Network Working Group
Internet-Draft
Expires: October 20, 2012

M. Xu
Y. Cui
S. Yang
J. Wu
Tsinghua University
C. Metz
G. Shepherd
Cisco Systems

April 18, 2012

Softwire Mesh Multicast
draft-ietf-softwire-mesh-multicast-02

Abstract

The Internet needs to support IPv4 and IPv6 packets. Both address families and their attendant protocol suites support multicast of the single-source and any-source varieties. As part of the transition to IPv6, there will be scenarios where a backbone network running one IP address family internally (referred to as internal IP or I-IP) will provide transit services to attached client networks running another IP address family (referred to as external IP or E-IP). It is expected that the I-IP backbone will offer unicast and multicast transit services to the client E-IP networks.

Softwires Mesh is a solution to E-IP unicast and multicast support across an I-IP backbone. This document describes the mechanisms for supporting Internet-style multicast across a set of E-IP and I-IP networks supporting softwires mesh.

Status of this Memo

Table of Contents

1. Introduction

The Internet needs to support IPv4 and IPv6 packets. Both address families and their attendant protocol suites support multicast of the single-source and any-source varieties. As part of the transition to IPv6, there will be scenarios where a backbone network running one IP address family internally (referred to as internal IP or I-IP) will provide transit services to attached client networks running another IP address family (referred to as external IP or E-IP).

The preferred solution is to leverage the multicast functions, inherent in the I-IP backbone, to efficiently and scalably tunnel encapsulated client E-IP multicast packets inside an I-IP core tree, which roots at one or more ingress AFBR nodes and branches out to one or more egress AFBR leaf nodes.

[6] outlines the requirements for the softwires mesh scenario including the multicast. It is straightforward to envisage that client E-IP multicast sources and receivers will reside in different client E-IP networks connected to an I-IP backbone network. This requires that the client E-IP source-rooted or shared tree should traverse the I-IP backbone network.

One method to accomplish this is to re-use the multicast VPN approach outlined in [10]. MVPN-like schemes can support the softwire mesh scenario and achieve a "many-to-one" mapping between the E-IP client multicast trees and transit core multicast trees. The advantage of this approach is that the number of trees in the I-IP backbone network scales less than linearly with the number of E-IP client trees. Corporate enterprise networks and by extension multicast VPNs have been known to run applications that create a large amount of (S,G) states. Aggregation at the edge contains the (S,G) states that need to be maintained by the network operator supporting the customer VPNs. The disadvantage of this approach is the possible inefficient bandwidth and resource utilization when multicast packets are delivered to a receiver AFBR with no attached E-IP receiver.

Internet-style multicast is somewhat different in that the trees tends to be relatively sparse and source-rooted. The need for multicast aggregation at the edge (where many customer multicast trees are mapped into a few or one backbone multicast trees) does not exist and to date has not been identified. Thus the need for a basic or closer alignment with E-IP and I-IP multicast procedures emerges.

A framework on how to support such methods is described in [8]. In this document, a more detailed discussion supporting the "one-to-one" mapping schemes for the IPv6 over IPv4 and IPv4 over IPv6 scenarios will be discussed.

2. Terminology

An example of a softwire mesh network supporting multicast is illustrated in Figure 1. A multicast source S is located in one E-IP client network, while candidate E-IP group receivers are located in the same or different E-IP client networks that all share a common I-IP transit network. When E-IP sources and receivers are not local to each other, they can only communicate with each other through the I-IP core. There may be several E-IP sources for some multicast group residing in different client E-IP networks. In the case of shared trees, the E-IP sources, receivers and RPs might be located in different client E-IP networks. In the simple case the resources of the I-IP core are managed by a single operator although the inter-provider case is not precluded.

```
                 ._._._._.            ._._._._.
                |         |          |         |   --------
                |  E-IP   |          |  E-IP   |--|Source S|
                | network |          | network |   --------
                 ._._._._.            ._._._._.
                    |                    |
                   AFBR             upstream AFBR
                    |                    |
                  __+____________________+__
                 /   :   :           :   :  \
                |    :      :      :     :   |  E-IP Multicast
                |    : I-IP transit core :   |  message should
                |    :     :       :     :   |  get across the
                |    :   :            :  :   | I-IP transit core
                 \_._._._._._._._._._._._._./
                     +                   +
                downstream AFBR    downstream AFBR
                     |                   |
                  ._._._._            ._._._._
     --------    |        |          |        |   --------
    |Receiver|-- |  E-IP  |          |  E-IP  |--|Receiver|
     --------    |network |          |network |   --------
                  ._._._._            ._._._._
```

