

Network Working Group
Request for Comments: 5384
Category: Standards Track

A. Boers
I. Wijnands
E. Rosen
Cisco Systems, Inc.
November 2008

The Protocol Independent Multicast (PIM) Join Attribute Format

Status of This Memo

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

Copyright Notice

Copyright (c) 2008 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.

Abstract

A "Protocol Independent Multicast - Sparse Mode" Join message sent by a given node identifies one or more multicast distribution trees that that node wishes to join. Each tree is identified by the combination of a multicast group address and a source address (where the source address is possibly a "wild card"). Under certain conditions it can be useful, when joining a tree, to specify additional information related to the construction of the tree. However, there has been no way to do so until now. This document describes a modification of the Join message that allows a node to associate attributes with a particular tree. The attributes are encoded in Type-Length-Value format.

Table of Contents

1. Introduction	2
2. Specification of Requirements	3
3. Use of Join Attributes	3
3.1. Sending Join Attributes	3
3.2. The Join Attribute Option in the PIM Hello	4
3.3. Receiving Join Attributes	4
3.3.1. General Considerations	4
3.3.2. Transitive and Non-Transitive Attributes	5
3.3.3. Conflicting Attributes	5
3.3.4. Attribute Change	6
3.4. PIM Attribute Packet Format	7
3.4.1. PIM Join Packet Format	7
3.4.2. PIM Join Attribute Hello Option	8
4. IANA Considerations	8
5. Security Considerations	9
6. Acknowledgments	9
7. Normative References	9
8. Informative References	9

1. Introduction

In the protocol known as "Protocol Independent Multicast - Sparse Mode" [RFC4601] (henceforth referred to as "PIM"), a Join message sent by a given node may identify one or more multicast distribution trees that that node wishes to join. Each tree is identified by the combination of a multicast group address and a source address (where the source address is possibly a "wild card"). Under certain conditions it can be useful, when joining a tree, to specify additional information related to the construction of the tree. However, there has been no way to do so until now. This document describes a modification of the Join message that allows a node to associate an attribute, encoded in Type-Length-Value (TLV) format, with a particular tree that it wishes to join. These attributes are known as "PIM Join Attributes".

In the PIM Join message, the Source Address is identified by being encoded as an "Encoded-Source Address" ([RFC4601], section 4.9.1). Each Encoded-Source Address occurs in the context of a particular group address, represented as an "Encoded-Group Address". Together, the Encoded-Source Address and the Encoded-Group Address identify a multicast distribution tree. The Encoded-Source Address contains an "encoding type" field. The only value defined in [RFC4601] is 0. This specification is the first to assign another encoding type value.

In order to associate TLVs with a particular tree, this specification defines a new encoding type for the Encoded-Source Address: type 1. When type 1 is used, the Encoded-Source Address may contain a sequence of "Join Attributes", each of which is encoded as a TLV. Then the type 1 Encoded-Source Address, in the context of the associated Encoded-Group Address, identifies a multicast distribution tree and specifies (via the Join Attribute TLVs) the attributes that apply to the tree. Apart from the fact that the type 1 Encoded-Source Address may contain Join Attributes, it is otherwise identical to the type 0 Encoded-Source Address.

This document does not contain the specification for any particular Join Attribute. It specifies how Join Attributes are to be encoded into the Join messages and it specifies generic procedures that are common to all Join Attributes. The content and purpose of any particular Join Attribute is outside the scope of this document.

The use of Join Attributes in "Protocol Independent Multicast - Dense Mode" [RFC3973] is not considered.

2. Specification of Requirements

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

3. Use of Join Attributes

3.1. Sending Join Attributes

Join Attributes are encoded as TLVs into the Encoded-Source Address field of a PIM Join message, as specified in section 3.4.1 below. Each attribute applies to the same multicast distribution tree that is identified by the combination of the Encoded-Source Address and the associated Encoded-Group Address. The multicast distribution tree may be either a source-specific tree or a shared tree.

The encoding of the "source address" field within the Encoded-Source Address is exactly the same for a type 1 Encoded-Source Address as for a type 0 Encoded-Source Address, specified in [RFC4601].

A type 1 Encoded-Source Address MUST contain at least one Join Attribute. The way to specify that there are no Join Attributes for a particular tree is to use the type 0 Encoded-Source Address.

Multiple Join Attributes of the same type or of different types may occur within a single Encoded-Source Address. This specification does not require all attributes of a given type to occur contiguously. There is no header field that specifies the number of attributes; rather the last attribute is specially marked as such.

Any PIM router that does not understand the type 1 Encoded-Source Address will not be able to process a PIM Join message that contains it. Further, if the use of any particular Join Attribute affects the construction of the multicast distribution tree, the tree may not be formed correctly unless the attribute is understood by all PIM routers that receive it. As a consequence, attributes are only useful within a single administrative domain (or perhaps a small set of contiguous, cooperating administrative domains) where it can be determined a priori that all deployed PIM routers understand the type 1 Encoded-Source Address, as well as whatever specific attributes are in use.

