# Chapter 7

# Simulation Environment for Anycast Routing

This chapter presents the details of building and testing a simulation environment for anycast routing protocol using JaNetSim (Lim, S.H. et al., 2001). Several phases of implementation are required as the process of building the simulation environment for anycast routing is inherently complex. Several protocols of IPv6 will be implemented and tested in the process.

## 7.1 Phases of Implementations

Several phases are required in the building of the simulation environment for anycast routing protocol:

1) Creates IPv6 packet according to IPv6 Protocol Specification by Deering, S. and Hinden, R. (1998), including the IPv6 header and IPv6 extension headers.
2) Prepare the basic IPv6 environment, such as implementataion of Neighbor Discovery (Narten, T. et al., 1998), Stateless Address Autoconfiguration (Thompson, S. and Narten, T., 1998) and Internet Control Message Protocol Version 6 (ICMPv6) (Conta, A. and Deering, S., 1995).
3) Building the Unicast routing protocol, the Routing Information Protocol next generation (RIPng) (Malkin, G. and Minnear R., 1997).
4) Building the group management protocol for IPv6, Multicast Listener Discovery (Deering, S. et al., 1999).
5) Implement RIPng extension for anycast routing.

6) Implement PIM-SM extension.

7) Implement the load-balancing schemes: shortest, round robin and fuzzy.

8) Implement the proposed nearest PIM-SM extension

## 7.2 Overview of Simulation Environments

Simulation allows the experiments and analysis of a system without the need to construct the actual system, as it is usually too expensive and impractical to build an actual system for every experiment. Simulation is an effective approach in the design, evaluation and experimentation of new networking protocols, theories or algorithms.

JaNetSim Network Simulator, a discrete event, Java-based object-oriented network simulator developed by Lim, S.H. et al. (2001) is used as the base engine for the simulation. The basic objects of the simulator are shown in Figure 7.1.

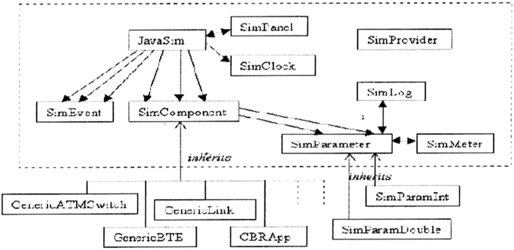


***Figure 7.1*** *JaNetSim Network Simulator Objects*

Major components involved in the simulations are Asynchronous Transfer Mode Label Switching Router (ATM LSR), IPv6 Broadband Terminal Equipment

(IPv6BTE), Generic Link, Variable Bit Rate IPv6 Sender Application (VBRIPv6SenderApp) and VBR IPv6 Application (VBRIPv6App). As the original network simulator is running in an IPv4 environment, work was done to enable the network simulator to run in an IPv6 environment.

A Generic Link will connect the routers in the simulation, the ATM LSRs to form a domain. IPv6 Broadband Terminal Equipment represents the aggregate traffic from the customer site. One or more applications generating network traffic will be created and connected to the IPv6 B-TE. Each IPv6 B-TE is required to be connected to a router through a generic link. IPv6 Sender Applications will generate the network traffic in the simulation and the IPv6 Applications that serve as the anycast service provider will be responding to each request.

## 7.3 Network Simulation Components

Several components are created or enhanced during the process of preparing the simulation environment for the anycast routing.

**Existing components**

1) The Asynchronous Transfer Mode Label Switching Router (ATM LSR)
    - Enriched features to support basic IPv6 features, such as Neighbor Discovery, Stateless Address Autoconfiguration, Internet Control Message Protocol Version 6 (ICMPv6), Multicast Listener Discovery (MLD).

- Support for Protocol Independent Multicast Sparse Mode (PIM-SM), such as BootStrap Router and PIM-SM messages handling..
- Support multiple load-balancing schemes.

2) Generic Link
   - Include additional link cost in the properties list.

