

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 30, 2015

P. Pfister
October 27, 2014

Source Specific support for Bidirectional Protocol Independent Multicast
draft-pfister-pim-ssbidir-00

Abstract

This document adds Source-Specific capabilities to the Bidirectional Protocol Independent Multicast (PIM-BIDIR) routing protocol. Similarly to PIM-BIDIR, multicast traffic is always forwarded to the RP Link, but packets are also filtered based on their source address and source-specific subscription state. This operation mode is backwards compatible with PIM-BIDIR, provides a simpler alternative to PIM-SM and does not suffer when the multicast source location cannot be determined.

Status of This Memo

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

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

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

This Internet-Draft will expire on April 30, 2015.

Copyright Notice

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

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

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Protocol Overview	3
3. Protocol Specifications	4
3.1. PIM-SSBIDIR Protocol State	5
3.1.1. Per Neighbor State	5
3.1.2. (*,G) State	5
3.1.3. (S,G) State	5
3.1.4. (S,G,rpt) State	6
3.1.5. Designated Forwarder Election	7
3.1.6. State Summerization Macro	7
3.2. Forwarding Rules	9
3.3. PIM Join/Prune Messages	9
3.3.1. Receiving (*,G) Join/Prune Messages	9
3.3.2. Receiving (S,G) Join/Prune Messages	9
3.3.3. Receiving (S,G,rpt) Join/Prune Messages	10
3.3.4. Sending (*,G) Join/Prune Messages	11
3.3.5. Sending (S,G) Join/Prune Messages	12
3.3.6. Sending (S,G,rpt) Join/Prune in (*,G) messages	12
3.3.7. Sending triggered (S,G,rpt) Join/Prune Messages	13
3.4. SSBIDIR-PIM Packet Formats	15
3.4.1. Encoded-Group Format	15
3.4.2. SSBIDIR Capable PIM-Hello Option	15
4. PIM-BIDIR Compatibility	16
5. Security Considerations	16
6. IANA Considerations	16
7. References	17
7.1. Normative References	17
7.2. Informative References	17
7.3. URIs	17
Appendix A. Acknowledgments	17
Author's Address	17

1. Introduction

The Protocol Independent Multicast (PIM - [RFC4601]) Sparse-Mode provides All-Source (ASM) and Source-Specific multicast (SSM) routing using an optimal source-specific routing path, but at the cost of two major drawbacks:

1. It relies on packet tunneling for All-Source Multicast (Pretty complex in terms of implementation).

2. It does not provide native multicast when some multicast source's location cannot be determined (e.g. when multiple default routes are in the RIB).

PIM BIDIR was specified as an alternative to PIM-SM: Simpler in terms of implementation and execution, at the cost of dropping source-specific multicast support. It is particularly suited in situations where multiple multicast sources also subscribe to other sources' sent packets. This document specifies the Source-Specific Bidirectional Mode (PIM-SSBIDIR), which adds source-specificity support to PIM-BIDIR. Unlike PIM-SM, all the traffic, including the source specific traffic, is forwarded on the RP-tree. Therefore, PIM-SSBIDIR does not provide source-specific path optimization, but still filters traffic according to multicast packet's source address, and thus reduces undesired traffic forwarding.

PIM-SSBIDIR, specified in this document, has two major advantages compared to PIM-SM.

1. Implementation and execution is simpler.
 - * A single RP-rooted forwarding tree is used.
 - * Assert state machine and messages are not used.
 - * Register and Register-Stop messages are not used (No tunneling).
2. Packets are forwarded natively even when the source's location can't be determined.

This extension was designed for multi-homed home networks ([I-D.ietf-homenet-arch] - a typical case in which source-location cannot be determined for ISP-originated traffic because of the multiple default routes). PIM-SM using PIM RPF Vectors [RFC5496] could be another alternative, but it is not clear how routers should behave when they receive conflicting RPF vectors. The protocol simplicity was also an important consideration as home routers usually have little resources and software reliability is important. Inputs from PIM and Homenet working group regarding other possible solutions are very welcome.

