Project Number:	**IST-2001-32603**
Project Title:	**6NET**
CEC Deliverable Number:	**32603/Consortium GARR (IIT-CNR)/DS/3.4.4/A1**
Contractual Date of Delivery to the CEC:	**June 2003**
Actual Date of Delivery to the CEC:	**July 15th, 2003**
Title of Deliverable:	**IPv4/IPv6 Multicast Interoperability**
Work package contributing to Deliverable:	**WP3**
Type of Deliverable*:	**R**
Deliverable Security Class**:	**PU**
Editors:	**Lorenzo Rossi, Antonio Pinizzotto** **(Consortium GARR / IIT-CNR)**
Contributors:	**Stig Venaas (UNINETT)**

* Type:	P - Prototype, R - Report, D - Demonstrator, O - Other
** Security Class:	PU- Public, PP – Restricted to other programme participants (including the Commission), RE – Restricted to a group defined by the consortium (including the Commission), CO – Confidential, only for members of the consortium (including the Commission)

Abstract:

This document describes how both IPv4 and IPv6 multicast can be deployed together in the same network, and also how IP hosts that do not support the same IP protocol, can still have multicast connectivity.

Keywords:

gateway, interoperability, IPv4, IPv6, multicast, reflector, scope, SSM.

Table of Contents

1. Introduction

IPv4 and IPv6 will co-exist for many years, possibly decades, so finding ways for the two to interoperate is important. There are a number of transition techniques that might be of help, and a large number of reports and papers are written on this subject. But most of these have focused on unicast, while very little has been written on multicast. Here we will describe several multicast issues and some possible solutions.

If all parties intending to communicate have applications and host operating systems that support the same IP protocol, there is usually no serious problem; except that one may have to establish tunnels through parts of the network. There is not much difference between unicast and multicast in this aspect. Not all tunnel techniques support multicast, but many do. However, for some of the techniques one may need some additional specifications and code.

For unicast one can try to make servers dual-stack so that IPv4-only and IPv6-only hosts can easily access it. Similarly, a multicast source could stream to both an IPv4 and an IPv6 group to allow all hosts to reach it. If the unicast server cannot be made dual-stack, hosts that support only the other protocol will not be able to access it unless some translation is performed somewhere on the path between them. The same can be said for multicast, but avoiding translation is very difficult for some applications; for instance, in a conference where each host in the group is sending to and receiving from all the others.

Several types of translation solutions are available for unicast; here we will study some multicast translation solutions. However, if possible one should avoid translation so first we will try to look for other alternatives. The alternative is what we are calling application solutions.

2. Multicast in dual-stack networks

When deploying multicast in a dual-stack network, one will probably want to deploy both IPv4 and IPv6 multicast. They are both mostly deployed using PIM-SM and the classical Any-Source Multicast (ASM) model with Rendezvous Points (RPs) and to some extent Source-Specific Multicast (SSM). There is no problem deploying both. Both routers and hosts can simultaneously do IPv4 and IPv6 multicast, and both can be done over the same network links. A router could also be an RP for both IPv4 and IPv6 groups.

2.1. Scopes in IPv4 and IPv6

Scopes are defined differently in IPv4 and IPv6. They both have defined some scopes for link, organization, etc, but IPv6 has many more scope values defined. The administrative boundaries for a site, an organization, etc should be the same for IPv4 and IPv6 in fully dual-stack networks. The scope boundaries used for multicast should be in accordance with this, e.g. so that the organization scope boundaries for IPv4 and IPv6 are in exactly the same places.

[RFC 2365] defines the following table 2.1 which might be of help:

Table 2.1: RFC 2365 scopes definitions table

IPv6 Scope	RFC 1884 Description	IPv4 Prefix
0	reserved	
1	node-local scope	
2	link-local scope	224.0.0.0/24
3	(unassigned)	239.255.0.0/16
4	(unassigned)	
5	site-local scope	
6	(unassigned)	
7	(unassigned)	
8	organization-local scope	239.192.0.0/14
A	(unassigned)	
B	(unassigned)	
C	(unassigned)	
D	(unassigned)	
E	global scope	224.0.1.0-238.255.255.255
F	reserved	
	(unassigned)	239.0.0.0/10
	(unassigned)	239.64.0.0/10
	(unassigned)	239.128.0.0/10

3. Application solutions

As we stated in the introduction, we would like to avoid translation if possible. We will here try to describe the possible alternatives we have at the application level.

