Internet Engineering Task Force (IETF)
Request for Comments: 6559
Category: Experimental
ISSN: 2070-1721

D. Farinacci
IJ. Wijnands
S. Venaas
Cisco Systems
M. Napierala
AT&T Labs

March 2012

# A Reliable Transport Mechanism for PIM

## Abstract

This document defines a reliable transport mechanism for the PIM protocol for transmission of Join/Prune messages. This eliminates the need for periodic Join/Prune message transmission and processing. The reliable transport mechanism can use either TCP or SCTP as the transport protocol.

## Status of This Memo

This document is not an Internet Standards Track specification; it is published for examination, experimental implementation, and evaluation.

This document defines an Experimental Protocol for the Internet community. This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6559.

Table of Contents

## 1. Introduction

The goals of this specification are:

- To create a simple incremental mechanism to provide reliable PIM Join/Prune message delivery in PIM version 2 for use with PIM Sparse-Mode (PIM-SM) [RFC4601], including PIM Source-Specific Multicast (PIM-SSM), and Bidirectional PIM [RFC5015].

- When a router supports this specification, it need not use the reliable transport mechanism with every neighbor. It can be negotiated on a per-neighbor basis.

The explicit non-goals of this specification are:

- Making changes to the PIM message formats as defined in [RFC4601].

- Providing support for automatic switching between the reliable transport mechanism and the regular PIM mechanism defined in [RFC4601]. Two routers that are PIM neighbors on a link will always use the reliable transport mechanism if and only if both have enabled support for the reliable transport mechanism.

This document will specify how periodic Join/Prune message transmission can be eliminated by using TCP [RFC0793] or SCTP [RFC4960] as the reliable transport mechanism for Join/Prune messages. The destination port number is 8471 for both TCP and SCTP.

This specification enables greater scalability in terms of control-traffic overhead. However, for routers connected to multi-access links, scalability comes at the price of increased PIM state and the overhead required to maintain this state.

In many existing and emerging networks, particularly wireless and mobile satellite systems, link degradation due to weather, interference, and other impairments can result in temporary spikes in the packet loss rate. In these environments, periodic PIM joining can cause join latency when messages are lost, causing a retransmission only 60 seconds later. By applying a reliable transport, a lost Join is retransmitted rapidly. Furthermore, when the last user leaves a multicast group, any lost Prune is similarly repaired, and the multicast stream is quickly removed from the wireless/satellite link. Without a reliable transport, the multicast transmission could otherwise continue until it timed out, roughly 3 minutes later. As network resources are at a premium in many of these environments, rapid termination of the multicast stream is critical for maintaining efficient use of bandwidth.

This is an experimental extension to PIM. It makes some fundamental changes to how PIM works in that Join/Prune state does not require periodic updates, and it partly turns PIM into a hard-state protocol. Also, using reliable delivery for PIM messages is a new concept, and it is likely that experiences from early implementations and deployments will lead to at least minor changes in the protocol. Once there is some deployment experience, making this a Standards Track protocol should be considered. Experiments using this protocol only require support by pairs of PIM neighbors, and need not be constrained to isolated networks.

### 1.1. Requirements Notation

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].

### 1.2. Definitions

PORT: Stands for PIM Over Reliable Transport, which is the short form for describing the mechanism in this specification where PIM can use the TCP or SCTP transport protocol.

Periodic Join/Prune message: A Join/Prune message sent periodically to refresh state.

Incremental Join/Prune message: A Join/Prune message sent as a result of state creation or deletion events. Also known as a triggered message.

Native Join/Prune message: A Join/Prune message that is carried with an IP protocol type of PIM.

PORT Join/Prune message: A Join/Prune message using TCP or SCTP for transport.

Datagram Mode: The procedures whereby PIM encapsulates triggered or periodic Join/Prune messages in IP packets.

PORT Mode: The procedures used by PIM and defined in this specification for sending Join/Prune messages over the TCP or SCTP transport layer.

## 2. Protocol Overview

PIM Over Reliable Transport (PORT) is a simple extension to PIMv2 for refresh reduction of PIM Join/Prune messages. It involves sending incremental rather than periodic Join/Prune messages over a TCP/SCTP connection between PIM neighbors.

PORT only applies to PIM Sparse-Mode [RFC4601] and Bidirectional PIM [RFC5015] Join/Prune messages.

This document does not restrict PORT to any specific link types. However, the use of PORT on, e.g., multi-access LANs with many PIM neighbors should be carefully evaluated. This is due to the facts that there may be a full mesh of PORT connections and that explicit tracking of all PIM neighbors is required.

PORT can be incrementally used on a link between PORT-capable neighbors. Routers that are not PORT-capable can continue to use PIM in Datagram mode. PORT capability is detected using new PORT-Capable PIM Hello Options.

Once PORT is enabled on an interface and a PIM neighbor also announces that it is PORT enabled, only PORT Join/Prune messages will be used. That is, only PORT Join/Prune messages are accepted from, and sent to, that particular neighbor. Native Join/Prune messages are still used for PIM neighbors that are not PORT enabled.

PORT Join/Prune messages are sent using a TCP/SCTP connection. When two PIM neighbors are PORT enabled, both for TCP or both for SCTP, they will immediately, or on demand, establish a connection. If the connection goes down, they will again immediately, or on demand, try to reestablish the connection. No Join/Prune messages (neither Native nor PORT) are sent while there is no connection. Also, any received native Join/Prune messages from that neighbor are discarded, even when the connection is down.

When PORT is used, only incremental Join/Prune messages are sent from downstream routers to upstream routers. As such, downstream routers do not generate periodic Join/Prune messages for state for which the Reverse Path Forwarding (RPF) neighbor is PORT-capable.

For Joins and Prunes that are received over a TCP/SCTP connection, the upstream router does not start or maintain timers on the outgoing interface entry. Instead, it keeps track of which downstream routers have expressed interest. An interface is deleted from the outgoing interface list only when all downstream routers on the interface no longer wish to receive traffic. If there also are native Joins/ Prunes from a non-PORT neighbor, then a router can maintain timers on

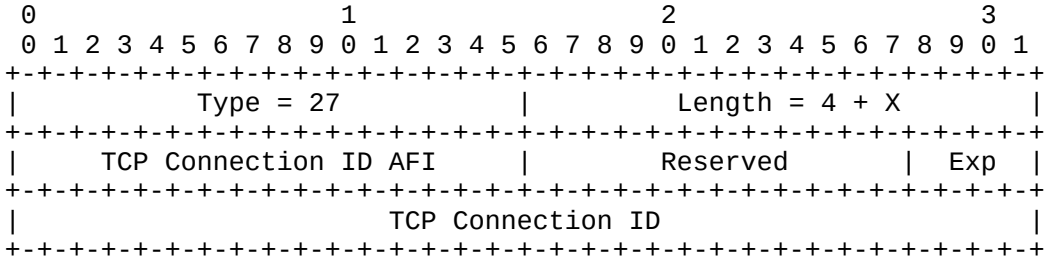
the outgoing interface entry as usual, while at the same time keep track of each of the downstream PORT Joins/Prunes.

