2010

Performance evaluation of multicast routing on IPv4 and IPv6 networks

Suneetha Hari

Recommended Citation

Hari, Suneetha, "Performance evaluation of multicast routing on IPv4 and IPv6 networks" (2010). Thesis. Rochester Institute of Technology. Accessed from

Performance Evaluation of Multicast Routing on IPv4 and IPv6 Networks

By

Suneetha Hari

Thesis submitted in partial fulfillment of the requirements
for the degree of
Master of Science in
Networking and Systems Administration

Rochester Institute of Technology

B. Thomas Golisano College
of
Computing and Information Sciences

18th November, 2010

Rochester Institute of Technology

B. Thomas Golisano College
of
Computing and Information Sciences

Master of Science in
Networking and Systems Administration

Thesis Approval Form

Student Name: Suneetha Hari

Thesis Title: Performance Evaluation of Multicast Routing on IPv4 and IPv6 Networks

Thesis Committee

Name	Signature	Date
Prof. Sumita Mishra Chair		
Prof. Sharon Mason Committee Member		
Prof. Charles Border Committee Member		

Thesis Reproduction Permission Form

Rochester Institute of Technology

B. Thomas Golisano College
of
Computing and Information Sciences

Master of Science in
Networking and Systems Administration

Performance Evaluation of Multicast Routing on IPv4 and IPv6 Networks

I, Suneetha Hari, hereby grant permission to the Wallace Library of the Rochester Institute of Technology to reproduce my thesis in whole or in part. Any reproduction must not be for commercial use or profit.

Date: ___________ Signature of Author: ___________________________

Abstract

Even though the transition from IPv4 to IPv6 has not been realized at the pace that it was anticipated, eventually with the depletion of IPv4 address space and the ever-growing demands of the Internet, the transition is inevitable. In the rapidly evolving world of technology, multimedia applications and voice/video conferencing are fast finding their ways into the Internet and corporate networks. Multicast routing protocols run over unicast routing protocols to provide efficient routing of such applications. This thesis was aimed at understanding how the transition from IPv4 to IPv6 would impact multicast routing. The multicast routing protocol Protocol Independent Multicast – Sparse Mode (PIM - SM) was used over both IPv4 and IPv6 networks and a mixed IPv4-IPv6 network. Parameters such as protocol overheads, throughput and jitter were evaluated in a lab environment using jperf.

Acknowledgements

I would like to take this opportunity to thank my committee members without whose constant support and encouragement I would not have been able to complete this research.

Prof. Mishra, Prof. Mason and Prof. Border were involved in my research from the start and provided guidance at every step till the completion.

I am grateful to my family – my parents, sister and my husband whose motivation and support enabled me to complete this thesis and degree.

Contents

List of Figures

List of Tables

1 Introduction

The Internet has grown tremendously over the last few years. What started as an experiment has grown to the worldwide network that we know today. Large numbers of users subscribe to online multimedia services such as video streaming. Messenger services such as Skype and Gtalk are replacing traditional phones for long distance calls across urban areas in many countries.

Information exchange can broadly be classified as unicast (one-to-one), broadcast (one-to-all) and multicast (one-to-many). A typical example of multicasting is Yahoo Messenger where multiple hosts subscribe to the service and the server communicates only with those hosts that have subscribed to it. One of the biggest advantages of multicasting is the conservation of bandwidth. The multicast server sends out only one packet and the router then generates multiple packets to reach each of the receivers. In this manner the network resources are used efficiently. Also, multicasting ensures timely reception of the data by the receivers [1]. In unicast routing, the server sends out a packet to each of the receivers. A more recent variation of multicast is anycast. It is a one-to-“one-of-many” distribution. There may be multiple recipients of an anycast message, but the sender sends the message only to the node that is logically or topologically the closest to it. The figure below is a comparison of unicast, broadcast, multicast and anycast.

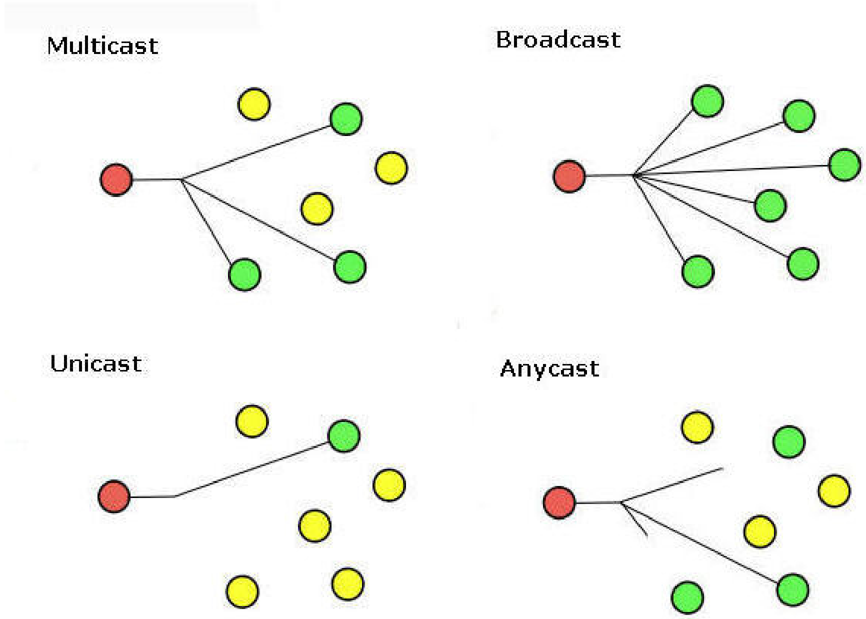


Figure 1: Unicast, multicast, broadcast and anycast depiction
Courtesy: www.trainsignaltraining.com

As part of this research, a survey was conducted to learn if enterprises deployed multicast routing in their networks and if so, for what kind of applications. The survey was also aimed at gathering which multicast routing protocols were used widely. The results of the survey indicated enterprises used multicast applications. Protocol Independent Multicast (PIM) was the multicast routing protocol preferred by most enterprise network administrators, since it is independent of the underlying unicast routing protocol in the network. Unlike Distance Vector Multicast Routing Protocol (DVMRP) that can be used only in networks that use a distance-vector unicast routing protocol, PIM can be used whether the unicast routing protocol is a distance-vector or link-state protocol. For this reason, PIM was chosen for this study.

This thesis was a quantitative one involving gathering results from laboratory experiments. The laboratory experimental setup consisted of four Cisco 2811 routers connected back to back using Cisco serial WAN Interface Cards (WICs). The first and the last routers in the chain were connected to hubs. Each hub had two PCs connected to it. One of the PCs was the source for the multicast traffic and the other three were receivers. The underlying unicast routing protocol chosen was Open Shortest Path First (OSPF), a popularly used routing protocol in enterprise networks. This network was maintained across all four scenarios, which were

1. IPv4 network
2. IPv6 network
3. IPv4-IPv6 network using dual-stack
4. IPv4-IPv6 network using Generic Routing Encapsulation (GRE) tunneling

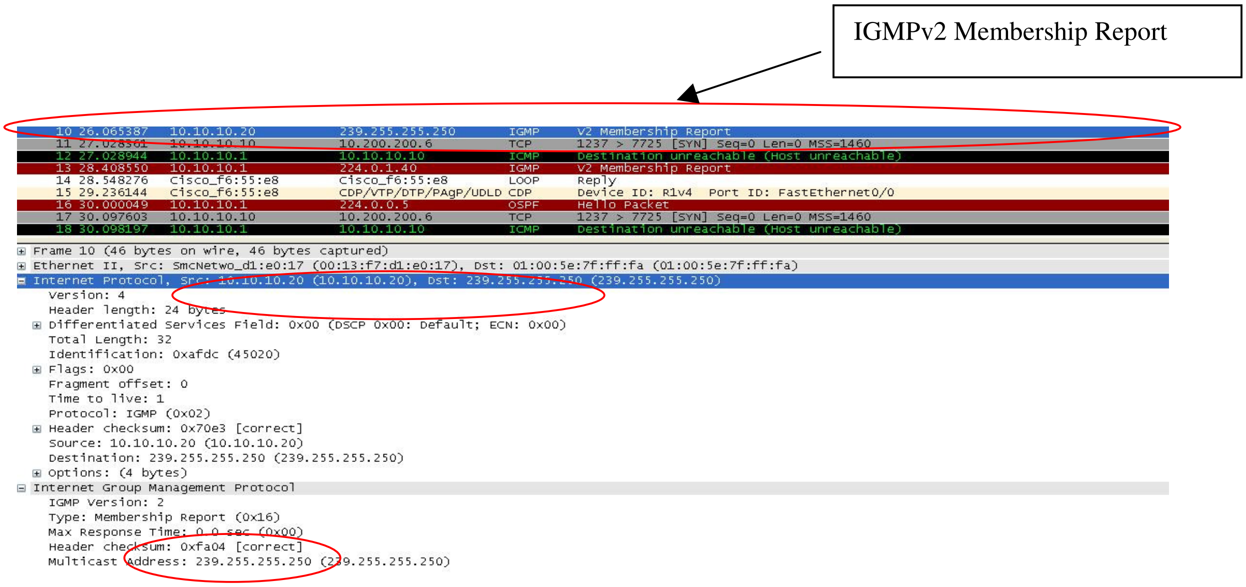
Jperf [2, 3] was used for generating multicast traffic and obtaining graphical results.

2 IPv4 multicast and IGMP

In IPv4, host membership to multicast group(s) is governed by the Internet Group Management Protocol (IGMP) [4]. The switches that the hosts connect to should have IGMP enabled. The multicast querying router is a chosen router on the network that periodically sends out group membership queries to all hosts connected to its local network. Any host that is interested in joining a multicast group sends a join request or membership report to that group. Any traffic destined to that multicast group address is then sent to the host. IP multicast is very dynamic and any host can join or leave a group at any time. A querying router need not be aware of all the hosts that belong to a particular multicast group. The router only needs to know that there is at least one member in each of the groups attached to its local network, so that it ensures that the multicast traffic destined for that group reaches the group.

IGMPv3 [5] is the latest version of IGMP. The significant difference between IGMPv1 and IGMPv2 is that in IGMPv2, a host that wishes to leave a multicast group has to explicitly send a Leave message to the querying router. This can significantly reduce bandwidth usage in bandwidth intensive applications. The major improvement of IGMPv3 over IGMPv2 is that in IGMPv3, source-specific multicast is supported. So a host can specify the host or hosts from which it wants to receive multicast traffic from.