If one wants to avoid translation in the network and still wants to have some IPv4 and IPv6 interoperability, one will need to make sure that sources send packets using an IP protocol supported by all receivers. This is a very strong requirement.

3.1. Single-source scenario

The easiest and also very common scenario is that of a single source sending data to a number of clients. For instance, this can be audio and video streaming from a conference, from television, etc. Note however that if RTP/RTCP is used, the receivers are also sources because they send RTCP reports. It may not be critical to receive the reports though.

If the streaming is within a closed environment where one knows that all potential receivers support the same IP protocol, one only needs to use the same protocol for the source.

In a more open environment one may not know who the potential receivers are and which IP protocols they support. To ensure that all of them will be able to receive, one will need one source for IPv4 and one for IPv6, unless some sort of translation is done. The important thing here is to source the same data in v4 and v6; it is not important whether this is done using one single dual-stack host, or two different hosts, which may not be dual-stack. If for instance one is streaming video, one can just make sure that the video signal is transmitted to two different hosts, both doing encoding, one sending v4 and one sending v6. If a single host is being used, one can perform the encoding once before the encoded data is transmitted as both v4 and v6. This means that one can send both v4 and v6 without using much extra processing power. However the amount of bandwidth consumed is doubled, at least if there are both v4 and v6 receivers.

Typically one would be sending both IPv4 and IPv6 on the local network where the source(s) is/are located. The multicast router(s) on the network will usually know whether there are receivers and only forward data if there are any. For example if there are IPv6 listeners only, the IPv4 data will not need to be forwarded, so the bandwidth used outside the local network is not doubled. Also if one is using SSM (Source-Specific Multicast), there is a solution called [MSNIP] which makes it possible for the sources to know whether there are receivers. In this case one can avoid the double bandwidth on the local network as well. But one must expect there to be both IPv4 and IPv6 receivers, and then one cannot avoid doubling the bandwidth.

There is a possible problem with announcements. For this one typically uses [SDP]. But in the standard SDP protocol there is no way to say that the same content is available from multiple addresses and that the receiver can choose one. We would like to choose between IPv4 and IPv6, but there might also be cases where it makes sense to choose between addresses from the same protocol. [RFC 3388] might be a solution for this. Of course, one might also try to distribute different SDP messages to potential IPv4 and IPv6 receivers. This might be problematic, but if one uses [SAP] for distributing SDP, one can distribute a message with only IPv4 addresses in SAP over IPv4, and only IPv6 addresses in SAP over IPv6.

3.2. Multi-source scenario

With just a few sources one may be able to ensure that all are dual-stack and streaming both IPv4 and IPv6 as described in the single-source scenario. But imagine a video conference where everyone (or almost everyone) wants to both receive and send data. Again, if all support the same IP protocol, this is simple. Then one simply uses a multicast group address of that protocol for the session. However, one might easily end up with some participants in the conference having IPv4-only hosts, and some others IPv6-only hosts. Some sort of translation is then the only option.

4. Translation solutions

Translation can be performed in several ways. The idea is to have one or more translation devices on the path(s) between sources that use one IP protocol and receivers that use another IP protocol. Note that in some rare cases translation might also be carried out within the sending or receiving host. This can be useful on a dual-stack host where the application used only supports one IP protocol. In many cases the hardest part, when using translation devices, is to make sure that the device actually will be on the data path. This can be more difficult for multicast than unicast.

As for unicast, translation can be done in several ways. Here we will describe two different solutions used in [6NET].

4.1. IPv4 - IPv6 multicast reflector

This is a quite simple and powerful tool implemented by Konstantin Kabassanov and is available from [REFLECT]. Rather than reflecting between unicast and multicast, this tool reflects between IPv4 and IPv6 multicast. For a given IPv4 group and port, and a given IPv6 group and port, one runs this tool with these arguments, and it will join both groups and listen to the corresponding ports. All data received from one group will be re-sent to the other. Note that it actually also listens and re-sends for the port above so that it works with RTP/RTCP without requiring an extra instance.

This tool is very useful, the main limitation is that one needs to run one instance of this per session, and that it will receive and re-send data also if there are no receivers. For a content provider it can be quite useful when one streams some content and the host or the application does not support both IP protocols. It might also be useful near the receivers in the case where the provider only streams using one protocol and receivers can only use the other.