**New Components**

1) IPv6BTE
   - A component that inherits the SimComponent, act as the Broadband Terminal Equipment (B-TE) in the IPv4 environment.
   - Represents the aggregate traffic from the customer site.
2) SimParamIPv6
   - A component that inherits the SimParameter, to store the new IPv6 address.
3) SimParamIPv6Prefix
   - A component that inherits the SimParameter, to store the IPv6 prefix.
   - Allow fast IPv6 prefix configuration, like ff::1/64 as stated in the IPv6 addressing architecture.
4) SimParamIPv6Rtable
   - A component that inherits SimParameter that stores the route table information for IPv6.
5) SimParamPIMSMv2Gtable
   - A component that inherits SimParameter that stores the anycast group member information for all anycast group.

6) VBRIPv6SenderApp

- A component that inherits the SimComponent, which acts as the source of the IPv6 anycast.

7) VBRIPv6App

- A component that inherits the SimComponent, which acts as the service provider of the Ipvv6 anycast service.

## 7.4 Objects and classes

Besides the new components added, several helper java files are included as well in the JaNetSim network simulator. The java files added are listed as follows:

1) Prefix.java

- Provide details information of a Prefix, according to the description of prefixes in the ICMPv6 for IPv6 Specification.

2) GMAC.java

- Generate a MAC address for each component that has an interface.

3) IPv6Packet.java

- Define the structure of an IPv6 packet, including the IPv6 header and all the IPv6 extension headers.

4) IPv6.java

- Define constants used in the IPv6 environment such as the IPv6 unspecified and loop back addresses, all nodes addresses, all routers addresses, router constants, host constants and node constants.
- Provide some handling of the IPv6 packet.
- Some helper functions that

5) IPv6Timer.java

- There are many occasions where timers are needed in the simulation environment. For example the Other Querier Present Timer in the ICMPv6 and the BootStrap Timer in the PIM-SM.
- The timer start-up time, stop time and the timeout period can be controlled via this java class.

6) ICMPv6Msg.java

- Define the format of the ICMPv6 messages such as Router Advertisement, Group Membership Query, Group Membership Report and Group Membership Done.

7) ICMPv6.java

- Handling of the ICMPv6 messages.
- Defines the mechanisms of the group management protocol for IPv6, the Multicast Listener Discovery Version 2 (MLDv2).

8) RIPng.java

- Defines the format of the RIPng messages.
- Provides the mechanisms for the RIPng, such as the periodic route table advertisements and the triggered update.

9) PatriciaTree.java

- Provides fast route table lookup algorithms and is built according to Sivarajan, Kumar N. (1999).

10) PIMSMv2Msg.java

- Defines the format of the PIM-SM messages such as the Join-Prune message and the BootStrap message.

11) PIMSMv2.java

- Defines the mechanisms for handling of the PIM-SM messages.
- Defines the PIM-SM mechanisms such as Rendezvous Point (RP) election, BootStrap Router (BSR) election, Designated Router (DR) election and load-balancing schemes.

12) EncodedAddress.java

- Defines the structure of the Encoded Unicast Address and Encoded Group Address used in the PIM-SM messages.

13) FuzzyRuleBase.java

- Initiates the fuzzy sets and rules
- Fuzzifies the inputs by implementing inference and composition methods (PRODUCT-MAX model)
- Performs defuzzification

## 7.5 Testing and Implementation

Testing of the components and protocols implemented is very important before running a simulation. Failure to do this would mean that the simulations might have to be rerun if problems are discovered later.

Link speed for the link between a B-TE and a router will be 25.0 MBits/sec and the link speed for the link between routers will be 7.0 MBits/sec. The link cost for all the links will be 1.0. The size for all output queues of the routers is 1000.

### 7.5.1 Neighbor Discovery and Stateless Autoconfiguration

**Overview**

Neighbor Discovery and Stateless Autoconfiguration are implemented to provide a basic simulation environment for IPv6. The routers will advertise the prefixes to each attached links. The B-TEs will capture the receive prefix and configure their interface addresses.