A sample of a receiver sending a report to the multicast querying router can be seen from the Wireshark capture below:



IGMPv2 Membership Report

```
 10 26.065387   10.10.10.20    239.255.255.250      IGMP   V2 Membership Report
 11 27.028361   10.10.10.10    10.200.200.6         TCP    1237 > 7725 [SYN] Seq=0 Len=0 MSS=1460
 12 27.028944   10.10.10.1     10.10.10.10          ICMP   Destination unreachable (Host unreachable)
 13 28.408550   10.10.10.1     224.0.1.40           IGMP   V2 Membership Report
 14 28.548276   Cisco_f6:55:e8 Cisco_f6:55:e8       LOOP   Reply
 15 29.236144   Cisco_f6:55:e8 CDP/VTP/DTP/PAgP/UDLD CDP   Device ID: R1v4  Port ID: FastEthernet0/0
 16 30.000049   10.10.10.1     224.0.0.5            OSPF   Hello Packet
 17 30.097603   10.10.10.10    10.200.200.6         TCP    1237 > 7725 [SYN] Seq=0 Len=0 MSS=1460
 18 30.098197   10.10.10.1     10.10.10.10          ICMP   Destination unreachable (Host unreachable)

⊞ Frame 10 (46 bytes on wire, 46 bytes captured)
⊞ Ethernet II, Src: SmcNetwo_d1:e0:17 (00:13:f7:d1:e0:17), Dst: 01:00:5e:7f:ff:fa (01:00:5e:7f:ff:fa)
⊟ Internet Protocol, Src: 10.10.10.20 (10.10.10.20), Dst: 239.255.255.250 (239.255.255.250)
    Version: 4
    Header length: 24 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 32
    Identification: 0xafdc (45020)
  ⊞ Flags: 0x00
    Fragment offset: 0
    Time to live: 1
    Protocol: IGMP (0x02)
  ⊞ Header checksum: 0x70e3 [correct]
    Source: 10.10.10.20 (10.10.10.20)
    Destination: 239.255.255.250 (239.255.255.250)
  ⊞ Options: (4 bytes)
⊟ Internet Group Management Protocol
    IGMP Version: 2
    Type: Membership Report (0x16)
    Max Response Time: 0.0 sec (0x00)
    Header checksum: 0xfa04 [correct]
    Multicast Address: 239.255.255.250 (239.255.255.250)
```

Figure 2: Wireshark capture showing IGMPv2 Membership Report

From the circled portions, it can be seen that the host 10.10.10.20 sends a membership report to the multicast group 239.255.255.250.

3 IPv6 multicast and MLD

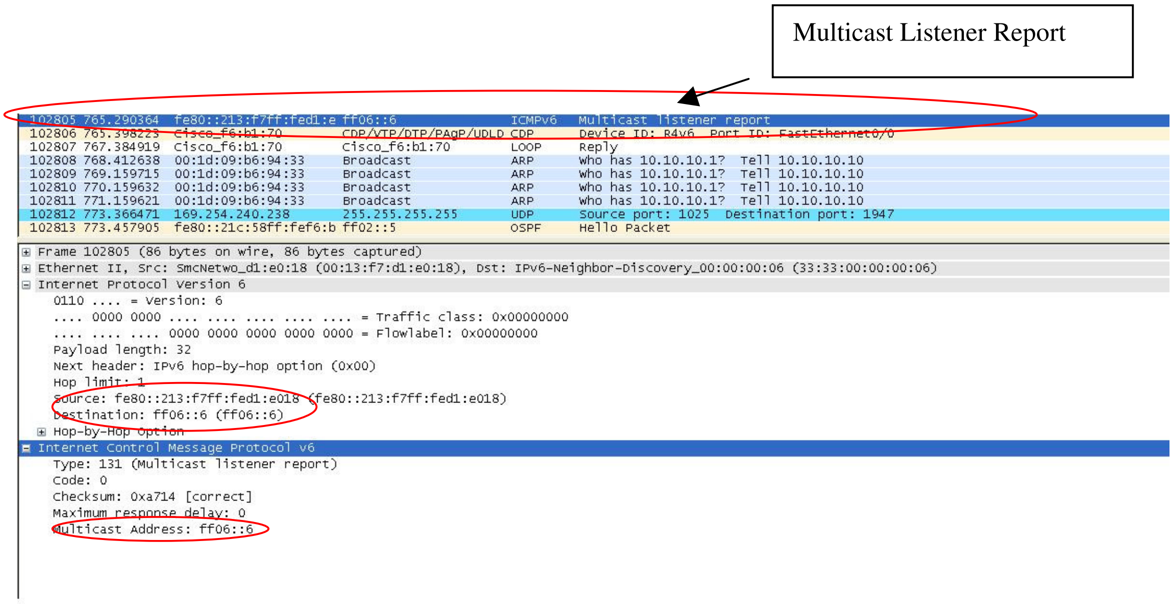
Multicast Listener Discovery [6] is the IGMP equivalent used in IPv6. MLD however uses Internet Control Message Protocol for IPv6 (ICMPv6). There are three types of MLD messages:

Multicast Listener Query – This is similar to the IGMP query sent by the router periodically for group memberships.

Multicast Listener Report – This is sent by the multicast host group in response to a router query or for the host to indicate that it wants to join a group.

Multicast Listener Done – This message is sent by the multicast host when it leaves a multicast group. The Done message is sent by the last group member so that the router is aware that there are no more hosts for the multicast traffic on that segment. This is similar to the IGMPv2 Leave Group message used in IPv4.

The Wireshark capture below shows an ICMPv6 Multicast Listener Report sent from a multicast receiver to a multicast group.



Multicast Listener Report

```
102805 765.290364  fe80::213:f7ff:fed1:e ff06::6              ICMPv6   Multicast listener report
102806 765.398223  Cisco_f6:b1:70        CDP/VTP/DTP/PAgP/UDLD CDP      Device ID: R4v6   Port ID: FastEthernet0/0
102807 767.384919  Cisco_f6:b1:70        Cisco_f6:b1:70         LOOP     Reply
102808 768.412638  00:1d:09:b6:94:33     Broadcast              ARP      Who has 10.10.10.1?  Tell 10.10.10.10
102809 769.159715  00:1d:09:b6:94:33     Broadcast              ARP      Who has 10.10.10.1?  Tell 10.10.10.10
102810 770.159632  00:1d:09:b6:94:33     Broadcast              ARP      Who has 10.10.10.1?  Tell 10.10.10.10
102811 771.159621  00:1d:09:b6:94:33     Broadcast              ARP      Who has 10.10.10.1?  Tell 10.10.10.10
102812 773.366471  169.254.240.238       255.255.255.255        UDP      Source port: 1025  Destination port: 1947
102813 773.457905  fe80::21c:58ff:fef6:b ff02::5                OSPF     Hello Packet

⊞ Frame 102805 (86 bytes on wire, 86 bytes captured)
⊞ Ethernet II, Src: SmcNetwo_d1:e0:18 (00:13:f7:d1:e0:18), Dst: IPv6-Neighbor-Discovery_00:00:00:06 (33:33:00:00:00:06)
⊟ Internet Protocol Version 6
    0110 .... = Version: 6
    .... 0000 0000 .... .... .... .... .... = Traffic class: 0x00000000
    .... .... .... 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
    Payload length: 32
    Next header: IPv6 hop-by-hop option (0x00)
    Hop limit: 1
    Source: fe80::213:f7ff:fed1:e018 (fe80::213:f7ff:fed1:e018)
    Destination: ff06::6 (ff06::6)
  ⊞ Hop-by-Hop Option
⊟ Internet Control Message Protocol v6
    Type: 131 (Multicast listener report)
    Code: 0
    Checksum: 0xa714 [correct]
    Maximum response delay: 0
    Multicast Address: ff06::6
```

Figure 3: Wireshark capture showing Multicast Listener Report

The source address seen in the capture is the Link Local address of the host's Ethernet interface. The multicast group to which it sends the Multicast Listener Report is ff06::6.

4 Statement of Purpose

Ever since the convergence of data and voice networking, applications such as video conferencing and Voice over IP (VoIP) have found their way into enterprise networks. Since such applications are bandwidth intensive, a multicast solution can be adopted when there are multiple recipients for the same data.

The IPv4 address space is expected to eventually deplete, since the Internet is growing every day. The migration to the 128-bit IPv6 address has already begun and would replace IPv4. While this transition is in its nascent stages, this thesis provides an opportunity to acquire working knowledge of IPv6, which is the future of the Internet. In essence, this thesis is aimed at evaluating multicast performance in the IPv4 era, the future IPv6 era and the transitional phase in which IPv6 forms the core or backbone and the edge devices are IPv4 compliant.

It is hypothesized in this thesis that the multicast routing overhead in an IPv6 network would be higher than in an IPv4 network due to the significantly larger address format of IPv6. It then becomes of experimental interest to verify the hypothesis.

5 Related Works

IP multicasting has been around for over two decades now. This section highlights some of the related work that has been conducted in the area of this research.

Research in the multicast domain started in the early 80s. Steve Deering invented multicasting and in 1989 RFC 1112 [7] was formulated. The initial works were on IGMP and DVMRP. DVMRP being a distance vector routing protocol had the same shortcomings as those faced by unicast distance vector protocols such as Routing Information Protocol (RIP), where the hop-count limited the protocols to be used only in smaller networks.

Link state routing protocols soon gained popularity as they could be deployed in larger networks. Open Shortest Path First (OSPF) had its own extension to support multicast called Multicast Open Shortest Path First (MOSPF – RFC 1584) [8]. This protocol failed and is not supported by lead vendors like Cisco.

In 1995, Steve Deering published his paper [1] on extending the multicasting used in Local Area Networks (LANs) to larger internetworks and the challenges involved due to the increased distances between the source and the receivers. The main objective of the multicast routing protocols discussed in the paper is to determine a low latency multicast routing protocol. This is important in an internetwork environment due to the possibility of lower link speeds when connecting several smaller networks.

In [9], Deborah Estrin et al. documented an Internet Draft for the IETF suggesting the interoperation of two different multicast routing protocols. It focused on the interconnectivity between a PIM-SM multicast domain and a DVMRP backbone by placing PIM Multicast Border Routers at the boundaries between the PIM- SM and DVMRP domains.