This document does not update the PIM Join/Prune packet format. In the procedures described in this document, each PIM Join/Prune message is included in the payload of a PORT message carried over TCP/SCTP. See Section 5 for details on the PORT message.

## 3. PIM Hello Options

### 3.1. PIM over the TCP Transport Protocol

Option Type: PIM-over-TCP-Capable




Assigned Hello Type values can be found in [HELLO-OPT].

When a router is configured to use PIM over TCP on a given interface, it MUST include the PIM-over-TCP-Capable Hello Option in its Hello messages for that interface. If a router is explicitly disabled from using PIM over TCP, it MUST NOT include the PIM-over-TCP-Capable Hello Option in its Hello messages.

All Hello messages containing the PIM-over-TCP-Capable Hello Option MUST also contain the Interface ID Hello Option, see Section 3.3.

Implementations MAY provide a configuration option to enable or disable PORT functionality. It is RECOMMENDED that this capability be disabled by default.

Length: Length in bytes for the value part of the Type/Length/Value encoding, where X is the number of bytes that make up the Connection ID field. X is 4 when AFI of value 1 (IPv4) [AFI] is used, 16 when AFI of value 2 (IPv6) [AFI] is used, and 0 when AFI of value 0 is used.

TCP Connection ID AFI: The AFI value to describe the address family of the address of the TCP Connection ID field. Note that this value does not need to match the address family of the PIM Hello message that carries it. When this field is 0, a mechanism outside the scope of this document is used to obtain the addresses used to establish the TCP connection.

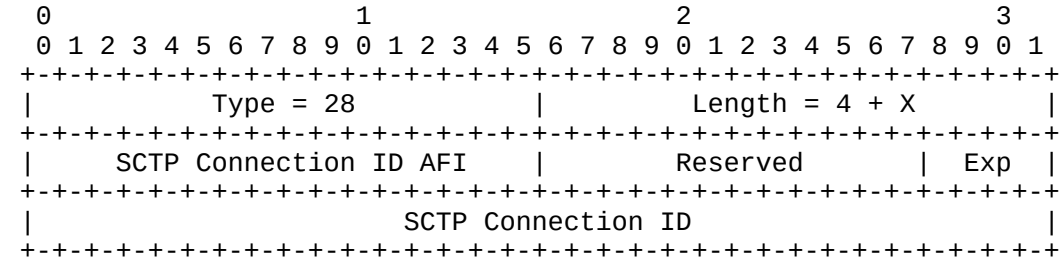
Reserved: Set to zero on transmission and ignored on receipt.

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 ignore the bits on receipt.

TCP Connection ID: An IPv4 or IPv6 address used to establish the TCP connection. This field is omitted (length 0) for the Connection ID AFI 0.

## 3.2. PIM over the SCTP Transport Protocol

Option Type: PIM-over-SCTP-Capable




Assigned Hello Type values can be found in [HELLO-OPT].

When a router is configured to use PIM over SCTP on a given interface, it MUST include the PIM-over-SCTP-Capable Hello Option in its Hello messages for that interface. If a router is explicitly disabled from using PIM over SCTP, it MUST NOT include the PIM-over-SCTP-Capable Hello Option in its Hello messages.

All Hello messages containing the PIM-over-SCTP-Capable Hello Option MUST also contain the Interface ID Hello Option; see Section 3.3.

Implementations MAY provide a configuration option to enable or disable PORT functionality. It is RECOMMENDED that this capability be disabled by default.

Length: Length in bytes for the value part of the Type/Length/Value encoding, where X is the number of bytes that make up the Connection ID field. X is 4 when AFI of value 1 (IPv4) [AFI] is used, 16 when AFI of value 2 (IPv6) [AFI] is used, and 0 when AFI of value 0 is used.

SCTP Connection ID AFI: The AFI value to describe the address family of the address of the SCTP Connection ID field. Note that this value does not need to match the address family of the PIM Hello message that carries it. When this field is 0, a mechanism outside the scope of this document is used to obtain the addresses used to establish the SCTP connection.

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 ignore the bits on receipt.

SCTP Connection ID: An IPv4 or IPv6 address used to establish the SCTP connection. This field is omitted (length 0) for the Connection ID AFI 0.

## 3.3. Interface ID

All Hello messages containing PIM-over-TCP-Capable or PIM-over-SCTP-Capable Hello Options MUST also contain the Interface ID Hello Option [RFC6395].

The Interface ID is used to associate a PORT Join/Prune message with the PIM neighbor from which it is coming. When unnumbered interfaces are used or when a single transport connection is used for sending and receiving Join/Prune messages over multiple interfaces, the Interface ID is used to convey the interface from Join/Prune message sender to Join/Prune message receiver. The value of the Interface ID Hello Option in Hellos sent on an interface MUST be the same as the Interface ID value in all PORT Join/Prune messages sent to a PIM neighbor on that interface.

The Interface ID need only uniquely identify an interface of a router; it does not need to identify to which router the interface belongs. This means that the Router ID part of the Interface ID MAY be 0. For details on the Router ID and the value 0, see [RFC6395].

## 4. Establishing Transport Connections

While a router interface is PORT enabled, a PIM-over-TCP-Capable or a PIM-over-SCTP-Capable Option MUST be included in the PIM Hello messages sent on that interface. When a router on a PORT-enabled interface receives a Hello message containing a PIM-over-TCP-Capable/ PIM-over-SCTP-Capable Option from a new neighbor, or an existing neighbor that did not previously include the option, it switches to PORT mode for that particular neighbor.

When a router switches to PORT mode for a neighbor, it stops sending and accepting Native Join/Prune messages for that neighbor. Any state from previous Native Join/Prune messages is left to expire as normal. It will also attempt to establish a transport connection (TCP or SCTP) with the neighbor. If both the router and its neighbor have announced both PIM-over-TCP-Capable and PIM-over-SCTP-Capable Options, SCTP MUST be used. This resolves the issue where two transports are both offered. The method prefers SCTP over TCP, because SCTP has benefits such as handling of call collisions and support for multiple streams, as discussed later in this document.

