
    when "../..//type = 'bridge'";
    type if:interface-ref;
    description "Outgoing interface in bridge forwarding";
  }

  leaf-list l2vpn-outgoing-ac {
    when "../..//type = 'l2vpn'";
    type l2vpn-instance-ac-ref;
    description "Outgoing AC in L2VPN forwarding";
  }

  leaf-list l2vpn-outgoing-pw {
    when "../..//type = 'l2vpn'";
    type l2vpn-instance-pw-ref;
    description "Outgoing PW in L2VPN forwarding";
  }

} // static-l2-multicast-group

} // instance-config-attributes-mld-snooping

grouping instance-state-group-attributes-igmp-mld-snooping {
  description
    "Attributes for both IGMP and MLD snooping groups.";

  leaf mac-address {
    type yang:phys-address;
    description "Destination MAC address for L2 multicast
forwarding.";
  }

  leaf expire {
    type uint32;
    units seconds;
    description
      "The time left before multicast group timeout.";
  }
}
```

```
    leaf up-time {
        type uint32;
        units seconds;
        description
            "The time elapsed since the device created L2 multicast
record.";
    }

} // instance-state-group-attributes-igmp-mld-snooping

grouping instance-state-attributes-igmp-snooping {

    description
        "State attributes for IGMP snooping for each VLAN or l2vpn
instance or EVPN instance.";

    uses instance-state-attributes-igmp-mld-snooping;

    list group {

        key "address";

        config false;

        description "IGMP snooping information";

        leaf address {
            type inet:ipv4-address;
            description
                "Multicast group IPV4 address";
        }

        uses instance-state-group-attributes-igmp-mld-snooping;

        leaf last-reporter {
            type inet:ipv4-address;
            description
                "Address of the last host which has sent report to join
the multicast group.";
        }

        list source {
            key "address";
            description "Source IPV4 address for multicast stream";
            leaf address {
                type inet:ipv4-address;
                description "Source IPV4 address for multicast stream";
            }
        }
    }
}
```

```
    uses instance-state-source-attributes-igmp-ml-d-snooping;

    leaf last-reporter {
        type inet:ipv4-address;
        description
            "Address of the last host which has sent report to join
the multicast group.";
    }

    list host {
        if-feature explicit-tracking;
        key "host-address";
        description
            "List of multicast membership hosts
            of the specific multicast source-group.";

        leaf host-address {
            type inet:ipv4-address;
            description
                "Multicast membership host address.";
        }
        leaf host-filter-mode {
            type enumeration {
                enum "include" {
                    description
                        "In include mode";
                }
                enum "exclude" {
                    description
                        "In exclude mode.";
                }
            }
            description
                "Filter mode for a multicast membership
                host may be either include or exclude.";
        }
    } // list host

} // list source
} // list group

} // instance-state-attributes-igmp-snooping

grouping instance-state-attributes-igmp-ml-d-snooping {
    description
        "State attributes for both IGMP and MLD Snooping of each
        VLAN or l2vpn instance or EVPN instance.";

    leaf entries-count {
```

```
        type uint32;
        config false;
        description
            "The number of L2 multicast entries in IGMP and MLD
Snooping.";
    }

    leaf-list bridge-mrouter-interface {

        when "../type = 'bridge'";
        type if:interface-ref;
        config false;
        description " mrouter interface in bridge forwarding";

    }

    leaf-list l2vpn-mrouter-interface-ac {

        when "../type = 'l2vpn'";
        type if:interface-ref;
        config false;
        description " mrouter interface whose type is interface in
l2vpn forwarding";

    }

    leaf-list l2vpn-mrouter-interface-pw {

        when "../type = 'l2vpn'";
        type l2vpn-instance-pw-ref;
        config false;
        description " mrouter interface whose type is pw in l2vpn
forwarding";

    }

} // instance-config-attributes-igmp-mld-snooping

grouping instance-state-attributes-mld-snooping {
    description
        "State attributes for MLD snooping of each VLAN.";

    uses instance-state-attributes-igmp-mld-snooping;

    list group {

        key "address";

        config false;

    }

}
```

```
    description "MLD snooping statistics information";

    leaf address {
        type inet:ipv6-address;
        description
            "Multicast group IPV6 address";
    }

    uses instance-state-group-attributes-igmp-mld-snooping;

    leaf last-reporter {
        type inet:ipv6-address;
        description
            "Address of the last host which has sent report to join
the multicast group.";
    }

    list source {
        key "address";
        description "Source IPV6 address for multicast stream";

        leaf address {
            type inet:ipv6-address;
            description "Source IPV6 address for multicast stream";
        }

        uses instance-state-source-attributes-igmp-mld-snooping;

        leaf last-reporter {
            type inet:ipv6-address;
            description
                "Address of the last host which has sent report to join
the multicast group.";
        }

        list host {
            if-feature explicit-tracking;
            key "host-address";
            description
                "List of multicast membership hosts
                of the specific multicast source-group.";

            leaf host-address {
                type inet:ipv6-address;
                description
                    "Multicast membership host address.";
            }

            leaf host-filter-mode {
                type enumeration {
                    enum "include" {
```



```
        description
            "In include mode";
        }
        enum "exclude" {
            description
                "In exclude mode.";
        }
    }
    description
        "Filter mode for a multicast membership
        host may be either include or exclude.";
    }
} // list host

} // list source
} // list group

} // instance-state-attributes-mld-snooping

grouping instance-state-source-attributes-igmp-mld-snooping {
    description
        "State attributes for both IGMP and MLD Snooping of each VLAN
or l2vpn instance or EVPN instance.";

    leaf-list bridge-outgoing-interface {
        when "../..../type = 'bridge'";
        type if:interface-ref;
        description "Outgoing interface in bridge forwarding";
    }

    leaf-list l2vpn-outgoing-ac {
        when "../..../type = 'l2vpn'";
        type l2vpn-instance-ac-ref;
        description "Outgoing AC in L2VPN forwarding";
    }

    leaf-list l2vpn-outgoing-pw {
        when "../..../type = 'l2vpn'";
        type l2vpn-instance-pw-ref;
        description "Outgoing PW in L2VPN forwarding";
    }

    leaf up-time {
        type uint32;
        units seconds;
        description "The time elapsed since the device created L2
multicast record";
    }
}
```

```
leaf expire {
    type uint32;
    units seconds;
    description
        "The time left before multicast group timeout.";
}

leaf host-count {
    if-feature explicit-tracking;
    type uint32;
    description
        "The number of host addresses.";
}

} // instance-state-source-attributes-igmp-mld-snooping

grouping general-statistics-error {
    description
        "A grouping defining statistics attributes for errors.";

    leaf checksum {
        type yang:counter64;
        description
            "The number of checksum errors.";
    }
    leaf too-short {
        type yang:counter64;
        description
            "The number of messages that are too short.";
    }
} // general-statistics-error

grouping general-statistics-sent-received {
    description
        "A grouping defining statistics attributes.";

    leaf query {
        type yang:counter64;
        description
            "The number of query messages.";
    }
    leaf membership-report-v1 {
        type yang:counter64;
        description
            "The number of membership report v1 messages.";
    }
    leaf membership-report-v2 {
        type yang:counter64;
        description
            "The number of membership report v2 messages.";
    }
}
```

```
    }
    leaf membership-report-v3 {
      type yang:counter64;
      description
        "The number of membership report v3 messages.";
    }
    leaf leave {
      type yang:counter64;
      description
        "The number of leave messages.";
    }
    leaf non-member-leave {
      type yang:counter64;
      description
        "The number of non member leave messages.";
    }
    leaf pim {
      type yang:counter64;
      description
        "The number of pim hello messages.";
    }
  } // general-statistics-sent-received
}

grouping interface-endpoint-attributes-igmp-snooping {
  description "interface attributes for igmp snooping";
  list host {

    if-feature explicit-tracking;

    key "host-address";

    config false;

    description
      "List of multicast membership hosts
      of the specific multicast source-group.";

    leaf host-address {
      type inet:ipv4-address;
      description
        "Multicast membership host address.";
    }
    leaf host-filter-mode {
      type enumeration {
        enum "include" {
          description
            "In include mode";
        }
      }
    }
  }
}
```

```
    }
    enum "exclude" {
        description
            "In exclude mode.";
    }
}
description
    "Filter mode for a multicast membership
    host may be either include or exclude.";
}
} // list host
} // interface-endpoint-attributes-igmp-snooping

grouping interface-endpoint-attributes-mld-snooping {
    description "interface endpoint attributes mld snooping";

    list host {

        if-feature explicit-tracking;

        key "host-address";

        config false;

        description
            "List of multicast membership hosts
            of the specific multicast source-group.";

        leaf host-address {
            type inet:ipv6-address;
            description
                "Multicast membership host address.";
        }
        leaf host-filter-mode {
            type enumeration {
                enum "include" {
                    description
                        "In include mode";
                }
                enum "exclude" {
                    description
                        "In exclude mode.";
                }
            }
            description
                "Filter mode for a multicast membership
                host may be either include or exclude.";
        }
    }
} // list host
} // interface-endpoint-attributes-mld-snooping
```

```
/*
 * igmp-snooping-instance
 */
container igmp-snooping-instances {
  description
    "igmp-snooping-instance list";

  list igmp-snooping-instance {
    key "name";
    description
      "IGMP Snooping instance to configure the igmp-
snooping.";

    leaf name {
      type string;
      description
        "Name of the igmp-snooping-instance to configure the igmp
snooping.";
    }

    leaf type {
      type enumeration {
        enum "bridge" {
          description "bridge";
        }
        enum "l2vpn" {
          description "l2vpn";
        }
      }
      description "The type indicates bridge or l2vpn.";
    }
  }

  uses instance-config-attributes-igmp-snooping {
    if-feature per-instance-config;
  }

  uses instance-state-attributes-igmp-snooping;
} //igmp-snooping-instance
} //igmp-snooping-instances


/*
 * mld-snooping-instance
 */
container mld-snooping-instances {
```

```
    description
      "mld-snooping-instance list";

    list mld-snooping-instance {
      key "name";
      description
        "MLD Snooping instance to configure the mld-snooping.";

      leaf name {
        type string;
        description
          "Name of the mld-snooping-instance to configure the mld
snooping.";
      }

      leaf type {
        type enumeration {
          enum "bridge" {
            description "bridge";
          }
          enum "l2vpn" {
            description "l2vpn";
          }
        }
        description "The type indicates bridge or l2vpn.";
      }
    }

    uses instance-config-attributes-mld-snooping {
      if-feature per-instance-config;
    }

    uses instance-state-attributes-mld-snooping;
  } //mld-snooping-instance
} //mld-snooping-instances

container bridges {
  description
    "Apply igmp-mld-snooping instance in the bridge scenario";

  list bridge {
    key name;

    description
      "bridge list";
```

```

        leaf name {
            type name-type;
            description
                "bridge name";
        }
        leaf igmp-snooping-instance {
            type igmp-snooping-instance-ref;
            description "Configure igmp-snooping instance under the
bridge view";
        }
        leaf mld-snooping-instance {
            type mld-snooping-instance-ref;
            description "Configure mld-snooping instance under the
bridge view";
        }
        list component {
            key "name";
            description
                " ";

            leaf name {
                type string;
                description
                    "The name of the Component.";
            }
            container bridge-vlan {
                description "bridge vlan";
                list vlan {
                    key "vid";
                    description
                        " ";

                    leaf vid {
                        type vlan-index-type;
                        description
                            "The VLAN identifier to which this entry
applies.";
                    }
                }
            }
            leaf igmp-snooping-instance {
                type igmp-snooping-instance-ref;
                description "Configure igmp-snooping instance
under the vlan view";
            }
            leaf mld-snooping-instance {
                type mld-snooping-instance-ref;
                description "Configure mld-snooping instance
under the vlan view";
            }
        }

```

```
        } //vlan
      } //bridge-vlan
    } //component
  } //bridge
} //bridges

container l2vpn-instances {
  description "Apply igmp-mld-snooping instance in the l2vpn
scenario";

  list l2vpn-instance {
    key "name";
    description "An l2vpn service instance";

    leaf name {
      type string;
      description "Name of l2vpn service instance";
    }

    leaf igmp-snooping-instance {
      type igmp-snooping-instance-ref;
      description "Configure igmp-snooping instance under the
l2vpn-instance view";
    }
    leaf mld-snooping-instance {
      type mld-snooping-instance-ref;
      description "Configure mld-snooping instance under the
l2vpn-instance view";
    }
  }
}

/* augments */

augment "/if:interfaces/if:interface" {
  description "Augment interface for referencing attributes which
only fit for interface view.";

  container igmp-mld-snooping {
    description
      "igmp-mld-snooping related attributes under interface view";

    leaf enable {
      if-feature admin-enable;
      type boolean;
      default false;
      description
```



"Set the value to true to enable IGMP and MLD Snooping in the VLAN instance.";

```
    }

    leaf version {
      type uint8 {
        range "1..3";
      }
      description "IGMP and MLD Snooping version.";
    }

    leaf type {
      type enumeration {
        enum "bridge" {
          description "bridge";
        }
        enum "l2vpn" {
          description "l2vpn";
        }
      }
      description "The type indicates bridge or l2vpn.";
    }

    container static-mrouter-interface {
      description
        "Container for choice static-mrouter-interface";

      choice static-mrouter-interface {
        description
          "Configure static multicast router interface under the
interface view";

        case bridge {
          when "../type = 'bridge'" {
            description
              "Applies to bridge scenario.";
          }
          description
            "Applies to bridge scenario.";

          leaf bridge-name {
            type string;
            description
              "The name for a bridge. Each interface
belongs to only one bridge.";
          }
        }
        leaf-list vlan-id {
          type uint32;
          description
```

"The vlan ids for bridge. If you don't specify vlan id here, the interface serves as the mrouter interface for all the vlans in this bridge.";

```

    }
  case l2vpn {
    when "../type = 'l2vpn'" {
      description
        "Applies to l2vpn scenario.";
    }
    description
      "Applies to l2vpn scenario.";

    leaf l2vpn-instance-name {
      type string;
      description
        "The l2vpn instance name applied in the
interface";
    }
  }

} // choice static-mrouter-interface
} // container static-mrouter-interface

container static-l2-multicast-group {
  description
    "Container for static-l2-multicast-group";

  choice static-l2-multicast-group {
    description
      "Configure static l2 multicast group under the
interface view";

    case bridge {
      when "../type = 'bridge'" {
        description
          "Applies to bridge scenario.";
      }
      description
        "Applies to bridge scenario.";

      leaf bridge-name {
        type string;
        description
          "bridge name.";
      }
    }
  }
}

```

```
list bridge-group-v4 {

    key "group source-addr";
    description
        "A static multicast route, (*,G) or (S,G).";

    leaf group {
        type inet:ipv4-address;
        description
            "Multicast group IPV4 address";
    }

    leaf source-addr {
        type source-ipv4-addr-type;
        description
            "Multicast source IPV4 address.";
    }

    leaf-list vlan-id {
        type uint32;
        description
            "vlan id.";
    }
}

list bridge-group-v6 {
    key "group source-addr";
    description
        "A static multicast route, (*,G) or (S,G).";

    leaf group {
        type inet:ipv6-address;
        description
            "Multicast group IPV6 address";
    }

    leaf source-addr {
        type source-ipv6-addr-type;
        description
            "Multicast source IPV6 address.";
    }

    leaf-list vlan-id {
        type uint32;
        description
            "vlan id.";
    }
}
```

```

    }
    case l2vpn {
        when "../type = 'l2vpn'" {
            description
                "Applies to l2vpn scenario.";
        }
        description
            "Applies to l2vpn scenario.";

        list l2vpn-group-v4 {
            key "group source-addr";
            description "A static multicast route, (*,G) or
(S,G).";

            leaf group {
                type inet:ipv4-address;
                description
                    "Multicast group IPV4 address";
            }

            leaf source-addr {
                type source-ipv4-addr-type;
                description
                    "Multicast source IPV4 address.";
            }

            leaf l2vpn-instance-name {
                type string;
                description
                    "The l2vpn instance name applied in the
interface";
            }
        }
        list l2vpn-group-v6 {
            key "group source-addr";

            description
                "A static multicast route, (*,G) or (S,G).";

            leaf group {
                type inet:ipv6-address;
                description
                    "Multicast group IPV6 address";
            }

            leaf source-addr {
                type source-ipv6-addr-type;

```

```
        description
            "Multicast source IPV6 address.";
    }

    leaf l2vpn-instance-name {
        type string;
        description
            "The l2vpn instance name applied in the
interface";
    }
}

} //choice static-l2-multicast-group
} // container static-l2-multicast-group

container statistics {
    config false;
    description
        "A collection of interface-related statistics objects.";

    uses general-state-attributes;
}

}

}

/*  RPCs  */

rpc clear-igmp-snooping-groups {
    if-feature rpc-clear-groups;
    description
        "Clears the specified IGMP Snooping cache tables.";

    input {

        leaf name {
            type string;
            description
                "Name of the igmp-snooping-instance";
        }

        leaf group {
            type inet:ipv4-address;
```

```
        description
            "Multicast group IPv4 address.
            If it is not specified, all IGMP snooping group tables
are
            cleared.";
        }

        leaf source {
            type inet:ipv4-address;
            description
                "Multicast source IPv4 address.
                If it is not specified, all IGMP snooping source-group
tables are
                cleared.";
        }
    }
} // rpc clear-igmp-snooping-groups

rpc clear-mld-snooping-groups {
    if-feature rpc-clear-groups;
    description
        "Clears the specified MLD Snooping cache tables.";

    input {
        leaf name {
            type string;
            description
                "Name of the mld-snooping-instance";
        }

        leaf group {
            type inet:ipv6-address;
            description
                "Multicast group IPv6 address.
                If it is not specified, all MLD snooping group tables are
                cleared.";
        }