In their paper Active Reliable Multicast [10], Li-wei H. Lehman, Stephen J. Garland and David L. Tennenhouse from the MIT Lab for Computer Science suggested a new scheme for loss recovery in large multicast networks. They also suggested performance improvement by reducing protocol overheads by the routers in both directions – upstream and downstream. Since latency is one of the challenges in large networks, the suggested mechanism employs a “local” retransmission by the router from its cache. Also, most multicast networks are very dynamic. Receivers subscribe and unsubscribe to groups often and a partial multicasting technique is suggested so that retransmissions are targeted only to those hosts that lost the packet.

XCAST6 is a novel multicast routing mechanism suggested by Yuji IMAI, Hiro Kishimoto, Myung-Ki SHIN and Young-Han KIM in their paper XCAST6: eXplicit Multicast on IPv6 [11].

The method suggests using IPv6 addresses' extended headers to include the multicast destinations instead of a multicast group address. Though the protocol claims to eliminate some of the complexities of the multicast routing protocols in use, it does have a big limitation in the number of members a group can have, as it is limited to the IPv6 header length.

Network Address Translation – Protocol Translation (NAT-PT) is one of the suggested ways for protocol translation between IPv4 and IPv6 networks [12]. As mentioned in the RFC one of the limitations of NAT-PT is its inability to translate between IPv4 and IPv6 multicast traffic. Referenced in the RFC are two solutions to overcome this issue.

In his paper [13]**,** Kazuaki Tsuchiya suggests an IPv6/IPv4 multicast proxy that would do both proxying and translation. IPv6 clients would send a request to the translator to join the multicast group and the translator sends a proxy IGMP request to the IPv4 router. On receiving the multicast traffic, the translator translates IPv4 to IPv6 and sends it to the IPv6 client/receiver. Similarly, IPv4 hosts can also join and receive traffic from an IPv6 multicast source.

Also referred in RFC 4966, An IPv4 - IPv6 Multicast Gateway [14], Venaas suggests embedding of IPv4 addresses in IPv6 addresses. The IPv4 multicast group address is mapped to an IPv6 multicast group address, by attaching prefixes to the IPv4 group address. Every IPv4 multicast group is mapped in this manner to an IPv6 multicast group.

Performance- Comparison Testing of IPv4 and IPv6 Throughput and Latency on Key Cisco Router Platforms – A Summary of findings, [15] *is* a whitepaper published by Cisco with test results on IPv4 only, IPv6 only and dual-stack networks in response to a query raised by federal agencies to understand if there would be any impact on the network performance on migrating to IPv6 networks.

Cisco conducted the tests using Spirent TestCenter across multiple router platforms on 100% IPv4, 100% IPv6 and a mix of IPv4 and IPv6 traffic on dual-stack networks and concluded from the results that only when the frame size was very small (< 256 bytes) was the throughput on IPv6 networks less than IPv4 networks. With larger frames, the performance was identical across all the network types studied. The paper also found that the variation in latency was negligible and in some cases, there was an improvement with the introduction of IPv6 traffic. In my paper, a performance comparison is done for multicast traffic.

Research on the performance of both IPv4 and IPv6 on different Windows operating systems was performed in 2008 [16] and the results indicate that in most cases, IPv4 outperforms IPv6.This is in line with the hypothesis of this paper, where it is presumed that IPv6 would have more overhead and therefore result in a performance degradation in a network. Since there is little work done on the performance comparison of IPv4 and IPv6 in multicast routing, this research is aimed at presenting experimental analysis to study this area.

Protocol Independent Multicast (PIM) [17, 18] is a multicast routing protocol that is most widely used. As the name suggests, this protocol is independent of the underlying unicast routing protocol. Irrespective of whether the unicast routing protocol is a distance vector or link-state protocol, PIM can be used.

PIM operates in two main modes- Sparse Mode (SM) [18] and Dense Mode (DM) [17]. In dense mode the multicast group receivers are highly concentrated in an area, whereas in sparse mode, large numbers of users are distributed over a wide area such as the Internet. PIM follows a tree-based structure. PIM- DM uses a source tree structure where the root of the tree is the source router and the rest of the tree structure is built from the root. In the sparse mode [19], the distribution tree has its root in the Rendezvous Point (RP) and all control messages from the source are sent to this RP. Every local network has a Designated Router (DR), which manages the PIM control messages. Multicast receivers that want to become part of a group send the request message to the RP. It is for this reason that SM is more efficient in the case of a wider area with scattered receivers.

6 Limitations

1. These experiments were conducted in a lab environment. The only traffic in the network was the simulated traffic generated by jperf.
2. Multicast traffic was generated using a traffic generator and not from real-world applications.
3. Switches could not be used in this study for consistency purposes. Hubs were used instead because the lab environment did not have switches that had support for Multicast Listener Discovery (MLD) which is the Internet Group Management Protocol (IGMP) equivalent in IPv6.
4. The impact of scalability on the performance was originally proposed in this study using simulation. OPNET Modeler was the chosen simulation tool. But the OPNET server in

the lab did not have licenses for IPv6 models and as a reason scalability could not be studied.

7 Experimental Setup

The hardware used for the lab experiments is as in the table below:

Device	**Quantity**
Cisco 2811 routers	4 (IOS 12.4 – Advanced IP Services)
NetGear 10/100 Mbps Hubs	2
Windows XP machines	4

Table 1: Hardware setup

The lab setup consisted of connecting four Cisco 2811 routers back-to-back using serial connections. NetGear hubs were connected to the fast Ethernet interface on Routers 1 and 4. Router 1 had 2 PCs connected to it via the hub. One of the PCs was the source of the multicast traffic. Two PCs were connected to Router 4 via another hub. The multicast group had three receivers.

The routers were configured to run OSPF as the unicast routing protocol. PIM-SM was configured on all the interfaces on all four routers. Jperf was used as the multicast traffic generator. The throughput and jitter were obtained using jperf, the Java based graphical front-end of iperf.

For each scenario, jperf was run for ten 10-minute periods and two 1-hour periods. For each test, jperf was transmitting 122 Kbytes per second at 1000 kbps.

The results were collected from two receivers – one on the same subnet as the source and the other on a different subnet. This was done in order to understand the impact of routing on the multicast traffic.

In jperf terminology, the client is the source of the multicast traffic and the servers are receivers of the multicast traffic. Also, it should be noted that the receivers have to join the multicast group before the source starts sending traffic, so that each of the receivers receives all the multicast traffic that was sent by the source and there is no packet loss.
Wireshark was used to capture packets at the network interface cards of the two receivers to gather additional information such as learning IGMP/MLD workings and the packets generated by PIM-SM.

8 Experimental Scenarios

This research was conducted in four different scenarios:

1. The present IPv4 only networks, which is the case in most enterprise networks.
2. The anticipated future IPv6 only networks.
3. The interim transitional phase where IPv4 and IPv6 co-exist. This dual network was set up using two different configurations:
    a. Dual-stack
    b. GRE tunneling

The rest of this section will explain in detail the four scenarios.

8.1 IPv4 only network

The network diagram and the IP addressing scheme for the IPv4 only network were as depicted in the figure below:

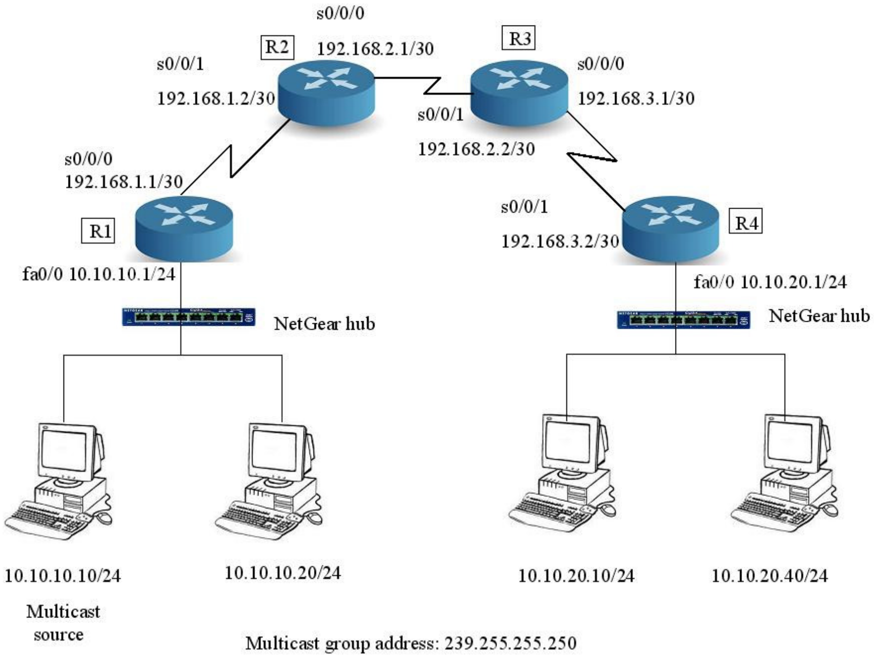


Figure 4: IPv4 only network diagram and addressing scheme

The source of the multicast traffic was 10.10.10.10 and the other three PCs were the receivers. The time-to-live (TTL) on the source was set to 10 (to account for the four routers that the traffic has to travel through to reach some of the multicast receivers). A sample router configuration can be seen at Appendix 12.1.

