

CHAPTER 7

IMPLEMENTATION AND TESTING

The solutions to enhance NAT-PT to support multicast have already been planned in chapter 6. This chapter will describe their implementation. This chapter presents a prototype implementation of NAT-PT with multicast support. Multicast address translation has been tested and verified.

This chapter consists of experimental design in section 7.1. The set up and network configuration are presented in section 7.2 and 7.3 respectively. Then the multicast address translation is tested in section 7.4 and the result is in section 7.5. We conclude with a summary in section 7.6.

7.1 Overview of the implementation

This part presents the prototype implementation of the multicast function for NAT-PT. The main focus is to add the enhanced multicast address translation module into a NAT-PT router. Then it is used to provide multicast communication between v4 and v6 worlds in a testbed network.

The enhancement has been inserted into the existing system that works on FreeBSD [32], a unix Operating System (OS). The original NAT-PT was implemented in the kernel of FreeBSD. NAT-PT was created by the KAME [33] project which has improved several OS in the family of Berkeley Software Distribution (BSD) to support IPv6.

The extension module has been located at a place that does not impact the existing system. It processes a multicast packet after the packet has been duplicated. At that point, the packet is ready to be forwarded to destination networks without further checking except finding the path from the routing table.

A NAT-PT router with multicast enabled functions as a normal multicast router. It has three main jobs to do with a multicast packet: receiving, processing and forwarding. However, the exclusive works of NAT-PT are address and packet translation. The multicast module has three main tasks.

1. To get packet

This function is used to get a multicast packet into the multicast module of NAT-PT. The existing code sequences for multicast packet handing needed to be modified to allow NAT-PT processing. In practice, getting v4 and v6 multicast packets into the NAT-PT code of router has been done in different ways because of the style of the existing code.

For v4 input, the system processes incoming v4 unicast packets before multicasting. NAT-PT is handled inside the unicast part. After that it starts to translate the other unicast packets. NAT-PT did not process multicast packets.

The existing NAT-PT code has been extended so that if a v4 multicast packet is detected, it will be translated as a multicast packet. Unicast packets continue to be translated as before. On the other hand, for v6 input, the system operates upon an incoming packet in a different way from IPv4. The packet is checked, and if it is multicast, multicast processing is performed, and processing ends. If not multicast, other packet processing occurs, including NAT-PT packet translation. Multicast v6 packets were never seen by the NAT-PT code.

The extended code is embedded into the multicast function. If a v6 input packet is multicast, the NAT-PT multicast code is invoked. Otherwise, the input packet continues to the unicast or other functions.

2. To translate packet

This function is used to check address of the received packet in order to process address and packet translation. NAT-PT uses the method from section 6.3.1 to translate a multicast packet. It provides multicast address translation following the mapping rules. When NAT-PT receives a multicast packet, it looks for the group address of that packet in the mapping rules. And then NAT-PT translates the addresses and packet. After that the translated packet is forwarded to all recipients.

3. To forward packet

This part deals with sending the translated packets to the recipients. In order to make sure that the translated packet is correctly forwarded, NAT-PT must interact correctly with the routing protocol. Multicast routing protocols for IPv4 and IPv6 have been deployed to find the appropriate paths. Any multicast routing protocol would do for the test network. The routing protocols on the existing system have been used to find the paths to send the translated packets. PIM-DM [41], for

IPv6, and DVMRP [40], for IPv4, have been chosen in the implementation. Each translated packet in each IP version is treated like a normal multicast packet.

The router has several interfaces to receive and forward packets. It must check incoming and outgoing interfaces before getting and sending the packets in order to avoid packet looping. PIM-DM and DVMRP are used to forward multicast v4 and v6 packets respectively. They rely on RPF to forward packet and use a unicast routing table to find the reverse path.

7.2 Testbed network design

The module for multicast address translation has already been added to NAT-PT. The function has been implemented and tested in a testbed network designed based on the assumptions enumerated in section 6.2.

This testbed network is similar to a network in the Internet. It has a multicast router to find appropriate paths and provide connections for v4 and v6 nodes. Multicast disciplines such as multicast routing protocol, IP multicast have been deployed to provide communication in this network. Because our aim is to determine whether NAT-PT can handle multicast, we deploy only one NAT-PT router. Assuming this succeeds, later work can test behavior with more NAT-PT enable multicast routers.

From these requirements, the testbed network has been designed with the following properties:

- The network consists of a v4-only and a v6-only network. They are connected by the router. Every node in each site attaches to the same link. The NAT-PT router is located at the boundary of v4 and v6 networks. The router is the gateway between the networks.
- The implementation has only one NAT-PT router in order to test only the basic functionality of multicast address translation and multicast packet forwarding.
- In testing, vic (Video Conferencing Tool) has been used to generate multicast packets. The vic program is easy to use. It can generate a stream of multicast packets. It also requires no external hardware, such as a microphone or camera, as it is able to use the system's frame buffer as its image source.
- The software to generate a schedule of multicast application is sdr (Session Description Tool).

- Multicast applications used for testing have been configured with global scope.

7.3 Setup and configuration

In order to test the extension function, the testbed network has been setup to provide multicast communication between v4 and v6 sites. The router must be enabled to support multicast. Then it requires network and routing configuration in order to forward the multicast packets and find the path to the recipients. Apart from that, NAT-PT must enable the multicast function. NAT-PT's configuration requires rules to translate multicast packets.

The implementation has two important settings: configuration of the network and of the mapping rules. The network configuration is presented in section 7.3.1 while section 7.3.2 gives an example of the mapping rules to translate multicast packets.

7.3.1 Network configuration

IP address and network configurations have been assigned to provide multicast communication between v4 and v6 networks. There are two main configurations: IP address and routing configuration.

7.3.1.1 IP address configuration

The particular IP addresses have been chosen for the configuration. There are two parts of IP address configurations: the end nodes – v4 and v6, and, NAT-PT router as in Figure 7.1.

v4 and v6 node: v4 and v6 nodes in each site have been configured with network address 172.30.21.0/24 and 3ffe:b80:53:1c6::0/64 respectively.

NAT-PT router: NAT-PT router has been configured IP address for v4 and v6 interfaces with address 172.30.21.1 and 3ffe:b80:53:1c6:2c1:28ff:fe02:5a54 respectively.

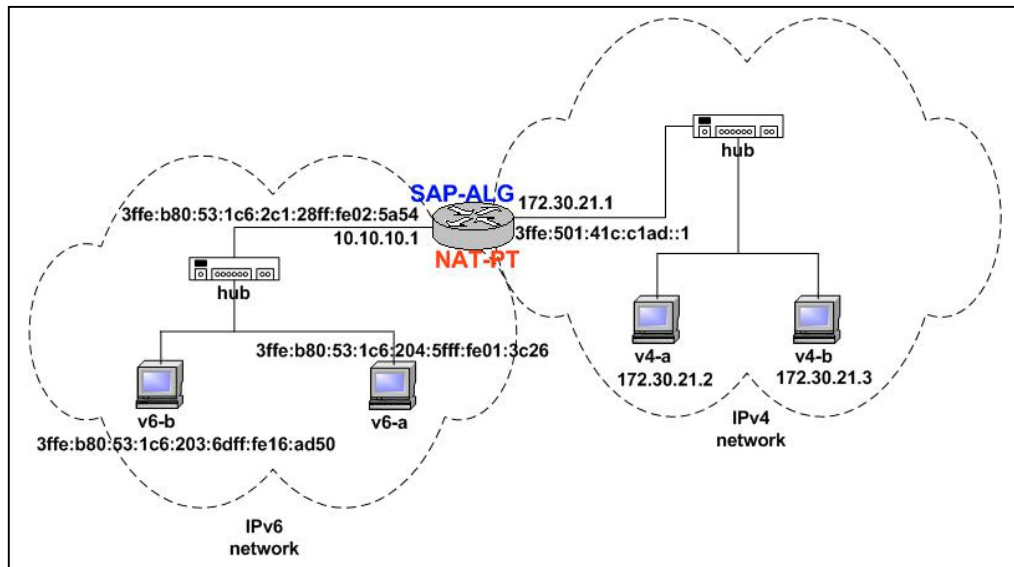


Figure 7.1 Testbed with network configuration

7.3.1.2 Routing configuration

Like any normal multicast router, the NAT-PT router requires a multicast routing protocol to determine the paths for multicast communication. Any multicast routing protocols which are available in the Internet can be deployed. These protocols do not affect our testing. An arbitrary multicast routing protocol can be used. In our implementation, we choose the program pim6dd based on PIM-DM on as the routing protocol for IPv6. We use the program mrouterd based on DVMRP as the multicast routing protocol for IPv4.