2. Protocol Overview

PIM-BIDIR specifies how a single RP-rooted tree can be maintained and used for forwarding multicast traffic to group subscribers. In the original document, only the (*,G) state machine was defined. This

document specifies (S,G) and (S,G,rpt) state machines when operating in BIDIR mode.

Similarly to PIM-SM, (S,G) Join/Prune messages allow subscribing to multicast traffic originated by a given source for a given group while (S,G,rpt) Join/Prune messages provide support for source-specific pruning. But unlike PIM-SM, SSBIDIR Join/Prune messages are always sent to the Designated Forwarder, and downstream multicast packets are always forwarded from the RP link to the subscribed hosts. Consequently, one major difference with PIM-SM is that (*,G) and (S,G,rpt) upstream states do not operate concurrently. A router's either operates in "include" mode using (S,G) states, or in "exclude" mode using (*,G) and (S,G,rpt) states.

Depending on the RP and sources locations, PIM-SSBIDIR does not always provide optimal routing of source-specific traffic. PIM-SM or PIM-SSM should be used in situations where source-specific path optimization is required.

PIM-SSBIDIR can operate with PIM-BIDIR neighbors. Neighbor's capabilities are determined using the new Source-Specific Bidirectional Capable PIM Hello option which is included in all Hello messages sent by SSBIDIR routers. When at least one neighbor, on a given link, does not support PIM-SSBIDIR, (S,G,rpt) states are not used, but (S,G) state can be. Additionally, when the Designated Forwarder does not support PIM-SSBIDIR, (S,G) downstream states are converted into (*,G) subscriptions. In any case, requested multicast traffic is always forwarded to the subscribed hosts. Although BIDIR-only router presence may increase undesired traffic overhead.

3. Protocol Specifications

The specification of PIM-SSBIDIR is broken into several parts:

- o Section 3.1 details the protocol state.
- o Section 3.1.5 recalls that the Designated Forwarder election must take place.
- o Section 3.2 specifies the multicast data forwarding rules.
- o Section 3.3 specifies the PIM-SSBIDIR Join/Prune generation and processing rules.

3.1. PIM-SSBIDIR Protocol State

This section describes the state that must be kept by routers operating in PIM-SSBIDIR mode, in addition to the state specified in PIM-BIDIR specifications.

3.1.1. Per Neighbor State

Routers MUST keep track of neighbor's SSBIDIR and BIDIR capabilities as specified in PIM Hello BIDIR and SSBIDIR capable options (Section 3.4.2).

This state is used by "ssbidir_neighbor(N)" macro defined in section Section 3.1.6.

3.1.2. (*,G) State

The (*,G) state is similar to the (*,G) state described in PIM-BIDIR's specifications.

3.1.3. (S,G) State

For every source S and group G operating in SSBIDIR mode, routers keep the following state:

(S,G) state:

For each interface:

Local Membership:

- State: One of {"NoInfo", "Include"}

PIM (S,G) Join/Prune:

- State: One of {"NoInfo", "Join", "Prune-Pending"}
- Prune-Pending Timer
- Join/Prune Expiry Timer

Not interface specific:

Upstream (S,G) Join/Prune State:

- State: One of {"NotJoined", "Joined"}
- Upstream (S,G) Join/Prune Timer

Local membership is the result of the local membership mechanism (such as MLD [RFC3810] or IGMP [RFC3376]) running on that interface. This information is used by `pim_include(S,G)` macro described in Section 3.1.6.

PIM (S,G) Join/Prune state is the result of receiving PIM (S,G) Join/Prune messages on this interface, and is specified in Section 3.3.2.