8.2 IPv6 only network

The IPv6 network connectivity and addressing scheme are shown in the figure below:

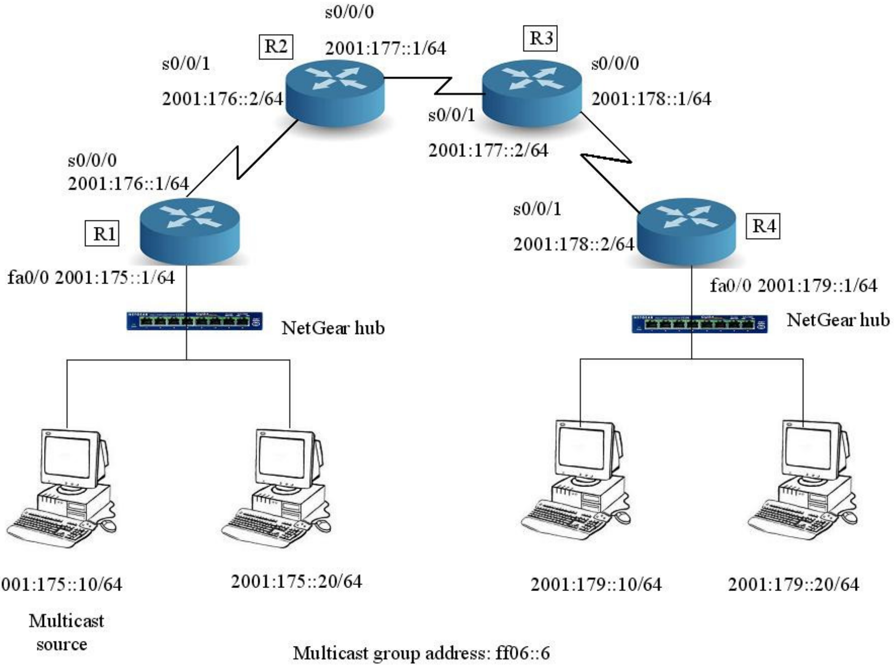


Figure 5: IPv6 only network diagram and addressing scheme

The source of the multicast traffic was 2001:175::10 and the other three PCs were the receivers. The time-to-live (TTL) on the source was set to 10 (to account for the four routers that the traffic has to travel through to reach some of the multicast receivers). A sample router configuration is presented at Appendix 12.2.

8.3 IPv4-IPv6 network – Dual-stack

In this scenario, the hosts and routers were configured with both IPv4 and IPv6 addresses. The multicast source generated two separate multicast streams – one for IPv4 and one for IPv6. Router R4 had an IPv4 receiver and an IPv6 receiver.

The network diagram and IPv4/v6 addressing scheme were as below:

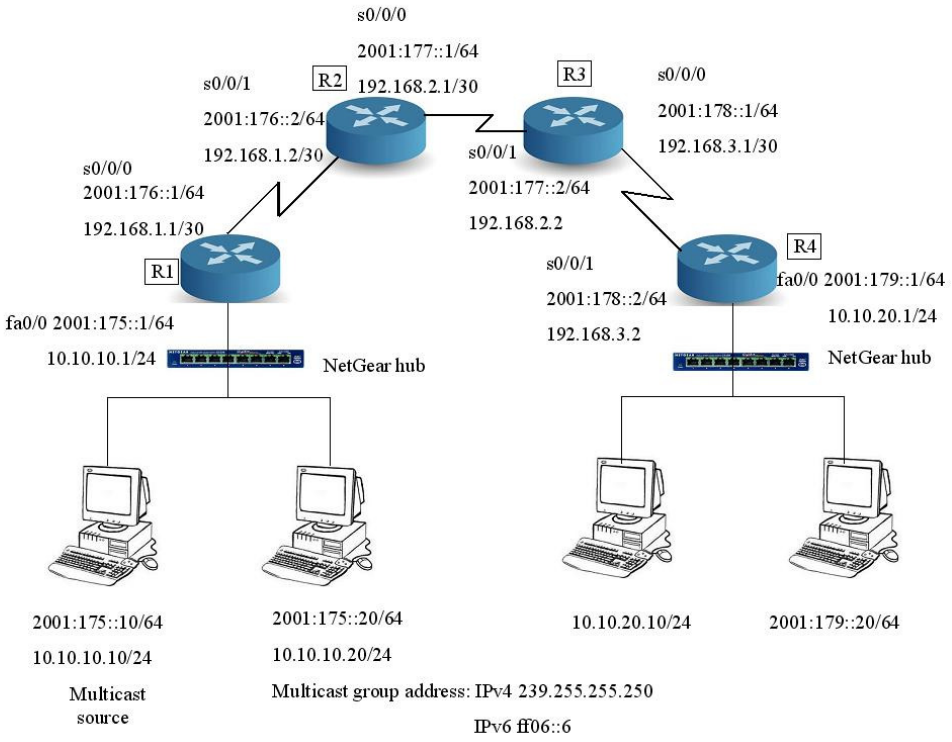


Figure 6: IPv4-IPv6 dual-stack network diagram and addressing scheme

A sample router configuration is included in Appendix 12.3.

8.4 IPv4-IPv6 network – GRE Tunneling

In this scenario, two IPv4 only networks were connected via an IPv6 only backbone network. For instance, during the migration period from IPv4 to IPv6, the backbone (ISPs) may migrate to

IPv6 before the edges. In such a case, the IPv4 end networks would communicate with each other via the IPv6 network. A GRE IPv6 tunnel was set up between the IPv4 only networks to encapsulate/decapsulate the IPv4 traffic.

GRE is a Cisco developed protocol that is used to connect networks running different protocols such as connecting an IP and IPX network and in this case connecting two IPv4 networks across an IPv6 backbone. In this scenario, a logical IPv6 GRE tunnel was configured. IPv4 packets entering the tunnel are encapsulated with an IPv6 header and decapsulated when the packet reaches the other end of the tunnel.

For the OSPF configuration, all the serial interfaces were in Area 0. The fast Ethernet interfaces of routers R1 and R4 and the GRE tunnel were in Area 1. A sample configuration of the tunneling edge router can be seen in Appendix 12.4.a and the core router configuration can be found at Appendix 12.4.b.

The network connectivity and IPv4/IPv6 addressing were as in the figure below:

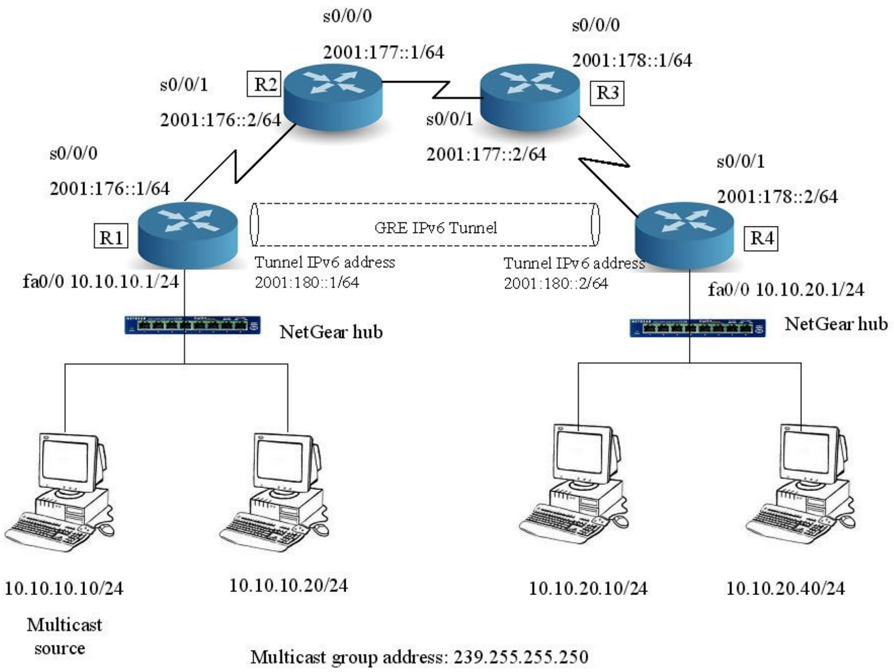


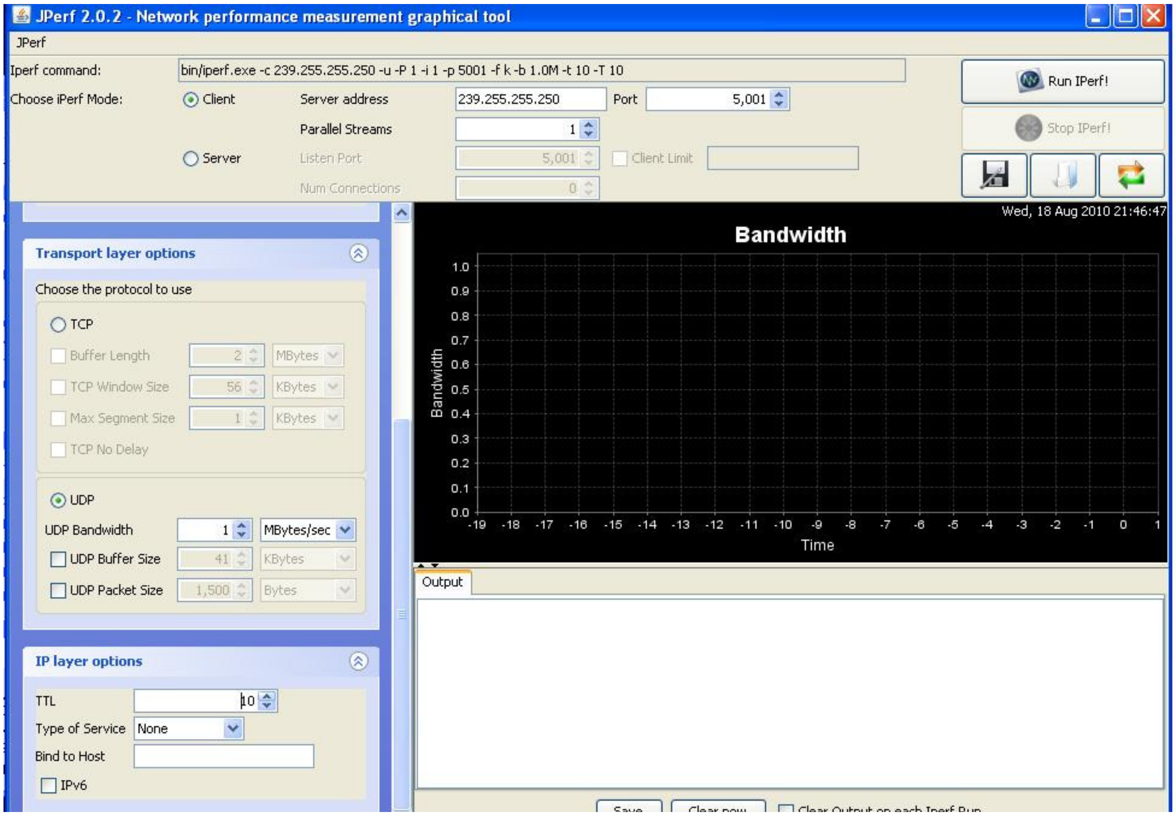
Figure 7: IPv4-IPv6 network diagram and addressing scheme – GRE tunneling

