IPv6 Multicast on JANET

Stig Venaas
UNINETT (Norway)

Tim Chown
School of Electronics
and Computer Science,
University of Southampton

Technical Guide

UKERNA Technical Guides

UKERNA Technical Guides are a series of user guides available to JANET customers. Technical Guides contain detailed technical information and are intended for technical support staff at JANET sites, network specialists, or those with a particular interest in the specialist area.

If you have any queries or comments about the Guides or would like to obtain copies, please contact:

JANET Customer Service
UKERNA
Atlas Centre, Chilton, Didcot
Oxfordshire, OX11 OQS

Tel: 0870 850 2212
Fax: 0870 850 2213
E-mail: service@janet.ac.uk

Further details of the documents in this series are available at:

http://www.ja.net/services/publications/technical-guides/

Authors and Contributors

This document was produced by the University of Southampton as complementary reading material for the IP Multicast tutorial at Networkshop 2006 in Hatfield, and was finalised to its current form after feedback and experience from the event.

The primary author is Stig Venaas of UNINETT, Norway. Stig had a one-year sabbatical at the School of Electronics and Computer Science at the University of Southampton, during which time the bulk of the text was produced. He also worked on the 6NET project (**http://www.6net.org**) and assisted in delivering the first UKERNA IPv6 Hands-On Workshop at Southampton in the summer of 2005.

Other contributions and editing came from Tim Chown of the University of Southampton.

The authors would like to thank Rob Evans (JANET NOSC), Steve Williams (UKERNA), David Price (University of Aberystwyth) and Duncan Rogerson (UKERNA) for their comments on the text, and Rina Samani (UKERNA) for her support in producing the guide.

Readers are assumed to have a basic knowledge of IPv6 and of IP Multicast. A companion IPv6 Technical Guide and an IPv4 Multicast Guide are available from the JANET Documentation repository (see **http://www.ja.net/services/publications/technical-guides**).

Contents

1 Introduction

1.1 Overview

In this guide we assume the reader has a working knowledge of both unicast IPv6 and IPv4 multicast; the reader is referred to the JANET Technical Guides on these subjects if they do not. The guide also assumes some knowledge of network planning and configuration issues.

IPv6 multicast is in many ways similar to IPv4 multicast. The basic mode of operation follows similar principles, e.g. there is a new MLD (Multicast Listener Discovery) protocol in IPv6 which is the equivalent of IGMP (Internet Group Management Protocol) for IPv4. PIM-SM (Protocol Independent Multicast – Sparse Mode) also largely works the same way, though the inter-domain PIM methods are different. IPv6 multicast introduces new concepts for PIM-SM, e.g. the idea of embedding the PIM RP (Rendezvous Point) address in the multicast group address.

We begin by explaining IPv6 multicast addresses and formats before introducing MLD and explaining how IPv6 multicast works on a LAN. IPv6's more explicit multicast address format makes the definition and management of scopes rather simpler than it is for IPv4. It is also much easier to generate your own globally unique multicast group addresses for applications.

We then move on to PIM-SM for IPv6, highlighting differences to IPv4. Then we discuss the new IPv6 deployment models, completing the inter-site picture for IPv6 multicast deployment. IPv6 also supports SSM (Source-Specific Multicast) and it is expected that SSM will be a popular model for IPv6 multicast deployment for many applications.

Finally we look at how IPv6 multicast is deployed on JANET and how to connect to that service, concluding with some example applications and information on troubleshooting.

1.2 Intended Audience

The guide is aimed at site or campus administrators and RNOs (Regional Network Operators) who are considering deploying IPv6 multicast support in their networks.

2 IPv6 Multicast Addresses and Scopes

We begin by looking at IPv6 multicast address formats. In IPv4 there is a specific range of addresses reserved for multicast use (224.0.0.0 to 239.255.255.255). In IPv6, multicast addresses are all addresses inside the prefix ff00::/8, or put another way an IPv6 address is a multicast address if and only if the first byte is 255 (ff in hex).

In this section we discuss the general address format as defined in [RFC3513]. This looks like this:

```
+--------+----+----+---------------------------------------+
|   8    |  4 |  4 |            112  bits                  |
+--------+----+----+---------------------------------------+
|11111111|    flgs|scop|            group   ID              |
+--------+----+----+---------------------------------------+
```

There are four bits each for flags and scope. There are 112 bits left to specify a group ID, which due to the increased size of IPv6 addresses is very generous compared to what is available in IPv4. As we will see later in Section 4 when we look at specific multicast address formats, the address size makes it much easier to generate globally unique multicast groups to use for applications than is the case with IPv4.

Of the four flag bits [RFC3513] only defines the low-order flag bit as the ‘T’ or ‘transient’ flag (i.e. if only this flag bit is set, the flags would have the binary value 0001). T = 0 indicates an address permanently assigned, by IANA (Internet Assigned Numbers Authority) [IANA]. T = 1 indicates a transient or non-permanently assigned address. When crafting your own IPv6 multicast group addresses for applications, this bit would be set to 1. The three other remaining flag bits are used to denote different types of addresses, as described in Section 4.

Note that for permanently assigned addresses (T = 0), the assignments are independent of scope.

The next four bits are scope bits. In IPv6, scopes are explicitly defined by these four scope bits, which specify the scope as follows:

```
0   reserved
1   interface-local scope
2   link-local scope
3   reserved
4   admin-local scope
5   site-local scope
6-7 (unassigned)
8   organization-local scope
9-D (unassigned)
E   global scope
F   reserved
```

Scopes are used to restrict the distribution of multicast packets, by filtering certain scopes at network boundaries. The idea is that the higher the value of the scope, the wider the distribution. The unassigned scopes are available for administrators to define as needed. Some of the scopes, like link-local, will be enforced automatically, while for most the administrators must configure these boundaries. We will look at ways of doing that later. This model is rather simpler than scoping in IPv4, where the scopes are tied to a specific prefix (under 239.0.0.0/8).

Let’s now look at some example multicast addresses.

- Consider the group ff02::101. We can deduce that this address is an IANA-assigned address of link-local scope. Using the IANA assignments page,[1] we can see that ff0X::

---

[1] http://www.iana.org/assignments/ipv6-multicast-addresses

101 is used to refer to NTP (Network Time Protocol) servers of scope 'X', so ff02::101 is all the NTP servers on a link (and thus ff05::101 would be all servers in a site, or there might be user-defined scopes in use, e.g. 6 or 7).

- An example of a transient address might be ff18::baad:cafe. Here the T-bit is set to 1 and the scope is organisational. This address might be used for a multicast application limited to an organisational network boundary.

3 Multicast Group IDs

As described above, a 128-bit IPv6 multicast address will include a multicast group ID. We previously wrote that there are 112 bits that can be used for the Group ID.

For an address starting with ff10::/12 (three first digits ff1), you can indeed choose these bits freely and things will work. However there is no guarantee that the address is not used by anyone else. Obviously, that is far from ideal, so in Section 4 we will look at specific address formats that do allow globally unique transient addresses to be crafted.

[RFC3307] describes how the Group ID should be selected. First of all it says that (as per [RFC3306]) the Group ID is the last 32 bits of the address. One reason for this is that the last 32 bits are used when mapping the IPv6 multicast address to the link layer destination address (e.g. the MAC address for Ethernet). Further, it defines ranges of the 32 bits to be used as follows:

```
0x00000001to0x3fffffff:UsedformulticastgroupsassignedbyIANA
0x40000000 to 0x7fffffff : Group IDs assigned by IANA
0x80000000 to 0xffffffff : Server/host allocation
```

This means that the first range is used for groups assigned by IANA. Those groups will also have a T-bit with a value of 0.