Figure 1: Softwire Mesh Multicast Framework

Terminology used in this document:

o Address Family Border Router (AFBR) - A dual-stack router interconnecting two or more networks using different IP address families. In the context of softwire mesh multicast, the AFBR runs

E-IP and I-IP control planes to maintain E-IP and I-IP multicast states respectively and performs the appropriate encapsulation/ decapsulation of client E-IP multicast packets for transport across the I-IP core. An AFBR will act as a source and/or receiver in an I-IP multicast tree.

o Upstream AFBR: The AFBR router that is located on the upper reaches of a multicast data flow.

o Downstream AFBR: The AFBR router that is located on the lower reaches of a multicast data flow.

o I-IP (Internal IP): This refers to the form of IP (i.e., either IPv4 or IPv6) that is supported by the core (or backbone) network. An I-IPv6 core network runs IPv6 and an I-IPv4 core network runs IPv4.

o E-IP (External IP): This refers to the form of IP (i.e. either IPv4 or IPv6) that is supported by the client network(s) attached to the I-IP transit core. An E-IPv6 client network runs IPv6 and an E-IPv4 client network runs IPv4.

o I-IP core tree: A distribution tree rooted at one or more AFBR source nodes and branched out to one or more AFBR leaf nodes. An I-IP core tree is built using standard IP or MPLS multicast signaling protocols operating exclusively inside the I-IP core network. An I-IP core tree is used to tunnel E-IP multicast packets belonging to E-IP trees across the I-IP core. Another name for an I-IP core tree is multicast or multipoint softwire.

o E-IP client tree: A distribution tree rooted at one or more hosts or routers located inside a client E-IP network and branched out to one or more leaf nodes located in the same or different client E-IP networks.

o uPrefix64: The /96 unicast IPv6 prefix for constructing IPv4-embedded IPv6 source address.

3. Scenarios of Interest

This section describes the two different scenarios where softwires mesh multicast will apply.

3.1. IPv4-over-IPv6

```
                   ._._._._.            ._._._._.
                  |  IPv4   |          |  IPv4   |   --------
                  | Client  |          | Client  |--|Source S|
                  | network |          | network |   --------
                   ._._._._.            ._._._._.
                      |                    |
                     AFBR             upstream AFBR(A)
                      |                    |
                    __+____________________+__
                   /   :   :           :   :  \
                  |    :      :      :     :   |
                  |    : IPv6 transit core :   |
                  |    :     :       :     :   |
                  |    :   :            :  :   |
                   \_._._._._._._._._._._._._./
                       +                   +
                  downstream AFBR(C)  downstream AFBR(D)
                       |                   |
                    ._._._._            ._._._._
       --------    |  IPv4  |          |  IPv4  |   --------
      |Receiver|-- | Client |          | Client |--|Receiver|
       --------    | network|          | network|   --------
                    ._._._._            ._._._._
```

Figure 2: IPv4-over-IPv6 Scenario

In this scenario, the E-IP client networks run IPv4 and I-IP core runs IPv6. This scenario is illustrated in Figure 2.

Because of the much larger IPv6 group address space, it will not be a problem to map individual client E-IPv4 tree to a specific I-IPv6 core tree. This simplifies operations on the AFBR because it becomes possible to algorithmically map an IPv4 group/source address to an IPv6 group/source address and vice-versa.

The IPv4-over-IPv6 scenario is an emerging requirement as network operators build out native IPv6 backbone networks. These networks naturally support native IPv6 services and applications but it is with near 100% certainty that legacy IPv4 networks handling unicast

and multicast should be accommodated.