        leaf source {
            type inet:ipv6-address;
            description
                "Multicast source IPv6 address.
                If it is not specified, all MLD snooping source-group
tables are
                cleared.";
        }
    }
} // rpc clear-mld-snooping-groups
}
<CODE ENDS>
```

#### 4. Security Considerations

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC5246].

The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

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

/ims:igmp-snooping-instances/ims:igmp-snooping-instance

/ims:mld-snooping-instances/ims:mld-snooping-instance

/if:interfaces/if:interface/ims:igmp-mld-snooping

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

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

/ims:igmp-snooping-instances/ims:igmp-snooping-instance

/ims:mld-snooping-instances/ims:mld-snooping-instance

/if:interfaces/if:interface/ims:igmp-mld-snooping

Unauthorized access to any data node of these subtrees can disclose the operational state information of IGMP & MLD Snooping on this device.

Some of the RPC operations in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control access to these operations. The IGMP & MLD Snooping Yang module support the "clear-igmp-snooping-groups" and "clear-mld-snooping-groups" RPCs. If it meets unauthorized RPC operation invocation, the IGMP and MLD Snooping group tables will be cleared unexpectedly.

## 5. IANA Considerations

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

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

-----  
URI: urn:ietf:params:xml:ns:yang:ietf-igmp-mld-snooping

Registrant Contact: The IESG.

XML: N/A, the requested URI is an XML namespace.  
-----

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

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



## 6. Normative References

- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.
- [RFC6021] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6021, October 2010.
- [RFC4541] M. Christensen, K. Kimball, F. Solensky, "Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches", RFC 4541, May 2006.
- [RFC2236] Fenner, W., "Internet Group Management Protocol, Version 2", RFC 2236, November 1997.
- [RFC2710] Deering, S., Fenner, W., and B. Haberman, "Multicast Listener Discovery (MLD) for IPv6", RFC 2710, October 1999.
- [RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, October 2002.
- [RFC3810] Vida, R. and L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, June 2004.
- [RFC4604] Holbrook, H., Cain, B., and B. Haberman, "Using InternetGroup Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Protocol Version 2 (MLDv2) for Source-Specific Multicast", RFC 4604, August 2006.
- [RFC4607] Holbrook, H. and B. Cain, "Source-Specific Multicast for IP", RFC 4607, August 2006.
- [draft-ietf-pim-igmp-ml-d-yang-01] X. Liu, F. Guo, M. Sivakumar, P. McAllister, A. Peter, "A YANG data model for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD)", draft-ietf-pim-igmp-ml-d-yang-01, October 28, 2016.
- [draft-ietf-pim-igmp-ml-d-yang-03] X. Liu, F. Guo, M. Sivakumar, P. McAllister, A. Peter, "A YANG data model for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD)", draft-ietf-pim-igmp-ml-d-yang-03, March 13, 2017.
- [draft-dsdt-nmda-guidelines-01] M. Bjorklund, J. Schoenwaelder, P. Shafer, K. Watsen, R. Wilton, "Guidelines for YANG Module Authors (NMDA)", draft-dsdt-nmda-guidelines-01, May 2017

[draft-bjorklund-netmod-rfc7223bis-00] M. Bjorklund, "A YANG Data Model for Interface Management", draft-bjorklund-netmod-rfc7223bis-00, August 21, 2017

[draft-bjorklund-netmod-rfc7277bis-00] M. Bjorklund, "A YANG Data Model for IP Management", draft-bjorklund-netmod-rfc7277bis-00, August 21, 2017

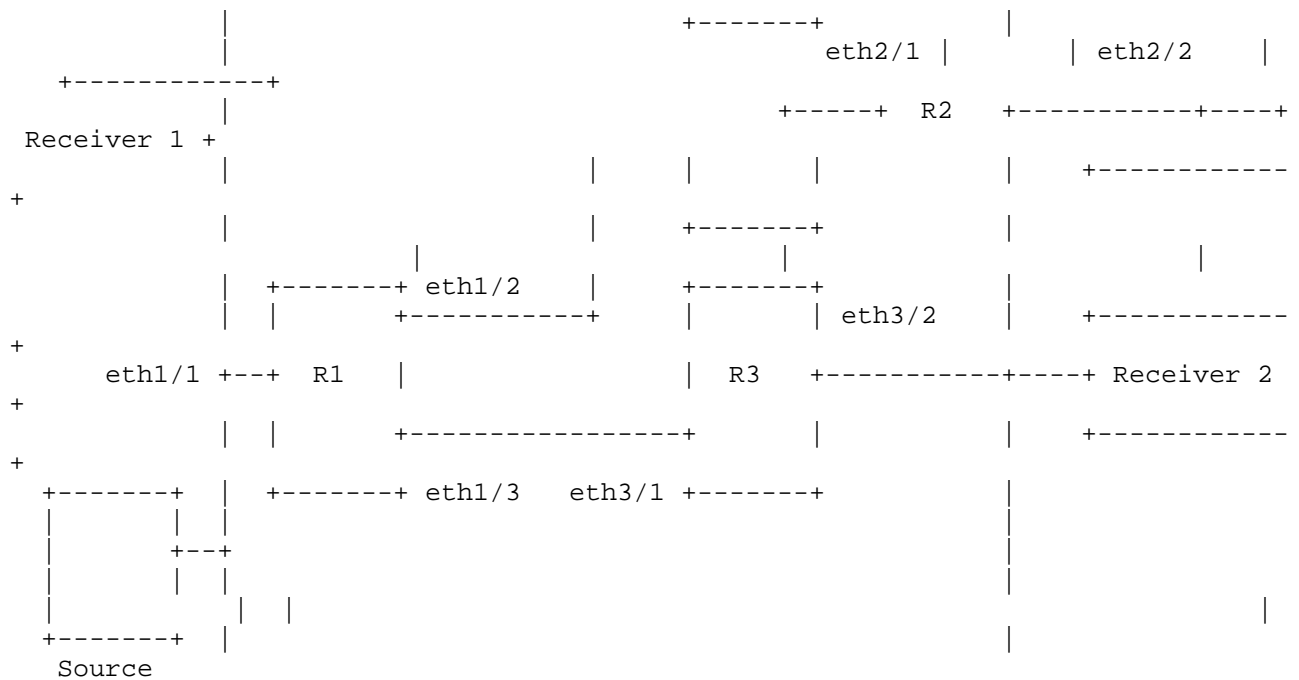
[draft-ietf-netmod-revised-datastores-03] M. Bjorklund, J. Schoenwaelder, P. Shafer, K. Watsen, R. Wilton, "Network Management Datastore Architecture", draft-ietf-netmod-revised-datastores-03, July 3, 2017

[draft-ietf-bess-evpn-yang-02] P. Brissette, A. Sajassi, H. Shah, Z. Li, H. Chen, K. Tiruveedhula, I. Hussain, J. Rabadan, "Yang Data Model for EVPN", draft-ietf-bess-evpn-yang-02, March 13, 2017

[draft-ietf-bess-l2vpn-yang-06] H. Shah, P. Brissette, I. Chen, I. Hussain, B. Wen, K. Tiruveedhula, "YANG Data Model for MPLS-based L2VPN", draft-ietf-bess-l2vpn-yang-06.txt, June 30, 2017

#### Appendix A. Data Tree Example

This section contains an example of an instance data tree in the JSON encoding [RFC7951], containing both configuration and state data.



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

```
{
  "ietf-igmp-ml-d-snooping:igmp-snooping-instances": {
    "igmp-snooping-instance": [
      {
        "name": "ins101",
        "type": "bridge",
        "enable": true
      }
    ]
  },
  "ietf-igmp-ml-d-snooping:mld-snooping-instances": {
    "mld-snooping-instance": [
      {
        "name": "ins102",
        "type": "bridge",
        "enable": true
      }
    ]
  },
  "ietf-igmp-ml-d-snooping:bridges": {
    "bridge": [
      {
        "name": "isp",
        "component": [
          {
            "name": "comp1",
            "bridge-vlan": {
              "vlan": [
                {
                  "vid": 101,
                  "igmp-snooping-instance": "ins101"
                },
                {
                  "vid": 102,
                  "mld-snooping-instance": "ins102"
                }
              ]
            }
          }
        ]
      }
    ]
  }
}
```

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

```
{
  "ietf-interfaces:interfaces": {
    "interface": [
      {
        "name": "1/1",
        "type": "iana-if-type:ethernetCsmacd",
        "admin-status": "up",
        "if-index": 214748,
        "oper-status": "up",
        "statistics": {
          "discontinuity-time": "2018-05-23T12:34:56-05:00"
        }
      },
      {
        "name": "1/2",
        "type": "iana-if-type:ethernetCsmacd",
        "admin-status": "up",
        "if-index": 214749,
        "oper-status": "up",
        "statistics": {
          "discontinuity-time": "2018-05-23T12:35:06-05:02"
        }
      }
    ]
  },
  "ietf-igmp-mld-snooping:igmp-snooping-instances": {
    "igmp-snooping-instance": [
      {
        "name": "ins101",
        "type": "bridge",
        "enable": true,
        "forwarding-mode": "ip",
        "explicit-tracking": false,
        "exclude-lite": false,
        "send-query": true,
        "immediate-leave": [null],
        "last-member-query-interval": 1,
        "query-interval": 125,
        "query-max-response-time": 10,
        "require-router-alert": false,
        "robustness-variable": 2,
        "entries-count": 1,
        "bridge-mrouter-interface": ["1/1"],
        "group": [
          {
            "address": "223.0.0.1",
            "mac-address": "01:00:5e:00:00:01",
            "expire": 120,
            "up-time": 180,

```

```

        "last-reporter": "100.0.0.1",
        "source": [
          {
            "address": "192.168.0.1",
            "bridge-outgoing-interface": ["1/2"],
            "up-time": 180,
            "expire": 120,
            "last-reporter": "100.0.0.1"
          }
        ]
      }
    ]
  },
  "ietf-igmp-ml-d-snooping:mld-snooping-instances": {
    "mld-snooping-instance": [
      {
        "name": "ins102",
        "type": "bridge",
        "enable": true,
        "forwarding-mode": "ip",
        "explicit-tracking": false,
        "exclude-lite": false,
        "send-query": true,
        "immediate-leave": [null],
        "last-member-query-interval": 1,
        "query-interval": 125,
        "query-max-response-time": 10,
        "require-router-alert": false,
        "robustness-variable": 2,
        "entries-count": 1,
        "bridge-mrouter-interface": ["1/1"],
        "group": [
          {
            "address": "FF0E::1",
            "mac-address": "01:00:5e:00:00:01",
            "expire": 120,
            "up-time": 180,
            "last-reporter": "2001::1",
            "source": [
              {
                "address": "3001::1",
                "bridge-outgoing-interface": ["1/2"],
                "up-time": 180,
                "expire": 120,
                "last-reporter": "2001::1"
              }
            ]
          }
        ]
      }
    ]
  }
}

```

## Authors' Addresses

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

Xufeng Liu  
Jabil  
8281 Greensboro Drive, Suite 200  
McLean VA 22102  
USA  
  
EMail: Xufeng.liu.ietf@gmail.com

Yisong Liu  
Huawei Technologies  
Huawei Bld., No.156 Beiqing Rd.  
Beijing 100095  
China  
  
Email: liuyisong@huawei.com

Anish Peter  
Individual  
  
EMail: anish.ietf@gmail.com

Mahesh Sivakumar  
Cisco Systems  
510 McCarthy Boulevard  
Milpitas, California  
USA

EMail: sivakumar.mahesh@gmail.com







PIM Working Group  
Internet Draft  
Intended status: Standards Track  
Expires: April 07, 2022

H. Zhao  
Ericsson  
X. Liu  
Volta Networks  
Y. Liu  
China Mobile  
M. Sivakumar  
Juniper  
A. Peter  
Individual

October 08, 2021

A Yang Data Model for IGMP and MLD Snooping  
draft-ietf-pim-igmp-mld-snooping-yang-20.txt

## Abstract

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

## Status of this Memo

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

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

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

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

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

This Internet-Draft will expire on April 07, 2022.

## Copyright Notice

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

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

## Table of Contents

1. Introduction.....	3
1.1. Terminology.....	3
1.2. Tree Diagrams.....	3
1.3. Prefixes in Data Node Names.....	4
2. Design of Data Model.....	4
2.1. Overview.....	5
2.2. Optional Capabilities.....	5
2.3. Position of Address Family in Hierarchy.....	6
3. Module Structure.....	6
3.1. IGMP Snooping Instances.....	6
3.2. MLD Snooping Instances.....	8
3.3. Using IGMP and MLD Snooping Instances.....	10
3.4. IGMP and MLD Snooping Actions.....	11
4. IGMP and MLD Snooping YANG Module.....	11
5. Security Considerations.....	31
6. IANA Considerations.....	33
6.1. XML Registry.....	33
6.2. YANG Module Names Registry.....	33
7. References.....	34
7.1. Normative References.....	34
7.2. Informative References.....	35
Appendix A. Data Tree Example.....	36
Authors' Addresses.....	39

## 1. Introduction

This document defines a YANG [RFC7950] data model for the management of Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping [RFC4541] devices.

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

### 1.1. Terminology

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

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

The following terminologies are used in this document:

- \* mrouter: multicast router, which is a router that has multicast routing enabled [RFC4286].
- \* mrouter interfaces: snooping switch ports where multicast routers are attached [RFC4541].

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

IGMP: Internet Group Management Protocol [RFC3376].

MLD: Multicast Listener Discovery [RFC3810].

### 1.2. Tree Diagrams

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

### 1.3. Prefixes in Data Node Names

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

Prefix	YANG module	Reference
inet	ietf-inet-types	[RFC6991]
yang	ietf-yang-types	[RFC6991]
if	ietf-interfaces	[RFC8343]
rt	ietf-routing	[RFC8349]
rt-types	ietf-routing-types	[RFC8294]
dot1q	ieee802-dot1q-bridge	[dot1Qcp]

Table 1: Prefixes and Corresponding YANG Modules

## 2. Design of Data Model

An IGMP/MLD snooping switch [RFC4541] analyzes IGMP/MLD packets and sets up forwarding tables for multicast traffic. If a switch does not run IGMP/MLD snooping, multicast traffic will be flooded in the broadcast domain. If a switch runs IGMP/MLD snooping, multicast traffic will be forwarded based on the forwarding tables to avoid wasting bandwidth. The IGMP/MLD snooping switch does not need to run any of the IGMP/MLD protocols. Because the IGMP/MLD snooping is independent of the IGMP/MLD protocols, the data model defined in this document does not augment, or even require, the IGMP/MLD data model defined in [RFC8652]. The model covers considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches [RFC4541].

IGMP and MLD snooping switches do not adhere to the conceptual model that provides the strict separation of functionality between different

communications layers in the ISO model, and instead utilize information in the upper level protocol headers as factors to be considered in processing at the lower levels [RFC4541].

IGMP Snooping switches utilize IGMP, and could support IGMPv1 [RFC1112], IGMPv2 [RFC2236], and IGMPv3 [RFC3376]. MLD Snooping switches utilize MLD, and could support MLDv1 [RFC2710] and MLDv2 [RFC3810]. The goal of this document is to define a data model that provides a common user interface to IGMP and MLD Snooping.

## 2.1. Overview

The IGMP and MLD Snooping YANG module defined in this document has all the common building blocks for the IGMP and MLD Snooping switches.

The YANG module includes IGMP and MLD Snooping instance definition, using instance in the L2 service type of BRIDGE [dot1Qcp]. It also includes actions for clearing IGMP and MLD Snooping group tables.

The YANG module doesn't cover L2VPN, which will be specified in a separated document.

## 2.2. Optional Capabilities

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

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

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

On the other hand, operational state parameters are not so widely designated as features, as there are many cases where the defaulting of an operational state parameter would not cause any harm to the system, and it is much more likely that an implementation without native support for a piece of operational state would be able to derive a suitable value for a state variable that is not natively supported.

### 2.3. Position of Address Family in Hierarchy

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

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

- \* The model can support IGMP Snooping (IPv4), MLD Snooping (IPv6), or both optionally and independently. Such flexibility cannot be achieved cleanly with a combined branch.
- \* The structure is consistent with other YANG data models such as [RFC8652], which uses separate branches for IPv4 and IPv6.
- \* Having separate branches for IGMP Snooping and MLD Snooping allows minor differences in their behavior to be modelled more simply and cleanly. The two branches can better support different features and node types.

### 3. Module Structure

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

```

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

```

The "igmp-snooping-instance" container instantiates an IGMP Snooping Instance. The "mld-snooping-instance" container instantiates an MLD Snooping Instance.

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

#### 3.1. IGMP Snooping Instances

The YANG module `ietf-igmp-mld-snooping` augments `/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol` to add the `igmp-snooping-instance` container.

All the IGMP Snooping related attributes have been defined in the `igmp-snooping-instance`. The read-write attributes represent configurable data. The read-only attributes represent state data.

One `igmp-snooping-instance` could be used in one `BRIDGE [dot1Qcp]` instance, and it corresponds to one `BRIDGE` instance.

Currently the value of `l2-service-type` in `igmp-snooping-instance` could only be set `bridge`. After it is set, `igmp-snooping-instance` could be used in the `BRIDGE` service.

The values of `bridge-mrouter-interface` is filled by the snooping device dynamically. It is different from `static-bridge-mrouter-interface` which is configured.

The attributes under the interfaces show the statistics of IGMP Snooping related packets.

```
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol:
    +--rw igmp-snooping-instance {igmp-snooping}?
      +--rw l2-service-type?                  l2-service-type
      +--rw enable?                          boolean
      +--rw forwarding-table-type?            enumeration
      +--rw explicit-tracking?                boolean
      |   {explicit-tracking}?
      +--rw lite-exclude-filter?              empty
      |   {lite-exclude-filter}?
      +--rw send-query?                      boolean
      +--rw fast-leave?                      empty {fast-leave}?
      +--rw last-member-query-interval?      uint16
      +--rw query-interval?                  uint16
      +--rw query-max-response-time?         uint16
      +--rw require-router-alert?            boolean
      |   {require-router-alert}?
      +--rw robustness-variable?              uint8
      +--rw static-bridge-mrouter-interface* if:interface-ref
      |   {static-mrouter-interface}?
      +--rw igmp-version?                    uint8
      +--rw querier-source?                  inet:ipv4-address
      +--rw static-l2-multicast-group* [group source-addr]
      |   {static-l2-multicast-group}?
      |   +--rw group
      |   |   rt-types:ipv4-multicast-group-address
      |   +--rw source-addr
      |   |   rt-types:ipv4-multicast-source-address
      |   +--rw bridge-outgoing-interface*   if:interface-ref
      +--ro entries-count?                   yang:gauge32
      +--ro bridge-mrouter-interface*        if:interface-ref
      +--ro group* [address]
      |   +--ro address
```

```

|         rt-types:ipv4-multicast-group-address
+--ro mac-address?      yang:phys-address
+--ro expire?           rt-types:timer-value-seconds16
+--ro up-time           uint32
+--ro last-reporter?    inet:ipv4-address
+--ro source* [address]
|   +--ro address
|   |         rt-types:ipv4-multicast-source-address
+--ro bridge-outgoing-interface*  if:interface-ref
+--ro up-time           uint32
+--ro expire?
|   rt-types:timer-value-seconds16
+--ro host-count?       yang:gauge32
|   {explicit-tracking}?
+--ro last-reporter?    inet:ipv4-address
+--ro host* [address] {explicit-tracking}?
|   +--ro address       inet:ipv4-address
|   +--ro filter-mode   filter-mode-type
+--ro interfaces
+--ro interface* [name]
|   +--ro name           if:interface-ref
+--ro statistics
|   +--ro discontinuity-time?  yang:date-and-time
+--ro received
|   +--ro query-count?        yang:counter64
|   +--ro membership-report-v1-count?  yang:counter64
|   +--ro membership-report-v2-count?  yang:counter64
|   +--ro membership-report-v3-count?  yang:counter64
|   +--ro leave-count?        yang:counter64
|   +--ro pim-hello-count?     yang:counter64
+--ro sent
|   +--ro query-count?        yang:counter64
|   +--ro membership-report-v1-count?  yang:counter64
|   +--ro membership-report-v2-count?  yang:counter64
|   +--ro membership-report-v3-count?  yang:counter64
|   +--ro leave-count?        yang:counter64
|   +--ro pim-hello-count?     yang:counter64

```

### 3.2. MLD Snooping Instances

The YANG module `ietf-igmp-ml-d-snooping` augments `/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol` to add the `mld-snooping-instance` container. The `mld-snooping-instance` could be used in the `BRIDGE [dot1Qcp]` service to enable MLD Snooping.

All the MLD Snooping related attributes have been defined in the `mld-snooping-instance`. The read-write attributes represent configurable data. The read-only attributes represent state data.



The mld-snooping-instance has similar structure as IGMP snooping. Some of leaves are protocol related. The mld-snooping-instance uses IPv6 addresses and mld-version, while igmp-snooping-instance uses IPv4 addresses and igmp-version. Statistic counters in each of the above snooping instances are also tailored to the specific protocol type. One mld-snooping-instance could be used in one BRIDGE instance, and it corresponds to one BRIDGE instance.

Currently the value of l2-service-type in mld-snooping-instance could only be set bridge. After it is set, mld-snooping-instance could be used in the BRIDGE service.

The value of bridge-mrouter-interface is filled by the snooping device dynamically. It is different from static-bridge-mrouter-interface which is configured.

The attributes under the interfaces show the statistics of MLD Snooping related packets.