Instead of assigning a group address, IANA can also assign a group ID. As we will see later it might be useful to be able to have a standardised ID independent of what the prefix is (this will be become clearer in the next section when we talk about unicast-prefix based addresses). Despite the ID being assigned by IANA, the multicast group itself is not, so the T-bit has value 0.

Server/host allocation covers all types of allocation protocols, including manual allocation (although the latter is not all that clear). Hence we suggest that if you need to pick a group address manually, the last 32 bits should be in this range.

The main reason for having these separate ranges is that one makes sure the addresses assigned by IANA and any other addresses will be translated to different link layer addresses.

4 Different Types of Multicast Addresses and Flag Definitions

We have seen that there is a flag T specifying whether an address is transient or not. The three other remaining flag bits are used to denote different types of addresses, as we will see in this section.

4.1 Unicast-Prefix-based IPv6 Multicast Addresses

If you want to create a multicast session, you need to choose a group address. There are enough bits to play with that collisions are unlikely if you pick a random address. There is, however, a mechanism for deriving multicast addresses from a unicast prefix, so that based on your globally unique IPv6 unicast prefix you can derive globally unique IPv6 multicast addresses. This is described in [RFC3306]. This is a notable advantage over IPv4 multicast, where techniques such as GLOP addressing are required.

This specific format is defined as follows:

```
+--------+----+----+--------+--------+----------------+--------+
|   8    | 4  | 4  |   8    |   8    |       64       |   32   |
+--------+----+----+--------+--------+----------------+--------+
|11111111|flgs|scop|reserved|plen|networkprefix|groupID|
+--------+----+----+--------+--------+----------------+--------+
```

The first 16 bits should be familiar to you. However, a new flag is introduced with this scheme to denote a unicast-prefix-based multicast address, so the flag bits are now:

```
+-+-+-+-+
|0|0|P|T|
+-+-+-+-+
```

The T bit has been described earlier. The new bit is the P bit. If set, it indicates the address is prefix-based: and of course, if not set, it is not. Note that prefix-based addresses are all regarded as transient (not IANA assigned), so when the P bit is set, the T bit must also be set and hence the address will begin ff30::/12.

The next four scope bits are as before, so a global scope unicast-prefix-based address would begin ff3e::/16. The reserved bits should all be zero. The interesting fields are *plen* (prefix length) and *network prefix*.

The idea is that you can take the /48 or /64 unicast prefix that you are using and use that to create a unique multicast address. So if we have, for example, a /64 prefix 2001:db8:1:2::/64, we put the value 64 (which is 40 in hex) into the *plen* field, and the 64 bits of the prefix into the *network prefix* field. If we pick the group ID ::baad:cafe and use global scope, the complete address becomes ff3e:40:2001:db8:1:2:baad:cafe.

Hosts, applications and end-users might easily pick unique addresses using this mechanism. They can see what the IPv6 address of the host is, and based on that derive something unique (at least, so long as someone else on the same 64-bit subnet has not picked the same 32-bit group ID).

A site administrator might want to take (for example) their /48 prefix, and derive some addresses for central services from that. With the site prefix 2001:db8:1::/48, the multicast address might become ff3e:30:2001:db8:1:0:baad:cafe. The *network prefix* prefix field is 64 bits wide, but here we only use 48 bits; the remaining bits should all be zero. Because a different prefix length is used for ‘central’ services, any multicast group addresses based on it will be different should users wish also to choose addresses based on their /64 subnet within the /48.

4.2 Addresses for Embedded-RP

There is a special technique called embedded-RP [RFC3956] that encodes the address of a PIM RP into the group address. Embedded-RP is an important new feature for IPv6 multicast. It is useful because if the RP address for a specific multicast group can be determined by a router simply from looking at the multicast group address being used, then the router can immediately know what the RP address is when it sees the group address. This removes the need for MSDP (Multicast Source Discovery Protocol) in IPv6.

The embedding of the RP address does mean that only certain unicast IPv6 addresses can be used for RPs; a large proportion of the RP address bits need to be zero so the RP address can be embedded in the group address. This ‘trick’ can only be done in IPv6, thanks to the size of an IPv6 address in relation to that of an IPv4 address. We will discuss use of embedded-RP in more detail in a later section: here we simply present the specific address format.

Embedded-RP addresses are based on a small modification of the unicast-prefix based addresses we discussed in the previous section. Instead of encoding a unicast prefix, we encode the RP address. To be able to encode the 128 bit RP address into part of the 128 bit group address, we obviously need to restrict what RP addresses can look like. The trick is to require that almost half of the bits in the RP address must be zero.

The first bits of the RP address are encoded as a *network prefix* and length *plen* where the length can be up to 64 (just like the unicast-prefix based addresses), but in addition the last 4 bits of the RP address are encoded as well. This means that the last 64 bits of the RP address must be zero, apart from the four at the very end. So some example RP addresses that would work with embedded-RP would be 2001:db8:1::f and 2001:db8:1:2::1.

The specific embedded-RP format is as follows:

```
+--------+----+----+----+----+----+-------------+----------+
|   8    | 4 |  4 |  4 |  4 | 8  |      64      |    32    |
+--------+----+----+----+----+----+-------------+----------+
|11111111|flgs|scop|rsvd|RIID|plen|networkprefix|groupID|
+--------+----+----+----+----+----+-------------+----------+
```

Another flag bit is introduced and used to denote an embedded RP address, the R bit, so the flags are now:

```
+-+-+-+-+
|0|R|P|T|
+-+-+-+-+
```

Leaving the high order flag bit 0 and setting R = 1 indicates embedded-RP. The embedded-RP specification says that then the P bit should be 1 too, and as we said earlier, that also then requires T = 1. So this means that we are using embedded-RP if and only if the four flag bits have the value 7.

The *rsvd* field has 4 bits which all must be zero. Then we have the *RIID* (RP Interface Identifier*)* field, which is where we store (embed) the last four bits of the RP address. The *plen* and *network prefix* fields contain the initial (up to) 64 bits just as they do for unicast-prefix based addresses. Finally, we still have 32 bits for the group ID.

Here are two example global scope group addresses that embed the two RP addresses we mentioned earlier:

- Group address ff7e:f30:2001:db8:1:0:baad:cafe for RP address 2001:db8:1::f
- Group address ff7e:140:2001:db8:1:2:baad:cafe for RP address 2001:db8:1:2::1

The scope bits and the group ID can be chosen freely (according to the desired scope, at least).

Users should not have to be told about RP addresses or how to construct them. Network administrators can, however, configure RPs with appropriate addresses, and then tell users, or possibly applications, to pick addresses from a certain /96 prefix.

4.3 Source-Specific Multicast Addresses

The addresses described so far give us what we need for doing ASM (Any Source Multicast). For SSM, a variation of the previously described unicast-prefix based addresses is used.

One basically follows the same method but uses a *plen* value of zero. This leaves us with the prefix ff3x:0::/32 (where ‘x’ is any scope). However the unused bits of the *network prefix* part should also be zero, so all 64 bits should be zero. This leaves us with ff3x::/96 for SSM, so we can only choose the 32-bit *group ID*. The *group ID* should be chosen as described in Section 3, to avoid conflicts at the link layer.