When a multicast packet is delivered into the NAT-PT router, it is determined whether it comes from the right place in order to avoid packet looping by the algorithm of RPF [39]. Then if the destination of the packet is in the list of mapping rules, it must be translated. At this time inside NAT-PT router, there are two kinds of packets: the original and the translated multicast packets. Both of them must be sent to destination networks that join the group via appropriate paths.

However, we must consider the way that these packets are sent. For the original, it must be treated, getting and sending, as a normal multicast packet. For the other packet, it is generated by NAT-PT. It is not necessary to check where the translated packet comes from and whether it arrives at the right path because it comes from NAT-PT itself.

Unfortunately the existing kernel has no interface to allow multicast packets to be forwarded as would an incoming packet, without the packet actually being an incoming packet, subject to RPF checks. While rearranging the code structure to add such an interface would be possible, that would have required more extensive kernel modifications than are required for this project.

However, we must identify for the routing table that the translated packet comes from NAT-PT itself. The packet must be forwarded to destination networks via appropriate paths. Although we cannot easily avoid path checking for translated multicast packet, we rectify this point by using an artificial interface marked as the receive interface for the translated multicast packet only.

We create an artificial interface with a unicast route to the translated source address. The interface is used by RPF checking for the translated packet only. This satisfies the RPF check which ideally would not be performed at all. For our implementation we have two interfaces. One connects to the v4 network and has only IPv4 addresses and routers. The other is only IPv6. To simplify our implementation we choose the v4 interface to be the artificial v6 interface, and the v6 interface to be the artificial interface for v4 packets. Thus, in practice, the source interface for a translated packet is set to be the interface over which it arrived before it was translated, though this is just a co-incidence.

We now explain the routing table configuration. This example was used in the implementation. Figure 7.1 shows a routing table which was used in this testing.

The router has an interface, x10, which connects to the v4 network. The interface is configured with IPv4 address 172.30.21.2 which is presented in record number 4. There is also the network prefix of this interface 172.30.21/24 which is seen in record number 3. Interface r10 is configured with IPv6 prefix address 3ffe:b80:53:1c6::/64 as in record number 6. The interface r10 has IPv6 address 3ffe:b80:53:1c6:2c1:28ff:fe02:5a54 presented in record number 8. It connects to the v6 network.

NAT-PT has been configured with magic IPv4 and IPv6 address as the artificial interfaces. The magic IPv4 and IPv6 number can be arbitrary addresses which are unique. In testing, we allocate the blocks of unicast IPv4 and IPv6 addresses 10.10.10.0/24 and 3ffe:501:41c:c1ad::0/64 respectively.

<i>Record no.</i>	<i>Internet: Destination</i>	<i>Gateway</i>	<i>Network if</i>
1	127.0.0.1	127.0.0.1	lo0
2	10.10.10/24	link#3	r10
3	172.30.21/24	link#1	x10
4	172.30.21.2	00:20:ed:59:8e:29	x10
	<i>Internet6: Destination</i>	<i>Gateway</i>	<i>Network if</i>
5	::1	::1	lo0
6	3ffe:b80:53:1c6::/64	link#1	r10
7	3ffe:501:41c:c1ad::/64	link#1	x10
8	3ffe:b80:53:1c6:2c1:28ff:fe02:5a54%r10	00:c1:28:02:5a54	lo0
9	fe80::2c1:28ff:fe02:5a54%r10	00:c1:28:02:5a54	r10

Figure 7.1 Routing table for translate multicast packet

We must configure the artificial IPv6 address on the v4 interface as in record number 7. The address is used by RPF checking to allow sending the translated packet to v6 network, for IPv4 to IPv6 translation. For the other direction, the artificial IPv4 address is configured on the v6 interface as in record number 2. However, this technique is a shortcut to testing only.

7.3.2 The mapping rules

NAT-PT requires mapping rules to translate addresses including multicast packets. In addition, NAT-PT does not care where the addresses used for mapping come from. There are two possible ways to generate the rules for multicast address mapping: manual configuration by a human or automatic configuration by software.

7.3.2.1 Manual configuration

Multicast address in the mapping rules can be assigned by the network administrator. The address can be an arbitrary number. We use the simple method of section 6.3.2 to assign addresses.

Figure 7.1 shows an example of the mapping rules and the addresses assigned.

IPv4 to IPv6

In testing, an arbitrary IPv4 number 224.200.200.200 is used to be the group address of multicast v4 applications. In order to provide these applications to v6 network, they require a temporary multicast IPv6 address for mapping. Any multicast IPv6 number can be chosen. It is not necessary to be a well-known or a permanently assigned multicast address.

Multicast IPv6 address, ff00::/8, with any group number can be used to be a temporary IPv6 address. However, a group address should not be duplicated anywhere in the network in order to avoid address clash. We apply the multicast IPv6 address assignment defined in RFC3306 [37] for our temporary IPv6 multicast address.

Then our temporary IPv6 multicast address has the prefix number ff3e:ff00::/96 with the multicast IPv4 address in the last 32 bits as the group ID. Following our assumptions in section 6.2, we require a global multicast IPv6 address in order to allow multicast sessions to be forwarded to any network in the Internet. So we set the scope field to identify global scope – its value is 1110 (e in hexadecimal number). We set both flags P and T to be value 1 to indicate this multicast address is transient address assigned according to the procedures of RFC3306. Then the IPv6 multicast flag has the value 0011 (3 in hexadecimal number). We want to indicate this prefix used to test the NAT-PT router with multicast in our testbed only. We enable the reserve field (always zero) to be value 1111 1111 without assign IPv6 unicast-prefix as a part of the multicast prefix because there is only one NAT-PT router. Then the prefix length is set to be zero.

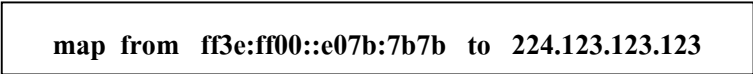
This choice of values uses (or abuses) the RFC3306 multicast address assignment technique in a way that should guarantee that no generated addresses conflict with any normally assigned RFC3306 multicast address. This technique also means that an IPv4 multicast group will be translated into the same IPv6 multicast group by any NAT-PT router configured this way. A mapping rule for IPv4 to IPv6 address translation in our testing is presented in Figure 7.3.

<p>map from 224.200.200.200 to ff3e:ff00::e0c8:c8c8</p>
--

Figure 7.1 Mapping rule for IPv4 to IPv6

IPv6 to IPv4

For IPv6 to IPv4 address mapping, we require a temporary IPv4 address to map to the original IPv6 address. An arbitrary block of multicast IPv4 addresses can be chosen. We select multicast IPv4 address 224.123.123.123 as an address to be used to translate IPv6 multicast addresses. The mapping rule for IPv6 to IPv4 is shown in Figure 7.4.



```
map from ff3e:ff00::e07b:7b7b to 224.123.123.123
```

Figure 7.2 Mapping rule for IPv6 to IPv4

7.3.2.2 Automatic configuration

In this configuration, address mapping has been generated by the multicast scheduling program, SAP-ALG. It provides multicast address allocation and address mapping.