```
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol:
    +--rw mld-snooping-instance {mld-snooping}?
      +--rw l2-service-type?                l2-service-type
      +--rw enable?                        boolean
      +--rw forwarding-table-type?          enumeration
      +--rw explicit-tracking?              boolean
      |   {explicit-tracking}?
      +--rw lite-exclude-filter?            empty
      |   {lite-exclude-filter}?
      +--rw send-query?                    boolean
      +--rw fast-leave?                    empty {fast-leave}?
      +--rw last-member-query-interval?     uint16
      +--rw query-interval?                 uint16
      +--rw query-max-response-time?        uint16
      +--rw require-router-alert?           boolean
      |   {require-router-alert}?
      +--rw robustness-variable?            uint8
      +--rw static-bridge-mrouter-interface* if:interface-ref
      |   {static-mrouter-interface}?
      +--rw mld-version?                    uint8
      +--rw querier-source?                 inet:ipv6-address
      +--rw static-l2-multicast-group* [group source-addr]
      |   {static-l2-multicast-group}?
      |   +--rw group
      |   |   rt-types:ipv6-multicast-group-address
      |   +--rw source-addr
      |   |   rt-types:ipv6-multicast-source-address
      |   +--rw bridge-outgoing-interface* if:interface-ref
      +--ro entries-count?                  yang:gauge32
      +--ro bridge-mrouter-interface*       if:interface-ref
      +--ro group* [address]
```

```

+--ro address
|   rt-types:ipv6-multicast-group-address
+--ro mac-address?      yang:phys-address
+--ro expire?           rt-types:timer-value-seconds16
+--ro up-time            uint32
+--ro last-reporter?    inet:ipv6-address
+--ro source* [address]
|   +--ro address
|   |   rt-types:ipv6-multicast-source-address
|   +--ro bridge-outgoing-interface*  if:interface-ref
|   +--ro up-time            uint32
|   +--ro expire?
|   |   rt-types:timer-value-seconds16
|   +--ro host-count?        yang:gauge32
|   |   {explicit-tracking}?
|   +--ro last-reporter?     inet:ipv6-address
|   +--ro host* [address] {explicit-tracking}?
|   |   +--ro address        inet:ipv6-address
|   |   +--ro filter-mode    filter-mode-type
+--ro interfaces
+--ro interface* [name]
+--ro name            if:interface-ref
+--ro statistics
+--ro discontinuity-time? yang:date-and-time
+--ro received
|   +--ro query-count?      yang:counter64
|   +--ro report-v1-count?  yang:counter64
|   +--ro report-v2-count?  yang:counter64
|   +--ro done-count?       yang:counter64
|   +--ro pim-hello-count?  yang:counter64
+--ro sent
+--ro query-count?      yang:counter64
+--ro report-v1-count?  yang:counter64
+--ro report-v2-count?  yang:counter64
+--ro done-count?       yang:counter64
+--ro pim-hello-count?  yang:counter64

```

### 3.3. Using IGMP and MLD Snooping Instances

The `igmp-snooping-instance` could be used in the service of `BRIDGE` [`dot1Qcp`] to configure the IGMP Snooping.

For the `BRIDGE` service this model augments `/dot1q:bridges/dot1q:bridge` to use `igmp-snooping-instance`. It means IGMP Snooping is enabled in the whole bridge.

It also augments `/dot1q:bridges/dot1q:bridge/dot1q:component/dot1q:bridge-vlan/dot1q:vlan` to use `igmp-snooping-instance`. It means IGMP Snooping is enabled in the specified VLAN on the bridge.

The mld-snooping-instance could be used in concurrence with igmp-snooping-instance to configure the MLD Snooping.

```
augment /dot1q:bridges/dot1q:bridge:
  +--rw igmp-snooping-instance?   igmp-mld-snooping-instance-ref
  +--rw mld-snooping-instance?    igmp-mld-snooping-instance-ref

augment /dot1q:bridges/dot1q:bridge/dot1q:component
  /dot1q:bridge-vlan/dot1q:vlan:
  +--rw igmp-snooping-instance?   igmp-mld-snooping-instance-ref
  +--rw mld-snooping-instance?    igmp-mld-snooping-instance-ref
```

### 3.4. IGMP and MLD Snooping Actions

IGMP and MLD Snooping actions clear the specified IGMP and MLD Snooping group tables. If both source X and group Y are specified, only source X from group Y in that specific instance will be cleared.

```
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol:
  +--rw igmp-snooping-instance {igmp-snooping}?
  +---x clear-igmp-snooping-groups {action-clear-groups}?
    +---w input
      +---w group      union
      +---w source     rt-types:ipv4-multicast-source-address

augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol:
  +--rw mld-snooping-instance {mld-snooping}?
  +---x clear-mld-snooping-groups {action-clear-groups}?
    +---w input
      +---w group      union
      +---w source     rt-types:ipv6-multicast-source-address
```

## 4. IGMP and MLD Snooping YANG Module

This module references [RFC1112], [RFC2236], [RFC2710], [RFC3376], [RFC3810], [RFC4541], [RFC5790], [RFC6636], [RFC6991], [RFC7761], [RFC8343], [dot1Qcp].

```
<CODE BEGINS> file ietf-igmp-mld-snooping@2021-10-08.yang
module ietf-igmp-mld-snooping {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-igmp-mld-snooping";

  prefix ims;

  import ietf-inet-types {
    prefix "inet";
```

```
    reference
      "RFC 6991: Common YANG Data Types";
  }

  import ietf-yang-types {
    prefix "yang";
    reference
      "RFC 6991: Common YANG Data Types";
  }

  import ietf-interfaces {
    prefix "if";
    reference
      "RFC 8343: A YANG Data Model for Interface Management";
  }

  import ietf-routing {
    prefix "rt";
    reference
      "RFC 8349: A YANG Data Model for Routing Management (NMDA
      Version)";
  }

  import ietf-routing-types {
    prefix "rt-types";
    reference
      "RFC 8294: Common YANG Data Types for the Routing Area";
  }

  import ieee802-dot1q-bridge {
    prefix "dot1q";
    reference
      "dot1Qcp: IEEE 802.1Qcp-2018 Bridges and Bridged Networks
      - Amendment: YANG Data Model";
  }

  organization
    "IETF PIM Working Group";

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

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

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

              Yisong Liu
              <mailto:liuyisong@chinamobile.com>
```

Anish Peter  
<mailto:anish.ietf@gmail.com>

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

";

description

"The module defines a collection of YANG definitions common for all devices that implement Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping which is described in RFC 4541.

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

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

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

```
revision 2021-10-08 {  
  description  
    "Initial revision.";  
  reference  
    "RFC XXXX: A YANG Data Model for IGMP and MLD Snooping";  
}
```

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

```
feature igmp-snooping {  
  description  
    "Support IGMP snooping.";  
  reference  
    "RFC 4541";  
}
```

```
feature mld-snooping {  
  description  
    "Support MLD snooping.";  
  reference  
    "RFC 4541";
```

```
}

feature fast-leave {
  description
    "Support configuration of fast leave. The fast leave feature
    does not send last member query messages to hosts.";
  reference
    "RFC 3376";
}

feature static-l2-multicast-group {
  description
    "Support configuration of static L2 multicast group.";
}

feature static-mrouter-interface {
  description
    "Support multicast router interface explicitly configured
    by management";
  reference
    "RFC 4541";
}

feature action-clear-groups {
  description
    "Support clearing statistics by action for IGMP & MLD snooping.";
}

feature require-router-alert {
  description
    "Support configuration of require-router-alert.";
  reference
    "RFC 3376";
}

feature lite-exclude-filter {
  description
    "Enable the support of the simplified EXCLUDE filter.";
  reference
    "RFC 5790";
}

feature explicit-tracking {
  description
    "Support configuration of per instance explicit-tracking.";
  reference
    "RFC 6636";
}

/* identities */
```

```
identity l2-service-type {
  description
    "Base identity for L2 service type in IGMP & MLD snooping";
}

identity bridge {
  base l2-service-type;
  description
    "This identity represents BRIDGE service.";
}

identity filter-mode {
  description
    "Base identity for filter mode in IGMP & MLD snooping";
}

identity include {
  base filter-mode;
  description
    "This identity represents include mode.";
}

identity exclude {
  base filter-mode;
  description
    "This identity represents exclude mode.";
}

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

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

/*
 * Typedefs
 */

typedef l2-service-type {
  type identityref {
    base "l2-service-type";
  }
  description "The L2 service type used with IGMP & MLD snooping ";
}
```

```
typedef filter-mode-type {
  type identityref {
    base "filter-mode";
  }
  description "The host filter mode";
}

typedef igmp-mld-snooping-instance-ref {
  type leafref {
    path "/rt:routing/rt:control-plane-protocols"+
        "/rt:control-plane-protocol/rt:name";
  }
  description
    "This type is used by data models which need to
    reference IGMP & MLD snooping instance.";
}

/*
 * Groupings
 */

grouping instance-config-attributes-igmp-mld-snooping {
  description
    "IGMP and MLD snooping configuration of each VLAN.";

  leaf enable {
    type boolean;
    default false;
    description
      "Set the value to true to enable IGMP & MLD snooping.";
  }

  leaf forwarding-table-type {
    type enumeration {
      enum "mac" {
        description
          "MAC-based lookup mode";
      }
      enum "ip" {
        description
          "IP-based lookup mode";
      }
    }
    default "ip";
    description "The default forwarding table type is ip";
  }

  leaf explicit-tracking {
    if-feature explicit-tracking;
    type boolean;
    default false;
  }
}
```



```
description
  "Track the IGMPv3 and MLDv2 snooping membership reports
   from individual hosts. It contributes to saving network
   resources and shortening leave latency.";
}

leaf lite-exclude-filter {
  if-feature lite-exclude-filter;
  type empty;
  description
    "For IGMP Snooping, the presence of this
     leaf enables the support of the simplified EXCLUDE filter
     in the Lightweight IGMPv3 protocol, which simplifies the
     standard versions of IGMPv3.
     For MLD Snooping, the presence of this
     leaf enables the support of the simplified EXCLUDE filter
     in the Lightweight MLDv2 protocol, which simplifies the
     standard versions of MLDv2.";
  reference
    "RFC 5790";
}

leaf send-query {
  type boolean;
  default false;
  description
    "When it is true, this switch will send out periodic
     IGMP General Query Message or MLD General Query Message.";
}

leaf fast-leave {
  if-feature fast-leave;
  type empty;
  description
    "When immediate leave is enabled, the IGMP software assumes
     that no more than one host is present on each VLAN port.";
}

leaf last-member-query-interval {
  type uint16 {
    range "10..10230";
  }
  units deciseconds;
  default 10;
  description
    "Last Member Query Interval, which may be tuned to modify
     the leave latency of the network.
     It is represented in units of 1/10 second.";
  reference "RFC 3376. Sec. 8.8.";
}
```

```
leaf query-interval {
    type uint16;
    units seconds;
    default 125;
    description
        "The Query Interval is the interval between General Queries
        sent by the Querier.";
    reference "RFC 3376. Sec. 4.1.7, 8.2, 8.14.2.";
}

leaf query-max-response-time {
    type uint16;
    units deciseconds;
    default 100;
    description
        "Query maximum response time specifies the maximum time
        allowed before sending a responding report.
        It is represented in units of 1/10 second.";
    reference "RFC 3376. Sec. 4.1.1, 8.3, 8.14.3.";
}

leaf require-router-alert {
    if-feature require-router-alert;
    type boolean;
    default false;
    description
        "When the value is true, router alert should exist
        in the IP header of IGMP or MLD packet. If it doesn't exist,
        the IGMP or MLD packet will be ignored.";
    reference "RFC 3376. Sec. 9.1, 9.2, 9.3.";
}

leaf robustness-variable {
    type uint8 {
        range "1..7";
    }
    default 2;
    description
        "Querier's Robustness Variable allows tuning for the
        expected packet loss on a network.";
    reference "RFC 3376. Sec. 4.1.6, 8.1, 8.14.1.";
}

leaf-list static-bridge-mrouter-interface {
    when 'derived-from-or-self(..l2-service-type,"ims:bridge")';
    if-feature static-mrouter-interface;
    type if:interface-ref;
    description "static mrouter interface in BRIDGE forwarding";
}
} // instance-config-attributes-igmp-mld-snooping
```

```
grouping instance-state-group-attributes-igmp-ml-d-snooping {
  description
    "Attributes for both IGMP and MLD snooping groups.";

  leaf mac-address {
    type yang:phys-address;
    description "Destination MAC address for L2 multicast.";
  }

  leaf expire {
    type rt-types:timer-value-seconds16;
    units seconds;
    description
      "The time left before multicast group timeout.";
  }

  leaf up-time {
    type uint32;
    units seconds;
    mandatory true;
    description
      "The time elapsed since L2 multicast record created.";
  }
} // instance-state-group-attributes-igmp-ml-d-snooping

grouping instance-state-attributes-igmp-ml-d-snooping {
  description
    "State attributes for IGMP & MLD snooping instance.";

  leaf entries-count {
    type yang:gauge32;
    config false;
    description
      "The number of L2 multicast entries in IGMP & MLD snooping";
  }

  leaf-list bridge-mrouter-interface {
    when 'derived-from-or-self(..//l2-service-type,"ims:bridge")';
    type if:interface-ref;
    config false;
    description
      "Indicates a list of mrouter interfaces dynamically learned in a
      bridge. When this switch receives IGMP/MLD queries from a
      multicast router on an interface, the interface will become
      mrouter interface for IGMP/MLD snooping.";
  }
} // instance-config-attributes-igmp-ml-d-snooping

grouping instance-state-source-attributes-igmp-ml-d-snooping {
```

```
description
  "State attributes for IGMP & MLD snooping instance.";

  leaf-list bridge-outgoing-interface {
    when 'derived-from-or-self ../../../../l2-service-
type, "ims:bridge")';
    type if:interface-ref;
    description "Outgoing interface in BRIDGE forwarding";
  }

  leaf up-time {
    type uint32;
    units seconds;
    mandatory true;
    description
      "The time elapsed since L2 multicast record created";
  }

  leaf expire {
    type rt-types:timer-value-seconds16;
    units seconds;
    description
      "The time left before multicast group timeout.";
  }

  leaf host-count {
    if-feature explicit-tracking;
    type yang:gauge32;
    description
      "The number of host addresses.";
  }
} // instance-state-source-attributes-igmp-ml-d-snooping

grouping igmp-snooping-statistics {
  description
    "The statistics attributes for IGMP snooping.";

  leaf query-count {
    type yang:counter64;
    description
      "The number of Membership Query messages.";
    reference
      "RFC 2236";
  }

  leaf membership-report-v1-count {
    type yang:counter64;
    description
      "The number of Version 1 Membership Report messages.";
    reference
      "RFC 1112";
  }
}
```

```
    leaf membership-report-v2-count {
      type yang:counter64;
      description
        "The number of Version 2 Membership Report messages.";
      reference
        "RFC 2236";
    }
    leaf membership-report-v3-count {
      type yang:counter64;
      description
        "The number of Version 3 Membership Report messages.";
      reference
        "RFC 3376";
    }
    leaf leave-count {
      type yang:counter64;
      description
        "The number of Leave Group messages.";
      reference
        "RFC 2236";
    }
    leaf pim-hello-count {
      type yang:counter64;
      description
        "The number of PIM hello messages.";
      reference
        "RFC 7761";
    }
  } // igmp-snooping-statistics

  grouping mld-snooping-statistics {
    description
      "The statistics attributes for MLD snooping.";

    leaf query-count {
      type yang:counter64;
      description
        "The number of Multicast Listener Query messages.";
      reference
        "RFC 3810";
    }
    leaf report-v1-count {
      type yang:counter64;
      description
        "The number of Version 1 Multicast Listener Report.";
      reference
        "RFC 2710";
    }
    leaf report-v2-count {
      type yang:counter64;
      description
```

```
        "The number of Version 2 Multicast Listener Report.";
    reference
        "RFC 3810";
}
leaf done-count {
    type yang:counter64;
    description
        "The number of Version 1 Multicast Listener Done.";
    reference
        "RFC 2710";
}
leaf pim-hello-count {
    type yang:counter64;
    description
        "The number of PIM hello messages.";
    reference
        "RFC 7761";
}
} // mld-snooping-statistics

augment "/rt:routing/rt:control-plane-protocols"+
    "/rt:control-plane-protocol" {
    when 'derived-from-or-self(rt:type, "ims:igmp-snooping")' {
        description
            "This container is only valid for IGMP snooping.";
    }
    description
        "IGMP snooping augmentation to control plane protocol
        configuration and state.";

    container igmp-snooping-instance {
        if-feature igmp-snooping;
        description
            "IGMP snooping instance to configure igmp-snooping.";

        leaf l2-service-type {
            type l2-service-type;
            default bridge;
            description
                "It indicates BRIDGE or other services.";
        }
    }

    uses instance-config-attributes-igmp-mld-snooping;

    leaf igmp-version {
        type uint8 {
            range "1..3";
        }
        default 2;
        description "IGMP version.";
    }
}
```

```
leaf querier-source {
  type inet:ipv4-address;
  description
    "The source address of IGMP General Query message,
    which is sent out by this switch.";
}

list static-l2-multicast-group {
  if-feature static-l2-multicast-group;
  key "group source-addr";
  description
    "A static multicast route, (*,G) or (S,G).";

  leaf group {
    type rt-types:ipv4-multicast-group-address;
    description
      "Multicast group IPv4 address";
  }

  leaf source-addr {
    type rt-types:ipv4-multicast-source-address;
    description
      "Multicast source IPv4 address.";
  }

  leaf-list bridge-outgoing-interface {
    when 'derived-from-or-self(..../l2-service-
type,"ims:bridge")';
    type if:interface-ref;
    description "Outgoing interface in BRIDGE forwarding";
  }
} // static-l2-multicast-group

uses instance-state-attributes-igmp-mld-snooping;

list group {

  key "address";

  config false;

  description "IGMP snooping information";

  leaf address {
    type rt-types:ipv4-multicast-group-address;
    description
      "Multicast group IPv4 address";
  }

  uses instance-state-group-attributes-igmp-mld-snooping;
```

```
leaf last-reporter {
    type inet:ipv4-address;
    description
        "Address of the last host which has sent report to join
        the multicast group.";
}

list source {
    key "address";
    description "Source IPv4 address for multicast stream";

    leaf address {
        type rt-types:ipv4-multicast-source-address;
        description "Source IPv4 address for multicast stream";
    }

    uses instance-state-source-attributes-igmp-ml-d-snooping;

    leaf last-reporter {
        type inet:ipv4-address;
        description
            "Address of the last host which has sent report
            to join the multicast group.";
    }

    list host {
        if-feature explicit-tracking;
        key "address";
        description
            "List of multicast membership hosts
            of the specific multicast source-group.";

        leaf address {
            type inet:ipv4-address;
            description
                "Multicast membership host address.";
        }

        leaf filter-mode {
            type filter-mode-type;
            mandatory true;
            description
                "Filter mode for a multicast membership
                host may be either include or exclude.";
        }
    } // list host
} // list source
} // list group

container interfaces {
    config false;
```



```
description
  "Contains the interfaces associated with the IGMP snooping
  instance";

list interface {
  key "name";

  description
    "A list of interfaces associated with the IGMP snooping
    instance";

  leaf name {
    type if:interface-ref;
    description
      "The name of interface";
  }

  container statistics {
    description
      "The interface statistics for IGMP snooping";

    leaf discontinuity-time {
      type yang:date-and-time;
      description
        "The time on the most recent occasion at which any one
        or more of the statistic counters suffered a
        discontinuity. If no such discontinuities have
        occurred since the last re-initialization of the local
        management subsystem, then this node contains the time
        the local management subsystem re-initialized
        itself.";
    }

    container received {
      description
        "Number of received snooped IGMP packets";

      uses igmp-snooping-statistics;
    }

    container sent {
      description
        "Number of sent snooped IGMP packets";

      uses igmp-snooping-statistics;
    }
  }
}

action clear-igmp-snooping-groups {
```

```
    if-feature action-clear-groups;
    description
        "Clear IGMP snooping cache tables.";

    input {
        leaf group {
            type union {
                type enumeration {
                    enum 'all-groups' {
                        description
                            "All multicast group addresses.";
                    }
                }
                type rt-types:ipv4-multicast-group-address;
            }
            mandatory true;
            description
                "Multicast group IPv4 address. If value 'all-groups' is
                 specified, all IGMP snooping group entries are cleared
                 for specified source address.";
        }
        leaf source {
            type rt-types:ipv4-multicast-source-address;
            mandatory true;
            description
                "Multicast source IPv4 address. If value '*' is specified,
                 all IGMP snooping source-group tables are cleared.";
        }
    }
} // action clear-igmp-snooping-groups
} // igmp-snooping-instance
} // augment

augment "/rt:routing/rt:control-plane-protocols"+
    "/rt:control-plane-protocol" {
    when 'derived-from-or-self(rt:type, "ims:mld-snooping")' {
        description
            "This container is only valid for MLD snooping.";
    }
    description
        "MLD snooping augmentation to control plane protocol
         configuration and state.";

    container mld-snooping-instance {
        if-feature mld-snooping;
        description
            "MLD snooping instance to configure mld-snooping.";

        leaf l2-service-type {
            type l2-service-type;
            default bridge;
        }
    }
}
```

