INTERNET-DRAFT R. Meng
Intended Status: Informational Huawei Technologies
Expires: May 20, 2018 November 20, 2017

An Enhancement of PIM-SM

draft-meng-pim-sm-enhancement-01.txt

Abstract

This document specifies an enhanced version of PIM-SM which works without requiring whole network deployment.

Status of this Memo

Copyright and License Notice

Table of Contents

1. Introduction

PIM-SM is a multicast routing protocol that can use the underlying unicast routing information base or a separate multicast-capable routing information base. It builds unidirectional shared trees rooted at a Rendezvous Point (RP) per group, and it optionally creates shortest-path trees per source.

However, PIM-SM must be deployed contiguously in the whole network, because a joining router can not join into a tree if the upstream neighbor does not support PIM-SM.

1.1. Terminology

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

2. Problem Description

PIM-SM protocol works generally as follows:

1)Multicast receivers express their interests in receiving traffic destined for multicast by using IGMP, MLD or other mechanisms.

2)One of a receiver's local routers is elected as the Designated Router (DR) for that subnet. On receiving the receiver's expression of interest, the DR then sends a PIM Join message towards the RP/Source for that multicast group. The Join message travels hop-by-hop towards the RP/Multicast source for the group, and in each router it passes through, multicast tree state for group G is instantiated. Eventually, the Join message either reaches the RP/Multicast source or reaches a router that already has Join state for that group.

3)In RFC 7761, all join/prune messages are multicast with TTL 1 to the 'ALL-PIM-ROUTERS' group, a message receiver would compare Unicast Upstream Neighbor Address carried in the message with the address of itself, the message will be processed only if the two addresses are the same.

As long as there is one router does not support PIM in the path from a joining router to the target RP/Source, the router can not join into the RPT/SPT.

3. Compatible Scheme

This scheme is based on RFC 7761.

New unicast Join/Prune messages(and their process procedures) will be introduced in this scheme and they will coexist with old Join/Prune messages.

3.1. Sending Join/Prune Messages

PIM-SM routers send join messages to join into multicast groups, send prune messages to leave multicast groups.

If upstream neighbor is a PIM-SM neighbor, old join/prune messages should be sent by the joining/pruning router even if unicast join/prune messages are being received.

Otherwise, new unicast join/prune messages should be sent as below:

1)If an RPT is being joined/pruned, the destination address of the unicast join/prune message should be the RP address, the source address of the unicast join/prune message should be the address of the joining/pruning router.

2)If an SPT is being joined/pruned, the destination address of the unicast join/prune message should be the multicast source address, the source address of the unicast join/prune message should be the address of the joining/pruning router, and there is no Joined/Pruned Source Address field in the message.

3.2. Receiving Join Messages

Both old join messages and new unicast join messages could be received:

1)Old Join messages can only be received by PIM-SM neighbors of the sender, they should be processed according to RFC7761.

2)Unicast Join messages could be received by PIM routers(other than RP/Multicast Source) through ACL or similar means, they could also be received by the destination(RP/Multicast Source) of the messages, receivers should create tunnels from themselves to senders along with new states.

Join messages should be processed as below in detail:

```
join_msg_arrives(msg) {

    if (msg.dst == 224.0.0.13) {

        //The message should be processed according to RFC 7761
```

```
       } else {

           S = multicast_source(msg);

           state = S ? get_state(*, G) : get_state(S, G);

           if (!state) {

               state = S ? new_state(*, G) : new_state(S, G);

               if (msg.dst != self_addr) {

                   if (upstream_neighbor_is_a_PIM-SM_neighbor) {

                       send_multicast_join_message;

                   } else {

                       new_msg = msg;

                       new_msg.src = OIF(new_msg).addr;

                       send(new_msg);

                   }

               }

           }

           add_IF_to_olist(state, create_tunnel(IIF(msg).addr,
   msg.src));

       }

   }

   add_IF_to_olist(state, IF) {

       if (/*IF is in state's olist*/){

           return;

       }

       add(state.olist, IF);

   }
```

```
IIF(msg) {

    return the input interface of msg;

}

OIF(msg) {

    return the output interface of msg;

}
```

3.3. Receiving Prune Messages

Old Prune messages should be processed according to RFC7761.