9 Experimental results and analyses

9.1 IPv4 only network

9.1.1 Throughput and Jitter

From the output obtained from jperf, it was seen that in all the ten 10-minute test periods there was no packet loss and the throughput was 100%. The jitter showed some variation. The jitter varied from 0 ms in some tests to a maximum of 7.792 ms. Sample screenshots and jperf output are shown below:


JPerf 2.0.2 - Network performance measurement graphical tool

Iperf command: bin/iperf.exe -c 239.255.255.250 -u -P 1 -i 1 -p 5001 -f k -b 1.0M -t 10 -T 10

Choose iPerf Mode: ● Client ○ Server

Server address 239.255.255.250 Port 5,001

Parallel Streams 1

Listen Port 5,001 ☐ Client Limit

Num Connections 0

Transport layer options

Choose the protocol to use

○ TCP

☐ Buffer Length 2 MBytes

☐ TCP Window Size 56 KBytes

☐ Max Segment Size 1 KBytes

☐ TCP No Delay

● UDP

UDP Bandwidth 1 MBytes/sec

☐ UDP Buffer Size 41 KBytes

☐ UDP Packet Size 1,500 Bytes

IP layer options

TTL 10

Type of Service None

Bind to Host

☐ IPv6

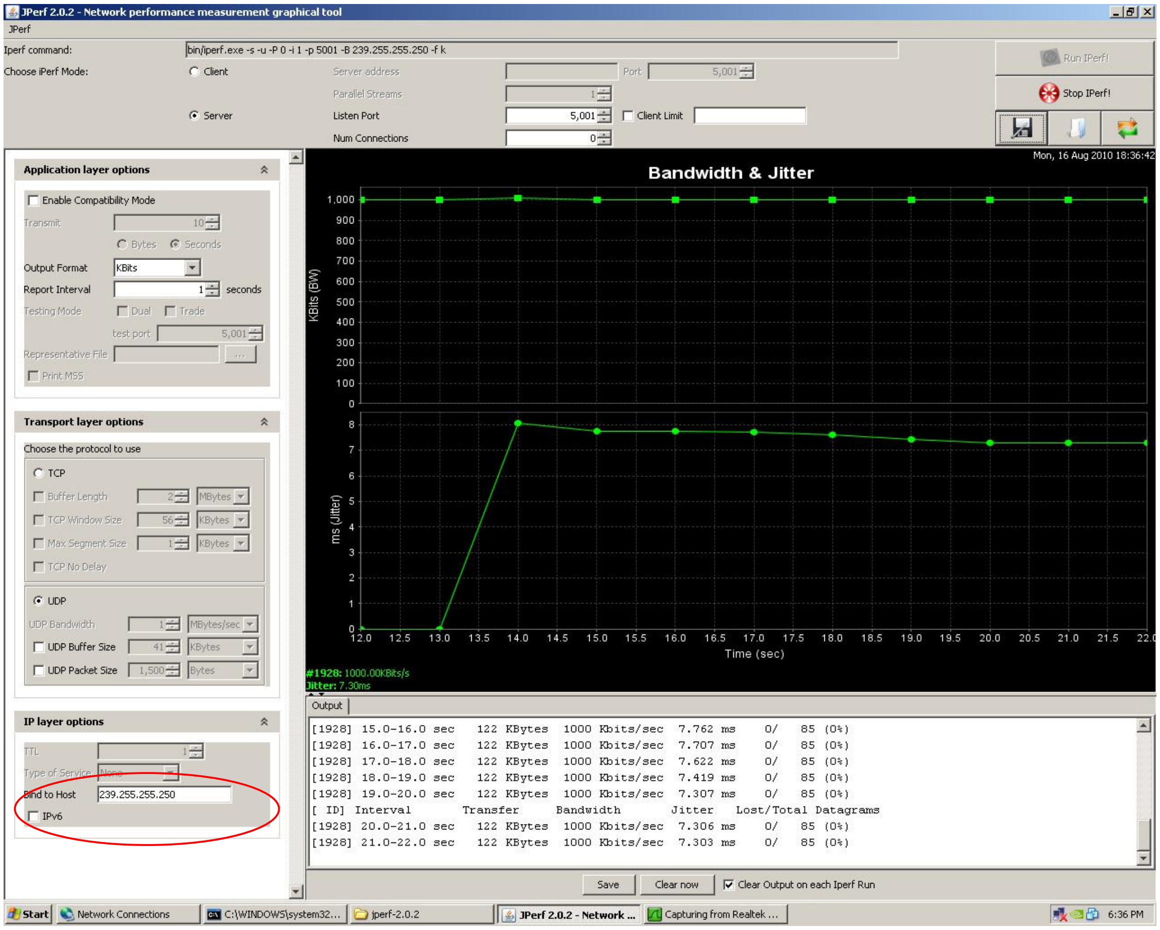
Figure 8: IPv4 Multicast Source

On starting jperf as the multicast receiver on the PC, the following information is displayed:

```
bin/Jperf.exe -s -u -P 0 -i 1 -p 5001 -B 239.255.255.250 -f k
------------------------------------------------------------
Server listening on UDP port 5001
Binding to local address 10.10.20.10
Receiving 1470 byte datagrams
UDP buffer size: 8.00 KByte (default)
```

Here we can see that the multicast group address is 239.255.255.250 to which the local host (10.10.20.10) binds.

The graphical output from jperf was captured at different points during the 10-minute period. It provides a real-time graph of the bandwidth and jitter. A sample of the screenshot is provided below:



Iperf command: `bin/iperf.exe -s -u -P 0 -i 1 -p 5001 -B 239.255.255.250 -f k`

Choose iPerf Mode: ○ Client ● Server

Listen Port: 5,001 ☐ Client Limit

Num Connections: 0

☐ Enable Compatibility Mode

Output Format: KBits

Report Interval: 1 seconds

● UDP

Bind to Host: 239.255.255.250

☐ IPv6

#1928: 1000.00KBits/s
Jitter: 7.30ms

```
[1928] 15.0-16.0 sec   122 KBytes  1000 Kbits/sec  7.762 ms    0/   85 (0%)
[1928] 16.0-17.0 sec   122 KBytes  1000 Kbits/sec  7.707 ms    0/   85 (0%)
[1928] 17.0-18.0 sec   122 KBytes  1000 Kbits/sec  7.622 ms    0/   85 (0%)
[1928] 18.0-19.0 sec   122 KBytes  1000 Kbits/sec  7.419 ms    0/   85 (0%)
[1928] 19.0-20.0 sec   122 KBytes  1000 Kbits/sec  7.307 ms    0/   85 (0%)
[ ID] Interval       Transfer     Bandwidth       Jitter   Lost/Total Datagrams
[1928] 20.0-21.0 sec   122 KBytes  1000 Kbits/sec  7.306 ms    0/   85 (0%)
[1928] 21.0-22.0 sec   122 KBytes  1000 Kbits/sec  7.303 ms    0/   85 (0%)
```

☑ Clear Output on each Iperf Run

Figure 9: Sample jperf screenshot from IPv4 multicast receiver

The last 10 seconds of the jperf output captured from a multicast receiver:

```
[ ID] Interval        Transfer    Bandwidth       Jitter    Lost/Total Datagrams
[1928] 591.0-592.0 sec   122 KBytes  1000 Kbits/sec  7.293 ms    0/   85 (0%)
[1928] 592.0-593.0 sec   122 KBytes  1000 Kbits/sec  7.282 ms    0/   85 (0%)
[1928] 593.0-594.0 sec   122 KBytes  1000 Kbits/sec  7.251 ms    0/   85 (0%)
[1928] 594.0-595.0 sec   122 KBytes  1000 Kbits/sec  7.212 ms    0/   85 (0%)
[1928] 595.0-596.0 sec   122 KBytes  1000 Kbits/sec  7.146 ms    0/   85 (0%)
```

```
[1928] 596.0-597.0 sec   122 KBytes  1000 Kbits/sec  6.955 ms    0/   85 (0%)
[1928] 597.0-598.0 sec   123 KBytes  1011 Kbits/sec  8.315 ms    0/   86 (0%)
[1928] 598.0-599.0 sec   122 KBytes  1000 Kbits/sec  8.314 ms    0/   85 (0%)
[1928] 599.0-600.0 sec   122 KBytes  1000 Kbits/sec  8.311 ms    0/   85 (0%)
[ ID] Interval       Transfer     Bandwidth       Jitter   Lost/Total Datagrams
[1928]  0.0-600.0 sec  73243 KBytes  1000 Kbits/sec  7.792 ms    0/51021 (0%)
```

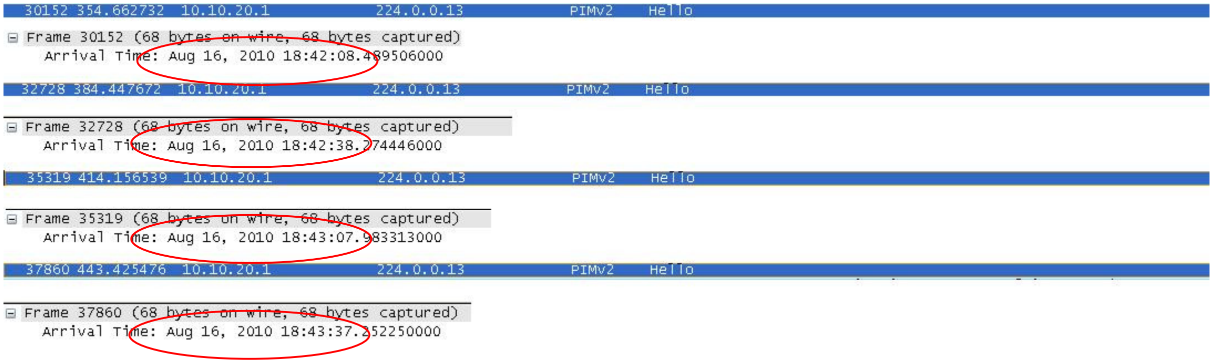
Table 2: 10-second jperf output from IPv4 multicast receiver

It can be observed from the output above, that over the 10-minute period, 73.244 MB of data was transferred at 1 Mbps. The jitter was 7.792 ms. The packet loss is 0% which implies a 100% throughput.

Two 1-hour test samples were also obtained from the multicast receiver. This was to simulate a real multicast application such as a 1-hour webinar. The jitter was 0 ms and 7.817 ms and the throughput was 100% in both the test cases.