5 Link-local IPv6 Multicast

Link-local multicast should be fairly straightforward as there is no routing involved. For IPv6 multicast to work on an Ethernet link, one simply maps the IPv6 multicast addresses to Ethernet MAC multicast addresses, and the link layer will do the job. However, Ethernet switches may cause problems. We will talk about switches later.

If you are using IPv6 over Ethernet, you are already using IPv6 link-local multicast. In fact, IPv6 cannot work properly on Ethernet without functioning multicast because in IPv6 the mapping from IP addresses to link layer addresses (the equivalent of IPv4 ARP (Address Resolution Protocol) for Ethernet) is done by a protocol called ND (Neighbour Discovery) [RFC2461] which uses IPv6 link-local multicast. This protocol is also used for IPv6 DAD (Duplicate Address Detection) and for SLAAC (Stateless Address Autoconfiguration) [RFC2462]. So, if you have working IPv6 unicast then link-local IPv6 multicast should also be working.

Neighbour discovery makes use of several link-local multicast addresses. There is the all-nodes multicast address ff02::1, all-routers multicast address ff02::2, and something called the solicited-node multicast address which we will explain shortly. The all-nodes and all-routers addresses are equivalents of the IPv4 addresses 224.0.0.1 and 224.0.0.2. Note that there is also a site-scoped all-routers address ff05::1 defined. This (on a multilink site) requires multicast routing, however.

A useful trick for debugging is to ping ff02::1 or ff02::2 on a link, although there are some systems that do not respond to multicast ping requests.

The solicited-node multicast address is defined in [RFC3513] as ff02::1:ffxx:xxxx, where the x's are the last 24 bits of a node's unicast address. If you want to ask for the link layer address for a given IP address (like ARP), you send the request to the solicited-node multicast address where the last 24 bits are taken from the unicast address, and the request is likely to go only to the host that needs it. In the worst case it will only go to a few hosts, while with IPv4 ARP (which uses broadcast) it would be picked up by everyone.

There are a number of other link-local addresses used by different protocols, but the above are the ones that are required for IPv6 operations. For the others, see the [IANA] list.[2]

If multicast on the link does not work properly, it is likely to be due to issues with Ethernet switches. Switches may try to restrict the flow of multicast only to the ports where there is interest, but this does not always work well. We will discuss that in more detail in the next section.

---

[2] http://www.iana.org/assignments/ipv6-multicast-addresses

6 Multicast Listener Discovery (MLD)

MLD (Multicast Listener Discovery) is the IPv6 protocol that is used between hosts and routers for reporting interest in receiving groups. It is the equivalent to IGMP for IPv4. There are two versions of MLD: MLDv1 [RFC2710] and MLDv2 [RFC3810]. MLDv1 is pretty similar to IGMPv2 while MLDv2 is similar to IGMPv3.

The basic operation of both IGMP and MLD protocols is such that when an application joins a multicast group, the host immediately sends a report to an on-link router to signal its interest in that group. In addition, the router periodically sends out queries checking whether there still is interest for the currently joined groups. In order to do SSM one needs MLDv2, since MLDv2 (like IGMPv3) is able to specify not just group addresses but also specific source addresses. Note that in addition to joining specific sources, MLDv2 also supports blocking specific sources.

We recommend using MLDv2 if possible; more and more routers and hosts have or are about to get support. MLDv2 is also backwards compatible with MLDv1, so we recommend configuring MLDv2 if available. The compatibility basically means that if one of the routers only does MLDv1 then MLDv2 routers and hosts will also use MLDv1. Also, if a host sends MLDv1 reports for a group then MLDv1 will be used for that group. Due to this compatibility, MLDv2 capable routers and hosts generally have MLDv2 enabled by default. For the exact details, please see [RFC3810].

We have observed cases where MLD reports or queries do not reach routers/hosts. We believe this must be due to switches not properly forwarding multicast. If someone cannot receive from a multicast group, the first debugging step should be to check that the router has seen the MLD report.

In a switched network it is desirable to restrict the flow of multicast to the ports where it is needed (i.e. one may want to not deliver multicast to an entire IP subnet (VLAN), but only to hosts that actually have signalled interest). There are generally three mechanisms for this. There is the CGMP (Cisco Group Multicast Protocol) protocol from Cisco® where the router signals the switch, the IEEE GARP (General Attributes Registration Protocol) GMRP (GARP Multicast Registration Protocol) which does signalling from host to switch, and finally there is IGMP/MLD snooping. CGMP is not available for IPv6, and few host stacks implement GMRP. Currently snooping is the most widely used mechanism, and IGMPv3 and MLDv2 have been designed with this in mind.

The basic idea of MLD snooping is that a switch watches for the MLD packets to learn who is interested. This might work well but it is not trivial to implement, and some switch implementations currently appear to get this wrong (though we expect this to improve with time). Another problem is that switches need to support the particular version of IGMP/MLD being used. At the time of writing very few switches do MLD (v1 or v2) snooping, which means that IPv6 multicast traffic might flow everywhere, while they may still do IGMP snooping and restrict IPv4 multicast. Similarly, should some MLDv3 protocol come along one day then switches doing MLDv2 might need to be upgraded. If you have switches doing only MLDv1 snooping then you may want to force hosts and routers to use MLDv1, but then you will not have SSM. Having said all that, we expect switch vendors generally to do MLDv2 snooping.

Note that switches may not do perfect filtering, e.g. they may not check the source addresses for SSM packets. They may also check only the destination MAC address, which means one should avoid using different groups with same MAC address. The Group ID restrictions described in Section 3 try to avoid this.

You might think that there is no reason to send MLD reports for link-local groups, since they never get forwarded off a subnet, but due to switches and snooping hosts should also send reports for such groups. The exception is the all-nodes multicast address ff02::1 which switches always should flood on all ports. There are two possible issues related to this. One is that some hosts are rumoured not to send MLD reports for link-locals; we are not aware of any specific case. Due to this, some switch vendors will flood at least some link-local multicast groups on all ports, in particular the solicited-node multicast addresses we mentioned earlier which are needed for Neighbour Discovery and IPv6 unicast to work.

There is another reason some vendors might wish to flood the solicited-node multicast groups. Every IPv6 host is supposed to join one (basically one group for each global address they have on the link), and in most cases all hosts will join different groups. This means that the switch ends up with quite a lot of state. There is a similar issue with IPv6 Node Information Queries (*draft-ietf-ipngwg-icmp-name-lookups*) that some host stacks support.

7 Protocol Independent Multicast - Sparse Mode (PIM-SM)

All IPv6 multicast deployments we are aware of use PIM-SM for routing. Although there might be other choices like Bidirectional PIM, we will discuss PIM-SM which we believe is what you are most likely to deploy.

In principle PIM-SM (and also Bidirectional PIM) is the same for IPv4 and IPv6. Indeed the first IPv6 PIM-SM implementations were exactly like the IPv4 ones, apart from the fact that link-local addresses are used for most PIM messages (apart from PIM Registers and BSR (Bootstrap Router) Candidate-RP Advertisements, PIM messages are all sent between routers on the same link and must use link-local addresses).