```
    description
      "It indicates BRIDGE or other services.";
  }

  uses instance-config-attributes-igmp-mld-snooping;

  leaf mld-version {
    type uint8 {
      range "1..2";
    }
    default 2;
    description "MLD version.";
  }

  leaf querier-source {
    type inet:ipv6-address;
    description
      "The source address of MLD General Query message,
       which is sent out by this switch.";
  }

  list static-l2-multicast-group {
    if-feature static-l2-multicast-group;
    key "group source-addr";
    description
      "A static multicast route, (*,G) or (S,G).";

    leaf group {
      type rt-types:ipv6-multicast-group-address;
      description
        "Multicast group IPv6 address";
    }

    leaf source-addr {
      type rt-types:ipv6-multicast-source-address;
      description
        "Multicast source IPv6 address.";
    }

    leaf-list bridge-outgoing-interface {
      when 'derived-from-or-self(..../l2-service-
type,"ims:bridge")';
      type if:interface-ref;
      description "Outgoing interface in BRIDGE forwarding";
    }
  } // static-l2-multicast-group

  uses instance-state-attributes-igmp-mld-snooping;

  list group {
    key "address";
```

```
config false;
description "MLD snooping statistics information";

leaf address {
  type rt-types:ipv6-multicast-group-address;
  description
    "Multicast group IPv6 address";
}

uses instance-state-group-attributes-igmp-mld-snooping;

leaf last-reporter {
  type inet:ipv6-address;
  description
    "Address of the last host which has sent report
    to join the multicast group.";
}

list source {
  key "address";
  description "Source IPv6 address for multicast stream";

  leaf address {
    type rt-types:ipv6-multicast-source-address;
    description "Source IPv6 address for multicast stream";
  }

  uses instance-state-source-attributes-igmp-mld-snooping;

  leaf last-reporter {
    type inet:ipv6-address;
    description
      "Address of the last host which has sent report
      to join the multicast group.";
  }

  list host {
    if-feature explicit-tracking;
    key "address";
    description
      "List of multicast membership hosts
      of the specific multicast source-group.";

    leaf address {
      type inet:ipv6-address;
      description
        "Multicast membership host address.";
    }
    leaf filter-mode {
      type filter-mode-type;
      mandatory true;
    }
  }
}
```

```
        description
            "Filter mode for a multicast membership
             host may be either include or exclude.";
    }
    } // list host
  } // list source
} // list group

container interfaces {
  config false;

  description
    "Contains the interfaces associated with the MLD snooping
     instance";

  list interface {
    key "name";

    description
      "A list of interfaces associated with the MLD snooping
       instance";

    leaf name {
      type if:interface-ref;
      description
        "The name of interface";
    }
  }

  container statistics {
    description
      "The interface statistics for MLD snooping";

    leaf discontinuity-time {
      type yang:date-and-time;
      description
        "The time on the most recent occasion at which any one
         or more of the statistic counters suffered a
         discontinuity. If no such discontinuities have
         occurred since the last re-initialization of the local
         management subsystem, then this node contains the time
         the local management subsystem re-initialized
         itself.";
    }
  }

  container received {
    description
      "Number of received snooped MLD packets";

    uses mld-snooping-statistics;
  }

  container sent {
```

```
        description
            "Number of sent snooped MLD packets";

        uses mld-snooping-statistics;
    }
}

action clear-mld-snooping-groups {
    if-feature action-clear-groups;
    description
        "Clear MLD snooping cache tables.";

    input {
        leaf group {
            type union {
                type enumeration {
                    enum 'all-groups' {
                        description
                            "All multicast group addresses.";
                    }
                }
                type rt-types:ipv6-multicast-group-address;
            }
            mandatory true;
            description
                "Multicast group IPv6 address. If value 'all-groups' is
                specified, all MLD snooping group entries are cleared
                for specified source address.";
        }
        leaf source {
            type rt-types:ipv6-multicast-source-address;
            mandatory true;
            description
                "Multicast source IPv6 address. If value '*' is specified,
                all MLD snooping source-group tables are cleared.";
        }
    }
} // action clear-mld-snooping-groups
} // mld-snooping-instance
} // augment

augment "/dot1q:bridges/dot1q:bridge" {
    description
        "Use IGMP & MLD snooping instance in BRIDGE.";

    leaf igmp-snooping-instance {
        type igmp-mld-snooping-instance-ref;
        description
            "Configure IGMP snooping instance under bridge view";
    }
}
```

```
    }

    leaf mld-snooping-instance {
      type igmp-mld-snooping-instance-ref;
      description
        "Configure MLD snooping instance under bridge view";
    }
  }

  augment "/dot1q:bridges/dot1q:bridge"+
    "/dot1q:component/dot1q:bridge-vlan/dot1q:vlan" {
    description
      "Use IGMP & MLD snooping instance in certain VLAN of BRIDGE";

    leaf igmp-snooping-instance {
      type igmp-mld-snooping-instance-ref;
      description
        "Configure IGMP snooping instance under VLAN view";
    }

    leaf mld-snooping-instance {
      type igmp-mld-snooping-instance-ref;
      description
        "Configure MLD snooping instance under VLAN view";
    }
  }
}
<CODE ENDS>
```

## 5. Security Considerations

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

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

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

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

ims:igmp-snooping-instance

ims:mld-snooping-instance

The subtrees under /dot1q:bridges/dot1q:bridge

ims:igmp-snooping-instance

ims:mld-snooping-instance

The subtrees under /dot1q:bridges/dot1q:bridge/dot1q:component  
/dot1q:bridge-vlan/dot1q:vlan

ims:igmp-snooping-instance

ims:mld-snooping-instance

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

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

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

ims:igmp-snooping-instance

ims:mld-snooping-instance

Unauthorized access to any data node of these subtrees can disclose the operational state information of IGMP & MLD Snooping on this device. The group/source/host information may expose multicast group memberships, and transitively the associations between the user on the host and the contents from the source which could be privately sensitive. Some of the action operations in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control access to these operations. These are the operations and their sensitivity/vulnerability:

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

ims:igmp-snooping-instance/ims:clear-igmp-snooping-groups

ims:mld-snooping-instance/ims:clear-mld-snooping-groups



Some of the actions in this YANG module may be considered sensitive or vulnerable in some network environments. The IGMP & MLD Snooping YANG module supports the "clear-igmp-snooping-groups" and "clear-mld-snooping-groups" actions. If unauthorized action is invoked, the IGMP and MLD Snooping group tables will be cleared unexpectedly. Especially when using wildcard, all the multicast traffic will be flooded in the broadcast domain. The devices that use this YANG module should heed the Security Considerations in [RFC4541].

## 6. IANA Considerations

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

### 6.1. XML Registry

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

---

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

---

### 6.2. YANG Module Names Registry

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

---

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

---

## 7. References

### 7.1. Normative References

- [dot1Qcp] IEEE, "Standard for Local and metropolitan area networks--Bridges and Bridged Networks--Amendment 30: YANG Data Model", IEEE Std 802.1Qcp-2018 (Revision of IEEE Std 802.1Q-2014), September 2018,  
<<https://ieeexplore.ieee.org/servlet/opac?punumber=8467505>>
- [RFC1112] Deering, S., "Host extensions for IP multicasting", STD 5, RFC 1112, August 1989.
- [RFC2236] W. Fenner, "Internet Group Management Protocol, Version 2", RFC 2236, November 1997.
- [RFC2710] Deering, S., Fenner, W., and B. Haberman, "Multicast Listener Discovery (MLD) for IPv6", RFC 2710, October 1999.
- [RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, October 2002.
- [RFC3688] Mealling, M., "The IETF XML Registry", RFC 3688, January 2004.
- [RFC3810] Vida, R. and L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, June 2004.
- [RFC4286] B. Haberman and J. Martin, "Multicast Router Discovery", RFC 4286, December 2005.
- [RFC4541] M. Christensen, K. Kimball, F. Solensky, "Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches", RFC 4541, May 2006.
- [RFC5790] H. Liu, W. Cao, H. Asaeda, "Lightweight Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Version 2 (MLDv2) Protocols", RFC 5790, February 2010.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.
- [RFC6241] R. Enns, Ed., M. Bjorklund, Ed., J. Schoenwaelder, Ed., A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, June 2011.

- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, June 2011.
- [RFC6636] H. Asaeda, H. Liu, Q. Wu, "Tuning the Behavior of the Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) for Routers in Mobile and Wireless Networks", RFC 6636, May 2012.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, July 2013.
- [RFC7761] B. Fenner, M. Handley, H. Holbrook, I. Kouvelas, R. Parekh, Z. Zhang, L. Zheng, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", RFC 7761, March 2016.
- [RFC7950] M. Bjorklund, Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, August 2016.
- [RFC8040] A. Bierman, M. Bjorklund, K. Watsen, "RESTCONF Protocol", RFC 8040, January 2017.
- [RFC8294] X. Liu, Y. Qu, A. Lindem, C. Hopps, L. Berger, "Common YANG Data Types for the Routing Area", RFC 8294, December 2017.
- [RFC8340] M. Bjorklund, and L. Berger, Ed., "YANG Tree Diagrams", RFC 8340, March 2018.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", RFC 8341, March 2018.
- [RFC8342] M. Bjorklund and J. Schoenwaelder, "Network Management Datastore Architecture (NMDA)", RFC 8342, March 2018.
- [RFC8343] M. Bjorklund, "A YANG Data Model for Interface Management", RFC 8343, March 2018.
- [RFC8349] L. Lhotka, A. Lindem, Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", RFC 8349, March 2018.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, August 2018.

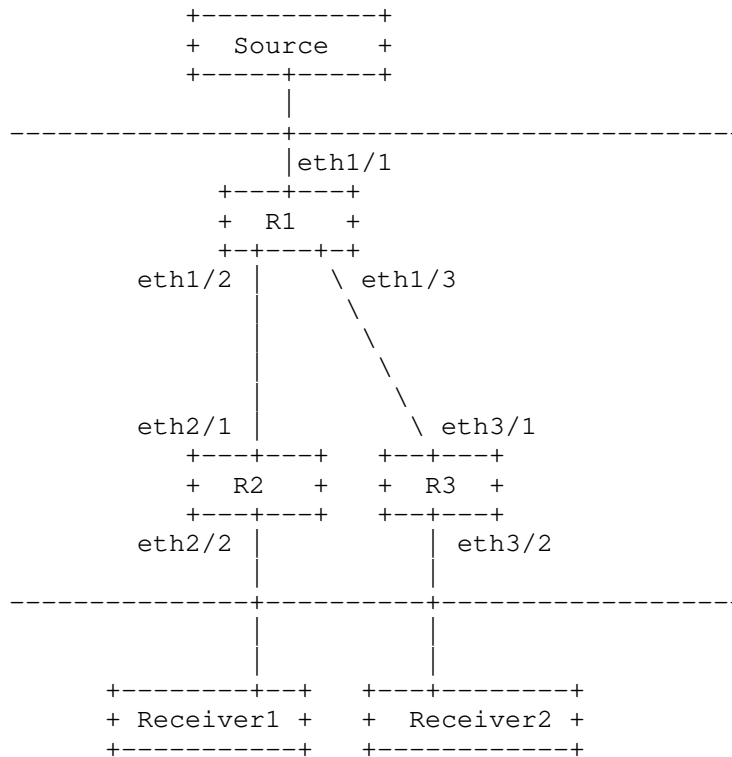
## 7.2. Informative References

- [RFC7951] L. Lhotka, "JSON Encoding of Data Modeled with YANG", RFC 7951, August 2016.
- [RFC8407] A. Bierman, "Guidelines for Authors and Reviewers of Documents Containing YANG Data Models", RFC 8407, October 2018.

[RFC8652] X. Liu, F. Guo, M. Sivakumar, P. McAllister, A. Peter, "A YANG Data Model for the Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD)", RFC 8652, November 2019.

## Appendix A. Data Tree Example

This section contains an example for bridge service in the JSON encoding [RFC7951], containing both configuration and state data.



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

```

{
  "ietf-interfaces:interfaces":{
    "interface":[
      {
        "name":"eth1/1",
        "type":"iana-if-type:ethernetCsmacd"
      }
    ]
  }
}

```

```

    },
    "ietf-routing:routing":{
      "control-plane-protocols":{
        "control-plane-protocol":[
          {
            "type":"ietf-igmp-ml-d-snooping:igmp-snooping",
            "name":"bis1",
            "ietf-igmp-ml-d-snooping:igmp-snooping-instance":{
              "l2-service-type":"ietf-igmp-ml-d-snooping:bridge",
              "enable":true
            }
          }
        ]
      }
    },
    "ieee802-dot1q-bridge:bridges":{
      "bridge":[
        {
          "name":"ispl",
          "address":"00-23-ef-a5-77-12",
          "bridge-type":"ieee802-dot1q-bridge:customer-vlan-bridge",
          "component":[
            {
              "name":"compl",
              "type":"ieee802-dot1q-bridge:c-vlan-component",
              "bridge-vlan":{
                "vlan":[
                  {
                    "vid":101,
                    "ietf-igmp-ml-d-snooping:igmp-snooping-instance":"bis1"
                  }
                ]
              }
            }
          ]
        }
      ]
    }
  }
}

```

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

```

{
  "ietf-interfaces:interfaces": {
    "interface": [
      {
        "name": "eth1/1",
        "type": "iana-if-type:ethernetCsmacd",
        "oper-status": "up",
        "statistics": {
          "discontinuity-time": "2018-05-23T12:34:56-05:00"
        }
      }
    ]
  }
}

```

```

    }
  }
],
},
"ietf-routing:routing": {
  "control-plane-protocols": {
    "control-plane-protocol": [
      {
        "type": "ietf-igmp-mld-snooping:igmp-snooping",
        "name": "bis1",
        "ietf-igmp-mld-snooping:igmp-snooping-instance": {
          "l2-service-type": "ietf-igmp-mld-snooping:bridge",
          "enable": true
        }
      }
    ]
  }
},
"ieee802-dot1q-bridge:bridges": {
  "bridge": [
    {
      "name": "isp1",
      "address": "00-23-ef-a5-77-12",
      "bridge-type": "ieee802-dot1q-bridge:customer-vlan-bridge",
      "component": [
        {
          "name": "comp1",
          "type": "ieee802-dot1q-bridge:c-vlan-component",
          "bridge-vlan": {
            "vlan": [
              {
                "vid": 101,
                "ietf-igmp-mld-snooping:igmp-snooping-instance": "bis1"
              }
            ]
          }
        }
      ]
    }
  ]
}
]
}
}

```

The following action is to clear all the entries whose group address is 225.1.1.1 for igmp-snooping-instance bis1.

```

POST /restconf/operations/ietf-routing:routing/control-plane-protocols/\
control-plane-protocol=ietf-igmp-mld-snooping:igmp-snooping,bis1/\
ietf-igmp-mld-snooping:igmp-snooping-instance/\
clear-igmp-snooping-groups HTTP/1.1
Host: example.com
Content-Type: application/yang-data+json

```