3.2. IPv6-over-IPv4

```
                    ._._._._.            ._._._._.
                   |  IPv6   |          |  IPv6   |   --------
                   | Client  |          | Client  |--|Source S|
                   | network |          | network |   --------
                    ._._._._.            ._._._._.
                       |                    |
                      AFBR             upstream AFBR
                       |                    |
                     __+____________________+__
                    /   :   :           :   :  \
                   |    :      :      :     :   |
                   |    : IPv4 transit core :   |
                   |    :     :       :     :   |
                   |    :   :            :  :   |
                    \_._._._._._._._._._._._._./
                        +                   +
                   downstream AFBR    downstream AFBR
                        |                   |
                     ._._._._            ._._._._
        --------    |  IPv6  |          |  IPv6  |   --------
       |Receiver|-- | Client |          | Client |--|Receiver|
        --------    | network|          | network|   --------
                     ._._._._            ._._._._
```

Figure 3: IPv6-over-IPv4 Scenario

In this scenario, the E-IP Client Networks run IPv6 while the I-IP core runs IPv4. This scenario is illustrated in Figure 3.

IPv6 multicast group addresses are longer than IPv4 multicast group addresses. It will not be possible to perform an algorithmic IPv6 - to - IPv4 address mapping without the risk of multiple IPv6 group addresses mapped to the same IPv4 address resulting in unnecessary bandwidth and resource consumption. Therefore additional efforts will be required to ensure that client E-IPv6 multicast packets can be injected into the correct I-IPv4 multicast trees at the AFBRs. This clear mismatch in IPv6 and IPv4 group address lengths means that it will not be possible to perform a one-to-one mapping between IPv6 and IPv4 group addresses unless the IPv6 group address is scoped.

As mentioned earlier, this scenario is common in the MVPN environment. As native IPv6 deployments and multicast applications

emerge from the outer reaches of the greater public IPv4 Internet, it is envisaged that the IPv6 over IPv4 softwire mesh multicast scenario will be a necessary feature supported by network operators.

4. IPv4-over-IPv6

4.1. Mechanism

Routers in the client E-IPv4 networks contain routes to all other client E-IPv4 networks. Through the set of known and deployed mechanisms, E-IPv4 hosts and routers have discovered or learned of (S,G) or (*,G) IPv4 addresses. Any I-IPv6 multicast state instantiated in the core is referred to as (S',G') or (*,G') and is certainly separated from E-IP multicast state.

Suppose a downstream AFBR receives an E-IPv4 PIM Join/Prune message from the E-IPv4 network for either an (S,G) tree or a (*,G) tree. The AFBR can translate the E-IPv4 PIM message into an I-IPv6 PIM message with the latter being directed towards I-IP IPv6 address of the upstream AFBR. When the I-IPv6 PIM message arrives at the upstream AFBR, it should be translated back into an E-IPv4 PIM message. The result of these actions is the construction of E-IPv4 trees and a corresponding I-IP tree in the I-IP network.

In this case it is incumbent upon the AFBR routers to perform PIM message conversions in the control plane and IP group address conversions or mappings in the data plane. It becomes possible to devise an algorithmic one-to-one IPv4-to-IPv6 address mapping at AFBRs.

4.2. Group Address Mapping

For IPv4-over-IPv6 scenario, a simple algorithmic mapping between IPv4 multicast group addresses and IPv6 group addresses is supported. [11] has already defined an applicable format. Figure 4 is the reminder of the format:

```
|   8    |  4 |  4 |     16     |  4 |        60         |    32    |
+--------+----+----+-----------+----+------------------+----------+
|11111111|0011|scop|00.......00|64IX|   sub-group-id   |v4 address|
+--------+----+----+-----------+----+------------------+----------+
                                              +-+-+-+-+
IPv4-IPv6 Interconnection bits (64IX):        |M|r|r|r|
                                              +-+-+-+-+
```

Figure 4: IPv4-Embedded IPv6 Multicast Address Format: SSM Mode

The high order bits of the I-IPv6 address range will be fixed for mapping purposes. With this scheme, each IPv4 multicast address can be mapped into an IPv6 multicast address(with the assigned prefix),

and each IPv6 multicast address with the assigned prefix can be mapped into IPv4 multicast address.