The upstream (S,G) Join/Prune State reflects the state of the upstream (S,G) state machine and the upstream (S,G) Join/Prune Timer is used to send periodic (S,G) Joins and override Prune(S,G) messages from peers on an upstream interface. Details are specified in Section 3.3.5.

3.1.4. (S,G,rpt) State

For every group G and source S operating in SSBIDIR mode, routers keep the following state:

(S,G,rpt) state:

For each interface:

Local Membership:

- State: One of {"NoInfo", "Exclude"}

PIM (S,G,rpt) Join/Prune:

- State: One of {"NoInfo", "Prune", "Prune-Pending", "PruneTmp", "Prune-Pending-Tmp"}
- Prune-Pending Timer
- Join/Prune Expiry Timer

Not interface specific:

Upstream (S,G,rpt) Join/Prune State:

- State: One of {"NotPruned", "Pruned"}
- Upstream (S,G) Override Timer

Local membership is the result of the local membership mechanism (such as MLD or IGMP) running on that interface. This information is used by `pim_include(S,G)` macro described in Section 3.1.6.

PIM (S,G,rpt) Join/Prune state is the result of receiving PIM (S,G,rpt) Join/Prune messages on this interface, and is specified in Section 3.3.3.

The upstream (S,G,rpt) Join/Prune State reflects the state of the upstream (S,G,rpt) state machine and the upstream (S,G,rpt) Join/Prune Timer is used to send periodic (S,G,rpt) Prunes and override Prune(S,G,rpt) messages from peers on the upstream interface. The upstream state machine's behavior is specified in Section 3.3.7.

3.1.5. Designated Forwarder Election

PIM-SSBIDIR routers MUST comply to the Designated Forwarder (DF) election process, as defined in PIM-BIDIR specifications.

3.1.6. State Summerization Macro

This section describes the macros used by PIM-SSBIDIR state machines.

RPA(G), RPF_interface(rpa) and RPF_neighbor(rpa) are related to the Designated Forwarder election. They are defined in PIM-BIDIR's specifications.

The following pseudo code describes the set of interfaces requesting (S,G) traffic.

```
olist(S,G) =  
  RPF_interface(RPA(G))  
  (+) joins(S,G)  
  (+) (joins(*,G) (-) prunes(S,G,rpt))  
  (+) pim_include(S,G)  
  (+) (pim_include(*,G) - pim_exclude(S,G))
```

The set "joins(*,G)" is the set of interfaces on which the router has received (*,G) Joins:

```
joins(*,G) =  
  { all interfaces I such that  
    I_am_DF(RPA(G),I) AND  
    DownstreamJPState(*,G,I) is either Join or Prune-Pending }
```

The set "joins(S,G)" is the set of interfaces on which the router has received (S,G) Joins:

```
joins(S,G) =  
  { all interfaces I such that  
    I_am_DF(RPA(G),I) AND  
    DownstreamJPState(S,G,I) is either Join or Prune-Pending }
```

The set "prunes(S,G,rpt)" is the set of interfaces on which the router has received (S,G,rpt) prunes:

```
prunes(S,G,rpt) =
  { all interfaces I such that
    I_am_DF(RPA(G),I) AND
    DownstreamJPState(S,G,rpt,I) is either Prune or PruneTmp }
```

The sets "pim_include(S,G)" and "pim_include(*,G)" indicate interfaces to which multicast traffic might be forwarded because of hosts that are local members on that interface. The set "pim_exclude(S,G)" is the set of interfaces where local members required to not receive traffic from source S:

```
pim_include(*,G) =
  { all interfaces I such that
    I_am_DF(RPA(G),I) AND
    local_receiver_include(*,G,I) }
```

```
pim_include(S,G) =
  { all interfaces I such that
    I_am_DF(RPA(G),I) AND
    local_receiver_include(S,G,I) }
```

```
pim_exclude(S,G) =
  { all interfaces I such that
    I_am_DF(RPA(G),I) AND
    local_receiver_exclude(S,G,I) }
```

Clauses "local_receiver_include(*,G,I)", "local_receiver_include(S,G,I)" and "local_receiver_exclude(S,G,I)" reflect local membership subscriptions, as defined in PIM-SM specifications.