7.4 Testing

There are two main parts of testing: the multicast module and SAP-ALG. Each part has been done in both directions – v4 to v6 and vice versa.

Firstly, the multicast extension module has been tested to provide address and packet translation. This testing requires multicast packets to determine correct operation of the multicast module. A multicast application, vic, is used to generate multicast packets. It creates the packets without requiring extra equipment such as a web camera or microphone. The vic program uses RTP to transmit data packets and RTCP to control the session. This part has been implemented in scenario 1 and 2.

Secondly, SAP-ALG has been tested to provide multicast address mapping. This part requires a program to generate SAP packets in order to exercise the SAP-ALG module. A suitable program is sdr which transmits SDP and SAP to provide session descriptions. It is used to create a schedule of multicast services and can generate group numbers. This part has been implemented in scenario 3 and 4.

The conditions of all testing in the first and second parts are based on the assumptions in section 6.2 as follows:

1. Multicast applications have been configured with global scope.
2. Only one NAT-PT router has been set up for testing.

Scenario 1: Manual configuration for IPv4 to IPv6 address translation

The objective of this testing is to verify the multicast function added into NAT-PT in order to provide multicast address translation and packet forwarding from IPv4 and IPv6. For NAT-PT to provide address and packet translation, the group address must be changed from an IPv4 to an IPv6 address. The expected result is that the v6 recipient can join the translated group and get the data packets from the v4 source.

In this scenario, the v4 source node provides a video stream by using the vic program and sends multicast data packets to the network. This vic session is configured with an IPv4 group address that has already been registered into the mapping rules. When these packets arrive at the gateway router, they are translated and forwarded by NAT-PT to the recipients in the v6 site.

NAT-PT relies on the address in the mapping rules to translate these multicast packets. The address in the rules was configured manually before the application was started.

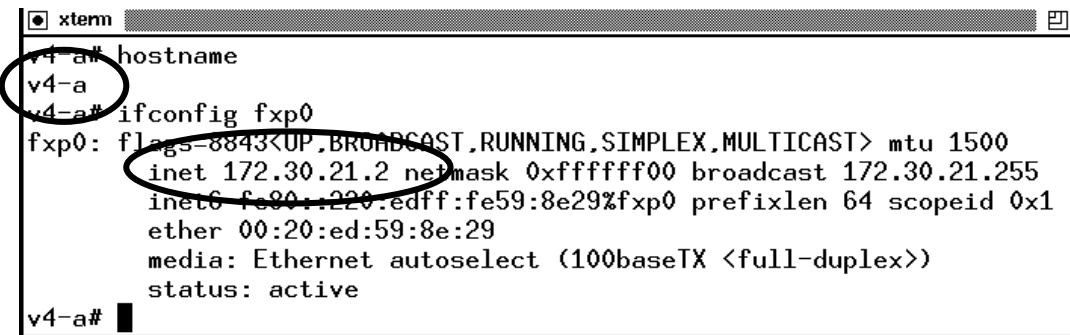
Procedure for this scenario is as follows:

1. **To send the packet to network:** v4-a node runs vic program and sends packets of a video stream to the network.
2. **To receive the packet:** IPv4 multicast routing protocol arranges for forwarding the v4 multicast packets. These packets are delivered into NAT-PT router.
3. **To translate address and packet:** The group address of packet is compared with the mapping rules. If the address is found in the list, the address and packet are translated using the address found in the rules. The procedures of address and packet translation are in section 6.5.
4. **To forward the translated packet to recipient:** The router forwards the translated data packets via every IPv6 interface to v6 recipients according to the needs of the v6 multicast routing protocol.
5. **To join group and get data packet:** At v6 site, when v6-a node runs the vic program it joins the group. The data packets are delivered to application layer and

displayed on the screen of the recipient.

Detail of testing:

A terminal window on v4-a has been sent as the data rather than video streaming from a web camera. The window presents name of source and network interface configuration for address checking at the other site. The terminal is shown in Figure 7.1.



```

xterm
v4-a# hostname
v4-a
v4-a# ifconfig fxp0
fxp0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    inet 172.30.21.2 netmask 0xfffff00 broadcast 172.30.21.255
    inet6 fe80::220:edff:fe59:8e29%fxp0 prefixlen 64 scopeid 0x1
    ether 00:20:ed:59:8e:29
    media: Ethernet autoselect (100baseTX <full-duplex>)
    status: active
v4-a#
  
```

Figure 7.1 Window terminal on the v4 source

V4-a has been configured with address 172.30.21.2. It runs vic application with the group number 224.200.200.200 and port 40000. This session is named manee. It is sent with the frame rate 7.0 f/s and bit rate 34 kb/s without any dropped packets as shown in Figure 7.2

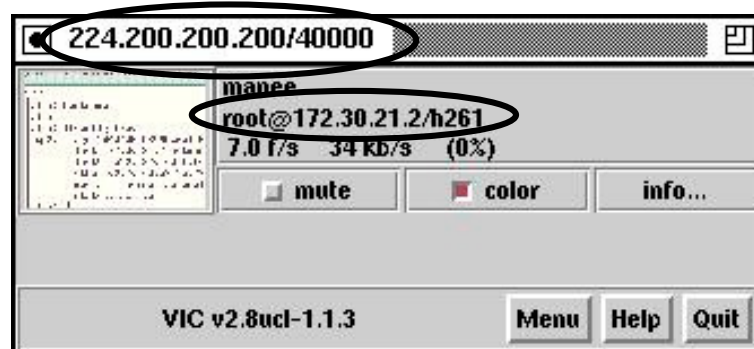


Figure 7.2 Menu of vic application on the v4 source

The information of the session is presented in Figure 7.3. The important fields are `cname` and `srcid`. `cname` is the name of creator who generated this session. It is used to describe the originator of the session. It looks like an e-mail address. The creator is user `root` on the node `172.30.21.2` and is using `h261` formats to encapsulate the session. `Srcid` is used to identify the session number and IP address of the source node.

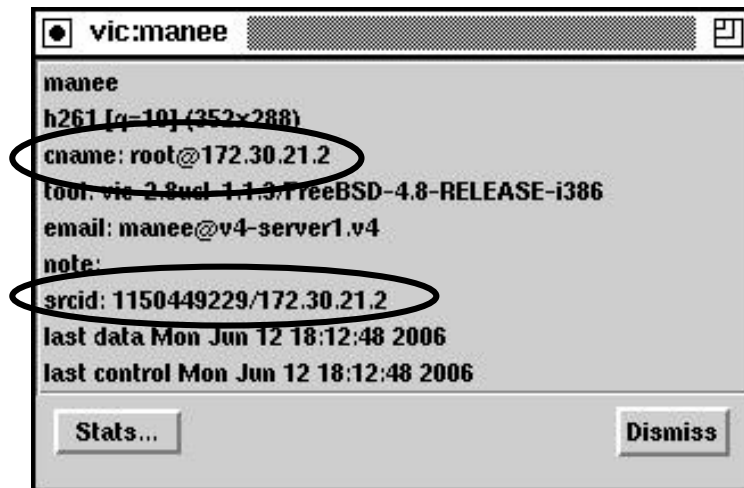


Figure 7.3 Detail of the v4 session on the v4 source

`cname` is a part of all session description entries to identify the originator. It is not necessary to be changed to IPv6 because it is not used as an address, but only as an identifier. The `srcid` consists of two numbers: a session identifier and the IP address of source from the network header. During a session, the identifier remains the same. The IP addresses – source and group, must be changed to IPv6 in order to allow the v4 recipient to join the group and exchange data packets.