```
{  
  "ietf-igmp-ml-d-snooping:input" : {  
    "group": "225.1.1.1",  
    "source": "*"   
  }  
}
```

#### Authors' Addresses

Hongji Zhao  
Ericsson (China) Communications Company Ltd.  
Ericsson Tower, No. 5 Lize East Street,  
Chaoyang District Beijing 100102, China

Email: hongji.zhao@ericsson.com

Xufeng Liu  
Volta Networks  
USA

EMail: xufeng.liu.ietf@gmail.com

Yisong Liu  
China Mobile  
China

Email: liuyisong@chinamobile.com

Anish Peter  
Individual

Email: anish.ietf@gmail.com

Mahesh Sivakumar  
Juniper Networks  
1133 Innovation Way  
Sunnyvale, California  
USA

Email: sivakumar.mahesh@gmail.com

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: November 2, 2018

A. Gupta  
Avi Networks  
S. Venaas  
Cisco Systems  
May 1, 2018

Use of PIM Address List Hello across address families  
draft-ietf-pim-ipv4-prefix-over-ipv6-nh-02.txt

## Abstract

In the PIM Sparse Mode standard there is an Address List Hello option used to list secondary addresses of an interface. Usually the addresses would be of the same address family as the primary address. In this document we provide a use case for listing secondary addresses that are from a different family. In particular, Multi-Protocol BGP (MP-BGP) has support for distributing next-hop information for multiple address families using one AFI/SAFI Network Layer Reachability Information (NLRI). When using this combined with PIM, the Address List Hello option can be used to determine which PIM neighbor to use as RPF neighbor.

## Status of This Memo

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

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

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

This Internet-Draft will expire on November 2, 2018.

## Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of



publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

#### Table of Contents

1. Introduction . . . . .	2
2. Solution . . . . .	4
3. Security Considerations . . . . .	4
4. IANA Considerations . . . . .	4
5. References . . . . .	4
5.1. Normative References . . . . .	4
5.2. Informative References . . . . .	4
Authors' Addresses . . . . .	5

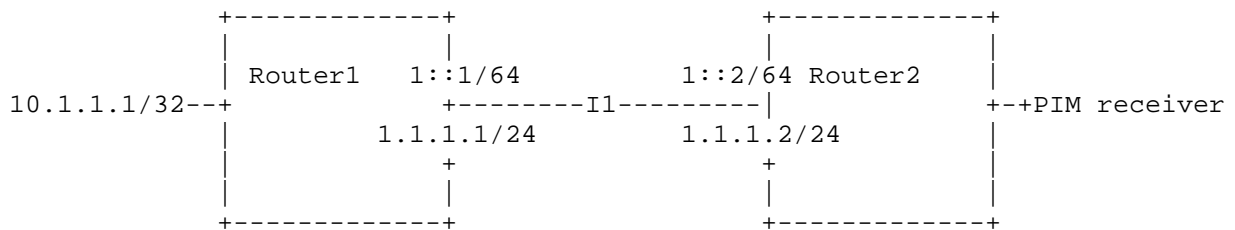
## 1. Introduction

The PIM Sparse Mode standard [RFC7761] defines an Address List Hello option used to list secondary addresses of an interface. It specifies that the addresses listed SHOULD be of the same address family as the primary address. It was not anticipated that it could be useful to list addresses of a different address family. This document describes a use-case for listing different address families.

While use of MP-BGP along with [RFC5549] enables one routing protocol session to exchange next-hop info for both IPv4 and IPv6 prefixes, forwarding plane needs additional procedures to enable forwarding in data-plane. For example, when a IPv4 prefix is learnt over IPv6 next-hop, forwarding plane resolves the MAC-Address (L2-Adjacency) for IPv6 next-hop and uses it as destination-mac while doing inter-subnet forwarding. While it's simple to find the required information for unicast forwarding, multicast forwarding in same scenario poses additional requirements.

Multicast traffic is forwarding on a tree build by multicast routing protocols such as PIM. Multicast routing protocols are address family dependent and hence a system enabled with IPv4 and IPv6 multicast routing will have two PIM sessions one for each of the AF. Also, Multicast routing protocol uses Unicast reachability information to find unique Reverse Path Forwarding Neighbor. Further it sends control messages such as PIM Join to form the tree. Now when a PIMv4 session needs to initiate new multicast tree in event of discovering new receiver It consults Unicast control plane to find next-hop information. While this multicast tree can be Shared or Shortest Path tree, PIMv4 will need a PIMv4 neighbor to send join. However, the Unicast control plane can provide IPv6 next-hop as explained earlier and hence we need certain procedures to find corresponding PIMv4 neighbor address. This address is vital for correct prorogation of join and furthermore to build multicast tree. This document describes various approaches along with their use-cases and pros-cons.

Figure 1: Example Topology



In example topology, Router1 and Router2 are PIMv4 and PIMv6 neighbors on Interface I1. Router2 learns prefix 10.1.1.1/32's next-hop as 1::1/64 on Interface I1 as advertised by Router1 using BGP IPV6 NLRI. But in order to send (10.1.1.1/32, multicast-group) PIMv4 join on Interface I1, Router2 needs to find corresponding PIMv4 neighbor. In case there are multiple PIMv4 neighbors on same Interface I1, problem is aggravated.

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 BCP 14, [RFC2119].

## 2. Solution

A PIM router can advertise its locally configured IPv6 addresses on the interface in PIMv4 Hello messages as per [RFC7761] section 4.3.4. Same applies for IPv4 address in PIMv6 Hello. PIM will keep this info for each neighbor in Neighbor-cache along with DR-priority, hold-time etc. Once IPv6 Next-hop is notified to PIMv4, it will look into neighbors on the notified RPF-interface and find PIMv4 neighbor advertising same IPv6 local address in secondary Neighbor-list. If such a match is found, that particular neighbor will be used as IPv4 RPF-Neighbor for initiating upstream join.

This method is valid for networks enabled with PIMv4 and PIMv6 both as well for the networks enabled with only PIMv4 with IPv6 BGP session or PIMv6 with IPv4 BGP session. This method doesn't require any additional config changes in the network.

## 3. Security Considerations

There are no new security considerations.

## 4. IANA Considerations

There are no IANA considerations.

## 5. References

### 5.1. Normative References

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

### 5.2. Informative References

- [RFC5549] Le Faucheur, F. and E. Rosen, "Advertising IPv4 Network Layer Reachability Information with an IPv6 Next Hop", RFC 5549, DOI 10.17487/RFC5549, May 2009, <<https://www.rfc-editor.org/info/rfc5549>>.
- [RFC7761] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I., Parekh, R., Zhang, Z., and L. Zheng, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", STD 83, RFC 7761, DOI 10.17487/RFC7761, March 2016, <<https://www.rfc-editor.org/info/rfc7761>>.

Authors' Addresses

Ashutosh Gupta  
Avi Networks  
5155 Old Ironsides Dr. Suite 100  
Santa Clara, CA 95054  
USA

Email: [ashutosh@avinetworks.com](mailto:ashutosh@avinetworks.com)

Stig Venaas  
Cisco Systems  
821 Alder Drive  
San Jose, CA 95035  
USA

Email: [stig@cisco.com](mailto:stig@cisco.com)

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: December 21, 2018

Mankamana. Mishra  
Cisco Systems  
June 19, 2018

PIM Backup Designated Router Procedure  
draft-mankamana-pim-bdr-00

Abstract

On a multi-access network, one of the PIM routers is elected as a Designated Router (DR). On the last hop LAN, the PIM DR is responsible for tracking local multicast listeners and forwarding traffic to these listeners if the group is operating in PIM-SM. In this document, we propose a mechanism to elect backup DR on a shared LAN. A backup DR on LAN would be useful for faster convergence. This draft introduces the concept of a Backup Designated Router (BDR) and the procedure to implement it.

Status of This Memo

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

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

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

This Internet-Draft will expire on December 21, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

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

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Terminology . . . . .	3
3. Applicability and deviation from draft PIM DR Improvement . .	4
4. Protocol Specification . . . . .	4
4.1. PIM Backup DR (BDR) election procedure . . . . .	4
4.2. Existing PIM DR failure . . . . .	4
4.3. Existing PIM BDR failure . . . . .	4
4.4. New PIM Router addition in network . . . . .	4
4.4.1. New PIM router eligible to be PIM DR on shared LAN .	4
4.4.2. New PIM router eligible to be PIM BDR on shared LAN .	5
4.4.3. New PIM router is not eligible to be PIM DR or BDR on shared LAN . . . . .	5
4.5. Initial case, All new PIM router coming up in shared LAN	5
4.6. Benefit . . . . .	6
5. Compatibility . . . . .	6
6. Manageability Considerations . . . . .	6
7. IANA Considerations . . . . .	6
8. Security Considerations . . . . .	6
9. Acknowledgement . . . . .	6
10. Normative References . . . . .	6
Author's Address . . . . .	7

## 1. Introduction

On a multi-access LAN such as an Ethernet, one of the PIM routers is elected as a DR. The PIM DR has two roles in the PIM-SM protocol. On the first hop network, the PIM DR is responsible for registering an active source with the Rendezvous Point (RP) if the group is operating in PIM-SM. On the last hop LAN, the PIM DR is responsible for tracking local multicast listeners and forwarding to these listeners if the group is operating in PIM-SM.

Consider the following last hop LAN in Figure 1:

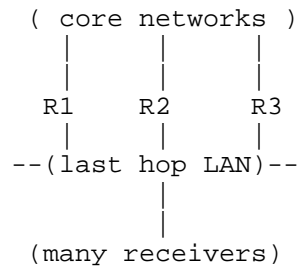


Figure 1: Last Hop LAN

Assume R1 is elected as the Designated Router. According to [RFC4601], R1 will be responsible for forwarding traffic to that LAN on behalf of any local member. In addition to keeping track of IGMP and MLD membership reports, R1 is also responsible for initiating the creation of source and/or shared trees towards the senders or the RPs.

There are multiple reasons for why network could potentially trigger DR re-election. Some of the reasons are

1. R1 going down
2. Access interface towards shared LAN going down
3. Config changed with lower DR priority

When any of above network event occurs, PIM DR re-election would be triggered. When a new DR is elected in shared LAN, new DR would be responsible to build a multicast tree towards source / RP. There are some cases, where traffic is crucial and the operator wants to have minimum traffic loss with DR failure. To address this requirement, this draft introduces a backup DR election procedure which would minimize traffic loss during PIM DR failure.

## 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] .

BDR - PIM Backup DR

With respect to PIM, this document follows the terminology that has been defined in [RFC4601] .

### 3. Applicability and deviation from draft PIM DR Improvement

[I-D.ietf-pim-dr-improvement] defines procedure to solve same problem which was stated in the introduction section of this draft.  
[I-D.ietf-pim-dr-improvement] introduces new PIM Hello options for election of backup PIM DR.

This draft provides mechanism to elect BDR without using any new PIM Hello.

### 4. Protocol Specification

#### 4.1. PIM Backup DR (BDR) election procedure

[RFC7761] defines procedure for PIM DR election. PIM DR is elected on interface "I" among all PIM routers for which "I" has received PIM Hello. BDR election follows the exact same procedure and the second best PIM DR on shared LAN to be chosen as BDR on interface "I"

BDR would perform each of the responsibility of PIM DR except it would not forward traffic on shared LAN.

#### 4.2. Existing PIM DR failure

When PIM DR fails, PIM DR re-election is triggered on shared LAN. Since BDR is second best DR in LAN, it MUST take over immediately and MUST start forwarding multicast traffic on shared LAN.

Again on a shared LAN, new BDR would be elected. and current BDR would be the new DR.

#### 4.3. Existing PIM BDR failure

When an existing PIM BDR fails, the shared LAN MUST have BDR re-election using the DR election procedure from [RFC7761].

#### 4.4. New PIM Router addition in network

When a new PIM router is added in shared LAN, It could be either one of the below defined roles.

##### 4.4.1. New PIM router eligible to be PIM DR on shared LAN

When a new PIM router is added in a shared LAN and has the highest PIM DR priority configured, if a new router starts propagating its configured DR priority right away, the existing PIM DR would give up its role. Then there would be potential traffic loss till the new DR



learns about membership states and builds a multicast tree to the source or RP.

To avoid any such traffic loss situation, new PIM router SHOULD send a PIM Hello with priority 0. After 2 (default value, SHOULD have way to configure) PIM Hello interval or IGMP Query Interval (Which ever is higher) it SHOULD start propagating its original configured DR priority.

Even though a new PIM router propagating its priority as 0, it MUST start building a multicast tree towards source / RP, This is So that traffic loss could be minimized once it starts sending Hello with configured DR priority.

For a brief amount of time, there would be multiple copies of flows present in the multicast core, but a user SHOULD be able to configure whether to send hello with 0 priority or a configured priority. Depending on the application tolerance (Traffic loss Vs Extra traffic in core) the operator can choose option whichever is suitable for network.

After a PIM Hello or IGMP Query interval, the network would get stable with only one DR and one BDR.

#### 4.4.2. New PIM router eligible to be PIM BDR on shared LAN

It SHOULD follow the exact same procedure defined in the previous section.

#### 4.4.3. New PIM router is not eligible to be PIM DR or BDR on shared LAN

First a PIM Hello MUST be sent with priority 0. Once it has gotten Hello from other PIM neighbors, it knows that it is not eligible to be PIM DR or BDR. It MUST send configured PIM DR priority immediately. It MUST not wait for next hello interval.

#### 4.5. Initial case, All new PIM router coming up in shared LAN

In this case, initially each of the PIM routers would send Hellos with priorities of 0. If a PIM router receives all Hellos with priorities 0, it MUST send out a Hello with a configured PIM DR priority. Since it is initial startup case, it would take up to one Hello interval to converge.

#### 4.6. Benefit

1. Easy to implement as it uses an existing PIM procedure to elect DR.
2. Does not introduce any new Hello option

#### 5. Compatibility

#### 6. Manageability Considerations

#### 7. IANA Considerations

#### 8. Security Considerations

#### 9. Acknowledgement

The author would like to thank Stig Venaas, Tharak Abraham, Anish Kachinthaya, Anvitha Kachinthaya for helping with original idea.

#### 10. Normative References

- [I-D.ietf-pim-dr-improvement]  
Zhang, Z., hu, f., Xu, B., and m. mishra, "PIM DR Improvement", draft-ietf-pim-dr-improvement-04 (work in progress), December 2017.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC4601] Fenner, B., Handley, M., Holbrook, H., and I. Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", RFC 4601, DOI 10.17487/RFC4601, August 2006, <<https://www.rfc-editor.org/info/rfc4601>>.
- [RFC7761] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I., Parekh, R., Zhang, Z., and L. Zheng, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", STD 83, RFC 7761, DOI 10.17487/RFC7761, March 2016, <<https://www.rfc-editor.org/info/rfc7761>>.

Author's Address

Mankamana Mishra  
Cisco Systems  
821 Alder Drive,  
MILPITAS, CALIFORNIA 95035  
UNITED STATES

Email: [mankamis@cisco.com](mailto:mankamis@cisco.com)

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: October 10, 2021

M. Mishra  
S. Santhanam  
A. Paramasivam  
J. Goh  
Cisco Systems  
G. Mishra  
Verizon Communications Inc. (VZ)  
April 8, 2021

PIM Backup Designated Router Procedure  
draft-mankamana-pim-bdr-05

Abstract

On a multi-access network, one of the PIM routers is elected as a Designated Router (DR). On the last hop LAN, the PIM DR is responsible for tracking local multicast listeners and forwarding traffic to these listeners if the group is operating in PIM-SM. In this document, we propose a mechanism to elect backup DR on a shared LAN. A backup DR on LAN would be useful for faster convergence. This draft introduces the concept of a Backup Designated Router (BDR) and the procedure to implement it.

Status of This Memo

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

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

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

This Internet-Draft will expire on October 10, 2021.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

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

## Table of Contents

1. Introduction . . . . .	2
2. Terminology . . . . .	3
3. Applicability and deviation from draft PIM DR Improvement . .	4
4. Protocol Specification . . . . .	4
4.1. PIM Backup DR (BDR) election procedure . . . . .	4
4.2. Existing PIM DR failure . . . . .	4
4.3. Existing PIM BDR failure . . . . .	4
4.4. New PIM Router addition in network . . . . .	4
4.4.1. New PIM router eligible to be PIM DR on shared LAN .	4
4.4.2. New PIM router eligible to be PIM BDR on shared LAN .	5
4.4.3. New PIM router is not eligible to be PIM DR or BDR on shared LAN . . . . .	5
4.5. Initial case, All new PIM router coming up in shared LAN	5
4.6. Benefit . . . . .	6
5. Compatibility . . . . .	6
6. Manageability Considerations . . . . .	6
7. IANA Considerations . . . . .	6
8. Security Considerations . . . . .	6
9. Acknowledgement . . . . .	6
10. Normative References . . . . .	6
Authors' Addresses . . . . .	7

## 1. Introduction

On a multi-access LAN such as an Ethernet, one of the PIM routers is elected as a DR. The PIM DR has two roles in the PIM-SM protocol. On the first hop network, the PIM DR is responsible for registering an active source with the Rendezvous Point (RP) if the group is operating in PIM-SM. On the last hop LAN, the PIM DR is responsible for tracking local multicast listeners and forwarding to these listeners if the group is operating in PIM-SM.

Consider the following last hop LAN in Figure 1:

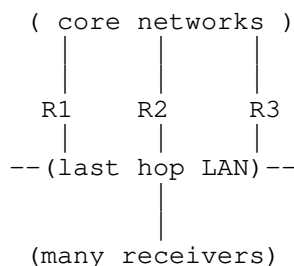


Figure 1: Last Hop LAN

Assume R1 is elected as the Designated Router. According to [RFC4601], R1 will be responsible for forwarding traffic to that LAN on behalf of any local member. In addition to keeping track of IGMP and MLD membership reports, R1 is also responsible for initiating the creation of source and/or shared trees towards the senders or the RPs.

There are multiple reasons for why network could potentially trigger DR re-election. Some of the reasons are

1. R1 going down
2. Access interface towards shared LAN going down
3. Config changed with lower DR priority

When any of above network event occurs, PIM DR re-election would be triggered. When a new DR is elected in shared LAN, new DR would be responsible to build a multicast tree towards source / RP. There are some cases, where traffic is crucial and the operator wants to have minimum traffic loss with DR failure. To address this requirement, this draft introduces a backup DR election procedure which would minimize traffic loss during PIM DR failure.

## 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] .

BDR - PIM Backup DR

With respect to PIM, this document follows the terminology that has been defined in [RFC4601] .

### 3. Applicability and deviation from draft PIM DR Improvement

[I-D.ietf-pim-dr-improvement] defines procedure to solve same problem which was stated in the introduction section of this draft.

[I-D.ietf-pim-dr-improvement] introduces new PIM Hello options for election of backup PIM DR.

This draft provides mechanism to elect BDR without using any new PIM Hello.

### 4. Protocol Specification

#### 4.1. PIM Backup DR (BDR) election procedure

[RFC7761] defines procedure for PIM DR election. PIM DR is elected on interface "I" among all PIM routers for which "I" has received PIM Hello. BDR election follows the exact same procedure and the second best PIM DR on shared LAN to be chosen as BDR on interface "I"