In order to operate with BIDIR routers, SSBIDIR routers keep track of neighbors' capabilities. The clause "ssbidir_neighbor(N)" is TRUE if the last received Hello from neighbor N contained a Source-Specific Bidirectional Capable Hello Option. It is FALSE otherwise. In addition, the "ssbidir_link(I)" is TRUE if "ssbidir_neighbor(N)" is TRUE for any neighbor reachable through the interface I. It is FALSE otherwise. These clauses are used in JoinDesired(G), JoinDesired(S,G) and PruneDesired(S,G,rpt) macro specified in Section 3.3.

3.2. Forwarding Rules

Similarly to PIM-BIDIR, forwarding takes place on the RP-tree, defined as the set of uplink interfaces `RPF_interface(RPA(G))` and interfaces where the router is elected Designated Forwarder.

The process of accepting a packet for forwarding, before building the outgoing interface list, is similar to PIM-BIDIR's as well.

In PIM-SSBIDIR, both group and source addresses are used in order to build the outgoing interface list, based on router's source-specific state. This differs from PIM-BIDIR, where only the group address is used.

A packet received on an interface for which we are the Designated Forwarder is always relayed upstream.

A packet received on the upstream interface, or on an interface on which we are the DF, is forwarded on all interfaces that requested the corresponding multicast traffic (Except the input interface).

In summary, on receipt of data from S to G on interface iif:

```
if( iif == RPA_interface(RPA(G)) || I_am_DF(RPA(G), iif) ) {  
    oiflist = olist(S,G) (-) iif  
    forward packet on all interfaces in oiflist  
}
```

3.3. PIM Join/Prune Messages

PIM SSBIDIR adds source-specific states and messaging to the existing PIM-BIDIR specifications. Therefore, this document only specifies (S,G) and (S,G,rpt) state machines and do not modify the existing BIDIR (*,G) state machine.

3.3.1. Receiving (*,G) Join/Prune Messages

The per-interface (*,G) state machine is similar to the state machine described in PIM-BIDIR.

3.3.2. Receiving (S,G) Join/Prune Messages

When a router receives a Join(S,G) or Prune(S,G) for which SSBIDIR is enabled, it MUST first check if the group's [B]idir flag is set. If not, the packet MUST be discarded.

The per-interface state machine for receiving (S,G) Join/Prune has three states.

NoInfo (NI)

The interface has no (S,G) Join state and no timers running.

Join (J)

One of the routers reachable on interface I has requested receiving source-specific traffic sent by S to G. If the router is the DF on this interface (`I_am_DF(RPA(G),I)` is TRUE), the Join state will cause us to forward packets sent by S to G on this interface.

PrunePending (PP)

The router has received a a Prune(S,G) message and is waiting for the PrunePending timer to timeout before moving to the NoInfo state. During that lapse of time, forwarding takes place exactly like in the Join state and on-link routers can override the Prune(S,G) by sending a Join(S,G) message.

(S,G) per-interface state machine events, transitions and timer values are similar to PIM-BIDIR's (*,G) (By replacing (*,G) by (S,G)). Details can be found in Section 3.4.1 [1] from [RFC5015].

3.3.3. Receiving (S,G,rpt) Join/Prune Messages

When a router receives a Join(S,G,rpt) or a Prune(S,G,rpt) for which SSBIDIR is enabled, it MUST first check if the group's [B]idir flag is set. If not, the packet MUST be discarded.

When a message is parsed for a given group G, (*,G) Joins must be processed before processing any (S,G,rpt) Prunes. Transient state "PruneTmp" and "Prune-Pending-Tmp" are used during the parsing process.

The per-interface state machine for receiving (S,G,rpt) Join/Prune messages has four states.