4.3. Source Address Mapping

There are two kinds of multicast --- ASM and SSM. Considering that I-IP network and E-IP network may support different kind of multicast, the source address translation rules could be very complex to support all possible scenarios. But since SSM can be implemented with a strict subset of the PIM-SM protocol mechanisms [5], we can treat I-IP core as SSM-only to make it as simple as possible, then there remains only two scenarios to be discussed in detail:

o E-IP network supports SSM

One possible way to make sure that the translated I-IPv6 PIM message reaches upstream AFBR is to set S' to a virtual IPv6 address that leads to the upstream AFBR. Figure 5 is the recommended address format based on [9]:

```
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| 0-------------32--40--48--56--64--72--80--88--96-----------127|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|     prefix    |v4(32)         | u | suffix    |source address |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|<------------------uPrefix64------------------>|
```

Figure 5: IPv4-Embedded IPv6 Virtual Source Address Format

In this address format, the "prefix" field contains a "Well-Known" prefix or an ISP-defined prefix. An existing "Well-Known" prefix is 64:ff9b, which is defined in [9]; "v4" field is the IP address of one of upstream AFBR's E-IPv4 interfaces; "u" field is defined in [4], and MUST be set to zero; "suffix" field is reserved for future extensions and SHOULD be set to zero; "source address" field stores the original S. We call the overall /96 prefix ("prefix" field and "v4" field and "u" field and "suffix" field altogether) "uPrefix64".

o E-IP network supports ASM

ASM and SSM have simalar PIM message format. The main differences between ASM and SSM are RP and (*,G) messages. To make this

scenario feasible, we must be able to translate (*,G) messages into (S',G') messages at downstream AFBRs, and translate it back at upstream AFBRs.

4.4. Routing Mechanism

In the mesh multicast scenario, routing information is required to distribute among AFBRs to make sure that PIM messages a downstream AFBR send reach the right upstream AFBR.

To make it feasible, the /32 prefix in "IPv4-Embedded IPv6 Virtual Source Address Format" must be known to every AFBR, and every AFBR should not only announce the IP address of one of its E-IPv4 interfaces presented in the "v4" field to other AFBRs by MPBGP, but also announce the corresponding uPrefix64 to the I-IPv6 network. Since every IP address of upstream AFBR's E-IPv4 interface is different from each other, every uPrefix64 that AFBR announces should be different either, and uniquely identifies each AFBR. As uPrefix64 is an IPv6 prefix, the distribution of uPrefix64 is the same as the distribution in mesh unicast scenario. But since "v4" field is an E-IPv4 address, and BGP messages are NOT tunneled through softwires or through any other mechanism as specified in [8], AFBRs MUST be able to transport and encode/decode BGP messages that are carried over I-IPv6, whose NLRI and NH are of E-IPv4 address family.

In this way, when a downstream AFBR receives an E-IPv4 PIM (S,G) message, it can translate it into (S',G') by looking up the IP address of the corresponding AFBR's E-IPv4 interface. Since the uPrefix64 of S' is unique, and is known to every router in the I-IPv6 network, the translated message will eventually arrive at the corresponding upstream AFBR, and the upstream AFBR can translate the message back to (S,G). When a downstream AFBR receives an E-IPv4 PIM (*,G) message, S' can be generated according to the format specified in Figure 4, with "source address" field setting to *(the IPv4 address of RP). The translated message will eventually arrive at the corresponding upstream AFBR. Since every PIM router within a PIM domain must be able to map a particular multicast group address to the same RP (see Section 4.7 of [5]), when this upstream AFBR checks the "source address" field of the message, it'll find the IPv4 address of RP, so this upstream AFBR judges that this is originally a (*,G) message, then it translates the message back to the (*,G) message and processes it.

4.5. Actions performed by AFBR

The following actions are performed by AFBRs:

o E-IPv4 (*,G) state maintenance

When an AFBR wishes to propagate a (*,G) Join/Prune message to an I-IPv6 upstream router, the AFBR MUST translate (*,G) Join/Prune messages into (S',G') Join/Prune messages following the rules specified above, then send the latter.

o E-IPv4 (S,G) state maintenance

When an AFBR wishes to propagate a (S,G) Join/Prune message to an I-IPv6 upstream router, the AFBR MUST translate (S,G) Join/Prune messages into (S',G') Join/Prune messages following the rules specified above, then send the latter.

o I-IPv6 (S',G') state maintenance

It is possible that there runs a pure I-IPv6 PIM-SSM in the I-IPv6 transit core. Since the translated souce address starts with the unique "Well-Known" prefix or the ISP-defined prefix that should not be used otherwise, mash multicast won't influnce pure PIM-SM multicast at all. When one AFBR receives a I-IPv6 (S',G') message, it should check S'. If S' starts with the unique prefix, it means that this message is actually a translated E-IPv4 (S,G) or (*,G) message, then the AFBR should translate this message back to E-IPv4 PIM message and process it.