BDR would perform each of the responsibility of PIM DR except it would not forward traffic on shared LAN.

#### 4.2. Existing PIM DR failure

When PIM DR fails, PIM DR re-election is triggered on shared LAN. Since BDR is second best DR in LAN, it MUST take over immediately and MUST start forwarding multicast traffic on shared LAN.

Again on a shared LAN, new BDR would be elected. and current BDR would be the new DR.

#### 4.3. Existing PIM BDR failure

When an existing PIM BDR fails, the shared LAN MUST have BDR re-election using the DR election procedure from [RFC7761].

#### 4.4. New PIM Router addition in network

When a new PIM router is added in shared LAN, It could be either one of the below defined roles.

##### 4.4.1. New PIM router eligible to be PIM DR on shared LAN

When a new PIM router is added in a shared LAN and has the highest PIM DR priority configured, if a new router starts propagating its configured DR priority right away, the existing PIM DR would give up its role. Then there would be potential traffic loss till the new DR

learns about membership states and builds a multicast tree to the source or RP.

To avoid any such traffic loss situation, new PIM router SHOULD send a PIM Hello with priority 0. After 2 (default value, SHOULD have way to configure) PIM Hello interval or IGMP Query Interval (Which ever is higher) it SHOULD start propagating its original configured DR priority.

Even though a new PIM router propagating its priority as 0, it MUST start building a multicast tree towards source / RP, This is So that traffic loss could be minimized once it starts sending Hello with configured DR priority.

For a brief amount of time, there would be multiple copies of flows present in the multicast core, but a user SHOULD be able to configure whether to send hello with 0 priority or a configured priority. Depending on the application tolerance (Traffic loss Vs Extra traffic in core) the operator can choose option whichever is suitable for network.

After a PIM Hello or IGMP Query interval, the network would get stable with only one DR and one BDR.

#### 4.4.2. New PIM router eligible to be PIM BDR on shared LAN

It SHOULD follow the exact same procedure defined in the previous section.

#### 4.4.3. New PIM router is not eligible to be PIM DR or BDR on shared LAN

First a PIM Hello MUST be sent with priority 0. Once it has gotten Hello from other PIM neighbors, it knows that it is not eligible to be PIM DR or BDR. It MUST send configured PIM DR priority immediately. It MUST not wait for next hello interval.

#### 4.5. Initial case, All new PIM router coming up in shared LAN

In this case, initially each of the PIM routers would send Hellos with priorities of 0. If a PIM router receives all Hellos with priorities 0, it MUST send out a Hello with a configured PIM DR priority. Since it is initial startup case, it would take up to one Hello interval to converge.



#### 4.6. Benefit

1. Easy to implement as it uses an existing PIM procedure to elect DR.
2. Does not introduce any new Hello option

#### 5. Compatibility

#### 6. Manageability Considerations

#### 7. IANA Considerations

#### 8. Security Considerations

#### 9. Acknowledgement

The author would like to thank Stig Venaas, Tharak Abraham, Anish Kachinthaya, Anvitha Kachinthaya for helping with original idea.

#### 10. Normative References

- [I-D.ietf-pim-dr-improvement]  
Zhang, Z., hu, f., Xu, B., and m. mishra, "PIM DR Improvement", draft-ietf-pim-dr-improvement-04 (work in progress), December 2017.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC4601] Fenner, B., Handley, M., Holbrook, H., and I. Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", RFC 4601, DOI 10.17487/RFC4601, August 2006, <<https://www.rfc-editor.org/info/rfc4601>>.
- [RFC7761] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I., Parekh, R., Zhang, Z., and L. Zheng, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", STD 83, RFC 7761, DOI 10.17487/RFC7761, March 2016, <<https://www.rfc-editor.org/info/rfc7761>>.

Authors' Addresses

Mankamana Mishra  
Cisco Systems  
821 Alder Drive,  
MILPITAS, CALIFORNIA 95035  
UNITED STATES

Email: [mankamis@cisco.com](mailto:mankamis@cisco.com)

Sridhar Santhanam  
Cisco Systems  
821 Alder Drive,  
MILPITAS, CALIFORNIA 95035  
UNITED STATES

Email: [sridsant@cisco.com](mailto:sridsant@cisco.com)

Aravind Paramasivam  
Cisco Systems  
821 Alder Drive,  
MILPITAS, CALIFORNIA 95035  
UNITED STATES

Email: [arparama@cisco.com](mailto:arparama@cisco.com)

Joseph Goh  
Cisco Systems  
SINGAPORE

Email: [hocgoh@cisco.com](mailto:hocgoh@cisco.com)

Gyan S. Mishra  
Verizon Communications Inc. (VZ)  
13101 Columbia Pike FDC1 Rm 304-D  
Silver Spring MD 20904  
UNITED STATES

Email: [gyan.s.mishra@verizon.com](mailto:gyan.s.mishra@verizon.com)

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: January 3, 2019

Mankamana. Mishra  
Stig. Venaas  
Cisco Systems  
Mahesh. Sivakumar  
juniper networks  
Zheng(Sandy). Zhang  
ZTE Corporation  
Mikael. Abrahamsson  
July 2, 2018

PIM Designated Router graceful shutdown  
draft-mankamana-pim-graceful-dr-shutdown-00

Abstract

On a multi-access network, one of the PIM routers is elected as a Designated Router (DR). On the last hop LAN, the PIM DR is responsible for tracking local multicast listeners and forwarding traffic to these listeners if the group is operating in PIM-SM. In case of a network maintenance, where we want to bring down the current DR, there is currently no way to gracefully handover the PIM DR role to a new DR on the shared LAN. In this document, we propose a modification to the PIM-SM protocol that allows PIM DR to gracefully shutdown or go down for maintenance. We also provide a procedure for PIM DR to gracefully handover its role to a new PIM DR in the network.

Status of This Memo

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

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

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

This Internet-Draft will expire on January 3, 2019.

## Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

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

## Table of Contents

1. Introduction . . . . .	2
2. Terminology . . . . .	3
3. Protocol Specification . . . . .	4
3.1. Proposed Mechanism . . . . .	4
3.2. Impact on the network . . . . .	4
3.2.1. Every PIM router supports the new specification on the shared LAN . . . . .	4
3.2.2. Hybrid shared LAN, some of PIM router does not support specification . . . . .	5
4. PIM Hello option . . . . .	5
5. IANA Considerations . . . . .	5
6. Security Considerations . . . . .	5
7. Acknowledgement . . . . .	5
8. Contributors . . . . .	6
9. References . . . . .	6
9.1. Normative References . . . . .	6
9.2. Informative References . . . . .	6
Authors' Addresses . . . . .	7

## 1. Introduction

On a multi-access LAN such as an Ethernet, one of the PIM routers is elected as a DR. The PIM DR represents the LAN segment/broadcast domain in the PIM topology tree and has two roles to play in the PIM-SM protocol. For sources connected to the segment, the PIM DR is responsible for registering one or more active sources with the Rendezvous Point (RP) if the group is operating in PIM-SM. In addition, on the last hop LAN, the PIM DR is responsible for tracking local multicast listeners and forwarding data traffic to these listeners if the group is operating in PIM-SM.

Consider the following last hop LAN in Figure 1:

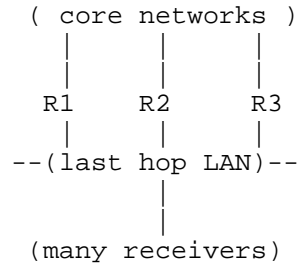


Figure 1: Last Hop LAN

Assume R1 is elected as the Designated Router. According to [RFC4601], R1 will be responsible for forwarding traffic to that LAN on behalf of any local members. In addition to keeping track of IGMP and MLD membership reports, R1 is also responsible for initiating the creation of source and/or shared trees towards the sources or the RPs.

If R1 needs to go on planned maintenance, the current approach is to lower the DR priority which would make sure that another PIM router on the LAN gets elected as the new DR and starts forwarding multicast traffic.

With this approach, R1 gives away DR role as soon as new priority is configured and a new PIM DR (lets assume R3) starts building a multicast tree and starts forwarding multicast traffic on the LAN. However, this could cause traffic disruption for the duration it takes for R3 to build the upstream multicast tree.

This draft defines a mechanism in the PIM protocol to handover DR role gracefully and as a result minimize traffic disruption.

## 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] .

With respect to PIM, this document follows the terminology that has been defined in [RFC4601] and [RFC7761] . Many places this draft would refer to PIM RFC [RFC4601] but it MUST be considered [RFC7761] as well.

### 3. Protocol Specification

In this draft, we define a new hello option to enable the graceful handover of a DR during planned maintenance. In Section 3.1, we describe the proposed mechanism. In Section 3.2, we evaluate the impact of the mechanism on the network under different conditions. Section 4 describes the proposed hello option.

#### 3.1. Proposed Mechanism

1. In Figure-1, assume that R1 is current PIM DR that needs to go on planned maintenance. R1 MUST send out a PIM Hello with option described in Section 4. The DR Priority MUST be set to 0. R1 MUST also set its assert metric to (PIM\_ASSERT\_INFINITY - 1)
2. The PIM assert metric modification would make sure that R1 does not become an assert winner
3. Sending DR priority as 0 would make sure to have default transition in case new DR does not support the new specification
4. The current PIM DR (R1 here) MUST not stop forwarding traffic to intended receivers unless it starts getting duplicate flows from newly elected PIM DR.
5. A failsafe timer SHOULD be used to stop forwarding multicast traffic towards receiver. It SHOULD be set to at least two PIM Hello intervals. But it SHOULD also be a configurable value.

#### 3.2. Impact on the network

This section covers impact of PIM hello with Section 4 option

##### 3.2.1. Every PIM router supports the new specification on the shared LAN

1. In Figure-1, if each of the PIM routers on shared LAN supported this specification, new DR election would be done as per [RFC4601]
2. The newly elected DR MUST start building the multicast tree towards the source/RP. It MUST start fail safe timer (default value 2 PIMHello interval) and MUST not generate a data driven assert. Once the timer expires, it can move back to the default assert mechanism. The reason to avoid an assert is to allow the old PIM DR on LAN to forward multicast traffic until such time the new DR is completely ready to forward multicast traffic.

3. It MUST forward multicast flow to receivers as soon as it gets the multicast flow from the source/RP
- 3.2.2. Hybrid shared LAN, some of PIM router does not support specification

There are two cases to consider,

1. If the new DR supports this specification, it would follow Section 3.1
  2. If the new DR does not support this specification, there is no need for any special handling as the new DR would take over as it does today. It would assert as soon as it gets elected as DR and the old DR would become the assert loser as it had already adjusted its assert metric to PIM\_ASSERT\_INFINITY - 1
4. PIM Hello option

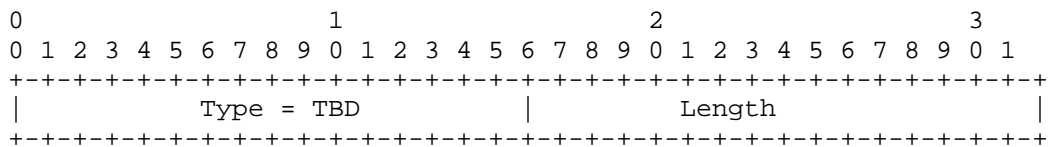


Figure 2: Graceful DR handoff Hello Option

where

Type : DR Graceful handoff

Length: 2

#### 5. IANA Considerations

A new PIM Hello option is TBD..

#### 6. Security Considerations

Security of the new PIM Hello Options is only guaranteed by the security of PIM Hello message, so the security considerations for PIM Hello messages as described in PIM-SM [RFC4601] apply here.

#### 7. Acknowledgement

## 8. Contributors

In addition to the authors listed on the front page, the following co-authors have also contributed to original idea.

Krishna Muddenahally Ananthamurthy

Cisco Systems

Sameer Gulrajani

Cisco systems

Rishabh Parekh

Cisco systems

## 9. References

### 9.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC4601] Fenner, B., Handley, M., Holbrook, H., and I. Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", RFC 4601, DOI 10.17487/RFC4601, August 2006, <<https://www.rfc-editor.org/info/rfc4601>>.
- [RFC6395] Gulrajani, S. and S. Venaas, "An Interface Identifier (ID) Hello Option for PIM", RFC 6395, DOI 10.17487/RFC6395, October 2011, <<https://www.rfc-editor.org/info/rfc6395>>.
- [RFC7761] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I., Parekh, R., Zhang, Z., and L. Zheng, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", STD 83, RFC 7761, DOI 10.17487/RFC7761, March 2016, <<https://www.rfc-editor.org/info/rfc7761>>.

### 9.2. Informative References

- [HELLO-OPT] IANA, "PIM Hello Options", IANA PIM-HELLO-OPTIONS, March 2007.



Authors' Addresses

Mankamana Mishra  
Cisco Systems  
821 Alder Drive,  
MILPITAS, CALIFORNIA 95035  
UNITED STATES

Email: [mankamis@cisco.com](mailto:mankamis@cisco.com)

Stig Venaas  
Cisco Systems  
821 Alder Drive,  
MILPITAS, CALIFORNIA 95035  
UNITED STATES

Email: [svenaas@cisco.com](mailto:svenaas@cisco.com)

Mahesh Sivakumar  
juniper networks  
1133 Innovation Way  
Sunnyvale, CALIFORNIA 94089  
UNITED STATES

Email: [sivakumar.mahesh@gmail.com](mailto:sivakumar.mahesh@gmail.com)

Zheng(Sandy) Zhang  
ZTE Corporation  
No. 50 Software Ave, Yuhuatai Distinct  
Nanjing  
China

Email: [zhang.zheng@zte.com.cn](mailto:zhang.zheng@zte.com.cn)

Mikael Abrahamsson

Email: [swmike@swm.pp.se](mailto:swmike@swm.pp.se)

PIM Working Group  
Internet-Draft  
Updates: 7761 (if approved)  
Intended status: Standards Track  
Expires: December 29, 2018

G. Mirsky  
ZTE Corp.  
J. Xiaoli  
ZTE Corporation  
June 27, 2018

Bidirectional Forwarding Detection (BFD) for Multi-point Networks and  
Protocol Independent Multicast - Sparse Mode (PIM-SM) Use Case  
draft-mirsky-pim-bfd-p2mp-use-case-02

## Abstract

This document discusses the use of Bidirectional Forwarding Detection (BFD) for multi-point networks to provide nodes that participate in Protocol Independent Multicast - Sparse Mode (PIM-SM) with the sub-second convergence. Optional extension to PIM-SM Hello, as specified in RFC 7761, to bootstrap point-to-multipoint BFD session. also defined in this document.

## Status of This Memo

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

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

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

This Internet-Draft will expire on December 29, 2018.

## Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

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

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
1.1. Conventions used in this document . . . . .	3
1.1.1. Terminology . . . . .	3
1.1.2. Requirements Language . . . . .	3
2. Problem Statement . . . . .	3
3. Applicability of p2mp BFD . . . . .	3
3.1. Multipoint BFD Encapsulation . . . . .	4
4. IANA Considerations . . . . .	5
5. Security Considerations . . . . .	5
6. Acknowledgments . . . . .	5
7. Normative References . . . . .	5
Authors' Addresses . . . . .	6

## 1. Introduction

Faster convergence in the control plane, in general, is beneficial and allows minimizing periods of traffic blackholing, transient routing loops and other scenarios that may negatively affect service data flow. That equally applies to unicast and multicast routing protocols.

[RFC7761] is the current specification of the Protocol Independent Multicast - Sparse Mode (PIM-SM) for IPv4 and IPv6 networks. Confirming implementation of PIM-SM elects a Designated Router (DR) on each PIM-SM interface. When a group of PIM-SM nodes is connected to shared-media segment, e.g. Ethernet, the one elected as DR is to act on behalf of directly connected hosts in context of the PIM-SM protocol. Failure of the DR impacts the quality of the multicast services it provides to directly connected hosts because the default failure detection interval for PIM-SM routers is 105 seconds. Introduction of Backup DR (BDR), proposed in [I-D.ietf-pim-dr-improvement] improves convergence time in the PIM-SM over shared-media segment but still depends on long failure detection interval.

Bidirectional Forwarding Detection (BFD) [RFC5880] had been originally defined to detect failure of point-to-point (p2p) paths - single-hop [RFC5881], multihop [RFC5883]. [I-D.ietf-bfd-multipoint] extends the BFD base specification [RFC5880] for multipoint and multicast networks, which precisely characterizes deployment scenarios for PIM-SM over LAN segment. This document demonstrates how point-to-multipoint (p2mp) BFD can enable faster detection of

PIM-SM router ailure and thus minimize multicast service disruption. The document also defines the extension to PIM-SM [RFC7761] to bootstrap a PIM-SM router to join in p2mp BFD session over shared-media link.

## 1.1. Conventions used in this document

### 1.1.1. Terminology

BFD: Bidirectional Forwarding Detection

BDR: Backup Designated Router

DR: Designated Router

p2mp: Pont-to-Multipoint

PIM-SM: Protocol Independent Multicast - Sparse Mode

### 1.1.2. Requirements Language

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

## 2. Problem Statement

[RFC7761] does not provide a method for fast, e.g. sub-second, failure detection of a neighbor PIM-SM router. BFD already has many implementations based on HW that are capable to support multiple sub-second session concurrently.

## 3. Applicability of p2mp BFD

[I-D.ietf-bfd-multipoint] may provide the efficient and scalable solution for the fast-converging environment that has head-tails relationships. Each such group presents itself as p2mp BFD session with its head being the root and other routers being tails of the p2mp BFD session. Figure 1 displays the new BFD Discriminator TLV [RFC7761] to bootstrap tail of the p2mp BFD session.

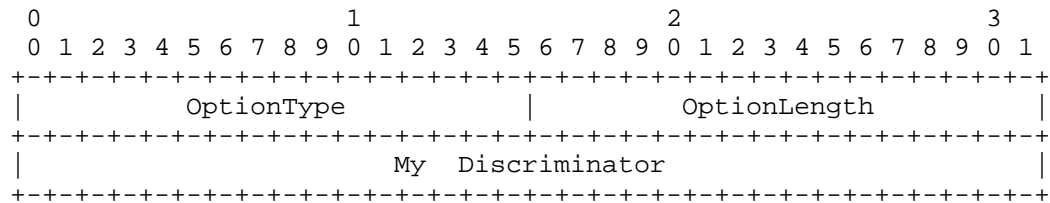


Figure 1: BFD Discriminator TLV to Bootstrap P2MP BFD session

where new fields are interpreted as:

OptionType is a value (TBA1) assigned by IANA Section 4 that identifies the TLV as BFD Discriminator TLV;

OptionLength value is always 4

My Discriminator - My Discriminator value allocated by the root of the p2mp BFD session.

If PIM-SM routers, that support this specification, are configured to use p2mp BFD for faster convergence, then the router to be monitored, referred to as 'head', MUST create BFD session MultipointHead, as defined in [I-D.ietf-bfd-multipoint]. The head MUST include BFD TLV in its PIM-Hello message and periodically transmit BFD control packets. Source IP address of the BFD control packet MUST be the same as the source IP address of the PIM-Hello with BFD TLV messages being transmitted by the head. The values of My Discriminator in the BFD control packet and My Discriminator field of the BFD TLV in PIM-Hello, transmitted by the head MUST be the same. When a PIM-SM router configured to monitor the head, referred to as 'tail', via p2mp BFD receives PIM-Hello packet with BFD TLV it MAY create p2mp BFD session as MultipointTail, as defined in [I-D.ietf-bfd-multipoint], and demultiplex p2mp BFD test session based on head's source IP address the My Discriminator value it learned from BFD Discriminator TLV. If the head ceased to include BFD TLV in its PIM-Hello message, tails MUST close the corresponding MultipointTail BFD session. If the tail detects MultipointHead failure it MUST remove the neighbor. If the failed head node was PIM-SM DR or BDR the tail MAY start DR Election process as specified in Section 4.3.2 [RFC7761] or in Section 4.1 [I-D.ietf-pim-dr-improvement] respectively.