When a packet has been sent to the network, the multicast routing protocol on the router containing the NAT-PT arranges for the packet to arrive. This is conventional multicast routing. NAT-PT checks the destination address. Here the address `224.200.200.200` is sought in the mapping rules. When found the destination (group) address is translated, in this case to `ff3e:ff00::e0c8:c8c8`. As the packet is being translated, the source address must also be processed. This is a unicast address and is handled as for any other unicast source address – if a mapping exists in the NAT-PT rules, that mapping is used, otherwise a new address is assigned (for v4 to v6 unicast no new mapping is created as the assignment algorithm is deterministic and repeatable. The

source address becomes 3ffe:501:41c:c1ad::ac1e:1502 in this example.

Vic relies on RTP and RTCP to transport a multicast session. RTP is used to deliver data packet. It works with RTCP to control and keep status of the session. The vic program picks the number 1062 to be the source port for sending data. It uses the port number 1063 to control the session. The port number 40000 is used to join this group and get the data. And the port number 40001 is used to keep status of the group and the using of data.

When the v4 source provides vic application, the packets are translated by NAT-PT following the rules. These addresses have been translated and kept in the mapping table as shown in Figure 7.4. The first record is the address mapping of RTP. The second is the mapping of RTCP. The port numbers are not changed because the rules were not defined to translate them.

Local Address (src)	Local Address (dst)	Remote Address (src)	Remote Address (dst)
172.30.21.2.1062	224.200.200.200.40000	3ffe:501:41c:c1ad::ac1e:1502.1062	ff3e:ff00::e0c8:c8c8.40000
172.30.21.2.1063	224.200.200.200.40001	3ffe:501:41c:c1ad::ac1e:1502.1063	ff3e:ff00::e0c8:c8c8.40001

Figure 7.4 Mapping table of IPv4 to IPv6 multicast address translation

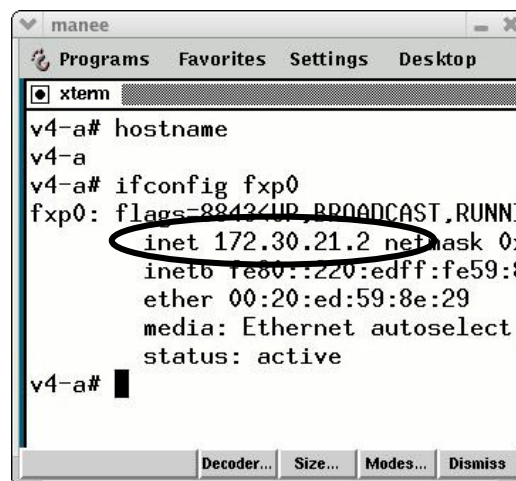
Result:

At the v6 site, host v6-a runs the vic program, the application joins the IPv6 group address ff3e:ff00::e0c8:c8c8 which as is the translated address. Vic receives the translated data packets as shown in the status display in Figure 7.5. It indicates a frame rate 8.1 f/s and bit rate 23 kb/s.



Figure 7.5 Menu of vic application on the v6 recipient

On the vic display, there are two video records. The first record presents that it comes from the v4 source address 172.30.21.2. The data streaming is an image of the terminal window on the v4-a node. It is displayed as in Figure 7.6.



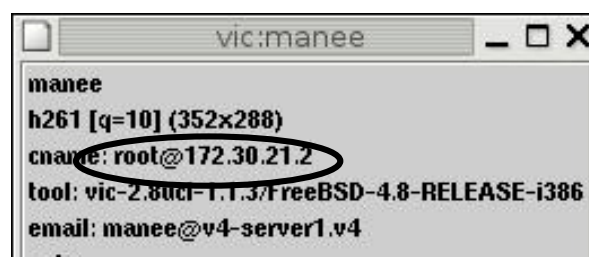
```

manee
Programs Favorites Settings Desktop
xterm
v4-a# hostname
v4-a
v4-a# ifconfig fxp0
fxp0: flags=8842<UP,BROADCAST,RUNNING>
    inet 172.30.21.2 netmask 0xffff0000
    inet6 fe80::220:edff:fe59:8e29
    ether 00:20:ed:59:8e:29
    media: Ethernet autoselect
    status: active
v4-a#
  
```

Figure 7.6 Display the v4 information on the v6 recipient

The information of this stream is presented in Figure 7.7. The field cname shows that the originator is the node 172.30.21.2. The result of address translation has been presented in the field srcid. NAT-PT translates the source address to the IPv6 address following the mapping rules. The result of the address mapping is presented in the mapping table Figure 7.4, which shows the active packet streams and their address mappings.

The source address in srcid is changed to the mapped address – IPv6 number 3ffe:b80:53:1c6::ac1e:1502. For the same session, the session identifier is the same number as used by vic on the v4 source.



```

vic:manee
manee
h261 [q=10] (352x288)
cname: root@172.30.21.2
tool: vic-2.8uci-1.1.3/FreeBSD-4.8-RELEASE-i386
email: manee@v4-server1.v4
  
```

Figure 7.7 Detail of the v4 session on the v6 recipient

In Figure 7.5, the second record belongs to the v6 recipient. The v6-a node shares its own session to the same IPv6 group with frame rate 8.0 f/s and bit rate 23 kb/s. The v6 session is named vic-v6a. At the v4 site, the v4-a source gets the shared session of the v6 recipient when the recipient joins the translated group. The result is shown on the vic display at the v4 node in Figure 7.8

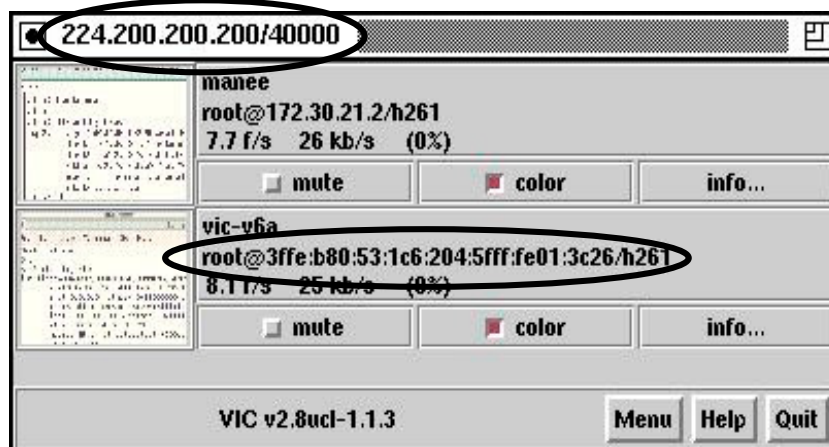


Figure 7.8 Menu of vic application on the v4 source after the v6 node joined the group

Scenario 2: Manual configuration for IPv6 to IPv4 address translation

The objective of this testing is similar to scenario 1. On the other hand it determines multicast address and packet translation in the other direction. If NAT-PT can provide address and

packet translation from IPv6 to IPv4, the v4 recipient can join the translated group and participate in multicast sessions from the v6 source.

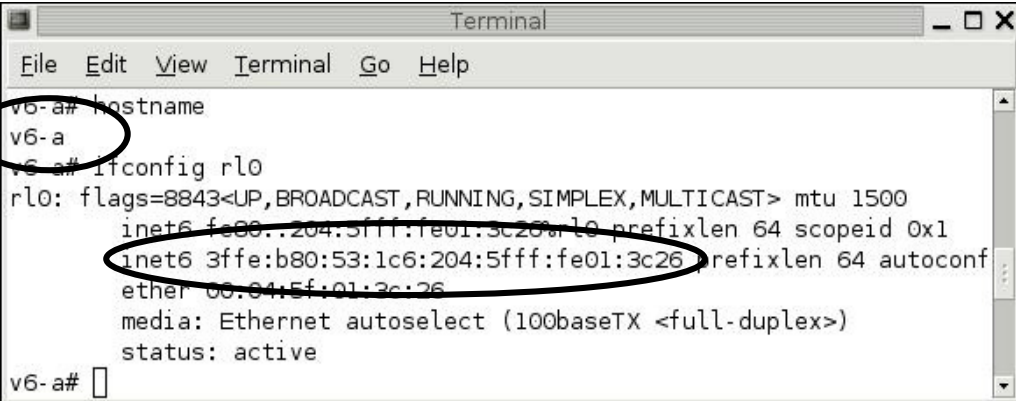
In this case, the v6 source node provides a multicast application. The v4 recipient wants to join the group of that application. The multicast IPv6 address, the v6 source address and the packet must be translated to IPv4. Then the multicast session can be forwarded to the v4 site. The addresses for translation have already been mapped in the rules before the application was started.