3.2. The Join Attribute Option in the PIM Hello

To ensure that a type 1 Encoded-Source Address is not sent to a PIM neighbor that does not understand this encoding, a new PIM Hello option, the "Join Attribute" option, is defined. This option **MUST** be included in the PIM Hellos of any PIM router that is willing to receive type 1 Encoded-Source Address. A PIM router **MUST NOT** send a type 1 Encoded-Source Address out any interface on which there is a PIM neighbor that has not included this option in its Hellos. (Even a router that is not the upstream neighbor must be able parse the packet in order to do Join suppression or overriding.)

Note that a PIM router that sends the "Join Attribute" Hello option does not necessarily understand every possible attribute type. As there is no immediate way to act on a neighbor's inability to process certain attribute types, it is not desired to have a Hello option for each possible attribute type.

3.3. Receiving Join Attributes

3.3.1. General Considerations

A PIM router that receives a type 1 Encoded-Source Address **MUST** examine all the attributes in the order in which they appear.

The specification for a given attribute type **MUST** specify the procedure to apply if there are multiple instances of that attribute type.

Processing an attribute may affect the following:

- the construction of the associated multicast distribution tree,
- the processing of other attributes of the same type that also occur in the type 1 Encoded-Source Address, and
- the forwarding (or not) of the attribute itself and/or other attributes of the same type that also occur in the type 1 Encoded-Source Address.

If the processing of a received attribute has any effect on the construction of the multicast distribution tree or on the set of attributes that are forwarded up the tree, then state **MUST** be maintained associating the received attribute with the adjacency or adjacencies from which it was received.

3.3.2. Transitive and Non-Transitive Attributes

If a PIM router understands a particular attribute type, the attribute is processed as specified above.

If a PIM router does not understand the type of a particular attribute, the PIM router either forwards that attribute or discards it, depending upon the setting of the attribute's F-bit. If the F-bit is set, then the router **MUST** forward the attribute; if the F-bit is clear, then the router **MUST** discard it.

If one or more non-transitive attributes are discarded, the rest of the Join Attributes (if any) are still forwarded. If there are no Join Attributes left to forward, a Join with a type 0 Encoded-Source Address field **MUST** be forwarded.

3.3.3. Conflicting Attributes

It is possible that a router receives conflicting attribute information from different downstream routers. Conflicts only occur with attributes of the same type.

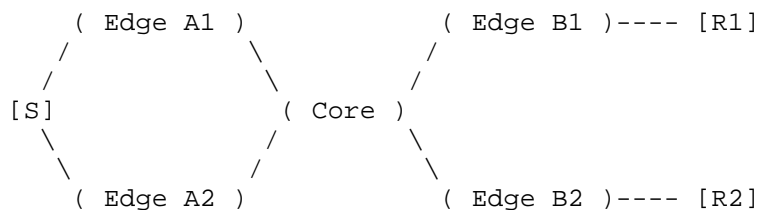


Figure 1

As an example, consider Figure 1 and suppose a Join Attribute is used to indicate a choice of exit router. There are 2 receivers for the same group connected to Edge B1 and B2. Suppose that edge router B1 prefers A1 and B2 prefers A2 as exit points to reach the source S. If both Edge B1 and B2 send a Join including an attribute to prefer their exit router in the network and they cross the same core router, the core router will get conflicting attribute information for the source. If this happens, we use the attribute from the PIM adjacency with the numerically smallest IP address. In the case of IPv6, the link local address will be used. When two neighbors have the same IP address, either for IPv4 or IPv6, the interface index MUST be used as a tie breaker. The attributes from other sending routers MAY be remembered; that way, if the adjacency that supplied the selected attribute gets pruned or expires, we are able to immediately use the attribute that was sent by the adjacency that is next in the order of preference. This enables us to converge quickly without waiting for the next periodic update.

When a particular attribute type is specified, the specification MAY include a conflict resolution procedure specific to that type. If so, that conflict resolution procedure MUST be used instead of the procedure described in this section.

It is possible that a router will receive, from two different adjacencies, transitive attributes of a given type. If the router does not understand attributes of that type and if the two adjacencies have not sent the exact same set of attributes of that type, then the conflict resolution procedure described in this section MUST be applied to those attributes. Two adjacencies are said to have sent the exact same set of attributes of a given type if they have sent the same number of instances of that attribute and if corresponding instances are byte-for-byte identical.

3.3.4. Attribute Change

A PIM router may decide to change the set of attributes it has associated with a given multicast distribution tree. This can happen if one of its downstream neighbors on the tree has changed the set of attributes. It can also happen as a result of processing the attributes. It can also happen for reasons outside the scope of this specification (such as a change in configuration).

If a PIM router needs to change the set of attributes for a given tree but does not change its upstream neighbor for that tree, it MUST send a new Join for that tree, specifying the new set of attributes. If the new set of attributes is the null set, the type 0 Encoded-Source format MUST be used.