**Topology Used**

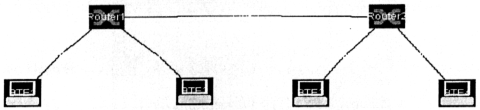

***Figure 7.2*** *Topology used in testing Neighbor Discovery and Stateless Autonconfiguration in IPv6*

**Parameter Settings and Configurations**

**Table 7.1** Parameter Settings for Router1

| Parameter | Value |
|---|---|
| EUI-64 identifiers for interface to Link1 | 0000:0000:0000:0002 |
| Prefix for interface to Link1 | 0001:0000:0000:0000:0000:0000:0000:0000/64 |
| EUI-64 identifiers for interface to Link2 | 0000:0000:0000:0003 |
| Prefix for interface to Link2 | 0002:0000:0000:0000:0000:0000:0000:0000/64 |
| EUI-64 identifiers for interface to Link5 | 0000:0000:0000:0009 |
| Prefix for interface to Link5 | 0005:0000:0000:0000:0000:0000:0000:0000/64 |

Table 7.2 Parameter Settings for Router2

| Parameter | Value |
|---|---|
| EUI-64 identifiers for interface to Link3 | 0000:0000:0000:0005 |
| Prefix for Interface to Link3 | 00030000:0000:0000:0000:0000:0000:0000/64 |
| EUI-64 identifiers for interface to Link4 | 0000:0000:0000:0006 |
| Prefix for Interface to Link4 | 0004:0000:0000:0000:0000:0000:0000:0000/64 |
| EUI-64 identifiers for interface to Link5 | 0000:0000:0000:000a |
| Prefix for Interface to Link5 | 0005:0000:0000:0000:0000:0000:0000:0000/64 |

Table 7.3 Parameter Settings for B-TEs

| BTE | Parameter | Value |
|---|---|---|
| BTE1 | EUI-64 identifiers for interface to Link1 | 0000:0000:0000:0001 |
| BTE2 | EUI-64 identifiers for interface to Link2 | 0000:0000:0000:0004 |
| BTE3 | EUI-64 identifiers for interface to Link3 | 0000:0000:0000:0007 |
| BTE4 | EUI-64 identifiers for interface to Link4 | 0000:0000:0000:0008 |

**Simulation Results**

Table 7.4 IPv6 addresses for B-TEs after simulation

| BTE | Parameter | Value |
|---|---|---|
| BTE1 | IPv6 Address for Link1 | 0001:0000:0000:0000:0000:0000:0000:000 1 |
| BTE2 | IPv6 Address for Link2 | 0002:0000:0000:0000:0000:0000:0000:000 4 |
| BTE3 | IPv6 Address for Link3 | 0003:0000:0000:0000:0000:0000:0000:000 7 |
| BTE4 | IPv6 Address for Link4 | 0004:0000:0000:0000:0000:0000:0000:000 8 |

The BTEs obtain their prefix information from their directly attached routers, performing autoconfiguration. Neighbor Discovery is not fully tested, as it is not a mandatory protocol for the implementation of anycast and constrained by this version of the network simulator. It is included so that it can be tested in the newer version of the network simulator.

### 7.5.2 RIPng

**Overview**

Routing Information Protocol next generation (RIPng) is implemented in this thesis as the unicast routing protocol for IPv6. The routers will advertise RIPng messages periodically and some triggered updates to build the route table.

**Topology Used**

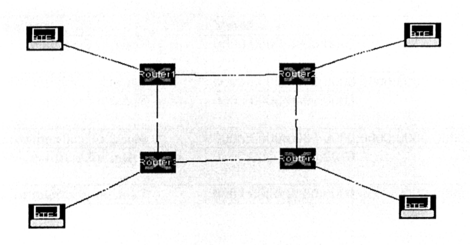


***Figure 7.3*** *Topology used for testing RIPng*

**Parameter Settings and Configurations**

**Table 7.5** Parameter Settings for Router1

| **Parameter** | **Value** |
|---|---|
| EUI-64 identifiers for interface to Link1 | 0000:0000:0000:0005 |
| Prefix for interface to Link1 | 0001:0000:0000:0000:0000:0000:0000:0000/64 |
| EUI-64 identifiers for interface to Link5 | 0000:0000:0000:0007 |
| Prefix for interface to Link5 | 0005:0000:0000:0000:0000:0000:0000:0000/64 |
| EUI-64 identifiers for interface to Link6 | 0000:0000:0000:0004 |
| Prefix for interface to Link6 | 0006:0000:0000:0000:0000:0000:0000:0000/64 |