o E-IPv4 (S,G,rpt) state maintenance

When an AFBR wishes to propagate a (S,G,rpt) Join/Prune message to an I-IPv6 upstream router, the AFBR MUST do as follows.

o Inter-AFBR signaling

(S,G,rpt) messages are not supported by I-IPv6 transit core since I-IPv6 transit core only works in SSM. As a result, we're unable to stop receiving data from any given S along the RP tree even if downstream AFBR has already switched over to the SPT, which may bring about a lot of redundancy. In order to solve this problem, we introduce a new mechanism for downstream AFBR to inform upstream AFBR to prune a given S from RPT, in order to reduce redundancy.

When a downstream AFBR wishes to propagate a (S,G,rpt) message to I-IPv6 upstream router, it should encapsulate the (S,G,rpt) message, then unicast the encapsulated message to the corresponding upstream AFBR, which we call it "RP'".

The encapsulated message will evevtually arrive at RP', but the

incoming interface of it may be different from the outcoming interface along the RP tree to the corresponding downstream AFBR that send this message, so RP' is unable to determine the (S,G,rpt) state of each I-IPv6 outgoing interface. To solve this problem, and keep the solution as simple as possible, we conceptually treat all the I-IPv6 outgoing interfaces as equal, and introduce a "virtual interface" as the representative of all the I-IPv6 outgoing interfaces, which is specified in Figure 6.

```
+-----------------------------------------+
|                                         |
|       +-----------+----------+          |
|       |  PIM-SSM  |    UDP   |          |
|       +-----------+----------+          |
|          ^                |             |
|          |                |             |
|          |                v             |
|       +----------------------+          |
|       |      Virtual I/F     |          |
|       +----------------------+          |
|   PIM    ^                | multicast   |
| messages |                |   data      |
|          |  +-------------+---+         |
|       +--+--|-----------+     |         |
|       |     v           |     v         |
|     +--------- +     +----------+       |
|     | I-IP I/F |     | I-IP I/F |       |
|     +----------+     +----------+       |
|        ^     |          ^     |         |
|        |     |          |     |         |
+--------|-----|----------|-----|---------+
         |     v          |     v
```

Figure 6: upstream AFBR virtual interface

The virtual interface has two responsibilities: On control plane, it should process the encapsulated (S,G,rpt) messages received from all the I-IPv6 interfaces, and work as a real interface that has joint (*,G). Since all the I-IPv6 interfaces are treated equal, the virtual interface only send (S,G,rpt) Prune messages to PIM-SSM module when every received encapsulated message has a (S,G,rpt) Prune inside, which means that no downstream AFBR want

to receive data from source S of group G along the RPT; On data plane, upon receiving a multicast data packet, the virtual interface should encapsulate it at first, then send to every I-IPv6 interface a copy of the encapsulated data.In this way, downstream AFBRS may receive some redundant data, but avoid black holes.

NOTICE: There may exist an E-IPv4 neighbor of RP' that has joint the RP tree, so the per-interface state machine for receiving E-IPv4 (S,G,rpt) Join/Prune messages should still take effect.

- o Process and forward multicast data

On receiving multicast data from upstream routers, the AFBR looks up its forwarding table to check the IP address of each outgoing interface. If there exists at least one outgoing interface whose IP address family is different from the incoming interface, the AFBR should encapsulate/decapsulate this packet and forward it to the outgoing interface(s), then forward the data to other outgoing interfaces without encapsulation/decapsulation.

Since all I-IP interfaces of upstream AFBR are treated equal, a AFBR may receive encapsulated data from S along the RP tree even if it has already switched over to SPT of S. At this time, the AFBR should silently drop this data.

- o SPT switchover

When a new AFBR expresses its interest in receiving traffic destined for a multicast group, it needs to receive all the data along the RP tree at first. But since downstream AFBRs in fact receive the union set of data needed by every downstream AFBR, RP' has to forward all the data from RP to all the downstream AFBRs. As a result, the downstream AFBRs that have already switched to the shortest-path tree will receive two copies of the same data, namely redundancy.

To reduce the redundancy, we recommend every AFBR's SwitchToSptDesired(S,G) function employ the "switch on first packet" policy. In this way, the delay of switchover to SPT is kept as little as possible, so is the redundancy.