When people started deploying IPv6 PIM-SM, one problem was found. Link-local addresses are used for PIM Hello messages, so neighbours learned one another's link-local addresses but not their global addresses. In some cases the routing tables the router uses for Reverse Path Forwarding checks only contain a global address for the next-hop. Since the router does not know which PIM neighbour has the next-hop global address, it does not know where to send PIM Join messages. To solve this, a new PIM Hello option was added so that when a router sends a hello message it can list all the addresses it has on the interface (in addition to the link-local address). Most IPv6 PIM-SM implementations now support this option.

A new version of PIM-SM is being worked on (*draft-ietf-pim-sm-v2-new*). There are several implementations and we expect it to become an RFC during 2006. On some router platforms the old PIM spec is still used for IPv4, while the new is used for IPv6.

We will discuss some of the differences you might encounter when deploying IPv6 multicast. One difference is the hello option we just discussed. The option is only part of the new specification; however, most old implementations have also added it.

Another difference is that a tunnel interface is used for PIM registers, at least conceptually. You will see on some implementations that when you configure a new RP on a router, a tunnel interface is created where the destination address is the RP address. Similarly on the RP itself, you may see a tunnel interface where it receives and decapsulates PIM registers. Note that in addition to manual configuration, a router may learn of a new RP through the BSR protocol or through use of embedded-RP, so these tunnel interfaces might come and go in a dynamic fashion. Routers supporting embedded-RP will install a new mapping when they see data packets or PIM join messages with a group address encoding a new RP. There is also some timeout so that the mappings and the tunnel interfaces will go away when no group addresses for that RP have been seen for a while.

One operational issue with tunnels coming and going dynamically (this is the case for embedded-RP and possibly also for BSR) is that monitoring tools may pick up the state changes and report it as a problem. Normally, an interface going down or coming up is a rare and possibly serious event. It may be difficult for the tools to tell the difference between register interfaces and other interfaces. Because of this, at least one implementation has a fixed tunnel for embedded-RP, allowing the same interface to be used for multiple destinations (RPs).

Some implementations of IPv6 PIM also implement BSR for automatic configuration of RPs. BSR is part of the old PIM specification but not the new one. There is a separate draft specifying a new version of BSR that includes support for scoping: see *draft-ietf-pim-sm-bsr-07*.

With IPv6 multicast addresses, scoping is done using the 4 scope bits. These are used for scoping in BSR. Some PIM implementations have support for specifying scope boundaries based on the 4-bit scope value, and then drop PIM messages (including BSR) and data packets as needed.

7.1 Enabling PIM-SM

If you have IPv6 unicast running on your network, turning on PIM-SM on a router is simple.

In Cisco IOS, PIM-SM is enabled with:

```
ip multicast-routing
```

This configuration is sufficient for using SSM and embedded-RP.

8 Inter-Domain IPv6 Multicast

There are several differences between IPv4 and IPv6 multicast, in particular when it comes to inter-domain IPv6 multicast. For IPv4 multicast, multicast administration is generally split into PIM domains, where organisations run their own RPs which their multicast routers use. A method is then needed for multicast to operate in the presence of multiple administrative PIM domains. In IPv4, there is MSDP, which allows multicast source information to be shared between multiple RPs.

The main difference is that IPv6 has no MSDP; instead it has embedded-RP, as described below. Embedded-RP makes inter-domain IPv6 multicast possible, despite the lack of MSDP, by making the RP address for a given multicast group implicit from the group address. These IPv4 and IPv6 solutions are very different from each other.

With IPv4 multicast, each domain generally has its own RP serving all groups. There are some exceptions, but generally speaking this is the case. Hence, there is a large number of RPs throughout the Internet serving the same groups. In order to have multicast connectivity between domains using different RPs, one needs to use MSDP to exchange information between the RPs. When someone using one of these RPs sends traffic to a group, this is signalled to all the other RPs in the network of MSDP peerings. While this method works, it does not scale very well.

With IPv6 multicast, embedded-RP should be used for inter-domain multicast. This leads to a very different deployment model. With embedded-RP, there is only one RP on the Internet for a given group. Each domain should still have at least one RP, but they would never be for the same groups and no signalling is needed between the RPs.

With embedded-RP we believe the organisation providing content or hosting or creating a session should have an RP, and use an embedded-RP group for that RP. There is then no requirement for a third party to support an RP.

MSDP is also used for doing anycast-RP for IPv4. If one wants to do anycast-RP with IPv6, this must then be done in some other way. An alternative way of doing this is proposed in *draft-ietf-pim-anycast-rp*. This is expected to become an RFC during 2006.

More and more router platforms are getting IPv6 multicast support but for a long time one will encounter situations where some routers do not. One then has a situation similar to when one deploys IPv6 and some routers do not support it. To provide IPv6 multicast to hosts, one may deploy IPv6 multicast routers alongside IPv6 unicast routers and have the multicast routers only doing multicast. Also, one may set up tunnels to bypass routers not supporting multicast inside the network. This leads to different topologies for IPv6 unicast and multicast. This should be avoided, at least for larger deployments or production services. It may take a lot of extra effort to manage such setups, but it can be done.

One common method for coping with different topologies for unicast and multicast is to use multiprotocol BGP with separate routes for unicast and multicast. This is done today for IPv4, and it can be used in the same way (but also across tunnels) for IPv6. This is at least a solution for inter-domain multicast. Inside a domain this is not so easy. One may also create static multicast routes, but this is indeed a management nightmare.

9 Deployment Guidelines

We will now discuss some concrete guidelines for how a backbone network connecting multiple sites should deploy IPv6 multicast, and in turn how end sites can deploy it.

9.1 For backbone networks

Deploying IPv6 multicast in a backbone network may be (relatively) simple. SSM should work with no configuration. For ASM, we recommend that only embedded-RP be used, apart from inside sites where one may need to configure other RPs for internal use. The most common routers supporting IPv6 multicast also support embedded-RP at the time of writing.

It is a serious problem if the backbone routers do not support embedded-RP, since each router on the shared tree must recognise the embedded-RP format and be able to 'decode' the RP address from the observed group address for the protocol to work. One should thus check this feature with your upstream provider(s), and also add embedded-RP capability into procurement tenders.

The only workaround if support is not present is to configure the backbone routers statically with the RP addresses for some commonly used RP addresses, but in effect this seriously limits the ASM connectivity. Some routers have embedded-RP enabled by default while some others have it off by default. Enabling or disabling embedded-RP should just be a per-router on/off toggle.

On Cisco IOS, both SSM and ASM (with embedded-RP) should work out of the box. On Juniper JunOS, embedded-RP must be enabled on each router.

There are two things that probably need to be configured in a backbone network. These are scope boundaries and BGP.

9.1.1 Scope boundaries

Sites or organisations that connect to a backbone will generally use the site and organisational scope values as discussed in Section 2, i.e. scope 5 for sites or scope 8 for organisations. Whether a connecting site uses scope 5, 8, or its own defined scope between these then the backbone should ensure that these scope boundaries are enforced by appropriate filtering. In addition it might be good to configure the same boundaries on the interfaces facing the sites. One can then protect against traffic leaking into the backbone in cases where a site has not configured this. For example, if scope 8 is used, this means that both the sites and the backbone should configure a scope 8 boundary on the interfaces facing each other, i.e. not pass traffic with this scope into each other's networks.

On Cisco IOS, the scope boundary is configured using the command

```
ipv6 multicast boundary scope 8
```

Note that until recently the command was

```
ipv6 zone boundary 8
```

so you may encounter that as well.