**Table 7.6** Parameter Settings for Router2

| Parameter | Value |
|---|---|
| EUI-64 identifiers for interface to Link2 | 0000:0000:0000:0009 |
| Prefix for interface to Link2 | 0002:0000:0000:0000:0000:0000:0000:0000/64 |
| EUI-64 identifiers for interface to Link5 | 0000:0000:0000:0008 |
| Prefix for interface to Link5 | 0005:0000:0000:0000:0000:0000:0000:0000/64 |
| EUI-64 identifiers for interface to Link8 | 0000:0000:0000:000b |
| Prefix for interface to Link8 | 0008:0000:0000:0000:0000:0000:0000:0000/64 |

**Table 7.7** Parameter Settings for Router3

| Parameter | Value |
|---|---|
| EUI-64 identifiers for interface to Link3 | 0000:0000:0000:0002 |
| Prefix for interface to Link3 | 0003:0000:0000:0000:0000:0000:0000:0000/64 |
| EUI-64 identifiers for interface to Link6 | 0000:0000:0000:0003 |
| Prefix for interface to Link6 | 0006:0000:0000:0000:0000:0000:0000:0000/64 |
| EUI-64 identifiers for interface to Link7 | 0000:0000:0000:0010 |
| Prefix for interface to Link7 | 0007:0000:0000:0000:0000:0000:0000:0000/64 |

**Table 7.8** Parameter Settings for Router4

| Parameter | Value |
|---|---|
| EUI-64 identifiers for interface to Link4 | 0000:0000:0000:000e |
| Prefix for interface to Link4 | 0004:0000:0000:0000:0000:0000:0000:0000/64 |
| EUI-64 identifiers for interface to Link7 | 0000:0000:0000:000d |
| Prefix for interface to Link7 | 0007:0000:0000:0000:0000:0000:0000:0000/64 |
| EUI-64 identifiers for interface to Link8 | 0000:0000:0000:000c |
| Prefix for interface to Link8 | 0008:0000:0000:0000:0000:0000:0000:0000/64 |

**Table 7.9** Parameter Settings for B-TEs

| BTE | Parameter | Value |
|---|---|---|
| BTE1 | EUI-64 identifiers for interface to Link1 | 0000:0000:0000:0006 |
| BTE2 | EUI-64 identifiers for interface to Link2 | 0000:0000:0000:000a |
| BTE3 | EUI-64 identifiers for interface to Link3 | 0000:0000:0000:0001 |
| BTE4 | EUI-64 identifiers for interface to Link4 | 0000:0000:0000:000f |

**Table 7.10** Route table for Router1

| Destination Network | Link Cost | Next Hop Address |
|---|---|---|
| 0001:0000:0000:0000:0000:0000:0000:0000 | 1 | 0001:0000:0000:0000:0000:0000:0000:000f |
| 0005:0000:0000:0000:0000:0000:0000:0000 | 1 | 0005:0000:0000:0000:0000:0000:0000:0011 |
| 0006:0000:0000:0000:0000:0000:0000:0000 | 1 | 0006:0000:0000:0000:0000:0000:0000:000e |
| 0002:0000:0000:0000:0000:0000:0000:0000 | 2 | 0005:0000:0000:0000:0000:0000:0000:0012 |
| 0003:0000:0000:0000:0000:0000:0000:0000 | 2 | 0006:0000:0000:0000:0000:0000:0000:000d |
| 0007:0000:0000:0000:0000:0000:0000:0000 | 2 | 0006:0000:0000:0000:0000:0000:0000:000d |
| 0008:0000:0000:0000:0000:0000:0000:0000 | 2 | 0005:0000:0000:0000:0000:0000:0000:0012 |
| 0004:0000:0000:0000:0000:0000:0000:0000 | 3 | 0005:0000:0000:0000:0000:0000:0000:0012 |

Table 7.10 only shows part of the information in the route table. The correctness of the route table information in the Router1 justifies that the RIPng is successfully implemented. Route tables for the other routers are not shown here to simplify the discussion.

## 7.5.3 RIPng extension

**Overview**

The RIPng extension is implemented for anycast routing. The RIPng implemented is based on the model reviewed by Park, Vicent D. and Macker, Joseph P. (1999b). The anycast packets sent by the sender will reach the nearest interface joining the anycast group.