5. IPv6-over-IPv4

5.1. Mechanism

Routers in the client E-IPv6 networks contain routes to all other client E-IPv6 networks. Through the set of known and deployed mechanisms, E-IPv6 hosts and routers have discovered or learned of (S,G) or (*,G) IPv6 addresses. Any I-IP multicast state instantiated in the core is referred to as (S',G') or (*,G') and is certainly separated from E-IP multicast state.

This particular scenario introduces unique challenges. Unlike the IPv4-over-IPv6 scenario, it's impossible to map all of the IPv6 multicast address space into the IPv4 address space to address the one-to-one Softwire Multicast requirement. To coordinate with the "IPv4-over-IPv6" scenario and keep the solution as simple as possible, one possible solution to this problem is to limit the scope of the E-IPv6 source addresses for mapping, such as applying a "Well-Known" prefix or an ISP-defined prefix.

5.2. Group Address Mapping

To keep one-to-one group address mapping simple, the group address range of E-IP IPv6 can be reduced in a number of ways to limit the scope of addresses that need to be mapped into the I-IP IPv4 space.

A recommended multicast address format is defined in [11]. The high order bits of the E-IPv6 address range will be fixed for mapping purposes. With this scheme, each IPv4 multicast address can be mapped into an IPv6 multicast address(with the assigned prefix), and each IPv6 multicast address with the assigned prefix can be mapped into IPv4 multicast address.

5.3. Source Address Mapping

There are two kinds of multicast --- ASM and SSM. Considering that I-IP network and E-IP network may support different kind of multicast, the source address translation rules could be very complex to support all possible scenarios. But since SSM can be implemented with a strict subset of the PIM-SM protocol mechanisms [5], we can treat I-IP core as SSM-only to make it as simple as possible, then there remains only two scenarios to be discussed in detail:

- o E-IP network supports SSM

   To make sure that the translated I-IPv4 PIM message reaches the upstream AFBR, we need to set S' to an IPv4 address that leads to the upstream AFBR. But due to the non-"one-to-one" mapping of

E-IPv6 to I-IPv4 unicast address, the upstream AFBR is unable to remap the I-IPv4 source address to the original E-IPv6 source address without any constraints.