Procedure for this scenario is as follows:

This case has the same procedure as scenario 1. Here the v6 source provides a video conference to network. It configures the vic program with the IPv6 group address in the mapping rules. When the data packets arrive at NAT-PT router, they are translated following the rules. And they are sent to the recipient in the v4 site.

Detail of testing:

A terminal window on v6-a is used as the source of data streaming of vic program. The window presents name of the v6 source and network interface configuration for address checking at the other site. The terminal of v6-a shows name and network configuration as in Figure 7.1.



```

Terminal
File Edit View Terminal Go Help
v6-a# hostname
v6-a
v6-a# ifconfig r10
r10: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    inet6 fe80::204:5fff:fe01:3c26%r10 prefixlen 64 scopeid 0x1
    inet6 3ffe:b80:53:1c6:204:5fff:fe01:3c26 prefixlen 64 autoconf
    ether 00:04:5f:01:3c:26
    media: Ethernet autoselect (100baseTX <full-duplex>)
    status: active
v6-a#
  
```

Figure 7.1 Window terminal on the v6 source

Node v6-a has been configured with IP address 3ffe:b80:53:1c6:204:5fff:fe01:3c26. It runs vic application with the group number ff3e:ff00::e07b:7b7b and port 50000. The description of this

session is shown in Figure 7.2. The session name is vic-v6a. It is created by user root on the v6 node with address 3ffe:b80:53:1c6:204:5fff:fe01:3c26. The data streaming is encapsulated using technique h261. This session is sent with the rate 8.0 frame per second and 31 kilobit per second.



Figure 7.2 Menu of vic application on the v6 source

The information of this session is presented in Figure 7.3. Like scenario 1, the fields cname and srcid are used to determine address translation of NAT-PT to provide multicast communication. The addresses of creator and group number must be translated to IPv4 addresses in order to allow the v4 recipient to join the group and get data packets from the v6 source. However, the session identifier must not be changed. It is used to indicate the identity of the session between the v6 source and the v4 recipient.

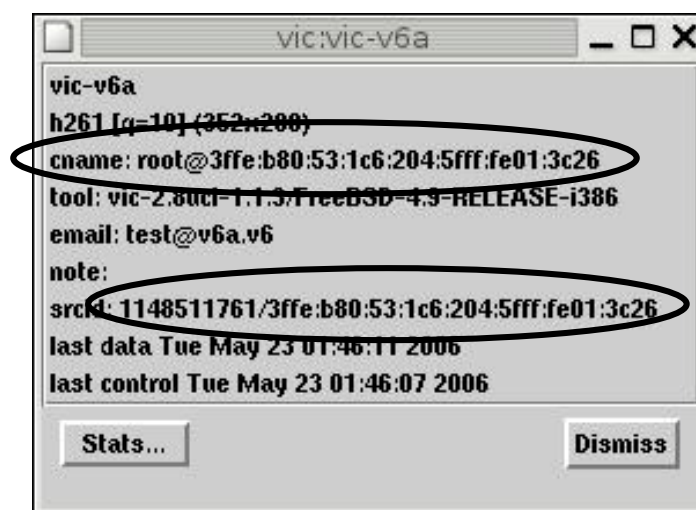


Figure 7.3 Detail of the v6 session on the v6 source

Cname looks like an e-mail address. It is used to describe the originator of the session. The creator is user root on the node 3ffe:b80:53:1c6:204:5fff:fe01:3c26. Srcid is used to identify the session number and IP address of node which creates this packet. It has been assigned the identifier number 1148511761. This packet is generated by the v6 source at IP address 3ffe:b80:53:1c6:204:5fff:fe01:3c26.

When a v6 packet has been sent to the network, the multicast routing protocol on NAT-PT arranges for the packet to arrive. NAT-PT checks destination address. It seeks the group address ff3e:ff00::e07b:7b7b in the mapping rules.

NAT-PT translates IPv6 source address and group (destination) address to IPv4 as indicated by Figure 7.4. IPv6 source 3ffe:b80:53:1c6:204:5fff:fe01:3c26 and group address ff3e:ff00::e07b:7b7b are mapped to the IPv4 addresses 10.10.10.17 and 224.123.123.123 respectively. The address mapping in the first record applies to RTP whereas the second applies to RTCP. The IPv6 source and group addresses are mapped to IPv4.

The vic program picks the number 1050 for the source port to send data. And it uses the port number 1051 to control the session. Port number 50000 is used to join this group and get the data, a port number 50001 is used to keep status of group RTCP. These ports are not translated in this case.

Local Address (src)	Local Address (dst)	Remote Address (src)	Remote Address (dst)
3ffe:b80:53:1c6:204:5fff:fe01:3c26.1050	ff3e:ff00::e07b:7b7b.50000	10.10.10.17.1050	224.123.123.123.50000
3ffe:b80:53:1c6:204:5fff:fe01:3c26.1051	ff3e:ff00::e07b:7b7b.50001	10.10.10.17.1051	224.123.123.123.50001

Figure 7.4 Mapping table of IPv6 to IPv4 multicast address translation

Result:

At the v4 site, the node v4-a runs the vic program and joins the group number 224.123.123.123 which is the translated address. Vic gets the translated data packets as shown in the display with frame rate 7.9 f/s and bit rate 21 kb/s without the dropped packet as in

Figure 7.5.

On the vic display, the first record is the IPv6 data stream. This session is named vic-v6a

and generated by the v6 node 3ffe:b80:53:1c6:204:5fff:fe01:3c26. The data streaming from the source node on vic program is presented in Figure 7.6. The detail of v6 session is in Figure 7.7. The second record is generated by the v4 recipient itself. It shares its own session to the same IPv4 group. At the v6 site, the v6-a node receives the date from the v4 node as shown on the display in Figure 7.8.