NoInfo (NI)

The interface has no (S,G,rpt) Prune state and no timers running.

Prune (P)

The per-interface (*,G) state is set to Join and we received a Prune(S,G,rpt) message. It means that all interface's neighbors that are interested in (*,G) do not want to receive packets sent by source S. If the router is the DF on this interface (`I_am_DF(RPA(G),I)` is TRUE), the Prune state will cause us to *not* forward packets sent by S to G on this interface.

PrunePending (PP)

The router has received a Prune(S,G,rpt) message and is waiting for the PrunePending timer to timeout before moving to the Prune state. During that lapse of time, forwarding takes place exactly like the NoInfo state and neighboring routers can override the Prune(S,G,rpt) by sending a Join(S,G,rpt) message.

PruneTmp (P') This state is a transient state that is used as we parse a Join/Prune message that contains a (*,G) Join. A (S,G,rpt) Prune present in the same message will reinstate the Prune state. However, if we reach the end of the message without encountering such a (S,G,rpt) Prune, we will revert to NoInfo state in this state machine. For forwarding purposes, PruneTmp behaves exactly like Prune.

Prune-Pending-Tmp (PP') This state is a transient state that is identical to P' except that it is associated with the PP state rather than the P state. For forwarding purposes, PP' behaves exactly like PP state.

(S,G,rpt) per-interface state machine events, transitions and timer values are similar to PIM-SM (S,G,rpt) per-interface state. Details can be found in Section 4.5.4 [2] from [RFC4601].

In addition, the state machine needs to react to the "Stop Being DF on I" event. When this event occurs, all (S,G,rpt) states associated with I are moved to "NoInfo" and PrunePending and Expiry timers are stopped (if running).

3.3.4. Sending (*,G) Join/Prune Messages

The Upstream (*,G) state machine is similar to the state machine described in PIM-BIDIR's specifications.

For BIDIR compatibility support, the JoinDesired(G) macro must be modified as follows:

```
bool JoinDesired(G) {
    out = joins(*,G) (+) pim_include(*,G)
    if(!ssbidir_neighbor(RPF_neighbor(RPA(G)))) {
        out = out (+) {olist(S,G) for any source S}
    }
    return (out (-) RPF_interface(RPA(G))) != {}
}
```

In case the upstream neighbor is not SSBIDIR capable, downstream (S,G) Joins are translated into upstream (*,G) Joins.

3.3.5. Sending (S,G) Join/Prune Messages

The upstream (S,G) state machine holds join state from downstream routers and local membership information. It controls (S,G) Join/Prune message emission.

This state machine has two possible states.

Not Joined

The downstream state machines and local membership information do not indicate that the router needs to send (S,G) Joins or override other routers' (S,G) prunes.

Joined

The downstream state machines and local membership information indicate that the router needs to send (S,G) Joins and override other routers' (S,G) prunes.

In addition, one timer JT(S,G) is kept that is used to trigger the sending of a Join(S,G) to the upstream Designated Forwarder RPF_neighbor(RPA(G)).

The detailed state machine is derived from PIM-BIDIR (*,G) upstream state machine by replacing (*,G) references with (S,G).

The JoinDesired(S,G) macro is defined as follows:

```
bool JoinDesired(S,G) {  
    if(!ssbidir_neighbor(RPF_neighbor(RPA(G))) OR  
        JoinDesired(G))  
        return FALSE  
    return (olist(S,G) (-) RPF_interface(RPA(G))) != {}  
}
```

In case the upstream neighbor is not SSBIDIR capable, downstream (S,G) Joins are translated into upstream (*,G) Joins.

3.3.6. Sending (S,G,rpt) Join/Prune in (*,G) messages

Whenever a Join(*,G) message is sent in response to a Prune(*,G), it must include a Prune(S,G,rpt) for every (S,G,rpt) upstream state that is set to Pruned.