The effect of this is to drop packets of scope 8 or less, and also to ignore PIM messages for groups of scope 8 or less. Some other vendors may have equivalent commands, or use access lists where arbitrary IPv6 group ranges might be defined. Note that standard access lists would be able to drop data packets but this is not sufficient for filtering PIM messages. The main concern is perhaps data packets but there are also issues with PIM registers that are PIM messages but may still transport data. If configuring access lists manually then we would still suggest filtering all scopes lower than 5 or 8, so one may need to configure several disjoint group ranges.

The backbone is probably connected to other backbone networks. If some multicast is used purely in the backbone, e.g. for some monitoring solution between monitoring nodes connected directly to the backbone, one should configure boundaries and RPs as for sites.

If the backbone and its connected sites form some unit where some multicast traffic is used between sites that should not leak outside the unit, then one should also configure scope boundaries for that. If scope 8 is used for each connecting site or organisation then one could use say scope 9 for this multicast. In the JANET context, this may typically be multicast restricted to an RNO and connected sites. In that case, the backbone should configure scope 9 as a boundary on the interfaces facing other backbone networks. In the previous paragraph we talked about using scope 5 or 8 for the boundary, but if the boundary is configured for scope 9 then one should block all scopes below, including both 5 and 8.

Specifics of IPv6 multicast on JANET are discussed in Section 10. UKERNA is expected to agree and publish recommended IPv6 multicast scopes for use in RNO networks and for JANET itself.

9.1.2 BGP

As we have discussed in the section on inter-domain multicast, one will typically use BGP for exchanging multicast routes. That is, if two domains have a BGP peering for unicast and they both do multicast then they should in general also have a multicast BGP peering. Configuring this peering is quite simple. On Cisco IOS there will be a section in the BGP config saying

```
address family ipv6 unicast
```

In order also to exchange multicast information, one may only need to add a section

```
address family ipv6 multicast
```

repeating what is in the unicast section for each of the multicast peers. One may want to have some different policies, though, so it is not necessarily exactly the same. The general principles are the same as for unicast.

On Cisco IOS one will by default perform multicast Reverse Path Forwarding using a combination of multicast BGP routes and routes learnt from other sources (excluding unicast BGP), where the most specific route is used; but if of equal length, then multicast BGP is used.

9.1.3 RP configuration

By only using embedded-RP and SSM for inter-domain multicast, a backbone network may not need to do any RP configuration. With embedded-RP we recommend that RPs are configured close to the edge, inside the different sites (see more on this in the next section). Due to the use of embedded-RP, the routers in the backbone will automatically know the RP addresses of the site's RP(s) when needed.

A backbone may however choose to configure some of its routers to be RPs for two reasons. One is if the backbone uses some multicast internally for (for example) monitoring purposes. We can then think of the backbone as a site with respect to the internal multicast usage; see the next section for guidance.

A backbone may also choose to offer RPs as a service to sites that do not have their own RPs. Ideally, one would recommend the sites to have their own instead, but not requiring a site to have their own RP may make it easier to get them started with multicast. But note that this is only really needed if the site creates its own multicast sessions, and then it might be reasonable also to require it to configure a router as an RP. If a site only joins sessions that other sites have created then it would join a group that has an RP address belonging to another site's RP encoded, so it would never use its own.

If a backbone wants to offer an RP service, it should use embedded-RP so that no RP configuration is needed on other routers. The backbone operator might configure just one router as an RP, or possibly several where different customers use different RPs. For instance if a backbone consists of several PoPs where groups of customers are connected to different PoPs, then one could consider one RP in each PoP. In the JANET context, the RNOs may be the PoPs.

If several sites are to use the same backbone RP for their sessions then each site should be informed by their service provider which group addresses to use, where each site can have its own unique group range. Ideally each site should have its own /96 prefix, giving it 32 bits for specifying groups.

In order to do this, it might be good to have, say, a /80 prefix leaving 16 bits to specify the site, and then the last 32 bits for each site to choose freely. Each site needs to be told which 16 bit value to use; this could perhaps be derived from the site's unicast prefix. For example, if the backbone belongs to an ISP with a /32 prefix and each site gets a /48 prefix, then there are 16 bits denoting the site. Obtaining a /80 prefix should be easy; it only requires the backbone to have at least a /48 prefix. Actually, a site would usually get a /48 prefix, which means a site could use similar techniques internally if desired.

We recommend that sites are encouraged run their own RP(s) where possible, but recognise that an interim provision may be helpful to help sites get started.

The group ranges that the sites are told to use would often be of global scope, but if the backbone uses (for example) scope A for limiting multicast to itself and its sites, then its operator might also tell the sites to use the same group ranges for that, but with scope A instead of (global) scope E.

9.2 For sites

First of all, we will focus on a typical small site. A large site may have their own backbone, and thus should also read the previous section. The same techniques can be used for a site but then scaled down to smaller scope values.

At a subnet level, hosts and routers need to support MLD. Note that MLDv2 is required for SSM operation. We recommend its use, and thus sites should mention MLDv2 support in procurement tenders.

In many situations it is undesirable if multicast is flooded over an entire LAN. This depends on size of the LAN, whether there are some slow links (e.g. wireless LAN), or some hosts that cannot cope with high multicast traffic rates. If possible, one should try to have switches that support MLDv2 snooping, but few support it at the time of writing (though this picture is changing). If available, MLD snooping should be enabled (ideally MLDv2).

For a site to be able to receive SSM from other sites, or send to other sites, no configuration is needed. If the site wishes to take part in external ASM sessions, it should make sure its routers support embedded-RP and have it enabled. If it has routers that do not support embedded-RP, it will need to configure the routers for all the external RPs it may use manually, which is not a good solution.

The site should configure a scope boundary of 5 or 8 (based on what its provider tells it to use) on its external router interface(s). This is only really needed if the site wishes to have some internal use of multicast (e.g. for internal video content distribution), but it is a good idea to have this in place anyway.

If the site wishes to create/host multicast sessions that are to be used outside the site, it will need to have its own RP, or have one provided by its backbone. We recommend the former.

If the site is to use ASM internally then an internal RP is needed. The same RP could be used for external groups. For internal use the site should consider embedded-RP. However, there might be a need for serving other groups. Some multicast applications may want fixed multicast addresses that are not in any way relying on which RP address is used, because that requires configuration. One example is DHCPv6 which makes use of the address 'FF05::1:3' for reaching all DHCP servers in the site. To support such applications one needs to configure an RP for (for example) FF05::/16.

If RPs that are not 'embedded' are used then the site must somehow configure all its routers with these RPs. That is, not only do the RP routers require configuration but also all the other routers. This can either be manually configured on all the site routers, or the site may use BSR as described in *draft-ietf-pim-sm-bsr-07*. Manual configuration might be fine if the site chooses the RP address with care, so that it can be moved around between routers, independent of the topology.

As for backbones, a site may choose not to have a central embedded-RP but have the RPs even closer to the edges. For example, a university might consist of different schools or departments, where each may have their own RP if needed. If the site uses a central RP then it will, as we described in the backbone section, try to form embedded-RP addresses from its /48 prefix (or /56) if possible, leaving 16 bits (or 8) to signify which part of the university is using which.

If the site uses an RP provided from the backbone, it may have only 32 bits. There may then be a need for co-ordinating internally regarding which part of the university can use which of these bits.

It might even be worth considering running RPs at edge routers. One might then form embedded-RP addresses based on a link's /64 prefix. In that case, one is left with 32 bits that only need to be unique on the link. So, avoiding collisions is then reduced to a per link issue.