When the router is using TCP, it will compare the TCP Connection ID it announced in the PIM-over-TCP-Capable Option with the TCP Connection ID in the Hello received from the neighbor. Unless connections are opened on demand (see below), the router with the lower Connection ID MUST do an active transport open to the neighbor Connection ID. The router with the higher Connection ID MUST do a passive transport open. An implementation MAY open connections only on demand; in that case, it may be that the neighbor with the higher Connection ID does the active open (see Section 4.5). If the router with the lower Connection ID chooses to only do an active open on demand, it MUST do a passive open, allowing for the neighbor to initiate the connection. Note that the source address of the active open MUST be the announced Connection ID.

When the router is using SCTP, the IP address comparison need not be done since the SCTP protocol can handle call collision.

The decisions whether to use PORT, which transport to use, and which Connection IDs to use are made independently for IPv4 and IPv6. Thus, if PORT is used both for IPv4 and IPv6, both IPv4 and IPv6 PIM Hello messages MUST be sent, both containing PORT Hello Options. If two neighbors announce the same transport (TCP or SCTP) and the same Connection IDs in the IPv4 and IPv6 Hello messages, then only one connection is established and is shared. Otherwise, two connections are established and are used separately.

The PIM router that performs the active open initiates the connection with a locally generated source transport port number and a well-known destination transport port number. The PIM router that performs the passive open listens on the well-known local transport port number and does not qualify the remote transport port number. See Section 5 for the well-known port number assignment for PORT.

When a transport connection is established (or reestablished), the two routers MUST both send a full set of Join/Prune messages for state for which the other router is the upstream neighbor. This is needed to ensure that the upstream neighbor has the correct state. When moving from Datagram mode, or when the connection has gone down, the router cannot be sure that all the previous Join/Prune state was received by the neighbor. Any state that was created before the connection was established (or reestablished) and that is not refreshed MUST be left to expire and be deleted. When the non-refreshed state has expired and been deleted, the two neighbors will be in sync.

When not running PORT, a full update is only needed when a router restarts; with PORT, it must be done every time a connection is established. This can be costly, although it is expected that a PORT connection will go up and down rarely. There may be a need for extensions to better handle this.

It is possible that a router starts sending Hello messages with a new Connection ID, e.g., due to configuration changes. A router MUST always use the last announced and last seen Connection IDs. A connection is identified by the local Connection ID (the one we are announcing on a particular interface), and the remote Connection ID (the one we are receiving from a neighbor on the same interface). When either the local or remote ID changes, the Connection ID pair we need a connection for changes. There may be an existing connection with the same pair, in which case the router will share that connection. Or, a new connection may need to be established. Note that for link-local addresses, the interface should be regarded as part of the ID, so that connection sharing is not attempted when the same link-local addresses are seen on different interfaces.

When a Connection ID changes, if the previously used connection is not needed (i.e., there are no other PIM neighborships using the same Connection ID pair), both peers MUST attempt to reset the transport connection. Next (even if the old connection is still needed), they MUST, unless a connection already exists with the new Connection ID pair, immediately or on demand attempt to establish a new connection with the new Connection ID pair.

Normally, the Interface ID would not change while a connection is up. However, if it does, the change does not affect the connection. It just means that when subsequent PORT Join/Prune messages are received, they should be matched against the last seen Interface ID.

Note that a Join sent over a transport connection will only be seen by the upstream router; thus, it will not cause non-PORT routers on the link with the upstream router to delay the refresh of Join state for the same state. Similarly, a Prune sent over a transport connection will only be seen by the upstream router; thus, it will never cause non-PORT routers on the link with the upstream router to send a Join to override this Prune.

Note also that a datagram PIM Join/Prune message for a said (S,G) or (*,G) sent by some router on a link will not cause routers on the same link that use a transport connection with the upstream router for that state to suppress the refresh of that state to the upstream router (because they don't need to periodically refresh this state) or to send a Join to override a Prune. The latter will not occur because the upstream router will only stop forwarding the traffic when all joined routers that use a transport connection have explicitly sent a Prune for this state, as explained in Section 6.

### 4.1. Connection Security

TCP/SCTP packets used for PORT MUST be sent with a TTL/Hop Limit of 255 to facilitate the enabling of the Generalized TTL Security Mechanism (GTSM) [RFC5082]. Implementations SHOULD provide a configuration option to enable the GTSM check at the receiver. This means checking that inbound packets from directly connected neighbors have a TTL/Hop Limit of 255, but implementations MAY also allow for a different TTL/Hop Limit threshold to check that the sender is within a certain number of router hops. The GTSM check SHOULD be disabled by default.

Implementations SHOULD support the TCP Authentication Option (TCP-AO) [RFC5925] and SCTP Authenticated Chunks [RFC4895].

### 4.2. Connection Maintenance

TCP is designed to keep connections up indefinitely during a period of network disconnection. If a PIM-over-TCP router fails, the TCP connection may stay up until the neighbor actually reboots, and even then it may continue to stay up until PORT tries to send the neighbor some information. This is particularly relevant to PIM since the flow of Join/Prune messages might be in only one direction and the downstream neighbor might never get any indication via TCP that the other end of the connection is not really there.

SCTP has a heartbeat mechanism that can be used to detect that a connection is not working, even when no data is sent. Many TCP implementations also support sending keep-alives for this purpose. Implementations MAY make use of TCP keep-alives, but the PORT keep-alive mechanism defined below allows for more control and flexibility.

One can detect that a PORT connection is not working by regularly sending PORT messages. This applies to both TCP and SCTP. For example, in the case of TCP, the connection will be reset if no TCP ACKs are received after several retries. PORT in itself does not require any periodic signaling. PORT Join/Prune messages are only sent when there is a state change. If the state changes are not frequent enough, a PORT Keep-Alive message (defined in Section 5.2) can be sent instead. For example, if an implementation wants to send a PORT message, to check that the connection is working, at least every 60 seconds, then whenever 60 seconds have passed since the previous message, a Keep-Alive message could be sent. If there were less than 60 seconds between each Join/Prune, no Keep-Alive messages would be needed. Implementations SHOULD support the use of PORT Keep-Alive messages. It is RECOMMENDED that a configuration option be available to network administrators to enable it when needed. Note that Keep-Alives can be used by a peer, independently of whether the other peer supports it.