### 3.1. Multipoint BFD Encapsulation

The MultipointHead of p2mp BFD session when transmitting BFD control packet:

MUST set TTL value to 1;

SHOULD use group address ALL-PIM-ROUTERS ('224.0.0.13' for IPv4 and 'ff02::d' for IPv6) as destination IP address

MAY use network broadcast address for IPv4 or link-local all nodes multicast group for IPv6 as the destination IP address;

MUST set destination UDP port value to 3784 when transmitting BFD control packets, as defined in [I-D.ietf-bfd-multipoint].

#### 4. IANA Considerations

IANA is requested to allocate a new OptionType value from PIM Hello Options registry according to:

Value Name	Length Number	Name Protocol	Reference
TBA	4	BFD Discriminator	This document

Table 1: BFD Discriminator option type

#### 5. Security Considerations

Security considerations discussed in [RFC7761], [RFC5880], and [I-D.ietf-bfd-multipoint], apply to this document.

#### 6. Acknowledgments

Authors cannot say enough to express their appreciation of comments and suggestions we received from Stig Venaas.

#### 7. Normative References

- [I-D.ietf-bfd-multipoint]  
 Katz, D., Ward, D., Networks, J., and G. Mirsky, "BFD for Multipoint Networks", draft-ietf-bfd-multipoint-18 (work in progress), June 2018.
- [I-D.ietf-pim-dr-improvement]  
 Zhang, Z., hu, f., Xu, B., and m. mishra, "PIM DR Improvement", draft-ietf-pim-dr-improvement-04 (work in progress), December 2017.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <<https://www.rfc-editor.org/info/rfc5880>>.
- [RFC5881] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop)", RFC 5881, DOI 10.17487/RFC5881, June 2010, <<https://www.rfc-editor.org/info/rfc5881>>.
- [RFC5883] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for Multihop Paths", RFC 5883, DOI 10.17487/RFC5883, June 2010, <<https://www.rfc-editor.org/info/rfc5883>>.
- [RFC7761] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I., Parekh, R., Zhang, Z., and L. Zheng, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", STD 83, RFC 7761, DOI 10.17487/RFC7761, March 2016, <<https://www.rfc-editor.org/info/rfc7761>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

## Authors' Addresses

Greg Mirsky  
ZTE Corp.

Email: [gregimirsky@gmail.com](mailto:gregimirsky@gmail.com)

Ji Xiaoli  
ZTE Corporation  
No.50 Software Avenue, Yuhuatai District  
Nanjing  
China

Email: [ji.xiaoli@zte.com.cn](mailto:ji.xiaoli@zte.com.cn)

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: August 31, 2018

V. Kamath  
R. Chokkanathapuram Sundaram  
Cisco Systems, Inc.  
February 27, 2018

PIM NULL Register packing  
draft-ramki-pim-null-register-packing-00

Abstract

In PIM-SM networks PIM registers are sent from the first hop router to the RP (Rendezvous Point) to signal the presence of Multicast source in the network. There are periodic PIM Null registers sent from first hop router to the RP to keep the state alive at the RP as long as the source is active. The PIM null register packet carry information about a single Multicast source and group. This document defines a standard to send multiple Multicast source and group information in a single pim null register packet and the interoperability between the PIM routers which do not understand the packet format with multiple Multicast source and group details.

Status of This Memo

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

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

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

This Internet-Draft will expire on August 31, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

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



carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
1.1. Conventions Used in This Document . . . . .	2
1.2. Terminology . . . . .	3
2. PIM Register Stop format with capability option . . . . .	3
3. New PIM Null register format . . . . .	4
4. New Packed PIM Register Stop format . . . . .	5
5. Protocol operation . . . . .	6
6. IANA Considerations . . . . .	7
7. Acknowledgments . . . . .	7
8. References . . . . .	7
8.1. Normative References . . . . .	7
8.2. Informative References . . . . .	8
Authors' Addresses . . . . .	8

## 1. Introduction

PIM Null registers are sent by First hop routers periodically for Multicast streams to keep the states active on the RP as long as the Multicast source is alive. As the number of multicast sources increase, the number of PIM null register packets that are sent increases at a given time. This results in more PIM packet processing at RP and FHR. The control plane policing(COPP), monitors the packets that gets processed by the control plane. Due to the high rate at which NULL registers are received at the RP, this can lead to COPP drops of Multicast PIM null register packets. This draft proposes a method to efficiently pack multiple PIM null registers and register stop into a single message as these packets anyway don't contain data. The draft also proposes interoperability with the routers that do not understand the new packet format.

### 1.1. Conventions Used in This Document

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].

## 1.2. Terminology

RP: Rendezvous Point

RPF: Reverse Path Forwarding

SPT: Shortest Path Tree

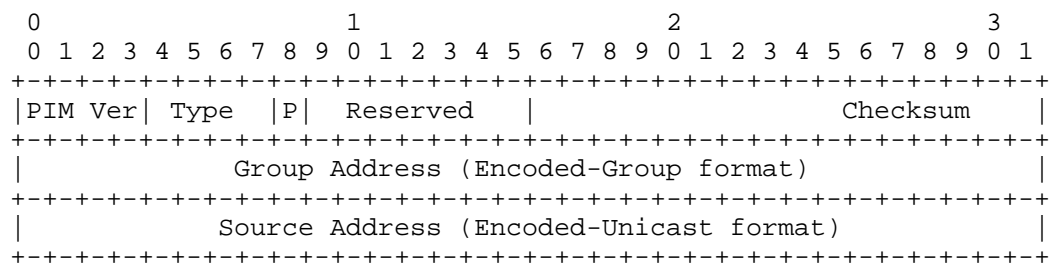
FHR: First Hop Router, directly connected to the source

LHR: Last Hop Router, directly connected to the receiver

## 2. PIM Register Stop format with capability option

A router (FHR) can decide to pack multiple NULL registers based on the capability received from the RP as part of Register Stop. This ensures compatibility with routers that don't support processing of the new format. The capability information can be indicated by the RP via the PIM register stop message sent to the FHR. Thus a FHR will switch to the new format only when it learns RP is capable of handling the packed null register messages. Conversely, a FHR that doesn't support the new format can continue generating the PIM NULL register the usual way since they don't check for the capability information present in the Register stop message. To exchange the capability information in the Register Stop message, the "reserved" field can be used to indicate this capability in those register stop messages. One bit of the reserved field is used to indicate the "packing" capability (P bit). The rest of the bits in the "Reserved" field will be retained for future use.

Figure 2: PIM Register Stop packet with capability option



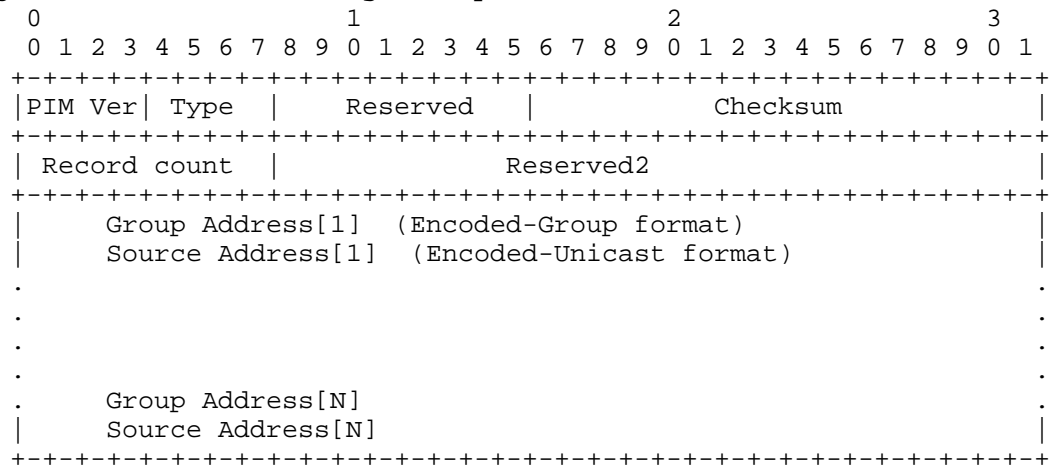
PIM Version, Reserved, Type, Checksum, Group Address, Source Address  
Same as RFC 7761 (Section 4.9.4)

P Capability bit used to indicate support for Packed NULL Register

### 3. New PIM Null register format

PIM null-register packet format is enhanced to include the count of the number of null-register records and pack multiple null-register records in the same packet. Currently the data part in the NULL register packet is a dummy IPv4 header which carries the source and group information and the other fields are unused. To indicate that the null register is in a new format the "Type" field in the PIM register packet format is used. To indicate the number of null register records a new field "record count" is introduced which can hold 8 bit value (max 255 records can be packed) which can be based on MTU. Even though null registers are supposed to be sent exactly every 60s, its fine to send a null register earlier, so as to merge the registers. When one register is sent, multiple registers can be packed together which are close enough in time.

Figure 1: New PIM NULL Register packet format



PIM Version, Reserved, Checksum  
Same as RFC 7761 (Section 4.9.3)

#### Type

The new packed NULL Register type value TBD

#### Record count

The count of the number of packed NULL register records.  
A record consists of Group and Source Address

#### Group Address

IP address of the Multicast Group

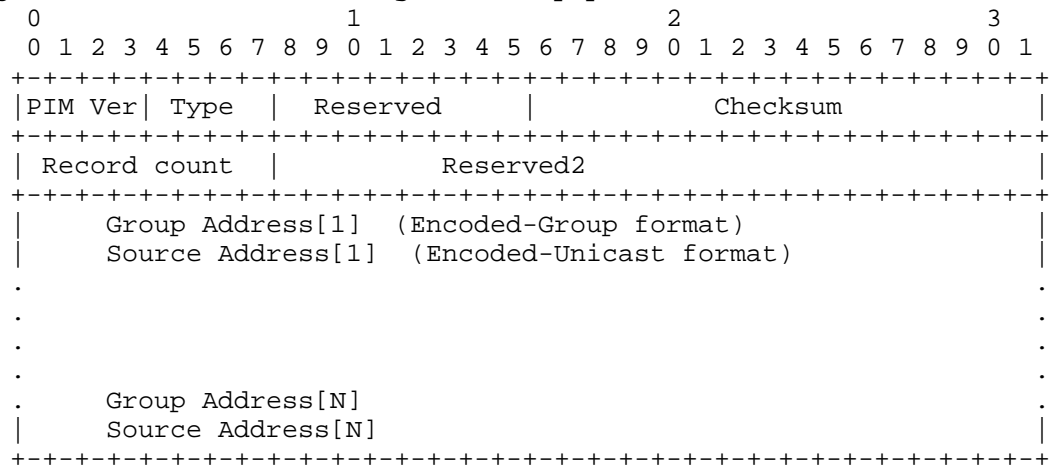
#### Source Address

IP Address of the Multicast Source

#### 4. New Packed PIM Register Stop format

The PIM register stop can be optimized to include multiple multicast group and source information. The Record count can indicate the number of S,G records that are packed and the Type value is used to indicate the new format.

Figure 3: New PIM Packed Register Stop packet formats



PIM Version, Reserved, Checksum  
Same as RFC 7761 (Section 4.9.3)

#### Type

The new packed Register Stop type value TBD

#### Record count

The count of the number of packed register stop records.  
A record consists of Group and Source Address

#### Group Address

IP address of the Multicast Group

#### Source Address

IP Address of the Multicast Source

## 5. Protocol operation

The following combinations exist -

FHR and RP both support the new PIM Register formats -

- a. FHR sends the PIM register towards the RP when a new source is detected
- b. RP sends a modified register stop towards the FHR that includes capability information by setting the P bit (Figure 2)
- c. Based on the receipt of the modified Register Stop, FHR will start packing of NULL registers using the new packed register format (Figure 1)
- d. RP processes the NULL registers and can generate Register Stop messages by packing multiple S,Gs towards the same FHR (Figure 3)

FHR supports but RP doesn't support new PIM Register formats-

- a. FHR sends the PIM register towards the RP
- b. RP sends a normal register stop without any capability information
- c. FHR then sends NULL registers in the old format

RP supports but FHR doesn't support the new PIM Register formats-

- a. FHR sends the PIM register towards the RP
- b. RP sends a modified register stop towards the FHR that includes capability information
- c. Since FHR doesn't support the new format, it sends NULL registers in the old format

## 6. IANA Considerations

This document requires the assignment of 2 new PIM message types for the packed pim register and pim register stop.

## 7. Acknowledgments

The authors would like to thank Stig Venaas and Umesh Dudani for contributing to the original idea and also their very helpful comments on the draft.

## 8. References

### 8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC7761] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I., Parekh, R., Zhang, Z., and L. Zheng, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", STD 83, RFC 7761, DOI 10.17487/RFC7761, March 2016, <<https://www.rfc-editor.org/info/rfc7761>>.

## 8.2. Informative References

[RFC3973] Adams, A., Nicholas, J., and W. Siadak, "Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised)", RFC 3973, DOI 10.17487/RFC3973, January 2005, <<https://www.rfc-editor.org/info/rfc3973>>.

### Authors' Addresses

Vikas Ramesh Kamath  
Cisco Systems, Inc.  
Tasman Drive  
San Jose CA 95134  
USA

Email: [vikkamat@cisco.com](mailto:vikkamat@cisco.com)

Ramakrishnan Cokkanathapuram Sundaram  
Cisco Systems, Inc.  
Tasman Drive  
San Jose CA 95134  
USA

Email: [ramaksun@cisco.com](mailto:ramaksun@cisco.com)

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: October 14, 2019

V. Kamath  
VMware  
R. Chokkanathapuram Sundaram  
R. Banthia  
Cisco Systems, Inc.  
April 12, 2019

PIM Null register packing  
draft-ramki-pim-null-register-packing-03

Abstract

In PIM-SM networks PIM registers are sent from the first hop router to the RP (Rendezvous Point) to signal the presence of Multicast source in the network. There are periodic PIM Null registers sent from first hop router to the RP to keep the state alive at the RP as long as the source is active. The PIM Null register packet carries information about a single Multicast source and group. This document defines a standard to send multiple Multicast source and group information in a single pim Null register packet and the interoperability between the PIM routers which do not understand the packet format with multiple Multicast source and group details.

Status of This Memo

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

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

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

This Internet-Draft will expire on October 14, 2019.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents



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

## Table of Contents

1. Introduction . . . . .	2
1.1. Conventions Used in This Document . . . . .	2
1.2. Terminology . . . . .	3
2. PIM Register Stop format with capability option . . . . .	3
3. New PIM Null register message . . . . .	4
4. New PIM Register Stop message format . . . . .	4
5. Protocol operation . . . . .	5
6. PIM Anycast RP considerations . . . . .	6
7. IANA Considerations . . . . .	6
8. Acknowledgments . . . . .	6
9. References . . . . .	6
9.1. Normative References . . . . .	7
9.2. Informative References . . . . .	7
Authors' Addresses . . . . .	7

## 1. Introduction

PIM Null registers are sent by First hop routers periodically for Multicast streams to keep the states active on the RP as long as the Multicast source is alive. As the number of multicast sources increases, the number of PIM Null register packets that are sent increases at a given time. This results in more PIM packet processing at RP and FHR. The control plane policing (COPP), monitors the packets that gets processed by the control plane. Due to the high rate at which Null registers are received at the RP, this can lead to COPP drops of Multicast PIM Null register packets. This draft proposes a method to efficiently pack multiple PIM Null registers and register stop into a single message as these packets anyway don't contain data. The draft also proposes interoperability with the routers that do not understand the new packet format.

### 1.1. Conventions Used in This Document

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].

## 1.2. Terminology

RP: Rendezvous Point

RPF: Reverse Path Forwarding

SPT: Shortest Path Tree

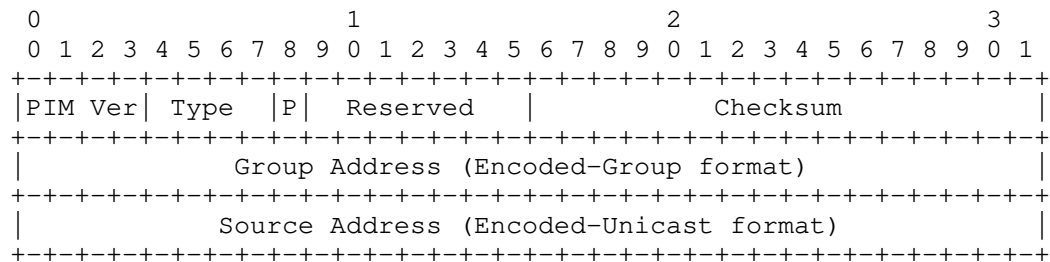
FHR: First Hop Router, directly connected to the source

LHR: Last Hop Router, directly connected to the receiver

## 2. PIM Register Stop format with capability option

A router (FHR) can decide to pack multiple Null registers based on the capability received from the RP as part of Register Stop. This ensures compatibility with routers that don't support processing of the new format. The capability information can be indicated by the RP via the PIM register stop message sent to the FHR. Thus a FHR will switch to the new format only when it learns RP is capable of handling the packed Null register messages. Conversely, a FHR that doesn't support the new format can continue generating the PIM Null register the current way. To exchange the capability information in the Register Stop message, the "reserved" field can be used to indicate this capability in those register stop messages. One bit of the reserved field is used to indicate the "packing" capability (P bit). The rest of the bits in the "Reserved" field will be retained for future use.

Figure 1: PIM Register Stop message with capability option



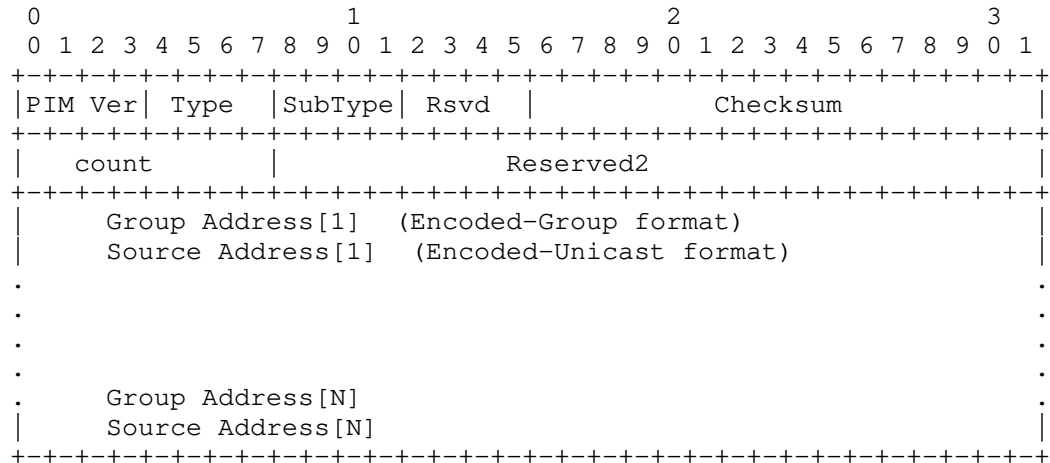
PIM Version, Reserved, Type, Checksum, Group Address, Source Address  
Same as RFC 7761 (Section 4.9.4)

P Capability bit used to indicate support for Packed Null Register

### 3. New PIM Null register message

New PIM Null register message format includes a count to indicate the number of Null register records in the message.

Figure 2: New PIM Null Register message format



PIM Version, Reserved, Checksum  
Same as RFC 7761 (Section 4.9.3)

Type, SubType  
The new packed Null Register Type and SubType values TBD

count  
The count of the number of packed Null register records.  
A record consists of Group and Source Address

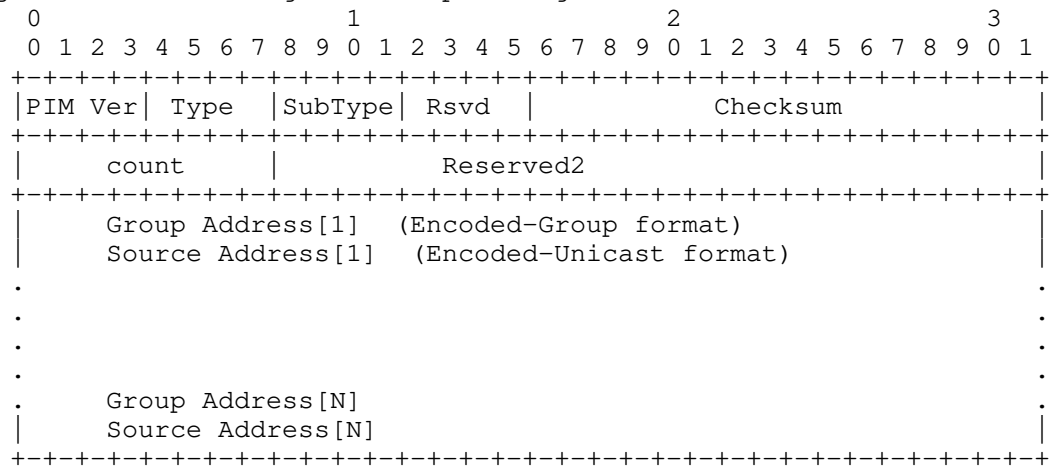
Group Address  
IP address of the Multicast Group

Source Address  
IP Address of the Multicast Source

### 4. New PIM Register Stop message format

The new PIM register stop is message includes a count to indicate the number of records that are present in the message.

Figure 3: New PIM Register Stop message format



PIM Version, Reserved, Checksum  
Same as RFC 7761 (Section 4.9.3)

Type  
The new Register Stop Type and SubType values TBD

Record count  
The count of the number of packed register stop records.  
A record consists of Group and Source Address

Group Address  
IP address of the Multicast Group

Source Address  
IP Address of the Multicast Source

## 5. Protocol operation

The following combinations exist -

FHR and RP both support the new PIM Register formats -

- a. FHR sends the PIM register towards the RP when a new source is detected
- b. RP sends a modified register stop towards the FHR that includes capability information by setting the P bit (Figure 2)
- c. Based on the receipt of new Register Stop, FHR will start packing of Null registers using the new packed register format (Figure 1)
- d. RP processes the new Null register message and can generate new register Stop messages by packing multiple S,Gs towards the same FHR (Figure 3)

FHR supports but RP doesn't support new PIM Register formats-

- a. FHR sends the PIM register towards the RP
- b. RP sends a normal register stop without any capability information
- c. FHR then sends Null registers in the old format

RP supports but FHR doesn't support the new PIM Register formats-

- a. FHR sends the PIM register towards the RP
- b. RP sends a modified register stop towards the FHR that includes capability information
- c. Since FHR doesn't support the new format, it sends Null registers in the old format

## 6. PIM Anycast RP considerations

The new PIM register format should be enabled only if its supported by all PIM anycast RP members in the RP set for the RP address.

## 7. IANA Considerations

This document requires the assignment of 2 new PIM message types for the packed pim register and pim register stop.

## 8. Acknowledgments

The authors would like to thank Stig Venaas and Umesh Dudani for contributing to the original idea and also their very helpful comments on the draft.

## 9. References

## 9.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC7761] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I., Parekh, R., Zhang, Z., and L. Zheng, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", STD 83, RFC 7761, DOI 10.17487/RFC7761, March 2016, <<https://www.rfc-editor.org/info/rfc7761>>.

## 9.2. Informative References

- [RFC3973] Adams, A., Nicholas, J., and W. Siadak, "Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised)", RFC 3973, DOI 10.17487/RFC3973, January 2005, <<https://www.rfc-editor.org/info/rfc3973>>.

## Authors' Addresses

Vikas Ramesh Kamath  
VMware  
3401 Hillview Ave  
Palo Alto CA 94304  
USA

Email: [vkamath@vmware.com](mailto:vkamath@vmware.com)

Ramakrishnan Chokkanathapuram Sundaram  
Cisco Systems, Inc.  
Tasman Drive  
San Jose CA 95134  
USA

Email: [ramaksun@cisco.com](mailto:ramaksun@cisco.com)

Raunak Banthia  
Cisco Systems, Inc.  
Tasman Drive  
San Jose CA 95134  
USA

Email: [rbanthia@cisco.com](mailto:rbanthia@cisco.com)

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: December 30, 2018

R. Chokkanathapuram  
R. Banthia  
Cisco Systems, Inc.  
June 28, 2018

PIM Router Graceful Insertion and Removal  
draft-raunak-pim-gir-support-00

Abstract

Graceful Insertion and Removal (GIR) of routers is often adopted by many network administrators as an alternative to ISSU. This document discusses various scenarios, requirements and possible solutions to make PIM gracefully shut down and isolate the multicast router with minimal network disruption when a router goes through maintenance procedures.