9.1.2 Protocol Overheads

PIM-SM was used as the multicast routing protocol. The protocol did not produce much of an overhead (deduced from the Wireshark captures). The PIMv2 Hello packets were sent out at 30-second intervals, as seen from time-stamps in the capture below. Apart from these Hello packets, the protocol was not very chatty in the IPv4 network.



```
30152 354.662732  10.10.20.1             224.0.0.13             PIMv2    Hello
⊟ Frame 30152 (68 bytes on wire, 68 bytes captured)
    Arrival Time: Aug 16, 2010 18:42:08.489506000
32728 384.447672  10.10.20.1             224.0.0.13             PIMv2    Hello
⊟ Frame 32728 (68 bytes on wire, 68 bytes captured)
    Arrival Time: Aug 16, 2010 18:42:38.274446000
35319 414.156539  10.10.20.1             224.0.0.13             PIMv2    Hello
⊟ Frame 35319 (68 bytes on wire, 68 bytes captured)
    Arrival Time: Aug 16, 2010 18:43:07.983313000
37860 443.425476  10.10.20.1             224.0.0.13             PIMv2    Hello
⊟ Frame 37860 (68 bytes on wire, 68 bytes captured)
    Arrival Time: Aug 16, 2010 18:43:37.252250000
```

Figure 10: PIM Hello packets for IPv4 multicast

9.2 IPv6 only network

9.2.1 Throughput and Jitter

On starting jperf on the receiver, the following is displayed:

```
bin/Jperf.exe -s -u -P 0 -i 1 -p 5001 -B ff06::6 -V -f k
------------------------------------------------------------
Server listening on UDP port 5001
Binding to local address ::
Receiving 1470 byte datagrams
UDP buffer size: 8.00 KByte (default)
```

The multicast group address is ff06::6.

As in the case of the IPv4 only network, results were obtained from a multicast receiver for ten 10-minute tests and two 1-hour tests.

It can be inferred from the results that IPv6 multicast does not introduce any significantly higher jitter or packet loss than in the case of an IPv4 only network. During the ten 10-minute tests, the jitter ranged from 0 ms to 9.487 ms. The throughput was 100% in all the ten tests. From these tests it can be concluded that the hypothesis of this research does not hold good.

For the two 1-hour tests, the jitter was 0 ms in one test and 7.299 in the second test with 100% throughput in both the tests.

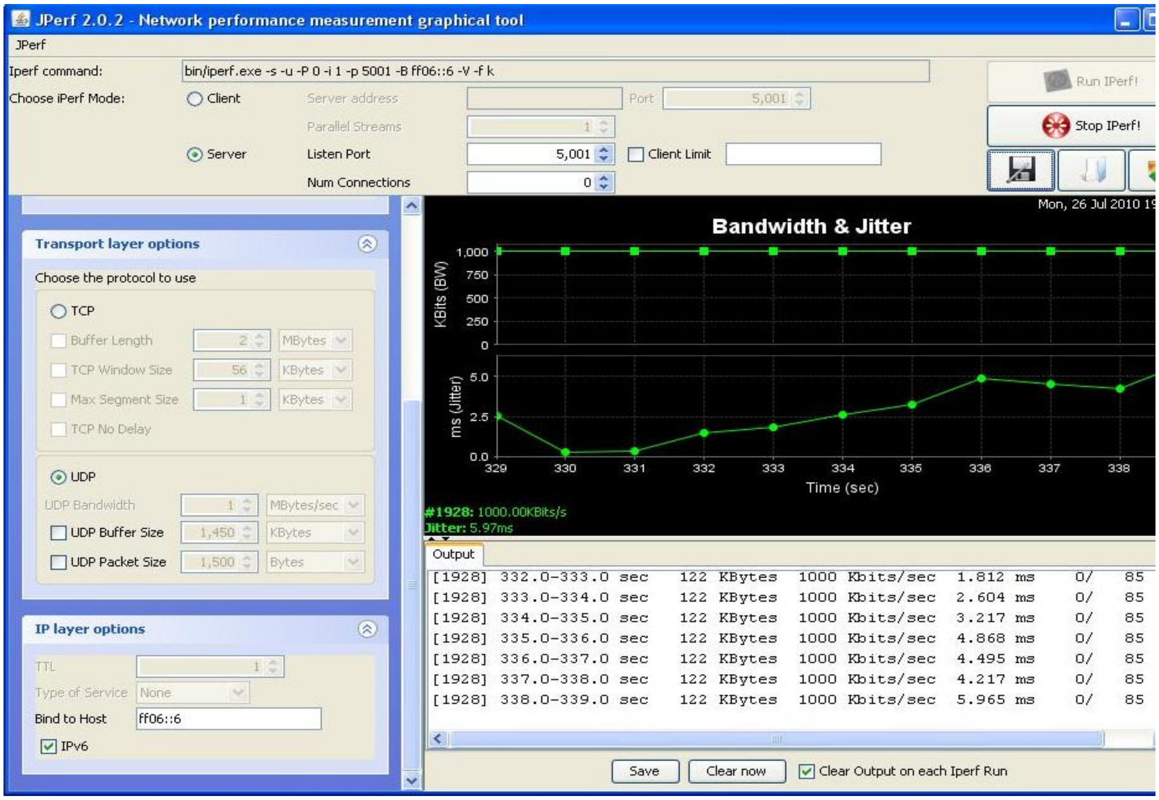
The last 10 seconds of the jperf output captured from a multicast receiver is pasted below:

```
[ ID]          Interval      Transfer     Bandwidth       Jitter   Lost/Total Datagrams
[1928] 591.0-592.0 sec   122 KBytes  1000 Kbits/sec  8.302 ms    0/   85 (0%)
[1928] 592.0-593.0 sec   122 KBytes  1000 Kbits/sec  8.292 ms    0/   85 (0%)
[1928] 593.0-594.0 sec   122 KBytes  1000 Kbits/sec  8.265 ms    0/   85 (0%)
[1928] 594.0-595.0 sec   122 KBytes  1000 Kbits/sec  8.230 ms    0/   85 (0%)
[1928] 595.0-596.0 sec   122 KBytes  1000 Kbits/sec  8.173 ms    0/   85 (0%)
[1928] 596.0-597.0 sec   122 KBytes  1000 Kbits/sec  8.005 ms    0/   85 (0%)
[1928] 597.0-598.0 sec   122 KBytes  1000 Kbits/sec  7.795 ms    0/   85 (0%)
[1928] 598.0-599.0 sec   122 KBytes  1000 Kbits/sec  7.794 ms    0/   85 (0%)
[1928] 599.0-600.0 sec   122 KBytes  1000 Kbits/sec  7.792 ms    0/   85 (0%)
[ ID] Interval      Transfer     Bandwidth       Jitter   Lost/Total Datagrams
[1928]  0.0-600.0 sec  73244 KBytes  1000 Kbits/sec  7.305 ms    0/51022 (0%)
```

Table 3: 10-second jperf output from IPv6 multicast receiver

From the output, it can be seen that over the 10-minute period, 73.244 MB of data was transferred at 1 Mbps with 0% packet loss. The jitter was 7.305 ms.

A screenshot of the live output from jperf is displayed below:



Iperf command: bin/iperf.exe -s -u -P 0 -i 1 -p 5001 -B ff06::6 -V -f k

Choose iPerf Mode: ○ Client ● Server

Server address

Port 5,001

Parallel Streams 1

Listen Port 5,001 ☐ Client Limit

Num Connections 0

Choose the protocol to use

○ TCP

☐ Buffer Length 2 MBytes

☐ TCP Window Size 56 KBytes

☐ Max Segment Size 1 KBytes

☐ TCP No Delay

● UDP

UDP Bandwidth 1 MBytes/sec

☐ UDP Buffer Size 1,450 KBytes

☐ UDP Packet Size 1,500 Bytes

TTL 1

Type of Service None

Bind to Host ff06::6

☑ IPv6

#1928: 1000.00KBits/s
Jitter: 5.97ms

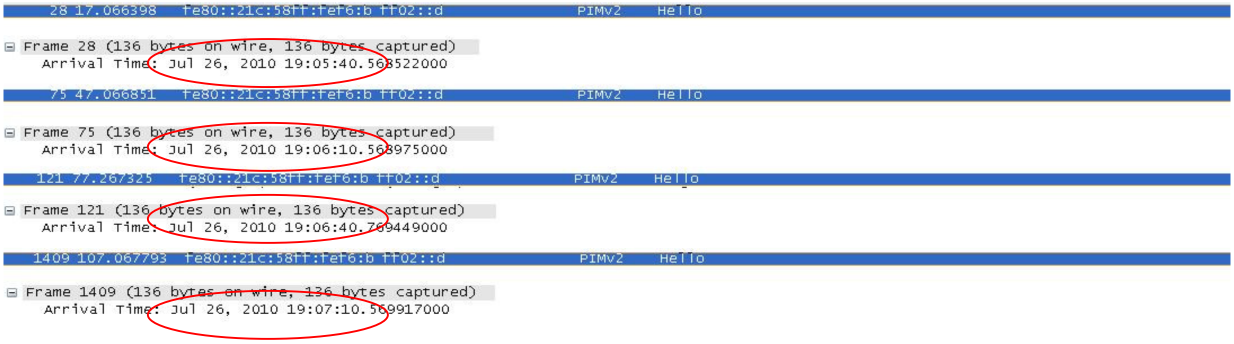
```
[1928] 332.0-333.0 sec   122 KBytes  1000 Kbits/sec  1.812 ms    0/   85
[1928] 333.0-334.0 sec   122 KBytes  1000 Kbits/sec  2.604 ms    0/   85
[1928] 334.0-335.0 sec   122 KBytes  1000 Kbits/sec  3.217 ms    0/   85
[1928] 335.0-336.0 sec   122 KBytes  1000 Kbits/sec  4.868 ms    0/   85
[1928] 336.0-337.0 sec   122 KBytes  1000 Kbits/sec  4.495 ms    0/   85
[1928] 337.0-338.0 sec   122 KBytes  1000 Kbits/sec  4.217 ms    0/   85
[1928] 338.0-339.0 sec   122 KBytes  1000 Kbits/sec  5.965 ms    0/   85
```