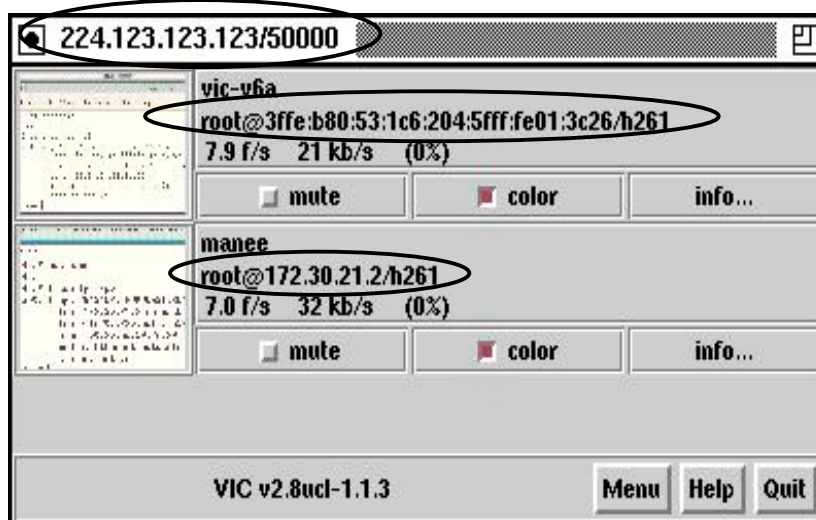


Figure 7.5 vic menu on the v4 recipient

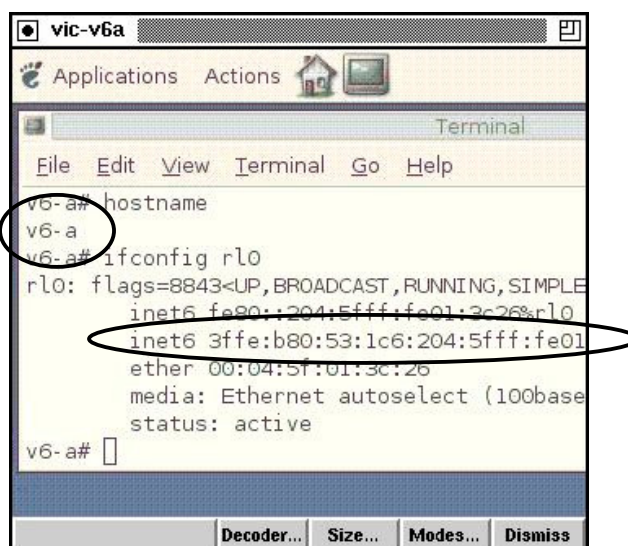


Figure 7.6 Display data of v6 session on the v4 recipient

The v4 recipient receives the data streaming from the v6 source. The description of the session in Figure 7.7 shows that the originator, in the field cname, who created this session is IPv6 address 3ffe:b80:53:1c6:204:5fff:fe01:3c26. Srcid contains the same session identifier as the session on the v6 source. However, srcid contains the IPv4 address rather than IPv6 because the source address was translated to the IPv4 address 10.10.10.17 by NAT-PT.

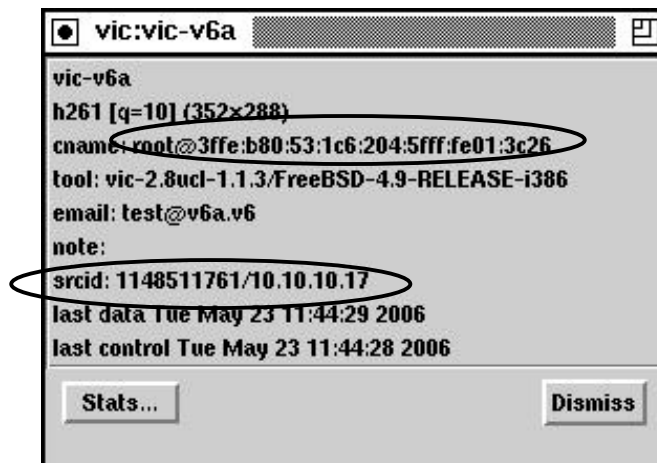


Figure 7.7 Detail of v6 session on the v4 recipient

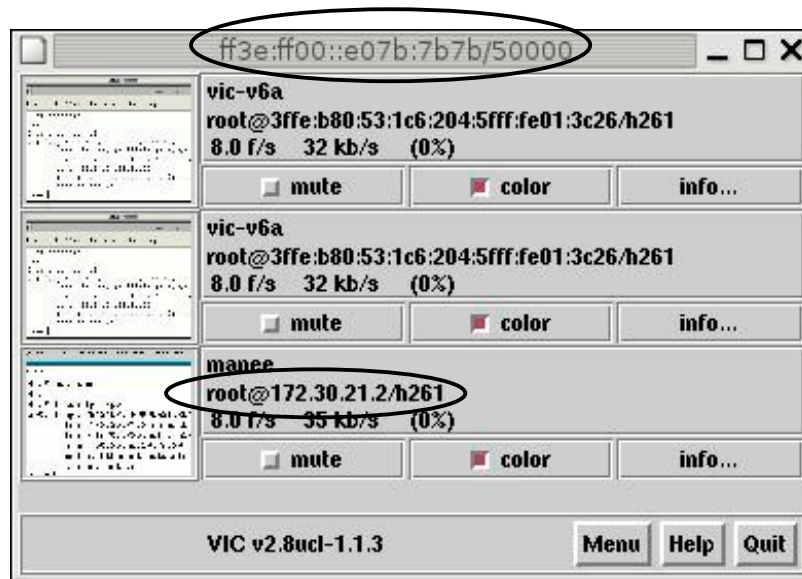


Figure 7.8 vic menu on the v6 source after join group

After the v4 recipient joins the translated group 224.123.123.123, it sends its own session to that group. At the v6 site, the v6 source gets the data streaming from the v4 recipient as seen in the third record in Figure 7.8. The v6 source sends the v6 data streaming to the v4 recipient as in the first record and, apparently due to a local bug on this v6 node, sends back to itself as in the second record.

Scenario 3: Automatic configuration for IPv4 to IPv6 address mapping

The objective of this testing is to determine the ability of the SAP-ALG to map multicast addresses from IPv4 to IPv6 and provide the appropriate information of the v4 application to the v6 listener.

The v4 announcer advertises a schedule of multicast services to the network. It uses the sdr program which relies on the SDR and SAP protocols to generate multicast information. Every node which receives this announcement knows the available multicast sessions and corresponding group addresses via the IPv4 SAP packet.

SAP-ALG on the router listens to the SAP packet. It intercepts and looks for group addresses in the packet. It generates the appropriate IPv6 group addresses and maps the v4

addresses from the SDP advertisement to the newly assigned v6 address. Then it advertises the mapped group to v6 world via a new generated IPv6 SAP packet. The ALG adds the mapped address to the mapping rules. NAT-PT relies on the mappings in the rules to translate multicast packets. The expected result is that the available multicast applications in v4 network are visible to the v6 listener.

Procedure for this scenario is as follows:

1. **To send the SAP packet to network:** The v4 source advertises multicast information to network via IPv4 SAP packet.
2. **To receive the packet:** IPv4 multicast routing protocol arranges for getting the v4 multicast packets, the IPv4 SAP packets. These packets are delivered into NAT-PT router.
3. **To provide address mapping:** SAP-ALG provides group address mapping. It creates a new IPv6 SAP packet that contains the mapped IPv6 group address to inform v6 sites. The way to provide address mapping has been presented in section 6.3.3.
4. **To transmit the new SAP packet:** NAT-PT sends the IPv6 SAP packet to v6 recipients.
5. **To get multicast data packet:** In the v6 world, v6 nodes discover the available multicast services by running the sdr program. When sdr is started, the v6 node joins SAP group. Available multicast sessions are displayed to the user by the sdr program. The v6 node can join any available session at the appropriate time.

Detail of testing:

Node v4-a uses the sdr program to create information of multicast services. The table of these v4 applications is named “Mcast Schedule in v4 world” as in Figure 7.1 It contains information of several multicast applications as in Figure 7.2 The group addresses highlighted by the rectangle are used to identify each application. The IP address in the circle belongs to the announcer who generated this advertisement. The announcer is node v4-a which has been configured with IP address 172.30.21.2.