Status of This Memo

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

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

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

This Internet-Draft will expire on December 30, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Terminology . . . . .	2
3. Applicability . . . . .	2
4. Graceful RPF change . . . . .	2
5. GIR removal procedure for a PIM router . . . . .	3
6. GIR insertion procedure for a PIM router . . . . .	4
7. Compatibility . . . . .	5
8. PIM GIR TLV . . . . .	5
9. IANA Considerations . . . . .	5
10. Security Considerations . . . . .	5
11. Normative References . . . . .	5
Authors' Addresses . . . . .	6

## 1. Introduction

When a network administrator wishes to perform maintenance activity on a router, a system maintenance mode command need to be configured. This isolates the router from the network by gracefully shutting down various protocols running on the router. Multicast protocols will also require graceful migration in order to achieve minimal traffic disruption when a maintenance activity is performed on a router. This document proposes a possible solution to perform GIR with PIM routers gracefully.

## 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] .

With respect to PIM, this document follows the terminology that has been defined in [RFC7761]

## 3. Applicability

The proposed change described in this specification applies to PIM routers only.

## 4. Graceful RPF change

Multicast routing protocol uses Unicast reachability information to find unique Reverse Path Forwarding Neighbor (RPF). Any change in unicast routing triggers multicast RPF changes. Multicast flows need



to change the RPF in a graceful manner to have minimal or no disruption in traffic flow. To achieve graceful RPF change, PIM should not change RPF immediately following unicast routing change. PIM should join the new path and wait for the traffic to arrive on the new path before pruning the old path. Until the packets arrive on the new path, the packets are accepted and forwarded on the old path. Since we have not changed the RPF to new one, we would see RPF failures. The RPF failures on the new path will indicate that the flow is available on the new path, upon which the RPF for the flows will be changed from old to new. Using this method, we will be able to achieve non-stop forwarding of multicast traffic thereby minimizing traffic disruption. The graceful RPF change, however, is not advisable in a normal RPF change scenario. This is because old path could be down due to link failures and the RPF change may take more time which could increase convergence time. Multicast flows can do a graceful RPF change in a GIR scenario since the flow will be available via the old path.

#### 5. GIR removal procedure for a PIM router

A multicast router undergoing a graceful insertion/removal must indicate the same to all the routers in PIM domain. This will ensure that all routers will gracefully change RPF for the multicast flows within the GIR window. This information needs to be propagated before the unicast metrics are altered by the GIR router. To achieve this, a PIM Flooding Mechanism message (PFM) [RFC8364] TLV is originated from the router undergoing GIR. The GIR details will be carried in the PFM TLV. This message is flooded periodically in the PIM domain and the RECOMMENDED interval to send this message is 60 secs. The propagation of this message will ensure that all the routers (in the PIM domain) knows the router undergoing GIR and can gracefully migrate flows from old path to new path when the unicast infinite metrics are advertised from the router undergoing GIR in the GIR window.

Procedure for PIM routers (GIR mode)-

1. The router undergoing GIR (GIR router) will send a PFM message with a new TLV option (GIR TLV) to all its PIM neighbors indicating that the router wishes to go to maintenance mode. The router could send more than one PFM message so that the loss of the PFM messages are minimized. The value fields in the TLV will be populated with the following hold-time values -
  1. graceful-rpf-start - This value indicates the seconds until the PIM router will start doing graceful RPF change

2. graceful-rpf-stop - This value indicates the seconds after which the PIM router will stop doing graceful RPF change. The time period between the graceful-rpf-start and graceful-rpf-stop indicates the duration during which the routers in the network will do graceful RPF changes for multicast flow.
2. Upon receipt of the PFM message with GIR TLV option from GIR router, PIM neighbors will compute the RPF towards the originator ip address in the incoming PFM message. If the RPF matches with the interface where this message is received, the router will perform the following, else pim will just drop the message.
  1. Start a timer for graceful-rpf-start, if not already started. Once the graceful-rpf-start timer expires, the routers will be in graceful RPF change mode until the hold-time period stop. During this time period, router will do Graceful RPF change (as described in section above) as soon as it receives unicast metric change. The unicast infinite metric change from the router undergoing GIR has to be sequenced between the advertised graceful-rpf-start and graceful-rpf-stop.
  2. Forward PFM message with the GIR TLV to its immediate PIM neighbors. This is to propagate PFM message with the GIR TLV to all the routers in the PIM domain.

The above method could be further optimized if PFM messages could carry the source prefixes of the multicast states present on the GIR router. The routers receiving the PFM GIR message can examine the source prefixes and move to Graceful RPF change mode only if the routers have the multicast state for the source prefixes. With this, not all routers needs to do Graceful RPF change. This will ensure Graceful RPF change occurs only on the routers that are impacted by the router undergoing GIR.

#### 6. GIR insertion procedure for a PIM router

The same method as described above will be used to gracefully insert a router with no traffic disruption after system maintenance mode. When a router is inserted into network, it will send the PFM GIR message with the graceful-rpf-start timer value set to 0, and graceful-rpf-stop timer set to seconds until Graceful RPF stop. During this window, the inserted router will start bringing up the unicast protocols. Every router in the PIM domain will examine the PFM message and do a Graceful RPF change for the window specified in the message. Once the GIR router is inserted and fully operational, it should send at least one message with both timers (graceful-rpf-start and graceful-rpf-stop) set to 0

## 7. Compatibility

The router undergoing GIR must set the Transitive bit to 1 in the PFM message so that when a router that doesn't support GIR, receives a PFM GIR TLV, it will forward the message to the PIM neighbors.

## 8. PIM GIR TLV

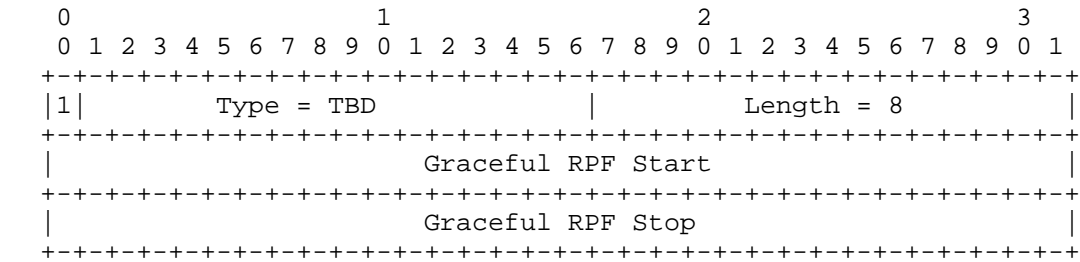


Figure 1: PIM GIR TLV

where

Type : TBD

Length : The length of the value in octets

Graceful RPF Start : Timer value in seconds until PIM router start doing graceful RPF change

Graceful RPF Stop : Timer value in seconds after which the PIM router will stop doing graceful RPF change

## 9. IANA Considerations

A new PIM PFM option is TBD for GIR.

## 10. Security Considerations

Security of the new PFM TLV is only guaranteed by the security of PFM message, so the security considerations for PFM message as described in [RFC8364] apply here.

## 11. Normative References

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

[RFC7761] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I., Parekh, R., Zhang, Z., and L. Zheng, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", STD 83, RFC 7761, DOI 10.17487/RFC7761, March 2016, <<https://www.rfc-editor.org/info/rfc7761>>.

[RFC8364] Wijnands, IJ., Venaas, S., Brig, M., and A. Jonasson, "PIM Flooding Mechanism (PFM) and Source Discovery (SD)", RFC 8364, DOI 10.17487/RFC8364, March 2018, <<https://www.rfc-editor.org/info/rfc8364>>.

#### Authors' Addresses

Ramakrishnan Chokkanathapuram  
Cisco Systems, Inc.  
Tasman Drive  
San Jose CA 95134  
United States of America

Email: [ramaksun@cisco.com](mailto:ramaksun@cisco.com)

Raunak Banthia  
Cisco Systems, Inc.  
Tasman Drives  
San Jose CA 95134  
United States of America

Email: [rbanthia@cisco.com](mailto:rbanthia@cisco.com)

Network Working Group  
Internet-Draft  
Updates: 3973, 5015, 6754, 7761, 8364  
(if approved)  
Intended status: Standards Track  
Expires: December 29, 2018

S. Venaas  
Cisco Systems, Inc.  
A. Retana  
Huawei R&D USA  
June 27, 2018

PIM reserved bits and type space extension  
draft-venaas-pim-reserved-bits-01

Abstract

The currently defined PIM version 2 messages share a common message header format. The common header definition contains eight reserved bits. This document specifies how these bits may be used by individual message types, and creates a registry containing the per message type usage. This document also extends the PIM type space by defining three new message types. For each of the new types, four of the previously reserved bits are used to form an extended type range.

This document Updates RFC7761 and RFC3973 by defining the use of the currently Reserved field in the PIM common header. This document further updates RFC7761 and RFC3973, along with RFC5015, RFC6754 and RFC8364, by specifying the use of the currently Reserved bits for each PIM message.

Status of This Memo

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

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

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

This Internet-Draft will expire on December 29, 2018.

## Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

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

## Table of Contents

1. Introduction . . . . .	2
2. Conventions used in this document . . . . .	3
3. PIM header common format . . . . .	3
4. Flag Bit definitions . . . . .	3
4.1. Flag Bits for Type 4 (Bootstrap) . . . . .	4
4.2. Flag Bits for Type 10 (DF Election) . . . . .	4
4.3. Flag Bits for Type 12 (PFM) . . . . .	4
4.4. Flag Bits for Type 13 (Type Space Extension) . . . . .	4
4.5. Flag Bits for Type 14 (Type Space Extension) . . . . .	4
4.6. Flag Bits for Type 15 (Type Space Extension) . . . . .	4
5. PIM Type Space Extension . . . . .	5
6. Security Considerations . . . . .	5
7. IANA considerations . . . . .	5
8. References . . . . .	6
8.1. Normative References . . . . .	6
8.2. Informative References . . . . .	7
Authors' Addresses . . . . .	7

## 1. Introduction

The currently defined PIM version 2 messages share a common message header format defined in the PIM Sparse Mode [RFC7761] and Dense Mode [RFC3973] specifications. The common header definition contains eight reserved bits. The message types defined in these documents all use this common header. However, several messages already make use of one or more bits, including the Bootstrap [RFC5059], DF-Election [RFC5015], and PIM Flooding Mechanism (PFM) [RFC8364] messages. There is no document formally specifying that these bits are to be used per message type.

This document refers to the bits specified as Reserved in the common PIM header [RFC7761] [RFC3973] as PIM message type flag bits, or simply flag bits, and it specifies that they are to be separately used on a per message type basis. It creates a registry containing the the per message type usage. For a particular message type, the usage of the flag bits can be defined in the document defining the message type, or a new document that updates that document.

The PIM message types as defined in the PIM Sparse Mode [RFC7761] and Dense Mode [RFC3973] specifications are in the range from 0 to 15. That type space is almost exhausted. Message type 15 was reserved by [RFC6166] for type space extension. In Section 5, this document specifies the use of the flag bits for message types 13, 14 and 15 in order to extend the PIM type space. The registration procedure for the extended type space is the same as for the existing type space, and the existing PIM message type registry is updated to include the extended type space.

## 2. Conventions used in this document

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

## 3. PIM header common format

The common PIM header is defined in section 4.9 of [RFC7761] and section 4.7.1 of [RFC3973]. This document updates the definition of the Reserved field and refers to that field as PIM message type flag bits, or simply flag bits. The new common header format is as below.

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	2	3	4	5	6	7	8	9
PIM Ver										Type										Flags Bits										Checksum									

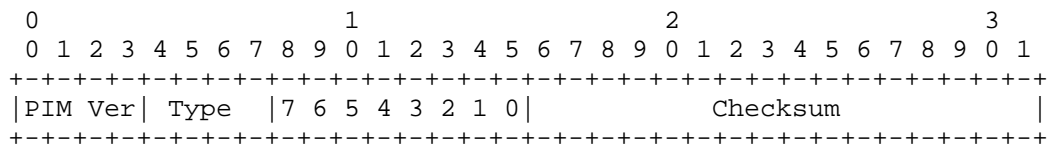
The Flags Bits field is defined in Section 4. All other fields remain unchanged.

## 4. Flag Bit definitions

Unless otherwise specified, all the flag bits for each PIM type are Reserved [RFC8126]. They MUST be set to zero on transmission, and they MUST be ignored upon receipt. The specification of a new PIM type, MUST indicate whether the bits should be treated differently.

Currently for the message types 0 (Hello), 1 (Register), 2 (Register Stop), 3 (Join/Prune), 5 (Assert), 6 (Graft), 7 (Graft-Ack), 8 (Candidate RP Advertisement), 9 (State Refresh) and 11 (ECMP Redirect), all flag bits are Reserved.

When defining flag bits it is helpful to have a well defined way of referring to a particular bit. The most significant of the flag bits, the bit immediately following the type field is referred to as bit 7. The least significant, the bit right in front of the checksum field is referred to as bit 0. This is shown in the diagram below.



#### 4.1. Flag Bits for Type 4 (Bootstrap)

PIM message type 4 (Bootstrap) [RFC5059] defines flag bit 7 as No-Forward. The usage of the bit is defined in that document. The remaining flag bits are Reserved.

#### 4.2. Flag Bits for Type 10 (DF Election)

PIM message type 10 (DF Election) [RFC5015] specifies that the four most significant flag bits (bits 4-7) are to be used as a sub-type. The remaining flag bits are currently Reserved.

### 4.3. Flag Bits for Type 12 (PFM)

PIM message type 12 (PFM) [RFC8364] defines flag bit 7 as No-Forward. The usage of the bit is defined in that document. The remaining flag bits are Reserved.

#### 4.4. Flag Bits for Type 13 (Type Space Extension)

This type and the flag bit usage is defined in Section 5.

#### 4.5. Flag Bits for Type 14 (Type Space Extension)

This type and the flag bit usage is defined in Section 5.

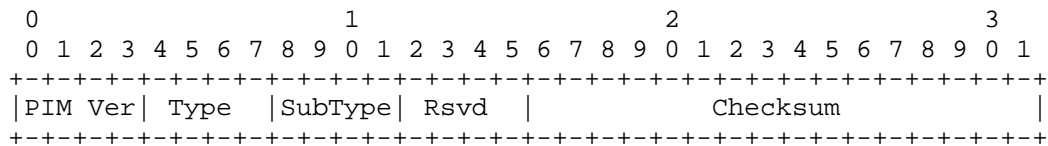
#### 4.6. Flag Bits for Type 15 (Type Space Extension)

This type and the flag bit usage is defined in Section 5.



## 5. PIM Type Space Extension

The type space defined by the existing PIM specifications is almost exhausted. This document defines types 13, 14 and 15 (Type Space Extension) allowing for 48 additional types by for each of the three types, using the four most significant flag bits (bits 4-7) as a new field to store the extended type. These types are referred to as types 13.0 to 13.15, 14.0 to 14.15 and 15.0 to 15.15 where the last number denotes the value stored in the new field. The remaining four flag bits (bits 0-3) are Reserved to be used by each extended type. The specification of a new PIM extended type MUST indicate whether the bits should be treated differently. The common header for the new types is shown in the diagram below. The "Type" field is set to 13, 14 or 15, and the extended type field "SubType" denotes the value after the dot.



## 6. Security Considerations

This document clarifies the use of the flag bits in the common PIM header and it extends the PIM type space. As such, there is no impact on security or changes to the considerations in [RFC7761] and [RFC3973].

## 7. IANA considerations

This document updates the PIM Message Types registry and also creates a PIM Message Type Flag Bits registry that shows which flag bits are defined for use by each of the PIM message types.

The following changes should be made to the existing PIM Message Types registry. For types 4 (Bootstrap) and 8 (Candidate RP Advertisement) a reference to RFC5059 should be added. For the currently unassigned types 13 and 14, and the reserved type 15, the name should be changed to "Type Space Extension", and reference this document. In addition, right underneath each of the rows for types 13, 14 and 15, there should be a new row where it says "13.0-13.15 Unassigned", "14.0-14.15 Unassigned" and "15.0-15.15 Unassigned", respectively.

A new registry called "PIM Message Type Flag Bits" should be created in the pim-parameters section with registration procedure "IETF"

Review" as defined in [RFC8126] with this document as a reference. The initial content of the registry should be as below.

Type	bit(s)	Name	Reference
0	0-7	Reserved	[RFC3973][RFC7761]
1	0-7	Reserved	[RFC3973][RFC7761]
2	0-7	Reserved	[RFC3973][RFC7761]
3	0-7	Reserved	[RFC3973][RFC7761]
4	0-6	Reserved	[RFC3973][RFC7761]
4	7	No-Forward	[RFC5059]
5	0-7	Reserved	[RFC3973][RFC7761]
6	0-7	Reserved	[RFC3973][RFC7761]
7	0-7	Reserved	[RFC3973][RFC7761]
8	0-7	Reserved	[RFC3973][RFC7761]
9	0-7	Reserved	[RFC3973][RFC7761]
10	0-3	Reserved	[RFC3973][RFC7761]
10	4-7	Sub-type	[RFC5015]
11	0-7	Reserved	[RFC6754]
12	0-6	Reserved	[RFC3973][RFC7761]
12	7	No-Forward	[RFC8364]
13	0-3	N/A (used by 13.0-13.15)	[this document]
13	4-7	Extended type	[this document]
13.0-13.15	0-3	Reserved	[this document]
14	0-3	N/A (used by 14.0-14.15)	[this document]
14	4-7	Extended type	[this document]
14.0-14.15	0-3	Reserved	[this document]
15	0-3	N/A (used by 15.0-15.15)	[this document]
15	4-7	Extended type	[this document]
15.0-15.15	0-3	Reserved	[this document]

## 8. References

### 8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3973] Adams, A., Nicholas, J., and W. Siadak, "Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised)", RFC 3973, DOI 10.17487/RFC3973, January 2005, <<https://www.rfc-editor.org/info/rfc3973>>.

- [RFC5015] Handley, M., Kouvelas, I., Speakman, T., and L. Vicisano, "Bidirectional Protocol Independent Multicast (BIDIR-PIM)", RFC 5015, DOI 10.17487/RFC5015, October 2007, <<https://www.rfc-editor.org/info/rfc5015>>.
- [RFC5059] Bhaskar, N., Gall, A., Lingard, J., and S. Venaas, "Bootstrap Router (BSR) Mechanism for Protocol Independent Multicast (PIM)", RFC 5059, DOI 10.17487/RFC5059, January 2008, <<https://www.rfc-editor.org/info/rfc5059>>.
- [RFC6754] Cai, Y., Wei, L., Ou, H., Arya, V., and S. Jethwani, "Protocol Independent Multicast Equal-Cost Multipath (ECMP) Redirect", RFC 6754, DOI 10.17487/RFC6754, October 2012, <<https://www.rfc-editor.org/info/rfc6754>>.
- [RFC7761] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I., Parekh, R., Zhang, Z., and L. Zheng, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", STD 83, RFC 7761, DOI 10.17487/RFC7761, March 2016, <<https://www.rfc-editor.org/info/rfc7761>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8364] Wijnands, IJ., Venaas, S., Brig, M., and A. Jonasson, "PIM Flooding Mechanism (PFM) and Source Discovery (SD)", RFC 8364, DOI 10.17487/RFC8364, March 2018, <<https://www.rfc-editor.org/info/rfc8364>>.

## 8.2. Informative References

- [RFC6166] Venaas, S., "A Registry for PIM Message Types", RFC 6166, DOI 10.17487/RFC6166, April 2011, <<https://www.rfc-editor.org/info/rfc6166>>.

## Authors' Addresses

Stig Venaas  
Cisco Systems, Inc.  
Tasman Drive  
San Jose CA 95134  
USA

Email: stig@cisco.com

Alvaro Retana  
Huawei R&D USA  
2330 Central Expressway  
Santa Clara CA 95050  
USA

Email: alvaro.retana@huawei.com