**Topology Used**

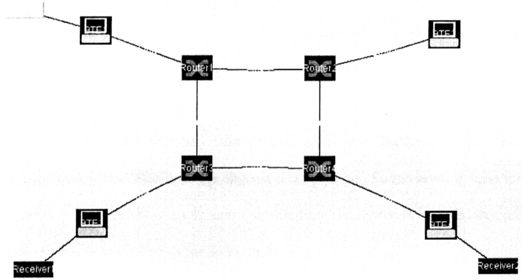


***Figure 7.4*** *Topology used in testing RIPng extension for anycast*

**Parameter Settings and Configurations**

The parameter settings for routers and B-TEs are same as the parameter settings in Table 6.5, 6.6, 6.7, 6.8 and 6.9.

**Table 7.11** Parameter Settings for Sender1

| **Parameter** | **Value** |
|---|---|
| Bit Rate (MBits/s) | 1.0 |
| Mean Burst Length (μsecs) | 5000.0 |
| Mean Interval Between Bursts (μsecs) | 15000.0 |
| Start Time (secs) | 70 |
| Number of MBits to be sent | 2.0 |
| Repeat count (-1=infinite) | 1 |
| Delay between calls (μsecs) | 3000000 |
| Destination IPv6 | 3e00:0000:0000:0000:fdff:ffff:ffff:fffe |

**Table 7.12** Parameter Settings for Receiver1 and Receiver2

| **Parameter** | **Value** |
|---|---|
| Group Address To Join | 3e00:0000:0000:0000:fdff:ffff:ffff:fffe |
| Join Group Time (secs) | 50 |
| Leave Group Time (secs) (-1=infinite) | -1 |

The anycast group here is 3e00:0000:0000:0000:fdff:ffff:ffff:fffe. It is allocated from the unassigned unicast address space (refer to Table 2.5) and the anycast ID used here is actually the anycast ID for Mobile IPv6 Home Agent (HA) (Johnson, D. and Deering, S., 1999). The reason that the join group time for the anycast receiver is 50 seconds is to allow the route table for the topology to stabilize before it joins the anycast group. The sender will only start generating network traffic after the route table for the topology is stable after the anycast receivers join. So the sending time for the sender is 70 seconds, 20 seconds later than the time the receivers join the anycast group. The simulation will be run for 80 seconds.

**Simulation Results**

**Table 7.13** Route table for Router1

| Destination Network | Link Cost | Next Hop Address |
|---|---|---|
| 0001:0000:0000:0000:0000:0000:0000:0000 | 1 | 0001:0000:0000:0000:0000:0000:0000:0005 |
| 0005:0000:0000:0000:0000:0000:0000:0000 | 1 | 0005:0000:0000:0000:0000:0000:0000:0007 |
| 0006:0000:0000:0000:0000:0000:0000:0000 | 1 | 0006:0000:0000:0000:0000:0000:0000:0004 |
| 0002:0000:0000:0000:0000:0000:0000:0000 | 2 | 0005:0000:0000:0000:0000:0000:0000:0008 |
| 0003:0000:0000:0000:0000:0000:0000:0000 | 2 | 0006:0000:0000:0000:0000:0000:0000:0003 |
| 0007:0000:0000:0000:0000:0000:0000:0000 | 2 | 0006:0000:0000:0000:0000:0000:0000:0003 |
| 0008:0000:0000:0000:0000:0000:0000:0000 | 2 | 0005:0000:0000:0000:0000:0000:0000:0008 |
| 0004:0000:0000:0000:0000:0000:0000:0000 | 3 | 0005:0000:0000:0000:0000:0000:0000:0008 |
| 3e00:0000:0000:0000:fdff:ffff:ffff:fffe | 2 | 0006:0000:0000:0000:0000:0000:0000:0003 |
| 3e00:0000:0000:0000:fdff:ffff:ffff:fffe | 3 | 0005:0000:0000:0000:0000:0000:0000:0008 |