When sending a periodic Join(*,G), it must include a Prune(S,G,rpt) for every (S,G,rpt) state set to Pruned which Override Timer is not running.

Note that if the message construction is made after processing the Join(*,G) event in the upstream (S,G,rpt) state machine, the first paragraph can be ignored and the second paragraph applied in all cases. That is because the Join(*,G) reception event cancels all (S,G,rpt) Override Timers.

3.3.7. Sending triggered (S,G,rpt) Join/Prune Messages

The upstream (S,G,rpt) state machine holds join state from downstream routers and local membership information. It controls (S,G,rpt) Join/Prune message emission. It is only used when upstream (*,G) state is Joined.

The upstream (S,G,rpt) state machine has two states:

NotPruned

The upstream (*,G) state is "Joined" but downstream routers and local membership indicates that at least one downstream interface should receive traffic sent from S to G.

Pruned

The upstream (*,G) state is "Joined" but not a single downstream interface requested traffic sent from S to G.

ASNotJoined(G)

The All-Sources (*,G) state is NotJoined. This state is kept because the (*,G) state machine takes care of sending (S,G,rpt) Prunes when JoinDesired(G) changes to true. Similarly, when JoinDesired(G) changes to false, all (S,G,rpt) states are moved back to ASNotJoined.

In addition, there is an (S,G,rpt) Override Timer, OT(S,G,rpt). In the NotPruned state, it is used to delay triggered Join(S,G,rpt) messages to prevent explosion of triggered messages. In the Pruned state, it is used to delay Prune(S,G,rpt) message origination whenever we received a Join(S,G,rpt) on that interface. This timer can be either cancelled, decreased to t_override or increased to t_suppressed (see PIM-BIDIR specifications for t_override and t_suppressed values).

The following tables show (S,G,rpt) upstream state machine transitions and events.

Event	State	
	ASNotJoined(NJ)	NotPruned(NP) & Pruned(P)
JoinDesired(G) -> true	If PD(S,G,rpt) -> P Else -> NP	—
JoinDesired(G) -> false	—	-> NP
Other events	Do nothing	

Event	State	
	NotPruned(NP)	Pruned(P)
PruneDesired(S,G,rpt) -> true	-> P; Cancel OT; Send Prune(S,G,rpt)	—
PruneDesired(S,G,rpt) -> false	—	-> NP; Cancel OT; Send Join(S,G,rpt)
See Prune(S,G,rpt) to RPF_DF(RPA(G))	Decrease OT to t_override	Cancel OT
See Join(S,G,rpt) to RPF_DF(RPA(G))	Cancel OT	Increase OT to t_suppress
See Prune(*,G) to RPF_DF(RPA(G))	Do nothing	Cancel OT
DF Changes	Cancel OT	Send Join/Prune to old/new DF
DF GenId changes	Do nothing	Decrease OT to t_override
Override Timer timeouts	Send Join(S,G,rpt)	Send Prune(S,G,rpt)

Upstream (S,G,rpt) state machine.

The (S,G,rpt) upstream state machine uses the following macro:

```
bool PruneDesired(S,G,rpt) {
    if(!ssbidir_link(RPF_interface(RPA(g))))
        return FALSE

    return (olist(S,G) (-) RPF_interface(RPA(G))) == {}
}
```

The PruneDesired(S,G,rpt) macro can return true even if JoinDesired(G) is false (And thus does not always means that Prune(S,G,rpt) should be sent upstream). That is to differentiate PruneDesired(S,G,rpt) transitions that are due to JoinDesired(G) transition and those which aren't.

3.4. SSBIDIR-PIM Packet Formats

This section describes the details of the packet formats for PIM-SSBIDIR control messages.

3.4.1. Encoded-Group Format

For all groups G operating in SSBIDIR mode, the [B]idirectional bit must be set in the Group-Encoded format included in Join/Prune messages.