An implementation that supports Keep-Alive messages acts as follows when processing a received PORT message. When processing a Keep-Alive message with a non-zero Holdtime value, it MUST set a timer to the value. We call this timer Connection Expiry Timer (CET). If the CET is already running, it MUST be reset to the new value. When processing a Keep-Alive message with a zero Holdtime value, the CET (if running) MUST be stopped. When processing a PORT message other than a Keep-Alive, the CET MUST be reset to the last received Holdtime value if running. If the CET is not running, no action is taken. If the CET expires, the connection SHOULD be shut down. This specification does not mandate a specific default Holdtime value. However, the dynamic congestion and flow control in TCP and SCTP can result in variable transit delay between the endpoints. When capacity varies, there may be loss in the network or variable link performance. Consistent behavior therefore requires a sufficiently large Holdtime value, e.g., 60 seconds to prevent premature termination.

It is possible that a router receives Join/Prune messages for an interface/link that is down. As long as the neighbor has not expired, it is RECOMMENDED to process those messages as usual. If they are ignored, then the router SHOULD ensure it gets a full update

for that interface when it comes back up. This can be done by changing the GenID (Generation Identifier; see [RFC4601]) or by terminating and reestablishing the connection.

If a PORT neighbor changes its GenID and a connection is established or is in the process of being established, the local side should generally tear down the connection and do as described in Section 4.3. However, if the connection is shared by multiple interfaces and the GenID changes for only one of them, the local side SHOULD simply send a full update, similar to other cases when a GenID changes for an upstream neighbor.

### 4.3. Actions When a Connection Goes Down

A connection may go down for a variety of reasons. It may be due to an error condition or a configuration change. A connection SHOULD be shut down as soon as there are no more PIM neighbors using it. That is, for the connection in question (and its associated local and remote Connection IDs), when there is no PIM neighbor with that particular remote Connection ID on any interface where we announce the local Connection ID, the connection SHOULD be shut down. This may happen when a new Connection ID is configured, PORT is disabled, or a PIM neighbor expires.

If a PIM neighbor expires, one should free connection state and downstream outgoing interface list (oif-list) state for that neighbor. A downstream router, when an upstream neighboring router has expired, will simply update the RPF neighbor for the corresponding state to a new neighbor where it would trigger Join/Prune messages. This behavior is according to [RFC4601], which defines the term "RPF neighbor". It is required of a PIM router to clear its neighbor table for a neighbor who has timed out due to neighbor Holdtime expiration.

When a connection is no longer available between two PORT-enabled PIM neighbors, they MUST immediately, or on demand, try to reestablish the connection following the normal rules for connection establishment. The neighbors MUST also start expiry timers so that all oif-list state for the neighbor using the connection gets expired after J/P_Holdtime, unless it later gets refreshed by receiving new Join/Prunes.

The value of J/P_Holdtime is 210 seconds. This value is based on Section 4.11 of [RFC4601], which says that J/P_HoldTime should be 3.5 * t_periodic where the default for t_periodic is 60 seconds.

### 4.4. Moving from PORT to Datagram Mode

There may be situations where an administrator decides to stop using PORT. If PORT is disabled on a router interface, or a previously PORT-enabled neighbor no longer announces any of the PORT Hello Options, the router follows the rules in Section 4.3 for taking down connections and starting timers. Next, the router SHOULD trigger a full state update similar to what would be done if the GenID changed in Datagram mode. The router SHOULD send Join/Prune messages for any state where the router switched from PORT to Datagram mode for the upstream neighbor.

### 4.5. On-Demand versus Pre-Configured Connections

Transport connections could be established when they are needed or when a router interface to other PIM neighbors has come up. The advantage of on-demand transport connection establishment is the reduction of router resources, especially in the case where there is no need for a full mesh of connections on a network interface. The disadvantage is additional delay and queueing when a Join/Prune message needs to be sent and a transport connection is not established yet.

If a router interface has become operational and PIM neighbors are learned from Hello messages, at that time, transport connections may be established. The advantage is that a connection is ready to transport data by the time a Join/Prune message needs to be sent. The disadvantage is there can be more connections established than needed. This can occur when there is a small set of RPF neighbors for the active distribution trees compared to the total number of neighbors. Even when transport connections are pre-established before they are needed, a connection can go down and an implementation will have to deal with an on-demand situation.

Note that for TCP, it is the router with the lower Connection ID that decides whether to open a connection immediately or on demand. The router with the higher Connection ID SHOULD only initiate a connection on demand, that is, if it needs to send a Join/Prune message and there is no currently established connection.

Therefore, this specification RECOMMENDS but does not mandate the use of on-demand transport connection establishment.

### 4.6. Possible Hello Suppression Considerations

Based on this specification, a transport connection cannot be established until a Hello message is received. Reasons for this are to determine if the PIM neighbor supports this specification and to determine the remote address to use for establishing the transport connection.

There are cases where it is desirable to suppress entirely the transmission of Hello messages. In this case, how to determine if the PIM neighbor supports this specification and how to determine out-of-band (i.e., outside of the PIM protocol) the remote address for establishing the transport connection are outside the scope of this document. In this case, the following is outside the scope of this document: how to determine if the PIM neighbor supports this specification as well as an out-of-band (outside of the PIM protocol) method to determine the remote address to establish the transport connection.

### 4.7. Avoiding a Pair of TCP Connections between Neighbors

To ensure that there is only one TCP connection between a pair of PIM neighbors, the following set of rules MUST be followed. Note that this section applies only to TCP; for SCTP, this is not an issue. Let nodes A and B be two PIM neighbors where A's Connection ID is numerically smaller than B's Connection ID, and each is known to the other as having a potential PIM adjacency relationship.

At node A:

- If there is already an established TCP connection to B, on the PIM-over-TCP port, then A MUST NOT attempt to establish a new connection to B. Rather, it uses the established connection to send Join/Prune messages to B. (This is independent of which node initiated the connection.)

- If A has initiated a connection to B, but the connection is still in the process of being established, then A MUST refuse any connection on the PIM-over-TCP port from B.

- At any time when A does not have a connection to B (which is either established or in the process of being established), A MUST accept connections from B.

At node B:

- If there is already an established TCP connection to A on the PIM-over-TCP port, then B MUST NOT attempt to establish a new connection to A. Rather, it uses the established connection to send Join/Prune messages to A. (This is independent of which node initiated the connection.)