Table 7.14 Route table for Router3

| Destination Network | Link Cost | Next Hop Address |
|---|---|---|
| 0003:0000:0000:0000:0000:0000:0000:0000 | 1 | 0007:0000:0000:0000:0000:0000:0000:0010 |
| 0006:0000:0000:0000:0000:0000:0000:0000 | 1 | 0006:0000:0000:0000:0000:0000:0000:0003 |
| 0007:0000:0000:0000:0000:0000:0000:0000 | 1 | 0003:0000:0000:0000:0000:0000:0000:0002 |
| 0001:0000:0000:0000:0000:0000:0000:0000 | 2 | 0006:0000:0000:0000:0000:0000:0000:0004 |
| 0004:0000:0000:0000:0000:0000:0000:0000 | 2 | 0007:0000:0000:0000:0000:0000:0000:000d |
| 0005:0000:0000:0000:0000:0000:0000:0000 | 2 | 0006:0000:0000:0000:0000:0000:0000:0004 |
| 0008:0000:0000:0000:0000:0000:0000:0000 | 2 | 0006:0000:0000:0000:0000:0000:0000:0004 |
| 0002:0000:0000:0000:0000:0000:0000:0000 | 3 | 0006:0000:0000:0000:0000:0000:0000:0004 |
| 3e00:0000:0000:0000:fdff:ffff:ffff:fffe | 1 | 0003:0000:0000:0000:0000:0000:0000:0001 |
| 3e00:0000:0000:0000:fdff:ffff:ffff:fffe | 2 | 0007:0000:0000:0000:0000:0000:0000:000d |

Table 7.15 Simulation results for testing RIPng extension

| Parameter | Sender1 | Receiver1 | Receiver2 |
|---|---|---|---|
| Total Packets Sent | 4716 | 0 | 0 |
| Total Received Packets | 0 | 4716 | 0 |

The simulation results justifies that the RIPng extension for anycast routing is successfully implemented. Receiver1 is the nearest interface for Sender1 (refer to Table 6.13 and Table 6.14). Router1 will route the packets to Router3 and then Router3 will route them towards Receiver1. Eventually all the packets for the anycast group 3e00:0000:0000:0000:fdff:ffff:ffff:fffe will be routed towards Receiver1. Receiver2 will receive no packets.

### 7.5.4 PIM-SM extension

**Overview**

PIM-SM extension is implemented as an alternative anycast routing protocol. Only the PIM-SM mechanisms that are necessary to the building of anycast routing is implemented. A BootStrap Router (BSR) and Rendezvous Point (RP) for the anycast group are selected. The Designated Router (DR), which discovers that there are directly attached receivers for the anycast group will create group information for the group, and then forward the anycast group joining messages towards the RP. An anycast tree will be built and used to distribute the traffic. The difference between PIM-SM extension with normal PIM-SM is that the PIM-SM extension will only forward the traffic to one of the group member rather than to all group members.

**Topology Used**

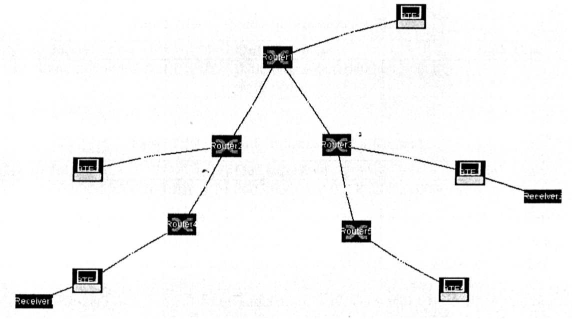


*Figure 7.5* *Topology used in testing PIM-SM extension*

**Parameter Settings and Configurations**

Router1 will be the BootStrap Router (BSR) and the Rendezvous Point (RP) for anycast group 3e00:0000:0000:0000:fdff:ffff:ffff:fffe. The simulation will be run for 80 seconds.