☑ Clear Output on each Iperf Run

Figure 11: Sample jperf screenshot from IPv6 multicast receiver

9.2.2 Protocol Overheads

When compared to IPv4, there was no difference in the protocol overhead that PIM adds when running over IPv6. Similar to IPv4, PIM sends out hello packets at 30-second intervals as can be seen from the Wireshark capture below.



```
    28 17.066398    fe80::21c:58ff:fef6:b ff02::d                    PIMv2     Hello

⊟ Frame 28 (136 bytes on wire, 136 bytes captured)
    Arrival Time: Jul 26, 2010 19:05:40.568522000

    75 47.066851    fe80::21c:58ff:fef6:b ff02::d                    PIMv2     Hello

⊟ Frame 75 (136 bytes on wire, 136 bytes captured)
    Arrival Time: Jul 26, 2010 19:06:10.568975000

   121 77.267325    fe80::21c:58ff:fef6:b ff02::d                    PIMv2     Hello

⊟ Frame 121 (136 bytes on wire, 136 bytes captured)
    Arrival Time: Jul 26, 2010 19:06:40.769449000

  1409 107.067793   fe80::21c:58ff:fef6:b ff02::d                    PIMv2     Hello

⊟ Frame 1409 (136 bytes on wire, 136 bytes captured)
    Arrival Time: Jul 26, 2010 19:07:10.569917000
```

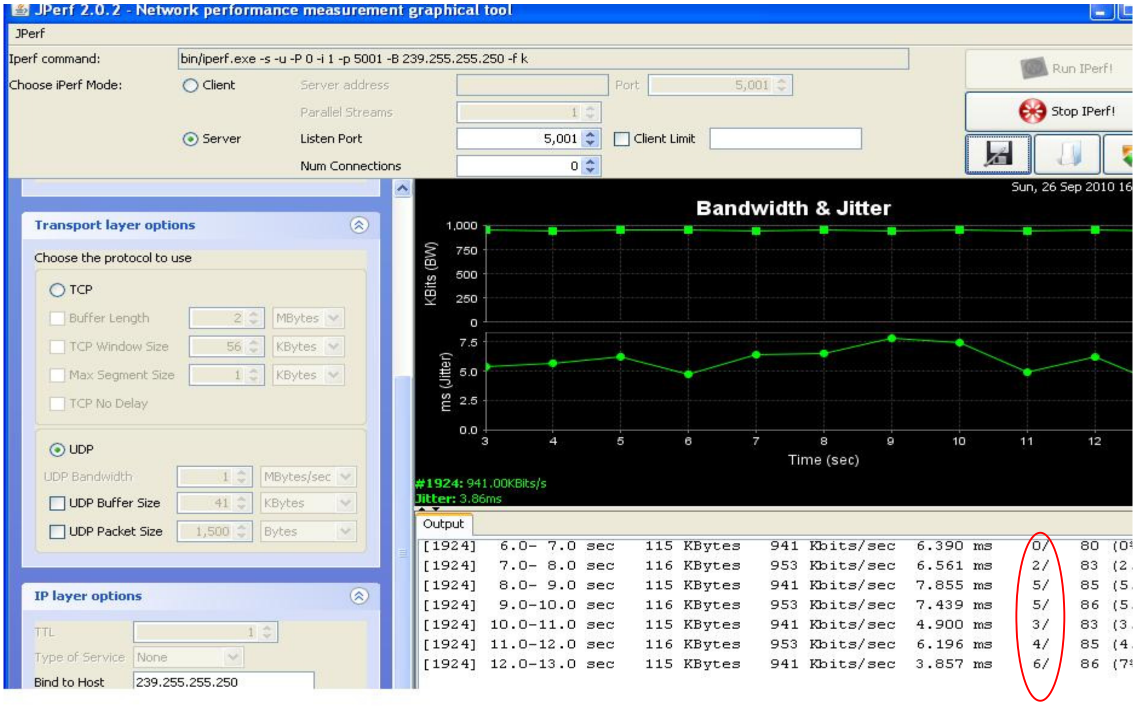
Figure 12: PIM Hello packets for IPv6 multicast

9.3 IPv4-IPv6 network – Dual-stack

9.3.1 Throughput and Jitter

For this scenario, an end-to-end dual-stack network was configured. Test outputs were obtained from an IPv4 only multicast receiver and an IPv6 only multicast receiver.

In this scenario, there was some jitter and packet loss in almost every test that was conducted. Sample screenshots and outputs from jperf are shown below:



Iperf command: bin/iperf.exe -s -u -P 0 -i 1 -p 5001 -B 239.255.255.250 -f k

Choose iPerf Mode: ○ Client ● Server

Server address

Port 5,001

Parallel Streams 1

Listen Port 5,001 ☐ Client Limit

Num Connections 0

Choose the protocol to use

○ TCP

☐ Buffer Length 2 MBytes

☐ TCP Window Size 56 KBytes

☐ Max Segment Size 1 KBytes

☐ TCP No Delay

● UDP

UDP Bandwidth 1 MBytes/sec

☐ UDP Buffer Size 41 KBytes

☐ UDP Packet Size 1,500 Bytes

TTL 1

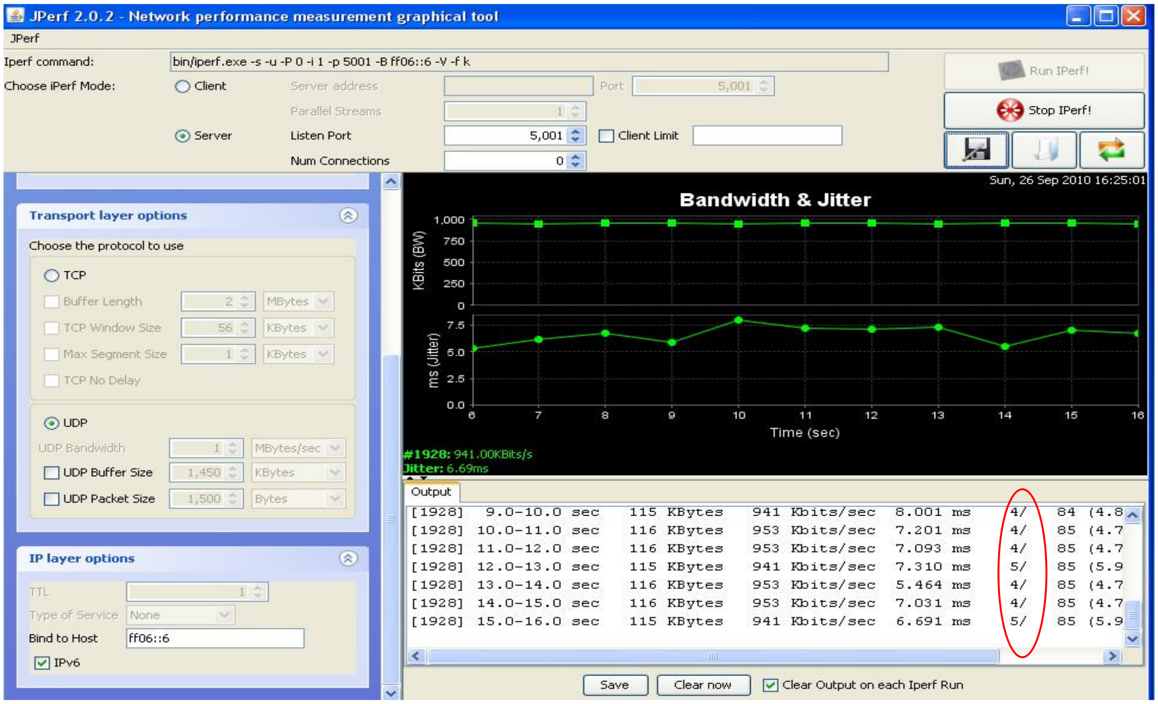
Type of Service None

Bind to Host 239.255.255.250

#1924: 941.00KBits/s
Jitter: 3.86ms

```
[1924]  6.0- 7.0 sec   115 KBytes   941 Kbits/sec  6.390 ms    0/   80 (0
[1924]  7.0- 8.0 sec   116 KBytes   953 Kbits/sec  6.561 ms    2/   83 (2.
[1924]  8.0- 9.0 sec   115 KBytes   941 Kbits/sec  7.855 ms    5/   85 (5.
[1924]  9.0-10.0 sec   116 KBytes   953 Kbits/sec  7.439 ms    5/   86 (5.
[1924] 10.0-11.0 sec   115 KBytes   941 Kbits/sec  4.900 ms    3/   83 (3.
[1924] 11.0-12.0 sec   116 KBytes   953 Kbits/sec  6.196 ms    4/   85 (4.
[1924] 12.0-13.0 sec   115 KBytes   941 Kbits/sec  3.857 ms    6/   86 (7
```

Figure 13: Sample jperf screenshot from IPv4 multicast receiver in dual-stack network



Iperf command: bin/iperf.exe -s -u -P 0 -i 1 -p 5001 -B ff06::6 -V -f k

Choose iPerf Mode: ○ Client Server address Port 5,001 Parallel Streams 1

● Server Listen Port 5,001 ☐ Client Limit

Num Connections 0

Sun, 26 Sep 2010 16:25:01

Transport layer options

Choose the protocol to use

○ TCP

☐ Buffer Length 2 MBytes

☐ TCP Window Size 56 KBytes

☐ Max Segment Size 1 KBytes

☐ TCP No Delay

● UDP

UDP Bandwidth 1 MBytes/sec

☐ UDP Buffer Size 1,450 KBytes

☐ UDP Packet Size 1,500 Bytes

IP layer options

TTL 1

Type of Service None

Bind to Host ff06::6

☑ IPv6

#1928: 941.00KBits/s
Jitter: 6.69ms

```
[1928]  9.0-10.0 sec   115 KBytes   941 Kbits/sec  8.001 ms    4/   84 (4.8
[1928] 10.0-11.0 sec   116 KBytes   953 Kbits/sec  7.201 ms    4/   85 (4.7
[1928] 11.0-12.0 sec   116 KBytes   953 Kbits/sec  7.093 ms    4/   85 (4.7
[1928] 12.0-13.0 sec   115 KBytes   941 Kbits/sec  7.310 ms    5/   85 (5.9
[1928] 13.0-14.0 sec   116 KBytes   953 Kbits/sec  5.464 ms    4/   85 (4.7
[1928] 14.0-15.0 sec   116 KBytes   953 Kbits/sec  7.031 ms    4/   85 (4.7
[1928] 15.0-16.0 sec   115 KBytes   941 Kbits/sec  6.691 ms    5/   85 (5.9
```