- If B has initiated a connection to A, but the connection is still in the process of being established, then if A initiates a connection too, B MUST accept the connection initiated by A and release the connection that it (B) initiated.

## 5. PORT Message Definitions

For scaling purposes, it may be desirable to allow Join/Prune messages from different PIM protocol families to be sent over the same transport connection. Also, it may be desirable to have a set of Join/Prune messages for one address family sent over a transport connection that is established over a different address-family network layer.

To be able to do this, we need a common PORT message format. This will provide both record boundary and demux points when sending over a stream protocol like TCP/SCTP.

A PORT message may contain PORT Options; see Section 5.3. We will define two PORT Options for carrying PIM Join/Prune messages -- one for IPv4 and one for IPv6. For each PIM Join/Prune message to be sent over the transport connection, we send a PORT Join/Prune message containing exactly one such option.

Each PORT message will have the Type/Length/Value format. Multiple different TLV types can be sent over the same transport connection.

To make sure PIM Join/Prune messages are delivered as soon as the TCP transport layer receives the Join/Prune buffer, the TCP Push flag will be set in all outgoing Join/Prune messages sent over a TCP transport connection.

PORT messages will be sent using destination TCP port number 8471. When using SCTP as the reliable transport, destination port number 8471 will be used. See Section 12 for IANA considerations.

PORT messages are error checked. This includes unknown/illegal type fields or a truncated message. If the PORT message contains a PIM Join/Prune Message, then that is subject to the normal PIM error

checks, including checksum verification. If any parsing errors occur in a PORT message, it is skipped, and we proceed to any following PORT messages.

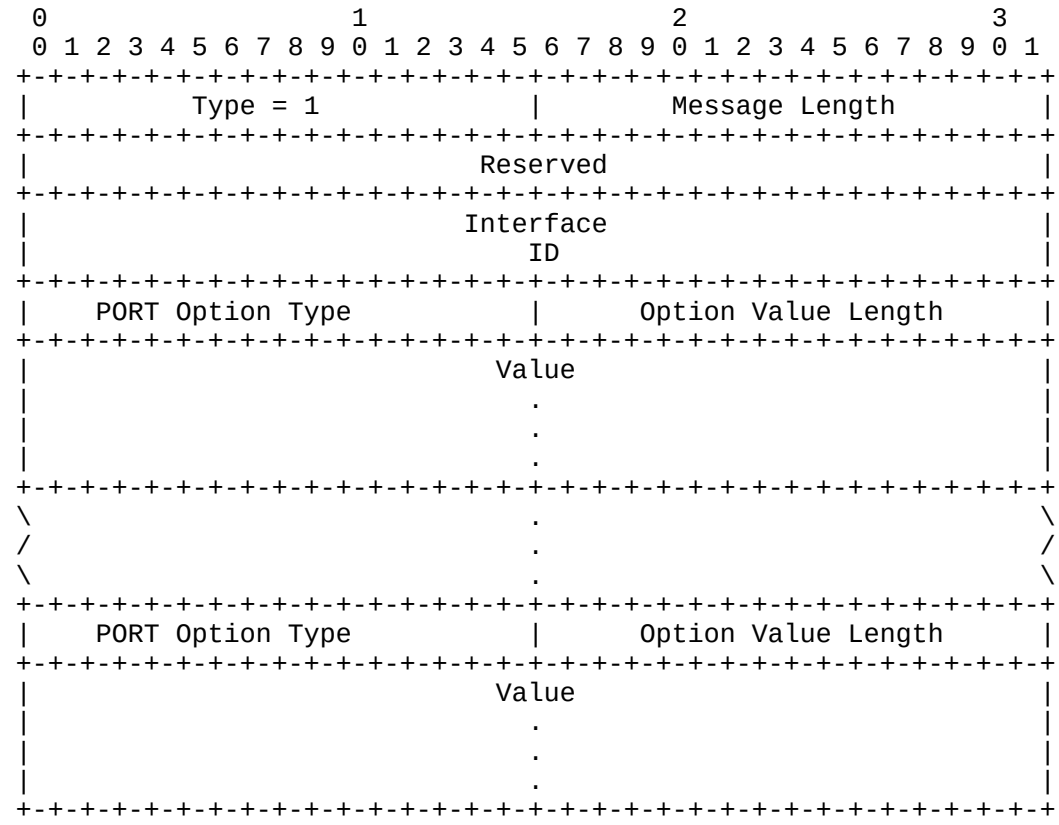
When an unknown type field is encountered, that message MUST be ignored. As specified above, one then proceeds as usual when processing further PORT messages. This is important in order to allow new message types to be specified in the future, without breaking existing implementations. However, if only unknown or invalid messages are received for a longer period of time, an implementation MAY alert the operator. For example, if a message is sent with a wrong length, the receiver is likely to see only unknown/ invalid messages thereafter.

The checksum of the PIM Join/Prune message MUST be calculated exactly as specified in Section 4.9 of [RFC4601]. For IPv6, [RFC4601] specifies the use of a pseudo-header. For PORT, the exact same pseudo-header MUST be used, but its source and destination address fields MUST be set to 0 when calculating the checksum.

The TLV type field is 16 bits. The range 65532 - 65535 is for experimental use [RFC3692].

This document defines two message types.

### 5.1. PORT Join/Prune Message




PORT Join/Prune Message

The PORT Join/Prune Message is used for sending a PIM Join/Prune.