If a PIM router needs to change the set of attributes for a given tree and as a result changes its upstream neighbor for that tree, it sends a Prune to the old upstream neighbor. The Prune does not need to carry any attributes.

When a PIM router receives a Join for a given tree and the Join does not contain exactly the same set of attributes as the prior Join, the set of attributes in the new Join becomes the entire new set of attributes. No attribute information from the prior Join is retained. There is no way to advertise incremental changes to the set of attributes; any attributes that are no longer present are considered to have been withdrawn. If, as the result of receiving a Join, a PIM router determines that the set of attributes has changed, it will need to send a new Join upstream that contains the new set of attributes. (Of course, the procedures for resolving attribute conflicts may need to be applied first.)

When a PIM router R1 receives a Prune for a given tree from a given downstream neighbor R2, where R2 had previously sent attributes applying to that tree, those attributes are considered to have been withdrawn. Depending on the attributes that R1 has received from its other downstream neighbors (if any) on the tree, R1 may determine that the set of attributes applying to the tree has changed, in which case it needs to send a new Join, with the new attribute set, to its upstream neighbor on the tree.

3.4. PIM Attribute Packet Format

3.4.1. PIM Join Packet Format

There is no space in the default PIM source encoding to include an attribute field. Therefore we introduce a new source encoding type. The attributes are formatted as TLVs. The new Encoded-Source Address looks like this:

```

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
+-----+-----+-----+-----+-----+-----+-----+-----+
| Addr Family | Encoding Type | Rsrvd  | S|W|R| Mask Len |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Source Address
+-----+-----+-----+-----+-----+-----+-----+-----+
|F|E| Attr_Type | Length           | Value
+-----+-----+-----+-----+-----+-----+-----+-----+
|F|E| Attr_Type | Length           | Value
+-----+-----+-----+-----+-----+-----+-----+-----+
:                                     :
:                                     :
:                                     :
```

- Encoding Type: 1
- F-bit, Transitive Attribute. If this bit is set, the attribute is a transitive attribute; otherwise, it is a non-transitive attribute. See section 3.3.2.
- E-bit, End of Attributes. If this bit is set, then this is the last Join Attribute appearing in this Encoded-Source Address field.
- "Attr_Type", a 6-bit field identifying the type of the Attribute.
- Length field, a 1-octet field specifying the length in octets, encoded as an unsigned binary integer, of the value field.

The other fields are the same as described in [RFC4601].

3.4.2. PIM Join Attribute Hello Option

```

      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
+-----+-----+-----+-----+-----+-----+-----+-----+
|      OptionType = 26      |      OptionLength = 0      |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

- Option type: 26.

4. IANA Considerations

A new IANA registry has been created for "PIM Join Attribute Types". These are values of the "Attr_Type" field depicted in section 3.4.1. Assignments are to be made according to the policy "IETF Review" as defined in [RFC5226].

IANA has assigned the PIM Hello option value 26 to the "Join Attribute" option, with this document as the reference.

[RFC4601] should have, but did not, create a registry for the "Encoding Type" field of the Encoded-Source Address format defined therein. IANA has set up a registry for this, referencing both this document and [RFC4601]. Assignments should be made according to the policy "IETF Review" as defined in [RFC5226]. Two encoding types are defined:

- The encoding type 0 has been allocated, defined as "native encoding for the address family", and [RFC4601] is the reference.

- The encoding type 1 has been allocated, defined as "native encoding for the address family, but with zero or more PIM Join Attributes present", and this document is the reference.

5. Security Considerations

Security of the Join Attribute is only guaranteed by the security of the PIM packet, so the security considerations for PIM Join packets as described in [RFC4601] apply here. Additional security considerations may apply to specific attributes; if so, these will need to be documented in the specification of those attributes.

Security considerations from [RFC5015] may apply as well.

6. Acknowledgments

The authors would like to thank Stig Venaas, James Lingard, Bharat Joshi, Marshall Eubanks, Pekka Savola, Tom Pusateri, and Elwyn Davies for their input.

7. Normative References

- [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.

8. Informative References

- [RFC3973] Adams, A., Nicholas, J., and W. Siadak, "Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised)", RFC 3973, January 2005.
- [RFC5015] Handley, M., Kouvelas, I., Speakman, T., and L. Vicisano, "Bidirectional Protocol Independent Multicast (BIDIR-PIM)", RFC 5015, October 2007.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

Authors' Addresses

Arjen Boers
Cisco Systems, Inc.
Avda. Diagonal, 682
Barcelona 08034

EMail: aboers@cisco.com

IJsbrand Wijnands
Cisco Systems, Inc.
De kleetlaan 6a
Diegem 1831
Belgium

EMail: ice@cisco.com

Eric C. Rosen
Cisco Systems, Inc.
1414 Massachusetts Avenue
Boxborough, MA, 01719

EMail: erosen@cisco.com