New Prune messages would be intercepted by PIM routers or be received be RP/Source, they should be processed as below.

```
prune_msg_arrives(msg) {

    if (msg.dst == 224.0.0.13) {

        //The message should be processed according to RFC 7761

    } else {

        S = multicast_source(msg);

        state = S ? get_state(*, G) : get_state(S, G);

        if (state) {

            IIF = tunnel(IIF(msg).addr, msg.src) ?
tunnel(IIF(msg).addr, msg.src) : IIF(msg);

            delete_IF_from_olist(state, IIF);

            if (state.olist_num == 0) {

                delete_state(state);

                if (msg.dst != self_addr) {

                    if (upstream_neighbor_is_a_PIM-SM_neighbor) {

                        send_multicast_prune_message;
```

```
                       } else {
                           new_msg = msg;
                           new_msg.src = OIF(new_msg).addr;
                           send(new_msg);
                       }
                   }
               }
           } else if (msg.dst != self_addr) {
               forward(msg);
           } else {
               //The prune message should be ignored
           }
       }
   }
   delete_IF_from_olist(state, IF) {
       if (/*IF is not in state's olist*/){
           return;
       }
       delete(state.olist, IF);
   }
   IIF(msg) {
       return the input interface of msg;
   }
   OIF(msg) {
```

```
        return the output interface of msg;

    }
```

4. Clean Slate Scheme

This scheme is a modification of RFC 7761:

1)Neighbor relationship between PIM routers will no longer be maintained.

2)Join/Prune messages(and their process procedures) in RFC 7761 will be replaced by Join/Prune messages introduced in this section.

4.1. Sending Join/Prune Messages

Join/Prune messages will no longer be multicast with TTL 1 to the 'ALL-PIM-ROUTERS' group, they will be unicast as below:

1)If an RPT is being joined/pruned, the destination address of the join/prune message should be the RP address, the source address of the join/prune message should be the address of the joining/pruning router.

2)If an SPT is being joined/pruned, the destination address of the join/prune message should be the multicast source address, the source address of the join/prune message should be the address of the joining/pruning router, and there is no Joined/Pruned Source Address field in the message.

4.2. Receiving Join Messages

Join messages could be received by PIM routers(other than RP/Multicast Source) through ACL or similar means, they could also be received by the destination(RP/Multicast Source) of the messages.

A receiver should create tunnel from itself to the sender along with new state only if it is the sender's neighbor which can be identified by TTL in IPv4 packet or Hop Limit in IPv6 packet.

Join messages would be intercepted by PIM routers or be received be RP/Source, they should be processed as below:

```
    join_msg_arrives(msg) {

        S = multicast_source(msg);

        state = S ? get_state(*, G) : get_state(S, G);
```

```
        if (!state) {

            state = S ? new_state(*, G) : new_state(S, G);

            if (msg.dst != self_addr) {

                new_msg = msg;

                new_msg.src = OIF(new_msg).addr;

                new_msg.ttl = 255;

                send(new_msg);

            }

        }

        IIF = (msg.ttl == 255) ? IIF(msg) : create_tunnel(IIF(msg).addr,
    msg.src);

        add_IF_to_olist(state, IIF);

    }

    add_IF_to_olist(state, IF) {

        if (/*IF is in state's olist*/){

            return;

        }

        add(state.olist, IF);

    }

    IIF(msg) {

        return the input interface of msg;

    }

    OIF(msg) {

        return the output interface of msg;

    }
```

4.3. Receiving Prune Messages

Prune messages would be intercepted by PIM routers or be received be RP/Source, they should be processed as below:

```
prune_msg_arrives(msg) {

    S = multicast_source(msg);

    state = S ? get_state(*, G) : get_state(S, G);

    if (state) {

        IIF = tunnel(IIF(msg).addr, msg.src) ? tunnel(IIF(msg).addr,
msg.src) : IIF(msg);

        delete_IF_from_olist(state, IIF);

        if (state.olist_num == 0) {

            delete_state(state);

            if (msg.dst != self_addr) {

                new_msg = msg;

                new_msg.src = OIF(new_msg).addr;

                send(new_msg);

            }

        }

    } else if (msg.dst != self_addr) {

        forward(msg);

    } else {

        //The prune message should be ignored

    }

}

delete_IF_from_olist(state, IF) {
```

```
        if (/*IF is not in state's olist*/){

            return;

        }

        delete(state.olist, IF);

    }

    IIF(msg) {

        return the input interface of msg;

    }

    OIF(msg) {

        return the output interface of msg;

    }
```

5. Packet Formats

There is only one modification about packet formats:

If an SPT is being joined/pruned, there will be no Joined/Pruned Source Address field in the joined/pruned message, and the Number of Joined Sources in the message is 1.

6. Security Considerations

To be perfected.

7. IANA Considerations

There is no IANA consideration in this specification.

8. References

8.1. Normative References

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

[RFC2119]   Bradner, S., "Key words for use in RFCs to Indicate

Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>..

[RFC7761] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I., Parekh, R., Zhang, Z., and L. Zheng, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", STD 83, RFC 7761, DOI 10.17487/RFC7761, March 2016, <http://www.rfc-editor.org/info/rfc7761>.

Authors' Addresses

Rui Meng
Huawei Technologies Co., Ltd
Huawei Campus, 156 Beiqing Road, Hai-dian District
Beijing 100089
China

EMail: mengrui@huawei.com