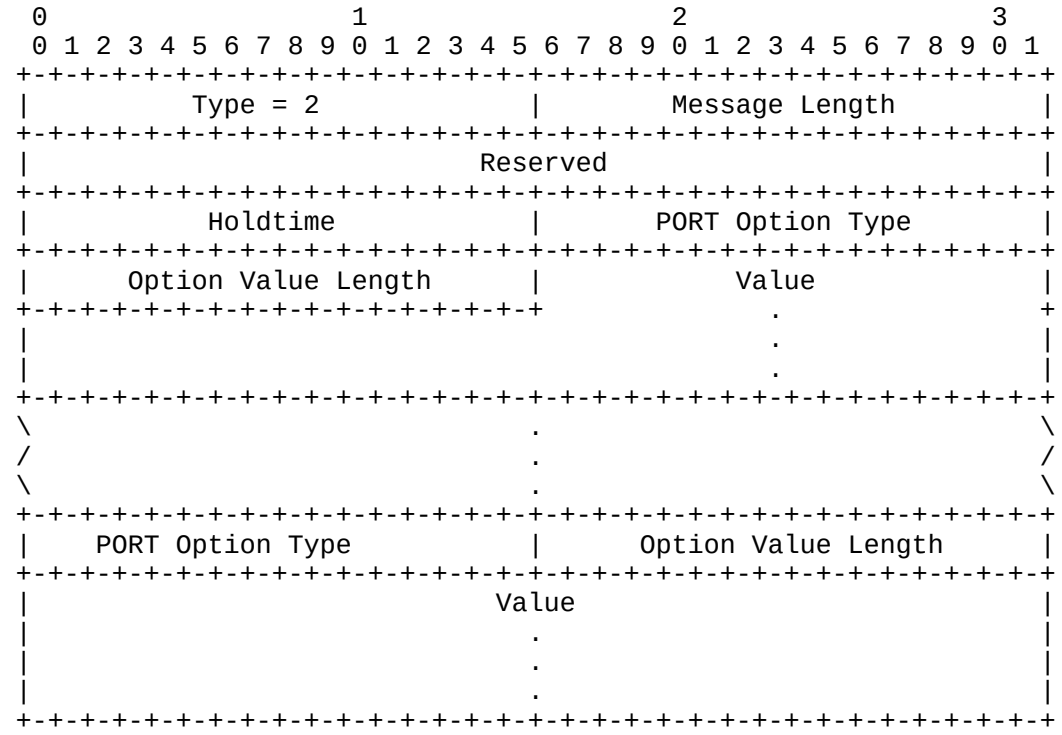
Message Length: Length in bytes for the value part of the Type/Length/Value encoding. If no PORT Options are included, the length is 12. If n PORT Options with Option Value lengths L1, L2, ..., Ln are included, the message length is 12 + 4*n + L1 + L2 + ... + Ln.

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

Interface ID: This MUST be the Interface ID of the Interface ID Hello Option contained in the PIM Hello messages that the PIM router is sending to the PIM neighbor. It indicates to the PIM neighbor what interface to associate the Join/Prune with. The Interface ID allows us to do connection sharing.

PORT Options: The message MUST contain exactly one PIM Join/Prune PORT Option, either one PIM IPv4 Join/Prune or one PIM IPv6 Join/Prune. It MUST NOT contain both. It MAY contain additional options not defined in this document. The behavior when receiving a message containing unknown options depends on the option type. See Section 5.3 for option definitions.

## 5.2. PORT Keep-Alive Message




PORT Keep-Alive Message

The PORT Keep-alive Message is used to regularly send PORT messages to verify that a connection is alive. They are used when other PORT messages are not sent at the desired frequency.

Message Length: Length in bytes for the value part of the Type/Length/Value encoding. If no PORT Options are included, the length is 6. If n PORT Options with Option Value lengths L1, L2, ..., Ln are included, the message length is 6 + 4*n + L1 + L2 + ... + Ln.

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

Holdtime: This specifies a Holdtime in seconds for the connection. A non-zero value means that the connection SHOULD be gracefully shut down if no further PORT messages are received within the specified time. This is measured on the receiving side by measuring the time from when one PORT message has been processed until the next has been processed. Note that this MUST be done for any PORT message, not just keep-alive messages. A Holdtime of 0 disables the keep-alive mechanism.

PORT Options: A keep-alive message MUST NOT contain any of the options defined in this document. It MAY contain other options not defined in this document. The behavior when receiving a message containing unknown options depends on the option type. See Section 5.3 for option definitions.

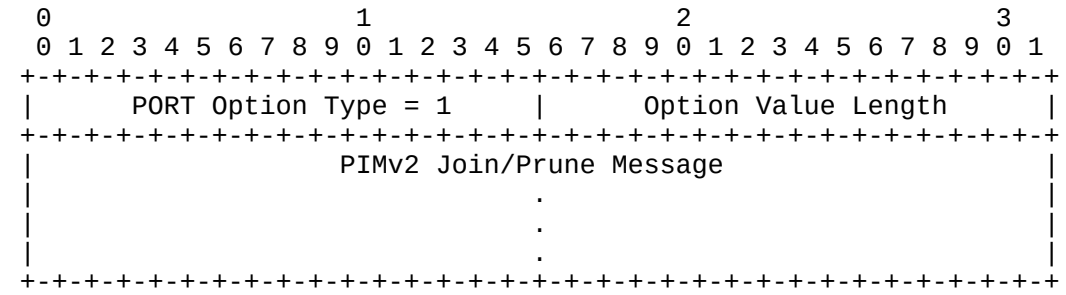
## 5.3. PORT Options

Each PORT Option is a TLV. The type is 16 bits. The PORT Option type space is split in two ranges. The types in the range 0 - 32767 (the most significant bit is not set) are for Critical Options. The types in the range 32768 - 65535 (the most significant bit is set) are for Non-Critical Options.

The behavior of a router receiving a message with an unknown PORT Option is determined by whether the option is a Critical Option. If the message contains an unknown Critical Option, the entire message must be ignored. If the option is Non-Critical, only that particular option is ignored, and the message is processed as if the option was not present.

PORT Option types are assigned by IANA, except the ranges 32764 - 32767 and 65532 - 65535, which are for experimental use [RFC3692]. The length specifies the length of the value in bytes. Below are the two options defined in this document.

### 5.3.1. PIM IPv4 Join/Prune Option



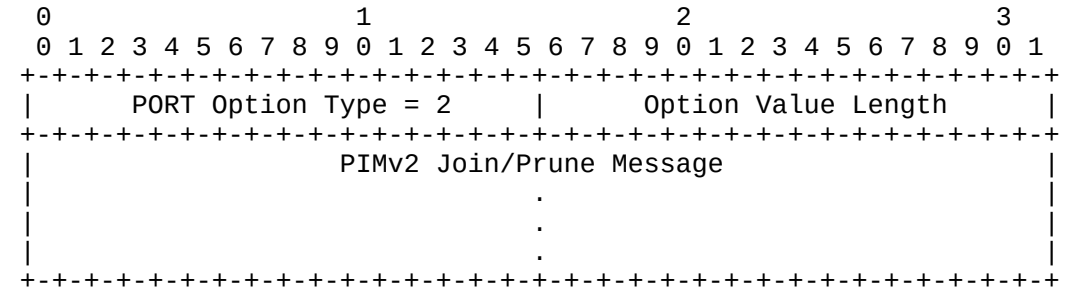

PIM IPv4 Join/Prune Option Format

The IPv4 Join/Prune Option is used to carry a PIMv2 Join/Prune message that has all IPv4-encoded addresses in the PIM payload.