**Simulation Results**

**Table 7.16** Group Information for Router1

| Group Joined | Out Interface | Link Cost |
|---|---|---|
| 3e00:0000:0000:0000:fdff:ffff:ffff:fffe | 0002:0000:0000:0000:0000:0000:0000:0019 | 2 |
| 3e00:0000:0000:0000:fdff:ffff:ffff:fffe | 0001:0000:0000:0000:0000:0000:0000:0016 | 3 |

**Table 7.17** Group Information for Router2

| Group Joined | Out Interface | Link Cost |
|---|---|---|
| 3e00:0000:0000:0000:fdff:ffff:ffff:fffe | 0003:0000:0000:0000:0000:0000:0000:0014 | 2 |

**Table 7.18** Group Information for Router3

| Group Joined | Out Interface | Link Cost |
|---|---|---|
| 3e00:0000:0000:0000:fdff:ffff:ffff:fffe | 0006:0000:0000:0000:0000:0000:0000:001b | 1 |

**Table 7.19** Group Information for Router4

| Group Joined | Out Interface | Link Cost |
|---|---|---|
| 3e00:0000:0000:0000:fdff:ffff:ffff:fffe | 0007:0000:0000:0000:0000:0000:0000:0012 | 1 |

Router 5 does not have any group information. Group information of Router1, Router2, Router3 and Router4 from Table 7.16, 7.17, 7.18 and 7.19 justifies that the PIM-SM extension is successfully implemented. An anycast tree is successfully established.

### 7.5.5 Load-balancing Scheme

**Overview**

Three load-balancing schemes will be implemented – shortest path, round robin and fuzzy shortest path. The anycast tree built by the PIM-SM extension will apply the load-balancing scheme. In short, the three load-balancing schemes discussed here are:

1) Shortest path – The packets will be routed to the nearest interface, regardless of the condition of the link that leads to that interface.
2) Round Robin – The packets will be routed to each interface having anycast group receivers.
3) Fuzzy shortest path – The packets will be routed to the nearest interface, provided that the link that leads to that interface is not congested. Otherwise, the other route is selected.

**Topology Used**

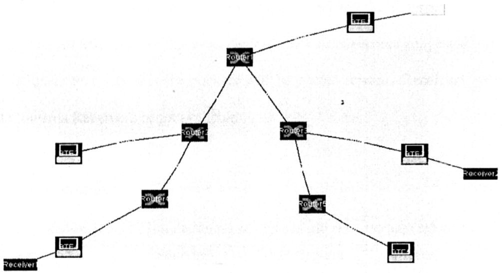


***Figure 7.6*** *Topology used in testing load-balancing scheme under normal condition*

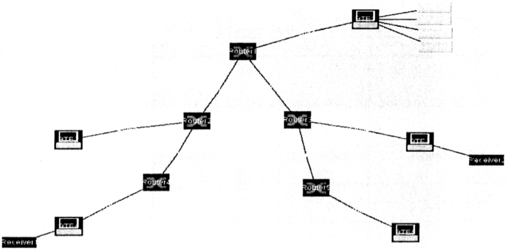

***Figure 7.7*** *Topology used in testing fuzzy shortest path in congestion*

**Parameter Settings and Configurations**

A simulation will be run for each load-balancing scheme using topology in Figure 7.6. The simulation will be run for 80 seconds. However, an extra simulation using topology in Figure 7.7 is required to test the fuzzy shortest path scheme. The output queue length of the Router1 to Router3 is changed to 3 and the link speed for Link2 is changed to 1.0 MBits/sec in this case, causing the link to become congested. The later simulation is to verify that the packets will be routed towards Receiver1 when the route towards Receiver2 is not available.

**Simulation Results**

**Table 7.20** Simulation results for testing shortest-path scheme

| **Parameter** | **Sender1** | **Receiver1** | **Receiver2** |
|---|---|---|---|
| Total Packets Sent | 4727 | 0 | 0 |
| Total Received Packets | 0 | 0 | 4727 |

**Table 7.21** Simulation results for testing shortest-path scheme

| Parameter | Sender1 | Receiver1 | Receiver2 |
|---|---|---|---|
| Total Packets Sent | 4727 | 0 | 0 |
| Total Received Packets | 0 | 2363 | 2364 |

**Table 7.22** Simulation results for testing shortest-path scheme under normal condition

| Parameter | Sender1 | Receiver1 | Receiver2 |
|---|---|---|---|
| Total Packets Sent | 4727 | 0 | 0 |
| Total Received Packets | 0 | 0 | 4727 |