We apply a fixed IPv6 prefix and static mapping to solve this problem. A recommended source address format is defined in [9]. Figure 7 is the reminder of the format:

```
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| 0-------------32--40--48--56--64--72--80--88--96-----------127|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                     uPrefix64                |source address |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

Figure 7: IPv4-Embedded IPv6 Source Address Format

In this address format, the "uPrefix64" field starts with a "Well-Known" prefix or an ISP-defined prefix. An existing "Well-Known" prefix is 64:ff9b/32, which is defined in [9]; "source address" field is the corresponding I-IPv4 source address.

- o E-IP network supports ASM

  ASM and SSM have similar PIM message format. The main differences between ASM and SSM are RP and (*,G) messages. To make this scenario feasible, we must be able to translate (*,G) messages into (S',G') messages at downstream AFBRs and translate it back at upstream AFBRs. Here, the E-IPv6 address of RP MUST follow the format specified in Figure 7. Assume RP' is the upstream AFBR that locates between RP and the downstream AFBR.

5.4. Routing Mechanism

In the mesh multicast scenario, routing information is required to distribute among AFBRs to make sure that PIM messages a downstream AFBR send reach the right upstream AFBR.

To make it feasible, the /96 uPrefix64 must be known to every AFBR, every E-IPv6 address of sources that support mesh multicast MUST follow the format specified in Figure 7, and the corresponding upstream AFBR should announce the I-IPv4 address in "source address" field to the I-IPv4 network. Since uPrefix64 is static and unique in IPv6-over-IPv4 scenario, there is no need to distribute it using BGP.

The distribution of "source address" field of multicast source addresses is a pure I-IPv4 process and no more specification is needed.

In this way, when a downstream AFBR receives a (S,G) message, it can translate the message into (S',G') by simply taking off the prefix in S. Since S' is known to every router in I-IPv4 network, the translated message will eventually arrive at the corresponding upstream AFBR, and the upstream AFBR can translate the message back to (S,G) by appending the prefix to S'. When a downstream AFBR receives a (*,G) message, it can translate it into (S',G') by simply taking off the prefix in *(the E-IPv6 address of RP). Since S' is known to every router in I-IPv4 network, the translated message will eventually arrive at RP'. And since every PIM router within a PIM domain must be able to map a particular multicast group address to the same RP (see Section 4.7 of [5]), RP' knows that S' is the mapped I-IPv4 address of RP, so RP' will translate the message back to (*,G) by appending the prefix to S' and propagate it towards RP.

5.5. Actions performed by AFBR

The following actions are performed by AFBRs:

o E-IPv6 (*,G) state maintenance

When an AFBR wishes to propagate a (*,G) Join/Prune message to an I-IPv4 upstream router, the AFBR MUST translate (*,G) Join/Prune messages into (S',G') Join/Prune messages following the rules specified above, then send the latter.

o E-IPv6 (S,G) state maintenance

When an AFBR wishes to propagate a (S,G) Join/Prune message to an I-IPv4 upstream router, the AFBR MUST translate (S,G) Join/Prune messages into (S',G') Join/Prune messages following the rules specified above, then send the latter.

o I-IPv4 (S',G') state maintenance

It is possible that there runs a pure I-IPv4 PIM-SSM in the I-IPv4 transit core. Since the translated souce address is known to the corresponding upstream AFBR, mash multicast won't influnce pure PIM-SM multicast at all. When one AFBR receives a (S',G') message whose S' is the "source address" field of an E-IPv6 source, which means that this message is actually a translated E-IPv6 (S,G) or (*,G) message, it should translate this message back to E-IPv6 PIM message and process it.

o E-IPv6 (S,G,rpt) state maintenance

When an AFBR wishes to propagate a (S,G,rpt) Join/Prune message to an I-IPv4 upstream router, the AFBR MUST do as follows.

o Inter-AFBR signaling

(S,G,rpt) messages are not supported by I-IPv4 transit core since I-IPv4 transit core only works in SSM. As a result, we're unable to stop receiving data from any given S along the RP tree even if downstream AFBR has already switched over to the SPT, which may bring about a lot of redundancy. In order to solve this problem, we introduce a new mechanism for downstream AFBR to inform upstream AFBR to prune a given S from RPT, in order to reduce redundancy.

When a downstream AFBR wishes to propagate a (S,G,rpt) message to I-IPv4 upstream router, it should encapsulate the (S,G,rpt) message, then unicast the encapsulated message to the corresponding upstream AFBR, which we call it "RP'".

The encapsulated message will evevtually arrive at RP', but the incoming interface of it may be different from the outcoming interface along the RP tree to the corresponding downstream AFBR that send this message, so RP' is unable to determine the (S,G,rpt) state of each I-IPv4 outgoing interface. To solve this problem, and keep the solution as simple as possible, we conceptually treat all the I-IPv4 outgoing interfaces as equal, and introduce a "virtual interface" as the representative of all the I-IPv4 outgoing interfaces, which is specified in Figure 6.

The virtual interface has two responsibilities: On control plane, it should process the encapsulated (S,G,rpt) messages received from all the I-IPv4 interfaces, and work as a real interface that has joint (*,G). Since all the I-IPv4 interfaces are treated equal, the virtual interface only send (S,G,rpt) Prune messages to PIM-SSM module when every received encapsulated message has a (S,G,rpt) Prune inside, which means that no downstream AFBR want to receive data from source S of group G along the RPT; On data plane, upon receiving a multicast data packet, the virtual interface should encapsulate it at first, then send to every I-IPv4 interface a copy of the encapsulated data.In this way, downstream AFBRS may receive some redundant data, but avoid black holes.