Option Value Length: The number of bytes that make up the PIMv2 Join/Prune message.

PIMv2 Join/Prune Message: PIMv2 Join/Prune message and payload with no IP header in front of it.

### 5.3.2. PIM IPv6 Join/Prune Option




PIM IPv6 Join/Prune Option Format

The IPv6 Join/Prune Option is used to carry a PIMv2 Join/Prune message that has all IPv6-encoded addresses in the PIM payload.

Option Value Length: The number of bytes that make up the PIMv2 Join/Prune message.

PIMv2 Join/Prune Message: PIMv2 Join/Prune message and payload with no IP header in front of it.

## 6. Explicit Tracking

When explicit tracking is used, a router keeps track of Join state for individual downstream neighbors on a given interface. This MUST be done for all PORT Joins and Prunes. Note that it may also be done for native Join/Prune messages, if all neighbors on the LAN have set the T bit of the LAN Prune Delay Option (see definition in Section 4.9.2 of [RFC4601]). The discussion below covers ET (explicit tracking) neighbors and non-ET neighbors. The set of ET neighbors MUST include the PORT neighbors. The set of non-ET neighbors consists of all the non-PORT neighbors, unless all neighbors have set the LAN Prune Delay T bit -- in which case, the ET neighbors set contains all neighbors.

For some link-types, e.g., point-to-point, tracking neighbors is no different than tracking interfaces. It may also be possible for an implementation to treat different downstream neighbors as being on different logical interfaces, even if they are on the same physical link. Exactly how this is implemented, and for which link types, is left to the implementer.

For (*,G) and (S,G) state, the router starts forwarding traffic on an interface when a Join is received from a neighbor on such an interface. When a non-ET neighbor sends a Prune, as specified in [RFC4601], if no Join is sent to override this Prune before the expiration of the Override Timer, the upstream router concludes that no non-ET neighbor is interested. If no ET neighbors are interested, the interface can be removed from the oif-list. When an ET neighbor sends a Prune, one removes the Join state for that neighbor. If no other ET or non-ET neighbors are interested, the interface can be removed from the oif-list. When a PORT neighbor sends a Prune, there can be no Prune Override, since the Prune is not visible to other neighbors.

For (S,G,rpt) state, the router needs to track Prune state on the shared tree. It needs to know which ET neighbors have sent Prunes, and whether any non-ET neighbors have sent Prunes. Normally, one would forward a packet from a source S to a group G out on an interface if a (*,G) Join is received, but no (S,G,rpt) Prune. With ET, one needs to do this check per ET neighbor. That is, the packet should be forwarded except in two cases: all ET neighbors that have sent (*,G) Joins have also sent (S,G,rpt) Prunes, and if a non-ET neighbor has sent a (*,G) Join, whether there also is non-ET (S,G,rpt) Prune state.

## 7. Support of Multiple Address Families

To allow for efficient use of router resources, one can mux Join/Prune messages of different address families on the same transport connection. There are two ways this can be accomplished -- using a common message format over a TCP connection or using multiple streams over a single SCTP connection.

Using the common message format described in this specification, and using different PORT Options, both IPv4- and IPv6-based Join/Prune messages can be encoded within the same transport connection.

When using SCTP multi-streaming, the common message format is still used to convey address-family information, but an SCTP association is used, on a per-family basis, to send data concurrently for multiple families. When data is sent concurrently, head-of-line blocking (which can occur when using TCP) is avoided.

## 8. Miscellany

There are no changes to processing of other PIM messages like PIM Asserts, Grafts, Graft-Acks, Registers, and Register-Stops. This goes for Bootstrap Router (BSR) and Auto-RP type messages as well.

This extension is applicable only to PIM-SM, PIM-SSM, and Bidirectional PIM. It does not take requirements for PIM Dense Mode (PIM-DM) into consideration.

## 9. Transport Considerations

As noted in the introduction, this is an experimental extension to PIM, and using reliable delivery for PIM messages is a new concept. There are several potential transport-related concerns. Hopefully, experiences from early implementations and deployments will reveal what concerns are relevant and how to resolve them.

One consideration is keep-alive mechanisms. We have defined an optional keep-alive mechanism for PORT; see Section 4.2. Also, SCTP and many TCP implementations provide keep-alive mechanisms that could be used. When to use keep-alive messages and which mechanism to use are unclear; however, we believe the PORT Keep-alive allows for better application control. It is unclear what Holdtimes are preferred for the PORT Keep-alives. For now, it is RECOMMENDED that administrators be able to configure whether to use keep-alives, what Holdtimes to use, etc.

In a stable state, it is expected that only occasional small messages are sent over a PORT connection. This depends on how often PIM Join/Prune state changes. Thus, over a long period of time, there may be only small messages that never use the entire TCP congestion window, and the window may become very large. This would then be an issue if there is a state change that makes PORT send a very large message. It may be good if the TCP stack provides some rate-limiting or burst-limiting. The congestion control mechanism defined in [RFC3465] may be of help.

With PORT, it is possible that only occasional small messages are sent (as discussed in the previous paragraph). This may cause problems for the TCP retransmit mechanism. In particular, the TCP Fast Retransmit algorithm may never get triggered. For further discussion of this and a possible solution, see [RFC3042].

There may be SCTP issues similar to the TCP issues discussed in the above two paragraphs.

## 10. Manageability Considerations

This document defines using TCP or SCTP transports between pairs of PIM neighbors. It is recommended that this mechanism be disabled by default. An administrator can then enable PORT TCP and/or SCTP on PIM-enabled interfaces. If two neighbors both have PORT SCTP (or both have PORT TCP), they will only use SCTP (or alternatively, TCP) for PIM Join/Prune messages. This is the case even when the connection is down (there is no fallback to native Join/Prune messages).

When PORT support is enabled, a router sends PIM Hello messages announcing support for TCP and/or SCTP and also Connection IDs. It should be possible to configure a local Connection ID, and also to see what PORT capabilities and Connection IDs PIM neighbors are announcing. Based on these advertisements, pairs of PIM neighbors will decide whether to try to establish a PORT connection. There should be a way for an operator to check the current connection state. Statistics on the number of PORT messages sent and received (including number of invalid messages) may also be helpful.

For connection security (see Section 4.1), it should be possible to enable a GTSM check to only accept connections (TCP/SCTP packets) when the sender is within a certain number of router hops. Also, one should be able to configure the use of TCP-AO.