The only other multicast related configuration that might be needed is multicast BGP. This should only be done if the site is currently using BGP and wishes to use some of the BGP peers for multicast connectivity.

We suggest that Designated Routers are configured with a PIM register rate-limit to avoid too much load on any given Designated Routers and on RPs.

10 Multicast in JANET

JANET is actively supporting IPv6 multicast deployment.

IPv6 multicast connectivity is available via the IPv6 Experimental Service[3] on the present JANET backbone. Connectivity can be arranged via the RNO and JANET Customer Service. (See Section 14.)

For the forthcoming JANET backbone upgrade at the end of 2006, IPv6 multicast will be supported on the backbone from launch. Sites or organisations interested in IPv6 multicast connectivity should still contact the RNO.

The JANET Experimental Service peers for IPv6 multicast with the m6bone network[4] and also with the (native) GÉANT IPv6 multicast service.

10.1 RNO support for RPs and Embedded RP

For the JANET backbone upgrade the RNOs are recommended by UKERNA to deploy an (IPv4) RP. An RNO may typically make such an RP available for use by its connected sites where those sites have not (yet) deployed their own RP, though UKERNA does not mandate that RNOs must offer such a service. Alternatively the RNO may encourage each site to deploy its own RP and establish MSDP peerings between these and the RNO’s RP.

The current contracts between UKERNA and the RNOs do not require deployment of an IPv6 RP, though this may change in the future. As mentioned above, IPv6 has no MSDP equivalent and thus only one global RP could be used per multicast IPv6 group. Instead, with IPv6 we expect to see use of embedded-RP, with sites (as discussed above) running their own RPs.

It is important that the RNO’s router supports embedded-RP, so that sites can use their own RPs. An RNO may still choose to provide an RP that can be used for a connected site’s multicast service that uses an embedded-RP based multicast group address.

10.2 Scopes

Some discussion of scope needs to be held within the JANET community.

One might suggest scope A for JANET (the same scope is used for Renater and UNINETT national research networks). Scope 9 could then be for Regional Networks and 8 for JANET-connected organisations. Each site would then have scopes 4-7 for internal scoping. Another option is scope 5 for sites, but we believe they need more than one extra scope (4) for internal scoping.

For UNINETT, if we use scope A and then C for GÉANT, then JANET might also use scope B, which would leave both 9 and A ‘spare’ for RNOs. With scope C for GÉANT, there would be room for one non-global scope (D) above GÉANT. This could possibly be used for all academic networks (GÉANT, Abilene and others).

Hopefully this discussion will be held soon and an updated IP JANET Multicast Addressing Policy will be published to included IPv6. Until then, it is safe to assume that sites/organisations can use up to scope 8, RNOs can use scope 9 and JANET can use at least scope A.

---

3. http://www.ja.net/development/ipv6/experimental-service.html
4. http://www.m6bone.net

10.3 IPv6 Multicast on GÉANT

Development of IPv6 multicast support on GÉANT has been aided significantly by piloting work done on the m6bone, within TERENA TF-NGN and via the 6NET project.

GÉANT now supports IPv6 multicast natively, via its Juniper infrastructure.

Embedded-RP was enabled on some routers early in 2006 and embedded-RP support should be pervasive by summer 2006. Applications using embedded-RP multicast groups have been tested between UK sites (Southampton) and US sites (New York), which relies on embedded-RP being supported through JANET, GÉANT and Internet2 (Abilene).

No GÉANT (European academic) IPv6 multicast scope value has as yet been agreed.

11 Monitoring and Debugging

In this section we discuss IPv6 multicast monitoring and debugging tools.

11.1 Monitoring IPv6 multicast

One important tool for monitoring the multicast service, and detecting and possibly narrowing down problems, is the multicast beacon. The idea is that sites running multicast can each deploy a beacon. The beacons should then run continuously, and as a result one will get a matrix showing which sites can receive multicast from which others, along with some measurements of loss, delay etc.

The best beacon to run for IPv6 is probably *dbeacon*, which can be downloaded from:

http://artemis.av.it.pt/~hsantos/dbeacon/

Another tool that is useful for end-users and one-off tests is *ssmping* and its integral *asmping* counterpart. By running these tools, one basically pings some host on the Internet (running the *ssmping*/*asmping* server) and one will see whether one can receive (unicast) and multicast from the server to the client host, as well as getting some further information. It might be good for sites, content providers etc. to run such a server so that end-users in the site or those wishing to receive content can use the *ssmping*/*asmping* to check multicast reception.

The *ssmping* package is available from :

http://www.venaas.no/multicast/ssmping/

UKERNA is in the process of extending the current JANET beacon infrastructure to include IPv6 for dbeacon and ssmping. Currently these tools are provided within JANET for IPv4 only at **www.beacon.ja.net** and **ssmping.beacon.ja.net**. Other activities to develop more granular monitoring of multicast traffic should encompass both IPv4 and IPv6 multicast.

11.2 Debugging IPv6 multicast

If an end user is not able to receive from a source, how does one track down the problem?

In principle, debugging IPv6 multicast problems is very similar to debugging IPv4 multicast problems. The tools described above can assist in this process.

There are some differences worth noting, however.

- MLD/MLDv2. An initial check is to verify that the subnet Designated Router is receiving MLD reports from the host (much as one would test for IGMP reception for IPv4). It is possible that switch ports may be filtering the packets. Snooping support for MLD and MLDv2 is in its infancy at the time of writing, so may possibly cause problems.
- Embedded-RP. When using an embedded-RP multicast group address, one requires all routers on the shared tree from the host to the RP to support the protocol, else the router will not know which address to forward data to (the RP address being ‘encoded’ in the group address).

Otherwise one would apply IPv4 multicast debugging procedures. For further information see section 7 of the JANET Multicast Guide:

http://www.ja.net/services/publications/technical-guides/ipv4-multicast-web.pdf

12 Applications

There are a number of IPv6 multicast applications that can be deployed and used today. These include:

- Mbone tools (vic, rat, etc): video and audio-based conferencing tools
  http://www-mice.cs.ucl.ac.uk/multimedia/software
- VideoLAN: video streaming
  http://www.videolan.org
- DVTS: digital video streaming
  http://www.sfc.wide.ad.jp/DVTS
- MAD Flute: reliable multicast file transfer
  http://www.atm.tut.fi

All these tools can be used now by sites wanting to pilot IPv6 multicast services. The VideoLAN package is probably the most impressive example.

12.1 ECS-TV and Surge Radio

At Southampton, we have also developed other applications, including:

- ECS-TV
  http://www.zepler.tv
- Surge Radio
  http://www.surgeradio.co.uk/listen/advanced.html

Both of these run over IPv6 multicast and use embedded-RP multicast groups with a locally hosted RP making the content available in UK, European and US research networks (where scoping permits).

VideoLAN's vlc application supports IPv6 multicast for transmission and reception of multimedia data over a network. With support for ASM, SSM and embedded-RP addressing schemes vlc becomes the ideal application for load testing high bandwidth multicast streams. ECS-TV at the School of Electronics and Computer Science, University of Southampton provides an example of one such deployment using this application. Being available across the university campus, many scopes can be used to control multicast traffic within the internal network. Channels are decoded via a Freeview receiver in a Linux system and are then transmitted as one MPEG Transport Stream per multicast group. Both ASM and embedded-RP addressing schemes have been used successfully using ff18:: and ff78:140:: prefixes respectively. The embedded-RP address scheme has a 64 bit prefix.