**Table 7.23** Simulation results for testing shortest-path scheme in congestion

| Parameter | Sender1 | Sender2 | Sender3 | Sender4 | Receiver1 | Receiver2 |
|---|---|---|---|---|---|---|
| Total Packets Sent | 4727 | 4744 | 4716 | 4719 | 0 | 0 |
| Total Received Packets | 0 | 0 | 0 | 0 | 5637 | 12762 |

Simulation results in Table 7.20 shows that the anycast packets will be routed towards the nearest interface joining the anycast group, the Receiver2 for shortest path scheme. Result in Table 7.21 shows that the anycast packets will be distributed evenly between the two receivers for round robin scheme. Result in Table 7.22 shows that the packets will be routed towards the nearest interface for fuzzy shortest path scheme when the traffic is light. However the result shows in Table 7.23 shows that fuzzy shortest path will allow some of the packets to be routed towards the other receiver when Link2 is congested or unavailable. These results justifies that the three load-balancing schemes are successfully implemented.

### 7.5.6 Nearest PIM-SM extension

**Overview**

The nearest PIM-SM extension will use the anycast tree built from the PIM-SM extension. For normal PIM-SM extension, the router will forward the anycast packets towards the Rendezvous Point (RP) for the anycast group when it is not joining the anycast tree for that group. However, the nearest PIM-SM extension will forward the anycast packets towards the nearest interface joining the anycast group, instead of the RP. Two simulations will be run using the same topology in Figure 6.8, one using PIM-SM extension and another one using nearest PIM-SM extension. Shortest-path load-balancing scheme is used in both of the simulations.

**Topology Used**

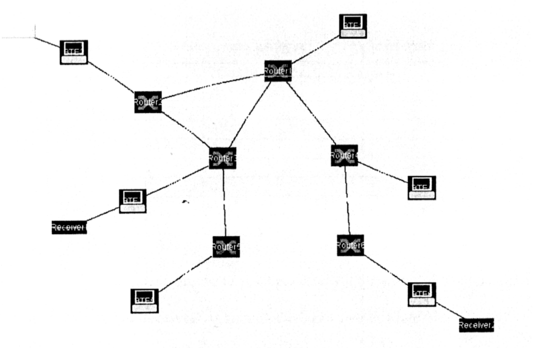

***Figure 7.8*** *Topology used in testing nearest PIM-SM extension*

**Parameter Settings and Configurations**

The link speed for the links between Routers is 7.0 MBit/sec and the link speed for the links between a router and a B-TE is 25.0 MBit/sec. Router1 is the BSR and RP for anycast group 3e00:0000:0000:0000:fdff:ffff:ffff:fffe in this simulation.

**Simulation Results**

A total of 4727 anycast packets were sent from Sender1.

**Table 7.24** Simulation results using PIM-SM extension

| Components | Number of Packets Received |
|---|---|
| Router1 | 4727 |
| Router2 | 4727 |
| Router3 | 4727 |
| Router4 | 0 |
| Router5 | 0 |
| Router6 | 0 |
| Receiver1 | 4727 |
| Receiver2 | 0 |

**Table 7.25** Simulation results using nearest PIM-SM extension

| Components | Number of Packets Received |
|---|---|
| Router1 | 0 |
| Router2 | 4727 |
| Router3 | 4727 |
| Router4 | 0 |
| Router5 | 0 |
| Router6 | 0 |
| Receiver1 | 4727 |
| Receiver2 | 0 |

In the case of PIM-SM extension, the anycast packets will be routed to Router1 (RP) first before distributed to Receiver1. Although there is a shorter route from Router2 to Reciever1 (Router2 -> Router3 -> BTE3 -> Receiver1), but the packets have to be sent to the RP first (Router2 -> Router1 -> Router3 -> BTE3 -> Receiver1) as it is an off-tree hit (refer to Section 6.2 for the explanation on off-tree hit). However, the nearest PIM-SM extension intended to minimize the delay, by allowing the packets to be sent through the shortest path from Router2 to Receiver3, without going through the Router1 (RP). By observing the results in Table 7.24 and Table 7.25, the differences between the nearest PIM-SM extension and PIM-SM extension can be recognized, as mention earlier in the section. The results show that the nearest PIM-SM extension is successfully implemented.