For connection maintenance (see Section 4.2), it is recommended to support Keep-Alive messages. It should be configurable whether to send Keep-Alives -- and if doing so, whether to use a Holdtime and what Holdtime to use.

There should be some way to alert an operator when PORT connections are going down or when there is a failure in establishing a PORT connection. Also, information like the number of connection failures, and how long the connection has been up or down, is useful.

## 11. Security Considerations

There are several security issues related to the use of TCP or SCTP transports. By sending packets with a spoofed source address, off-path attackers might establish a connection or inject packets into an existing connection. This might allow an attacker to send spoofed Join/Prune messages and/or reset a connection. Mechanisms that help protect against this are discussed in Section 4.1.

For authentication, TCP-AO [RFC5925] may be used with TCP, Authenticated Chunks [RFC4895] may be used with SCTP. Also, GTSM [RFC5082] can be used to help prevent spoofing.

## 12. IANA Considerations

This specification makes use of a TCP port number and an SCTP port number for the use of the pim-port service that has been assigned by IANA. It also makes use of IANA PIM Hello Options assignments that have been made permanent.

### 12.1. PORT Port Number

IANA previously had assigned a port number that is used as a destination port for pim-port TCP and SCTP transports. The assigned number is 8471. References to this document have been added to the Service Name and Transport Protocol Port Number Registry for pim-port.

### 12.2. PORT Hello Options

In the "PIM-Hello Options" registry, the following options have been added for PORT.

| Value | Length | Name | Reference |
|---|---|---|---|
| 27 | Variable | PIM-over-TCP-Capable | this document |
| 28 | Variable | PIM-over-SCTP-Capable | this document |

### 12.3. PORT Message Type Registry

A registry for PORT message types has been created. The message type is a 16-bit integer, with values from 0 to 65535. An RFC is required for assignments in the range 0 - 65531. This document defines two PORT message types: Type 1 (Join/Prune) and Type 2 (Keep-alive). The type range 65532 - 65535 is for experimental use [RFC3692].

The initial content of the registry is as follows:

| Type | Name | Reference |
|---|---|---|
| 0 | Reserved | this document |
| 1 | Join/Prune | this document |
| 2 | Keep-alive | this document |
| 3-65531 | Unassigned | |
| 65532-65535 | Experimental | this document |

### 12.4. PORT Option Type Registry

A registry for PORT Option types. The option type is a 16-bit integer, with values from 0 to 65535. The type space is split in two ranges, 0 - 32767 for Critical Options and 32768 - 65535 for Non-Critical Options. Option types are assigned by IANA, except the ranges 32764 - 32767 and 65532 - 65535 that are for experimental use [RFC3692]. An RFC is required for the IANA assignments. An RFC defining a new option type must specify whether the option is Critical or Non-Critical in order for IANA to assign a type. This document defines two Critical PORT Option types: Type 1 (PIM IPv4 Join/Prune) and Type 2 (PIM IPv6 Join/Prune).

The initial content of the registry is as follows:

| Type | Name | Reference |
|---|---|---|
| 0 | Reserved | this document |
| 1 | PIM IPv4 Join/Prune | this document |
| 2 | PIM IPv6 Join/Prune | this document |
| 3-32763 | Unassigned Critical Options | |
| 32764-32767 | Experimental | this document |
| 32768-65531 | Unassigned Non-Critical Options | |
| 65532-65535 | Experimental | this document |

## 13. Contributors

In addition to the persons listed as authors, significant contributions were provided by Apoorva Karan and Arjen Boers.

## 14. Acknowledgments

The authors would like to give a special thank you and appreciation to Nidhi Bhaskar for her initial design and early prototype of this idea.

Appreciation goes to Randall Stewart for his authoritative review and recommendation for using SCTP.

Thanks also goes to the following for their ideas and review of this specification: Mike McBride, Toerless Eckert, Yiqun Cai, Albert Tian, Suresh Boddapati, Nataraj Batchu, Daniel Voce, John Zwiebel, Yakov Rekhter, Lenny Giuliano, Gorry Fairhurst, Sameer Gulrajani, Thomas Morin, Dimitri Papadimitriou, Bharat Joshi, Rishabh Parekh, Manav Bhatia, Pekka Savola, Tom Petch, and Joe Touch.

A special thank you goes to Eric Rosen for his very detailed review and commentary. Many of his comments are reflected as text in this specification.

## 15. References

### 15.1. Normative References

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

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

[RFC4601] Fenner, B., Handley, M., Holbrook, H., and I. Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", RFC 4601, August 2006.

[RFC4895] Tuexen, M., Stewart, R., Lei, P., and E. Rescorla, "Authenticated Chunks for the Stream Control Transmission Protocol (SCTP)", RFC 4895, August 2007.

[RFC4960] Stewart, R., "Stream Control Transmission Protocol", RFC 4960, September 2007.

[RFC5015] Handley, M., Kouvelas, I., Speakman, T., and L. Vicisano, "Bidirectional Protocol Independent Multicast (BIDIR-PIM)", RFC 5015, October 2007.

[RFC5082] Gill, V., Heasley, J., Meyer, D., Savola, P., and C. Pignataro, "The Generalized TTL Security Mechanism (GTSM)", RFC 5082, October 2007.

[RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, June 2010.

[RFC6395] Gulrajani, S. and S. Venaas, "An Interface Identifier (ID) Hello Option for PIM", RFC 6395, October 2011.

## 15.2. Informative References

[AFI] IANA, "Address Family Numbers", <http://www.iana.org/assignments/address-family-numbers>.

[HELLO-OPT] IANA, "PIM-Hello Options", <http://www.iana.org/assignments/pim-parameters>.

[RFC3042] Allman, M., Balakrishnan, H., and S. Floyd, "Enhancing TCP's Loss Recovery Using Limited Transmit", RFC 3042, January 2001.

[RFC3465] Allman, M., "TCP Congestion Control with Appropriate Byte Counting (ABC)", RFC 3465, February 2003.

[RFC3692] Narten, T., "Assigning Experimental and Testing Numbers Considered Useful", BCP 82, RFC 3692, January 2004.

## Authors' Addresses

Dino Farinacci
Cisco Systems
Tasman Drive
San Jose, CA 95134
USA

EMail: dino@cisco.com

IJsbrand Wijnands
Cisco Systems
Tasman Drive
San Jose, CA 95134
USA

EMail: ice@cisco.com

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

EMail: stig@cisco.com

Maria Napierala
AT&T Labs
200 Laurel Drive
Middletown, New Jersey 07748
USA

EMail: mnapierala@att.com