Figure 7.1 Menu of sdr program on the v4 announcer

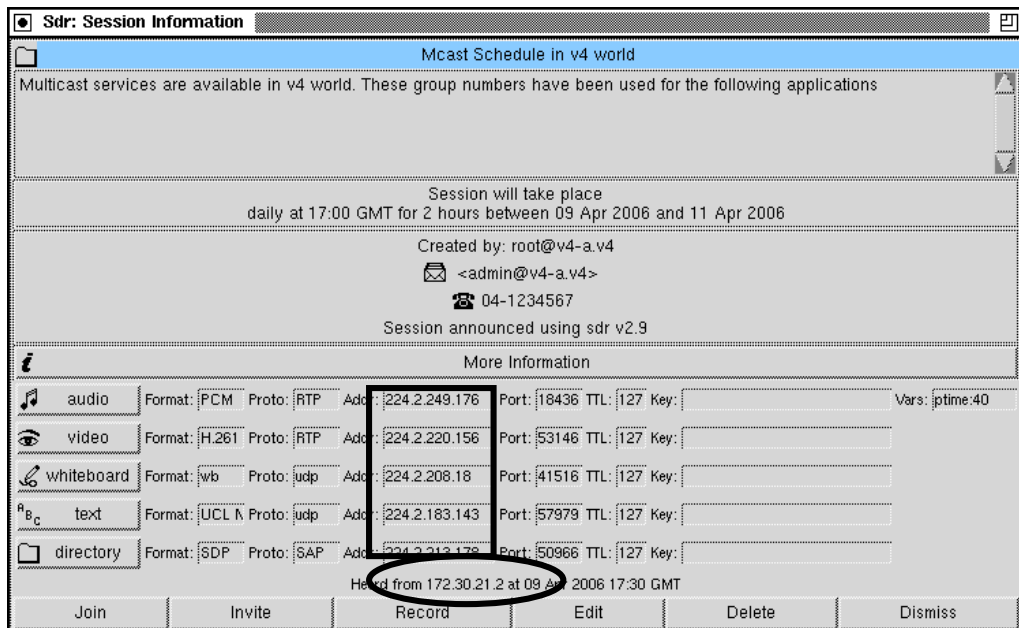


Figure 7.2 Information of each application on sdr of the v4 announcer

The information in Figure 7.2 is the session description. It is carried as the payload of SAP

packet and delivered to network. RFC2974 [45] defined a well known IP address for IPv4 SAP packets which is 224.2.127.254. The announcer, v4-a, sends the SAP packet to network to this destination address. The NAT-PT system joins the group when SAP-ALG is started.

The ALG on the router joins the SAP group and receives SAP packets. It copies the session description inside the IPv4 SAP payload (SDP). Then it provides address mapping for the groups for the multicast services advertised. The algorithm to provide address mapping from IPv4 to IPv6 has been presented in section 6.3.2.

After address mapping, SAP-ALG generates a new IPv6 SAP packet to contain the mapped address. The format of IPv6 SAP packet is defined in RFC2974 [45]. It consists of two parts: payload and header. The session description in the payload of an IPv6 SAP packet is similar to the original payload of an IPv4 SAP except for the group number and the IP address of the creator. The group number is the translated address – an IPv6 group, whereas the creator is IP address of the SAP-ALG node which generates the new IPv6 SAP packet. The ALG for this test is located at the NAT-PT router which has been configured with IPv6 address 3ffe:b80:53:1c6:2c1:28ff:fe02:5a54. Thus the ALG uses this IPv6 address to identify the source of the new IPv6 packet.

The IPv6 SAP header contains the IPv6 address of the SAP-ALG node (the NAT-PT router) as source, and the well known IPv6 SAP group identifier as destination address. The IPv6 SAP address is ff0e::2:7ffe. The NAT-PT system sends the IPv6 SAP packet to the v6 network.

Result:

At the v6 site, the node v6-b runs sdr to look for a table of multicast services. It joins the IPv6 SAP group. It gets the session named “Mcast Schedule in v4 world”. This session is displayed on sdr menu of the v6 recipient as in Figure 7.3 The session information of each service is presented in the sdr dialog as in Figure 7.4 The IPv6 group address of each application highlighted in the rectangle has already been mapped by SAP-ALG.

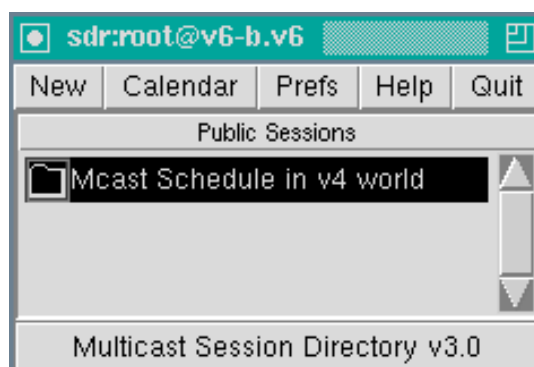


Figure 7.3 Menu of sdr program on v6 recipient

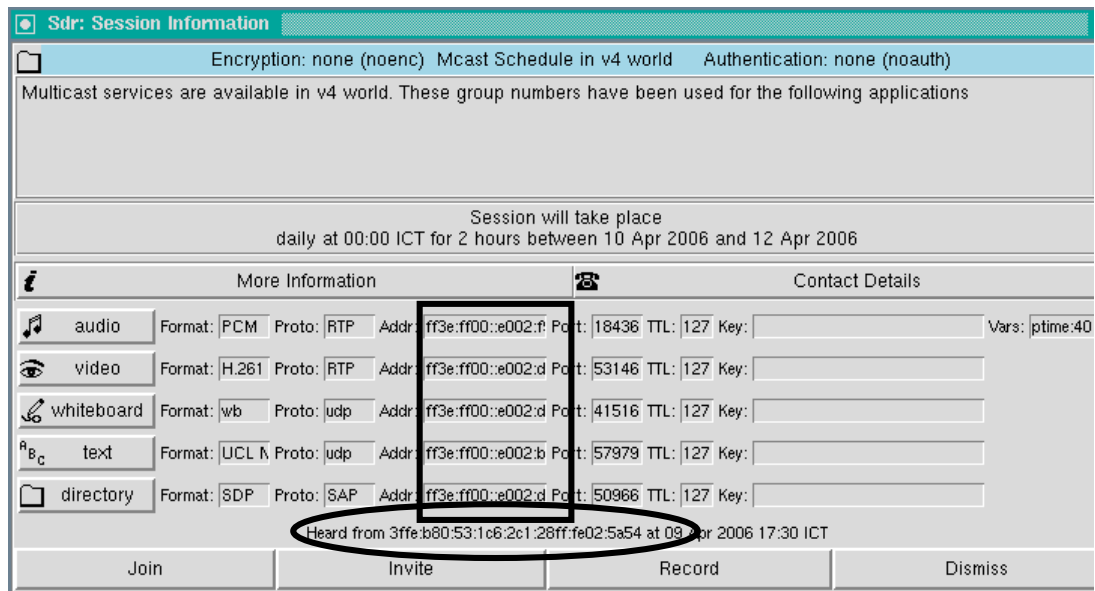


Figure 7.4 Information of each application on sdr of v6 recipient

Scenario 4: Automatic configuration for IPv6 to IPv4 address mapping

The objective of this testing is similar to scenario 3. However, this part determines work of SAP-ALG to map from multicast IPv6 to IPv4 address and provide the appropriate information of the sessions created by v6 nodes to the v4 listener.

The v6 announcer advertises a table of v6 multicast services to the network. It uses the sdr

program to generate a session description for each session. The expected result is that the SAP-ALG can provide group address mapping from IPv6 to IPv4. The multicast sessions in the v6 network should be visible to the v4 listener.

Procedure for this scenario is as follows:

The procedure to provide session description for a multicast application is similar to scenario 3. In this direction, the v6 announcer provides a multicast schedule to the network. The description is delivered to every v6 network. SAP-ALG generates IPv4 address mapping for the IPv6 groups in the IPv6 SDP payload of the SAP packet. Then it includes the mapped addresses in the mapping rules. After that the ALG generates a new IPv4 SAP packet to contain the mapped group address – a multicast IPv4 address. It sends the new packet to the v4 network.

Detail of testing:

The node v6-b generates multicast information advertisement by using the sdr program. On the v6 announcer, the schedule named “Mcast service in v6 site” as in Figure 7.1 is advertised to network. It contains the description of a video conference as in Figure 7.2.

The IPv6 group addresses in the rectangle are used to identify this session. This description is generated by the creator with IPv6 address 3ffe:b80:53:1c6:203:6dff:fe16:ad50.