NOTICE: There may exist an E-IPv6 neighbor of RP' that has joint the RP tree, so the per-interface state machine for receiving E-IPv6 (S,G,rpt) Join/Prune messages should still take effect.

o Process and forward multicast data

On receiving multicast data from upstream routers, the AFBR looks up its forwarding table to check the IP address of each outgoing interface. If there exists at least one outgoing interface whose IP address family is different from the incoming interface, the AFBR should encapsulate/decapsulate this packet and forward it to the outgoing interface(s), then forward the data to other outgoing interfaces without encapsulation/decapsulation.

Since all I-IP interfaces of upstream AFBR are treated equal, a AFBR may receive encapsulated data from S along the RP tree even if it has already switched over to SPT of S. At this time, the AFBR should silently drop this data.

o SPT switchover

When a new AFBR expresses its interest in receiving traffic destined for a multicast group, it needs to receive all the data along the RP tree at first. But since downstream AFBRs in fact receive the union set of data needed by every downstream AFBR, RP' has to forward all the data from RP to all the downstream AFBRs. As a result, the downstream AFBRs that have already switched to the shortest-path tree will receive two copies of the same data, namely redundancy.

To reduce the redundancy, we recommend every AFBR's SwitchToSptDesired(S,G) function employ the "switch on first packet" policy. In this way, the delay of switchover to SPT is kept as little as possible, so is the redundancy.

6. Other Consideration

6.1. Selecting a Tunneling Technology

The choice of tunneling technology is a matter of policy configured at AFBRs.

In most cases, the policy of choosing tunneling technologies will be very simple, such as all AFBRs use the same technology. But it's possible that there doesn't exist one technique that all AFBRs support. A recommended solution is described in [8], which divides AFBRs into one or more classes, and each of these classes is assigned a technology that every AFBR in this class supports. In this way, all the AFBRs in the same class can choose the right technology to communicate with each other.

6.2. Fragmentation

The encapsulation performed by upstream AFBR will increase the size of packets. As a result, the outgoing I-IP link MTU may not accommodate the extra size. As it's not always possible for core operators to increase every link's MTU, fragmentation and reassembling of encapsulated packets MUST be supported by AFBRs.

7. Security Considerations

The AFBR routers could maintain secure communications through the use of Security Architecture for the Internet Protocol as described in[RFC4301]. But when adopting some schemes that will cause heavy burden on routers, some attacker may use it as a tool for DDoS attack.

8. IANA Considerations

When AFBRs perform address mapping, they should follow some predefined rules, especially the IPv6 prefix for source address mapping should be predefined, so that ingress AFBR and egress AFBR can finish the mapping procedure correctly. The IPv6 prefix for translation can be unified within only the transit core, or within global area. In the later condition, the prefix should be assigned by IANA.

9. References

9.1. Normative References

[1] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, March 2000.

[2] Foster, B. and F. Andreasen, "Media Gateway Control Protocol (MGCP) Redirect and Reset Package", RFC 3991, February 2005.

[3] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 2373, July 1998.

[4] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

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

[6] Li, X., Dawkins, S., Ward, D., and A. Durand, "Softwire Problem Statement", RFC 4925, July 2007.

[7] Wijnands, IJ., Boers, A., and E. Rosen, "The Reverse Path Forwarding (RPF) Vector TLV", RFC 5496, March 2009.

[8] Wu, J., Cui, Y., Metz, C., and E. Rosen, "Softwire Mesh Framework", RFC 5565, June 2009.

[9] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", RFC 6052, October 2010.

9.2. Informative References

[10] Aggarwal, R. and E. Rosen, "Multicast in MPLS/BGP IP VPNs", draft-ietf-l3vpn-2547bis-mcast-10 (work in progress), January 2010.

[11] Boucadair, M., Qin, J., Lee, Y., Venaas, S., Li, X., and M. Xu, "IPv4-Embedded IPv6 Multicast Address Format", draft-ietf-mboned-64-multicast-address-format-01 (work in progress), February 2012.

Appendix A. Acknowledgements

Wenlong Chen, Xuan Chen, Alain Durand, Yiu Lee, Jacni Qin and Stig Venaas provided useful input into this document.

Authors' Addresses

Mingwei Xu
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing 100084
P.R. China

Phone: +86-10-6278-5822
Email: xmw@cernet.edu.cn

Yong Cui
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing 100084
P.R. China

Phone: +86-10-6278-5822
Email: cuiyong@tsinghua.edu.cn

Shu Yang
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing 100084
P.R. China

Phone: +86-10-6278-5822
Email: yangshu@csnet1.cs.tsinghua.edu.cn

Jianping Wu
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing 100084
P.R. China

Phone: +86-10-6278-5983
Email: jianping@cernet.edu.cn

Chris Metz
Cisco Systems
170 West Tasman Drive
San Jose, CA 95134
USA

Phone: +1-408-525-3275
Email: chmetz@cisco.com

Greg Shepherd
Cisco Systems
170 West Tasman Drive
San Jose, CA 95134
USA

Phone: +1-541-912-9758
Email: shep@cisco.com