3.4.2. SSBIDIR Capable PIM-Hello Option

SSBIDIR-PIM introduces one new PIM-Hello option.

- o OptionType TBD_BY_IANA: Source-Specific Bidirectional Capable.

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|           Type = TBD_BY_IANA           |           Length = 0           |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

This option MUST be included in all PIM Hello messages sent by SSBIDIR capable routers.

SSBIDIR capable routers MUST as well include the Bidirection PIM-BIDIR Hello Option in sent Hello messages.

4. PIM-BIDIR Compatibility

Compatibility problems between SSBIDIR and BIDIR routers occur when a BIDIR router joins (*,G) while an SSBIDIR router is pruning (S,G,rpt) on the same link: The BIDIR router will not suppress (S,G,rpt) Prunes, which will prevent it from receiving traffic from source S. Similarly, if the Designated Forwarder is not SSBIDIR capable, it will reject (S,G) and (S,G,rpt) messages.

PIM-SSBIDIR interoperates with PIM-BIDIR by detecting BIDIR-only neighbors and suppressing undesired behaviors. When at least one neighbor, on a given link, is BIDIR-only, (S,G,rpt) messages are not used anymore (all traffic to G will be requested). When the upstream router is BIDIR-only, both (S,G,rpt) and (S,G) messages are not used anymore and only All-Source Multicast is supported.

PIM-BIDIR compatibility SHOULD be supported. If an implementer chooses not to implement the compatibility support, its implementation MUST behave as if all neighbors were SSBIDIR capable. Additionally, an error MUST be logged in a rate-limited manner if a Hello message that does not include the SSBIDIR Capable Option is received.

PIM-BIDIR capable routers send (*,G) Join/Prune messages with the [B]idir bit set, but do not deal with (S,G) or (S,G,rpt) Join/Prune messages. The expected behavior from BIDIR capable routers in such situations is not specified. In order to support (S,G) subscriptions while operating in compatibility mode, we expect BIDIR implementations to silently ignore source-specific Join/Prunes.

5. Security Considerations

Similarly to PIM-SM, introducing Source-Specific support to PIM BIDIR makes it vulnerable to easy deny of service attacks by generating lots of (S,G) Join or (S,G,rpt) Prune states for different sources.

In order to operate securely, PIM messages SHOULD be authenticated and local subscriptions should be limited in rate and number (On a per-host basis if the link-layer provides authentication, on a per-link basis otherwise).

6. IANA Considerations

IANA is kindly asked to assign a new PIM-Hello Option Type to be used for Source-Specific Bidirectional BIDIR-PIM Hello options.

7. References

7.1. Normative References

- [RFC5015] Handley, M., Kouvelas, I., Speakman, T., and L. Vicisano, "Bidirectional Protocol Independent Multicast (BIDIR-PIM)", RFC 5015, October 2007.
- [RFC4601] Fenner, B., Handley, M., Holbrook, H., and I. Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", RFC 4601, August 2006.
- [RFC3810] Vida, R. and L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, June 2004.
- [RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, October 2002.

7.2. Informative References

- [I-D.ietf-homenet-arch]
Chown, T., Arkko, J., Brandt, A., Troan, O., and J. Weil, "IPv6 Home Networking Architecture Principles", draft-ietf-homenet-arch-11 (work in progress), October 2013.
- [RFC5496] Wijnands, IJ., Boers, A., and E. Rosen, "The Reverse Path Forwarding (RPF) Vector TLV", RFC 5496, March 2009.

7.3. URIs

- [1] <http://tools.ietf.org/html/rfc5015#section-3.4.1>
- [2] <http://tools.ietf.org/html/rfc4601#section-4.5.4>

Appendix A. Acknowledgments

The author would like to thank Steven Barth, Mohammed Hawari, Mark Townsley, Stig Venaas and IJsbrand Wijnands for their useful comments.

Author's Address

Pierre Pfister
Paris
France

Email: pierre@darou.fr