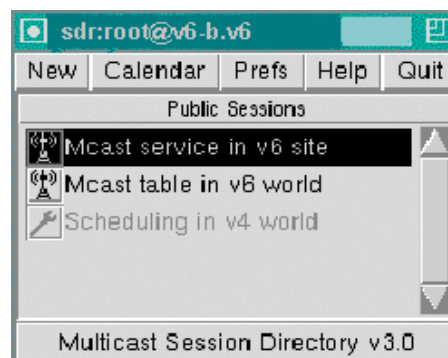


Figure 7.1 Menu of sdr program on the v6 source

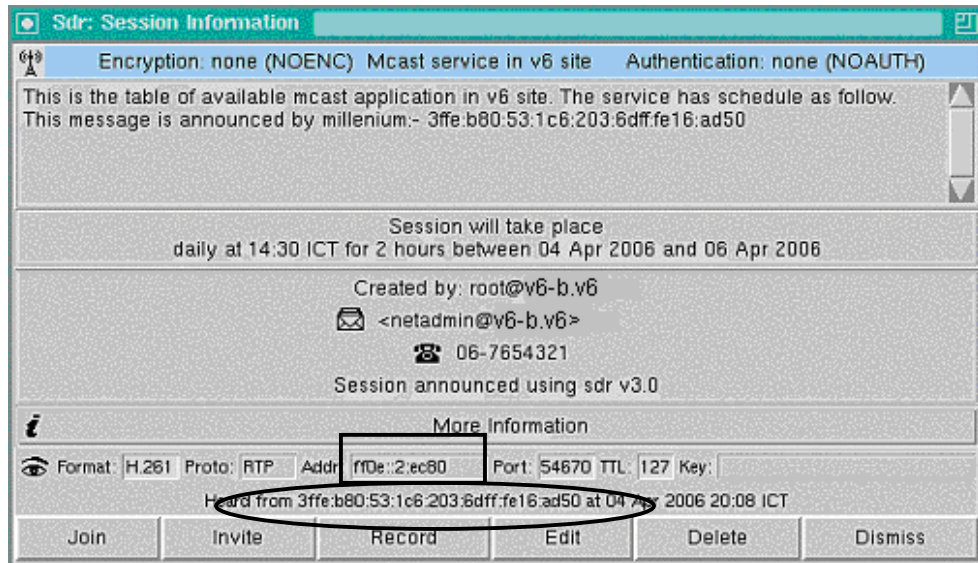


Figure 7.2 Information of each application on sdr of the v6 source

This session description is the payload of an IPv6 SAP packet. It is delivered to network by the IPv6 SAP. When the packet arrives at the NAT-PT router, SAP-ALG obtains the IPv6 group address from inside the SAP payload. Then the SAP-ALG provides group address mapping from IPv6 to IPv4. It generates a new IPv4 SAP packet that contains the mapped group number – an IPv4 multicast address. The new packet is advertised to the v4 network.

Result:

At the v4 site, the node v4-a runs the sdr program to find multicast services available in the network. It joins the IPv4 SAP group. The session named “Mcast service in v6 site” is displayed on the sdr menu as in Figure 7.27. The session information has only one service which is available from the v6 site. It is a videoconference as presented in Figure 7.28.

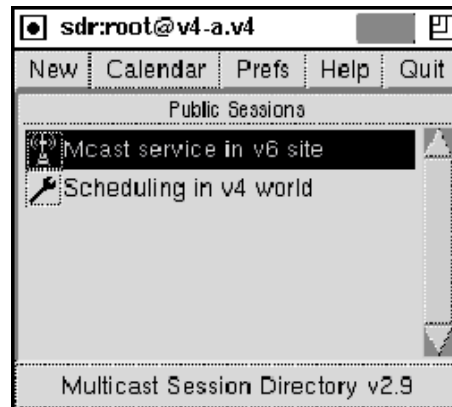


Figure 7.3 Menu of sdr program on v4 recipient

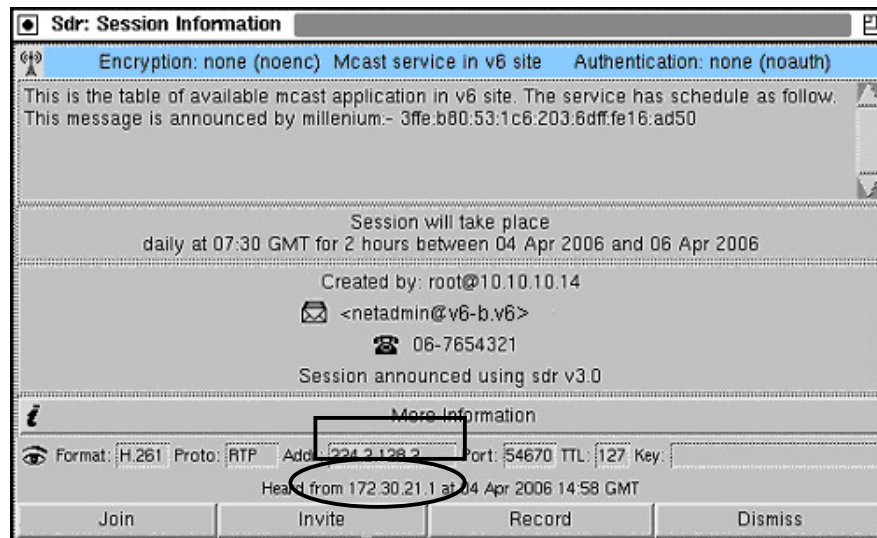


Figure 7.4 Information of each application on sdr of v4 recipient

The description of the multicast application in the IPv4 SAP packet is similar to the IPv6 SAP packet. However, the group address and the creator address in this packet are changed. The group number is the IPv4 multicast address which was mapped by SAP-ALG. Since the description is generated by the ALG, the address of creator is the IP address of the SAP-ALG node which was configured as 172.30.21.1.

7.5 Summary

The result of implementation and testing shows that NAT-PT can provide multicast

address and packet translations between IPv4 and IPv6. It can provide multicast sessions and communication between v4 and v6 nodes. The extension module works, and it enhances NAT-PT to support multicast.

SAP-ALG works well with IPv4 and IPv6 SAP. It can provide address mapping for multicast IPv4 and IPv6 groups. It can announce multicast schedules that available in the v4 network to v6 sites and vice versa. Multicast sessions in the networks running both IPv4 and IPv6 are available to systems in the other network.

However, the implementation is not complete. We have tested with only one NAT-PT router. A second NAT-PT could be added in either of two places. It could parallel the first NAT-PT, offering an alternative path and redundancy. Or it could connect to an entirely different network, with packets passing through both NAT-PTs being translated out of, and then back into, their original protocol.

The first of those cases allows for the possibility of looping packets – a packet originating at the v4 side might be translated to v6 then sent over the v6 network to the other NAT-PT router. There it could be translated back to v4 and transmitted into the v4 network, where it would be received, again, by the first NAT-PT router. Clearly a mechanism is required to prevent this from occurring.

This may seem simple – if no mapping for the translated-from-v4 IPv6 group back to a v4 group exists at the second NAT-PT, there is no problem. However, these mappings are created automatically by the SAP-ALG. The solution to the problem may lie in the ALG rather than in the kernel modules, but a solution is required.

The second case requires consideration as to the address mappings used. Is a group created in one IPv4 (or IPv6) world required to have the same group identifier (address) in the other world running the same protocol? In the net common to both NAT-PT routers, how do those routers cooperate to avoid causing address duplication when it should not occur, get deliberately both generate the same mapping when the two groups in the other networks are intended to be the same. Again, as address mapping are created by the SAP-ALG a solution to this problem needs to be found there.

As well as those issues, which require more research, there are simpler problems with the implementation that need work. The API to the kernel NAT-PT functions is poorly documented and seems poorly designed. This has caused this implementation to avoid any API interactions at all, relying instead entirely upon the supplied configuration tool to add NAT-PT mappings when required. While acceptable for human use, this method is not adequate for automatic updates from the SAP-ALG (or the DNS-ALG for unicast applications.) A new API is required.