This might be used as a general tool for re-sending multicast for many groups, but because of the limitations described above, one can only run this for a limited number of groups. If this is used somewhere in the network, the users would typically have to contact an administrator and ask for specific sessions to be reflected for a limited time. However, one might possibly use a web interface or other means for automating this.

4.2. IPv4 - IPv6 multicast gateway

The idea here is to embed IPv4 multicast addresses into IPv6 by prepending them with a specific /96 IPv6 prefix, such that for each IPv4 multicast address there is a respective IPv6 multicast address. The gateway is an IPv6 PIM SM multicast router and the RP for the /96 prefix used. An

IPv6 host in the PIM domain can receive or send to any IPv4 group by using the respective IPv6 group. Since the gateway is the RP for the prefix used, it will know when there are IPv6 receivers and sources present, and able to receive all the IPv6 data sent. IPv6 data will be sent to the corresponding IPv4 group, and if there are IPv6 receivers, it will join the IPv4 group and resend any data received from it. Using the gateway one can have e.g. a video conference with multiple participants in both IPv4 and IPv6, and have full two-way connectivity. The main limitation is that it doesn't offer IPv4 users access to IPv6 multicast sessions unless they happen to use a group inside the /96 prefix. It could perhaps be enhanced to offer some additional static mappings for specific groups, but the reflector described in the previous section is also an option for those groups. The gateway is further described in [MCGW].

This gateway could for instance be placed on the border between an IPv4-only and an IPv6-only network, but it could also be placed inside a dual-stack network. It could be used within a single site or organization, or offered as a service in a large network. One way to restrict it to a site might be to use a /96 prefix of site-level scope. A single gateway is currently deployed as a service to all of [6NET] and the [M6BONE] network. This works, but will probably not scale when there is a large number of users. It would be possible to deploy several gateways in the same network if desired. They would then use different /96 prefixes, and each would be the RP for their respective prefix.

To make it easier for the users to use this gateway, a SAP translator is also used. It listens to IPv4 SAP announcements and resends them as IPv6 SAP announcements where the IPv4 group addresses are rewritten to IPv6 group addresses by prepending with the appropriate /96 prefix. This means that an IPv6 user running sdr or a similar SAP listener can take part in an IPv4 session by simply clicking on the advertisement.

5. Conclusion

Deploying both IPv4 and IPv6 multicast in a network is quite easy. Hosts can also easily do both at the same time. They both live their completely separate lives. This is also part of the problem. Hosts doing IPv4 multicast cannot easily communicate with hosts doing IPv6 multicast. In order to do this you will need some translation solutions such as the reflector and the gateway we described. Translation should be avoided if possible though. As an alternative to translation we suggest that sources send both IPv4 and IPv6 multicast in cases where there might be both IPv4-only and IPv6-only receivers. This might be difficult with many receivers however, particularly if all receivers are also sources.

6. References

[RFC2365] RFC 2365: "Administratively Scoped IP Multicast", D. Meyer, July 1998 (Status: Best Current Practice)

[MSNIP] draft-ietf-magma-msnip-04.txt: "Multicast Source Notification of Interest Protocol (MSNIP)", B. Fenner, B. Haberman, H. Holbrook, I. Kouvelas, 19 June 2003; expires: December 2003

[SDP] RFC 2327: "SDP: Session Description Protocol"; M. Handley, V. Jacobson; April 1998 (Updated by RFC3266) (Status: Proposed Standard)

[RFC3388] RFC 3388: "Grouping of Media Lines in the Session Description Protocol (SDP)", G. Camarillo, G. Eriksson, J. Holler, H. Schulzrinne, December 2002 (Status: Proposed Standard)

[SAP] RFC 2974: “Session Announcement Protocol”; M. Handley, C. Perkins, E. Whelan, October 2000 (Status: Experimental Protocol for the Internet community)

[6NET] http://www.6net.org/

[REFLECT] http://www.kabassanov.com/reflectors/

[MCGW] draft-venaas-mboned-v4v6mcastgw-00.txt: "An IPv4 - IPv6 multicast gateway", S. Venaas, February 2003; expires: August 2003

[M6BONE] http://www.m6bone.net/