We have found some MLD incompatibilities as hinted at earlier in this report. Here older Linux kernels (pre 2.6.4) and Windows XP (SP2) fail to speak the MLDv2 protocol (for SSM) supported by the Cisco® infrastructure being used. Windows Vista will support MLDv2, however.

13 Configuration Examples

A number of configuration examples for IPv6 multicast can be found in the material produced for Networkshop 34, available at:

http://www.multicast.org.uk/nws34/

Here we include two examples of IOS and JunOS configurations to give a flavour for specific configurations. These are taken from the Networkshop dual-stack IPv4-IPv6 hands-on sessions; the reader is referred to the hand-out notes from the above web site for specifics of the network configurations. (Note: the workshop used IOS only. The Juniper example was tested at Networkshop but not used in the course itself.)

13.1 IOS

```
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname nws1a
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
resource policy
!
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
ip subnet-zero
ip cef
!
!
ip multicast-routing
!
ipv6 unicast-routing
ipv6 cef
ipv6 multicast-routing
!
voice-card 0
!
!
interface Loopback0
 ip address 193.61.85.9 255.255.255.255
 ipv6 address 2001:630:23F:7110::1/128
!
interface Loopback1
 ip address 193.61.85.13 255.255.255.255
 ipv6 address 2001:630:23F:7110::2/128
!
interface FastEthernet0/0
 ip address 193.61.85.230 255.255.255.252
 ip broadcast-address 193.61.85.231
 ip pim sparse-mode
 duplex auto
 speed auto
 ipv6 address 2001:630:23F:7001::2/64
 ipv6 pim bsr border
 ipv6 rip PS1 enable
 ipv6 multicast boundary scope 8
!
interface FastEthernet0/1
 ip address 193.61.85.1 255.255.255.252
 ip broadcast-address 193.61.85.3
 ip pim sparse-mode
 duplex auto
 speed auto
 ipv6 address 2001:630:23F:7100::1/64
 ipv6 rip PS1 enable
!
router bgp 64620
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 193.61.85.229 remote-as 64615
 neighbor 193.61.85.229 description nws7206
 !
 address-family ipv4 multicast
 neighbor 193.61.85.229 activate
 no auto-summary
 no synchronization
 network 193.61.85.0 mask 255.255.255.224
 exit-address-family
!
ip classless
ip route 0.0.0.0 0.0.0.0 193.61.85.229
ip route 193.61.85.16 255.255.255.2 40 193.61.85.2
ip route 193.61.85.0 255.255.255.224 Null0
!
!
ip http server
no ip http secure-server
ip pim rp-address 193.61.85.13
ip pim ssm default
ip msdp peer 193.61.85.229
ip msdp description 193.61.85.229 nws7206
ip msdp sa-filter in 193.61.85.229 list MSDP-FILTER
ip msdp sa-filter out 193.61.85.229 list MSDP-FILTER
ip msdp originator-id Loopback0
!
ip access-list extended MSDP-FILTER
 deny ip any 239.0.0.0 0.255.255.255
 permit ip any any
!
ipv6 router rip PS1
 redistribute connected
!
ipv6 pim rp-address 2001:630:23F:7110::2 v6rp
!
!
ipv6 access-list v6rp
 permit ipv6 any FF05::/16
 permit ipv6 any FF15::/16
 permit ipv6 any FF08::/16
 permit ipv6 any FF18::/16
!
end
```

13.2 JunOS

```
version 7.5R2.8;
system {
    host-name nws2b;
    authentication-order password            ;
    root-authentication {
        encrypted-password“$1$Inc1nNR3$cJFztow8ZJDwyUApbZqsA
0”; ## SECRET-DATA
    }
    login {
        user admin {
            uid 2000;
            class superuser;
            authentication {
                encrypted-password“$1$tDLApt97$imIw5UuI8jxy
RQhZGGhf7.”; ## SECRET-DATA
            }
        }
    }
    services {
        telnet;
        web-management {
            http;
        }
    }
    syslog {
        file messages {
            any any;
        }
    }
}
interfaces {
    fe-0/0/0 {
        unit 0 {
            family inet {
                address 193.61.85.3        4/30;
            }
            family inet6 {
                address 2001:630:23        f:7200::2/64;
            }
        }
    }
    se-0/0/0 { ## Serial Interfac            e
        unit 0 {
            family inet {
                address 192.168.9.4        8/24;
            }
        }
    }
    sp-0/0/0 {
        unit 0 {
            family inet;
            }
        }
        fe-0/0/1 {
            unit 0 {
                family inet {
                    address 193.61.85.49/29;
                }
                family inet6 {
                    address 2001:630:23f:7240::1/64;
                }
            }
        }
        fe-4/0/0 {
            unit 0 {
                family inet {
                    address 193.61.85.57/29;
                }
                family inet6 {
                    address 2001:630:23f:8240::1/64;
                }
            }
        }
        lo0 {
            unit 0 {
                family inet {
                    address 127.0.0.1/32;
                    address 193.61.85.42/32;
                }
            }
        }
    }
    routing-options {
        interface-routes {
            rib-group inet6 v6-unicast-multicast-group;
        }
        static {
            route 0.0.0.0/0 next-hop 193.61.85.33;
        }
        rib-groups {
            v6-unicast-group {
                export-rib inet6.0;
                import-rib inet6.0;
            }
            v6-multicast-group {
                export-rib inet6.2;
                import-rib inet6.2;
            }
            v6-unicast-multicast-group {
                export-rib inet6.0;
                import-rib [ inet6.0 inet6.2 ];
            }
        }
        router-id 193.61.85.42;
    }
    protocols {
        igmp {
            interface all;
        }
        mld {
            interface all;
        }
        router-advertisement {
            interface fe-0/0/1.0 {
                prefix 2001:630:23f:724          0::/64;
            }
            interface fe-4/0/0.0 {
                prefix 2001:630:23f:728          0::/64;
            }
        }
        sap;
        pim {
            rib-group inet6 v6-multica           st-group;
            rp {
                embedded-rp;
                static {
                    address 2001:630:23        f:7210::2;
                }
            }
            interface all {
                mode sparse;
            }
        }
    }
```

14 Further Information/Getting Help

General information on the use of IP multicast with networks belonging to the JANET community in the UK can be located on the JANET web site. Specific information concerning multicast and IPv6 on JANET is located at [JANETv6].

14.1 Regional Network Connection Procedure and Support

Requests for new IPv6 multicast connections and general queries should be sent to JANET Customer Service (JCS) at service@janet.ac.uk. At present this forms part of the IPv6 Experimental Service on JANET. The support level for IPv6 is changing (being made a production service) for the forthcoming JANET backbone upgrade.

14.2 Operational Queries and Fault Reporting

Operational queries and fault reporting on existing multicast connections must be sent to the JANET Operations Desk.

14.3 Customer Site Network Connection Procedure and Support

Requests for new multicast connections to customer sites should be directed via the appropriate Regional Network.

15 Glossary

Anycast-RP	A way of using the same RP-address on several RP-routers to do load balancing, and also fast fail-over, see RFC 3446.
ARP	Address Resolution Protocol.
ASM	Any Source Multicast. ASM is the classical multicast service model as described in RFC 1112 where any host can join a given multicast group G, and any host can send a packet with destination address G, and have it delivered to all members of the group G. The sender does not need to be a member of G. Compare with SSM.
Any Source Multicast	See ASM.
Backbone:	The basis of a large network, e.g. the JANET core
Bi-directional PIM	Uses shared trees, not only from RP towards receivers as in PIM-SM, but also in the other direction from sources towards the RP. There are no PIM registers. See *draft-ietf-pim-bidir-07*.
BSR	Bootstrap Router Protocol. A dynamic protocol for configuring the group-to-RP mappings in a multicast domain. See also the PIM-SM specification [RFC2362].
Bootstrap Router Protocol	See BSR
CGMP	Cisco Group Multicast Protocol.
Cisco Group Multicast Protocol	See CGMP.
Channel	This is used as a term for the source-group pair (S,G) in SSM; see SSM.
DAD	Duplicate Address Detection.
DR	Designated Router. The PIM-SM router on a link that is acting on behalf of the hosts on the link. When a host starts sending multicast, the DR sends register messages to the RP. When there are multiple PIM-SM routers on a link, one of them is elected as the DR. Initially the DR will also send join messages on behalf of the hosts and maintain tree state, but in some cases another PIM router on the link can take over; see last-hop router. See also the PIM-SM specification [RFC2362].
Designated Router	See DR.
Embedded-RP	A way of encoding the IPv6 RP address in an IPv6 multicast group address. By using groups derived from the RP-address, routers can compute the RP-address from the group address so that they do not need any prior RP configuration. See [RFC3956].
GARP	General Attributes Registration Protocol.
GMRP	GARP Multicast Registration Protocol.

IANA.	Internet Assigned Numbers Authority.
IGMP	Internet Group Management Protocol. Protocol used between hosts and multicast routers. It is used by IPv4 hosts to report multicast group membership to routers. The latest version is IGMPv3. For source-specific reports, like in SSM, IGMPv3 is required. For IPv6, see MLD.
Last-hop router	The PIM-SM router on a link that is responsible for sending join messages on behalf of the hosts and maintaining tree state. This is the last router to forward the packets before they reach the host. This is initially the DR, but with multiple PIM routers on the same link, another router may become the last-hop router. See also the PIM-SM specification [RFC2362].
MBGP	Multi Protocol BGP. This is often used for multicast. One may not always want the same topology for multicast and unicast. Using MBGP one can have BGP peerings exchanging both unicast and multicast prefixes independent of each other. See [RFC2858].
MLD	Multicast Listener Discovery. Protocol used between hosts and multicast routers. It is used by IPv6 hosts to report multicast group membership to routers. For MLD, see [RFC2710]. For source-specific reports, like in SSM, MLDv2 is required, see [RFC3810]. For IPv4, see IGMP.
MSDP	Multicast Source Discovery Protocol. An inter-domain protocol used only for IPv4 multicast. It connects RPs in different domains, so that information about new sources can be distributed between the RPs.
ND	Neighbour Discovery.
NTP	Network Time Protocol.
PIM	Protocol Independent Multicast. An inter-domain multicast routing protocol. Called protocol independent because it makes use of the unicast routing table for Reverse Path Forwarding, but is independent of which unicast routing protocols are used to populate the table.
PIM-SM	Protocol Independent Multicast – Sparse Mode. The sparse mode variant of PIM. Called sparse because it only forwards where requested. See also [RFC2362].
Reverse Path Forwarding	RPF is used to determine where to send join messages, or from whom a packet should arrive. The RPF neighbour for a given address is computed from the routing tables and is often the next-hop a unicast packet with that destination address would be forwarded to. PIM uses RPF to find out where to send join messages and also performs a so-called RPF check, discarding data packets arriving on the wrong interface. RPF is used by many multicast protocols and flooding mechanisms, and also BSR. See also the PIM-SM specification [RFC2362].
RIID	RP Interface Identifier.
RNO	Regional Network Operator.
RP	Rendezvous Point. For PIM-SM, this is used for source discovery. New sources register with the RP and initially traffic is forwarded on RPT which is rooted at the RP. See also the PIM-SM specification [RFC2362].

RPT - RP Tree (aka shared tree)	The tree that is rooted at the RP and created by (*,G) joins from last-hop routers. For ASM this tree is used at least initially; last-hop routers may create SPTs and switch to them. See also the PIM-SM specification [RFC2362].
SLAAC	Stateless Address Autoconfiguration.
SPT	Shortest Path Tree. This tree is rooted at the source, or rather at the source's DR, and the leaves are last-hop routers and/or the RP. It is built by (S,G)-joins. See also the PIM-SM specification [RFC2362].
SSM	Source Specific Multicast. Instead of joining a group G, hosts join a so-called channel (S,G) and the host will only receive from the source S. A host can join several sources with the same group. The source does not need to be member of the group. See *draft-ietf-ssm-arch-06*.

Appendix: Bibliography

A.1 General References

[IANA] Internet Assigned Numbers Authority
http://www.iana.org

[JANETv6] JANET IPv6
http://www.ja.net/development/ipv6

A.2 Internet RFCs

In this section we list RFCs that have some relevance to multicast IPv6. All RFCs can be found from:

http://www.ietf.org/rfc.html

These will be referenced elsewhere in this guide by entries of the form [RFC9999].

RFC2362	‘Protocol Independent Multicast – Sparse Mode (PIM-SM)’, June 1998.
RFC2461	‘Neighbour Discovery for IP Version 6 (IPv6)’, December 1998.
RFC2462	‘IPv6 Stateless Address Autoconfiguration’, December 1998.
RFC2710	‘Multicast Listener Discovery (MLD) for IPv6’, October 1999.
RFC2858	‘Multiprotocol Extensions for BGP-4’, June 2000.
RFC3306	‘Unicast Prefix-based IPv6 Multicast Addresses’, August 2002.
RFC3307	‘Allocation Guidelines for IPv6 Multicast Addresses’, August 2002.
RFC3513	‘Internet Protocol Version 6 Addressing Architecture’, April 2003.
RFC3810	‘Multicast Listener Discovery Version 2 (MLDv2) for IPv6’, June 2004.
RFC3956	‘Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address’, November 2004.

Tell us what you think

Technical Guides are produced and published by UKERNA for use within the JANET Community.

We welcome your comments on all aspects of this document and on any other UKERNA publication.

Please direct feedback to JANET Customer Service, at the address below, or e-mail us directly at:

documentation@janet.ac.uk

UKERNA manages the networking programme on behalf of the higher and further education and research community in the United Kingdom. JANET, the United Kingdom's academic and research network, is funded by the Joint Information Systems Committee (JISC).

For further information please contact:

JANET Customer Service
UKERNA
Atlas Centre, Chilton, Didcot
Oxfordshire, OX11 0QS

Tel: 0870 850 2212
Fax: 0870 850 2213
E-mail: service@janet.ac.uk

Copyright:

Trademarks:

JANET®, SuperJANET® and UKERNA® are registered trademarks of the Higher Education Funding Councils for England, Scotland and Wales. The JNT Association is the registered user of these trademarks.

Cisco® is a registered trademark of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries.

The term Linux® is a registered trademark of Linus Torvalds, the original author of the Linux kernel.

Microsoft® and Windows® are registered trademarks of Microsoft Corporation in the United States and/or other countries.

Disclaimer:

Availability:

Further copies of this document may be obtained from JANET Customer Service at the above address.

This document is also available electronically from: **http://www.ja.net/services/publications/technical-guides**

UKERNA

JISC

GD / JANET / TECH / 013