☑ Clear Output on each Iperf Run

Figure 14: Sample jperf screenshot from IPv6 multicast receiver in dual-stack network

```
[ ID] Interval       Transfer     Bandwidth       Jitter   Lost/Total Datagrams
[1924] 591.0-592.0 sec   116 KBytes   953 Kbits/sec  5.800 ms    3/   84 (3.6%)
[1924] 592.0-593.0 sec   115 KBytes   941 Kbits/sec  6.404 ms    5/   85 (5.9%)
[1924] 593.0-594.0 sec   116 KBytes   953 Kbits/sec  6.948 ms    4/   85 (4.7%)
[1924] 594.0-595.0 sec   115 KBytes   941 Kbits/sec  7.782 ms    5/   85 (5.9%)
[1924] 595.0-596.0 sec   115 KBytes   941 Kbits/sec  5.511 ms    4/   84 (4.8%)
[1924] 596.0-597.0 sec   116 KBytes   953 Kbits/sec  9.027 ms    5/   86 (5.8%)
[1924] 597.0-598.0 sec   116 KBytes   953 Kbits/sec  4.183 ms    4/   85 (4.7%)
[1924] 598.0-599.0 sec   115 KBytes   941 Kbits/sec  7.172 ms    6/   86 (7%)
[1924] 599.0-600.0 sec   116 KBytes   953 Kbits/sec  7.661 ms    4/   85 (4.7%)
[ ID] Interval       Transfer     Bandwidth       Jitter   Lost/Total Datagrams
[1924]  0.0-600.5 sec  69424 KBytes   947 Kbits/sec  4.873 ms 2600/50961 (5.1%)
```

Table 4: 10-second jperf output from IPv4 multicast receiver in dual-stack network

```
[ ID] Interval       Transfer     Bandwidth       Jitter   Lost/Total Datagrams
[1928] 591.0-592.0 sec   116 KBytes   953 Kbits/sec  5.237 ms    4/   85 (4.7%)
[1928] 592.0-593.0 sec   115 KBytes   941 Kbits/sec  6.996 ms    5/   85 (5.9%)
[1928] 593.0-594.0 sec   116 KBytes   953 Kbits/sec  5.481 ms    5/   86 (5.8%)
[1928] 594.0-595.0 sec   115 KBytes   941 Kbits/sec  5.986 ms    4/   84 (4.8%)
[1928] 595.0-596.0 sec   118 KBytes   964 Kbits/sec  6.764 ms    4/   86 (4.7%)
[1928] 596.0-597.0 sec   115 KBytes   941 Kbits/sec  6.676 ms    4/   84 (4.8%)
[1928] 597.0-598.0 sec   116 KBytes   953 Kbits/sec  5.115 ms    4/   85 (4.7%)
```

```
[1928] 598.0-599.0 sec   115 KBytes   941 Kbits/sec  6.401 ms    5/   85 (5.9%)
[1928] 599.0-600.0 sec   116 KBytes   953 Kbits/sec  5.288 ms    4/   85 (4.7%)
[ ID] Interval       Transfer     Bandwidth       Jitter   Lost/Total Datagrams
[1928]  0.0-600.4 sec  69519 KBytes   949 Kbits/sec  7.775 ms 2595/51022 (5.1%)
```

Table 5: 10-second jperf output from IPv6 multicast receiver in dual-stack network

From the screenshots it can be seen that for every interval of packet transmission, there is some packet loss.

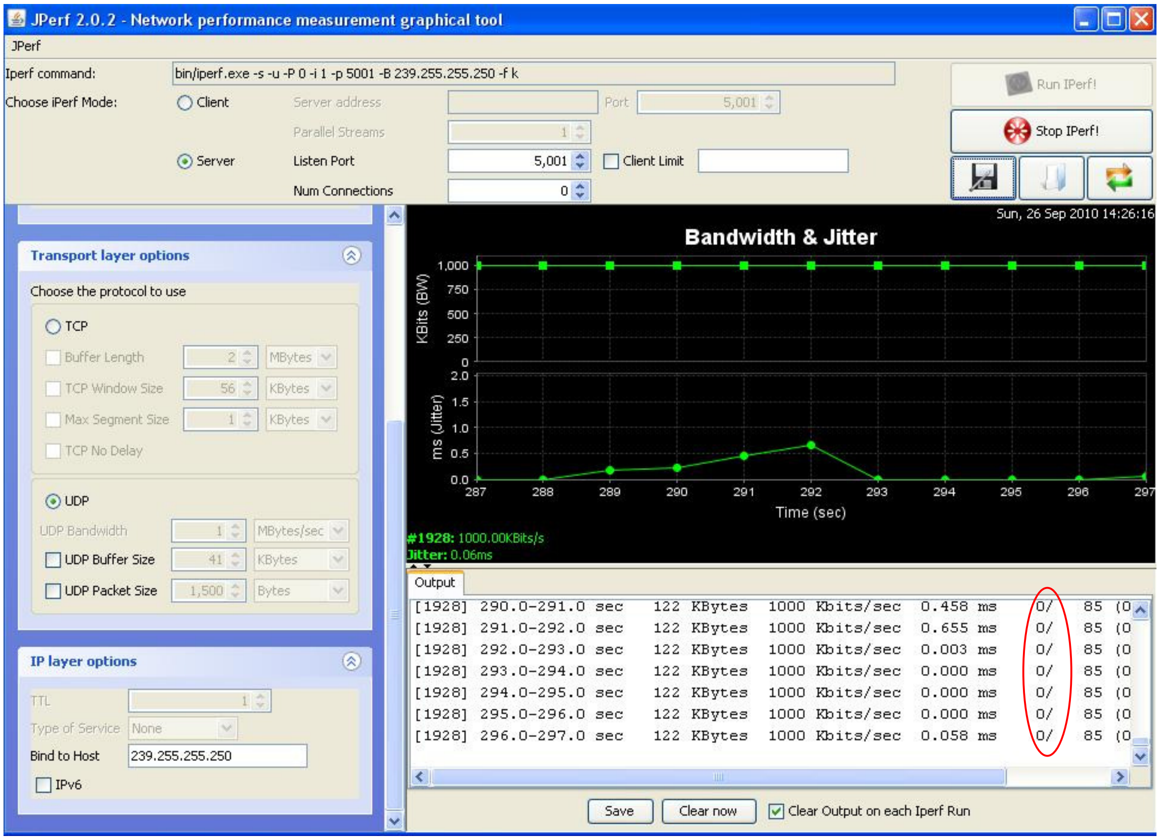
Two 1-hour tests were also conducted and packet loss was observed in both the test cases.

The table below shows the throughput for an IPv4 multicast receiver and an IPv6 multicast receiver for all the ten 10-minute tests:

10-minute test	Dual-stack IPv4 multicast receiver throughput (%)	Dual-stack IPv6 multicast receiver throughput (%)
1	94.84	94.988
2	94.871	94.966
3	94.863	94.914
4	94.898	94.932
5	94.88	94.959
6	94.913	94.931
7	94.837	94.934
8	94.79	94.955
9	94.844	94.962
10	94.897	94.952

Table 6: Throughput for IPv4 and IPv6 receivers in dual-stack network

For all the tests conducted in all the four scenarios, a few sample results were obtained from a multicast receiver in the same subnet as the source and consistently, the jitter was 0 ms in most cases and less than 2 ms in other cases. So it can be concluded that any variation in latency and packet loss was caused due to the routing of the multicast traffic across the four routers. This result is significant in this dual-stack scenario, where the multicast receiver residing in the same subnet as the source has negligible jitter and packet loss. A screenshot of an IPv4 host on the same subnet is shown below:



Iperf command: bin/iperf.exe -s -u -P 0 -i 1 -p 5001 -B 239.255.255.250 -f k

Choose iPerf Mode: ○ Client Server address Port 5,001 Parallel Streams 1

● Server Listen Port 5,001 ☐ Client Limit Num Connections 0

Choose the protocol to use

○ TCP
☐ Buffer Length 2 MBytes
☐ TCP Window Size 56 KBytes
☐ Max Segment Size 1 KBytes
☐ TCP No Delay

● UDP
UDP Bandwidth 1 MBytes/sec
☐ UDP Buffer Size 41 KBytes
☐ UDP Packet Size 1,500 Bytes

TTL 1
Type of Service None
Bind to Host 239.255.255.250
☐ IPv6

#1928: 1000.00KBits/s
Jitter: 0.06ms

```
[1928] 290.0-291.0 sec   122 KBytes  1000 Kbits/sec  0.458 ms   0/   85 (0
[1928] 291.0-292.0 sec   122 KBytes  1000 Kbits/sec  0.655 ms   0/   85 (0
[1928] 292.0-293.0 sec   122 KBytes  1000 Kbits/sec  0.003 ms   0/   85 (0
[1928] 293.0-294.0 sec   122 KBytes  1000 Kbits/sec  0.000 ms   0/   85 (0
[1928] 294.0-295.0 sec   122 KBytes  1000 Kbits/sec  0.000 ms   0/   85 (0
[1928] 295.0-296.0 sec   122 KBytes  1000 Kbits/sec  0.000 ms   0/   85 (0
[1928] 296.0-297.0 sec   122 KBytes  1000 Kbits/sec  0.058 ms   0/   85 (0
```

☑ Clear Output on each Iperf Run

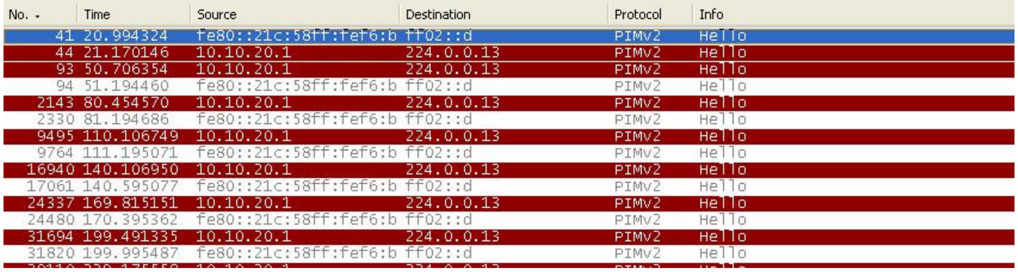
Figure 15: IPv4 multicast receiver in the same subnet as the source in dual-stack network

```
[ ID] Interval       Transfer     Bandwidth       Jitter   Lost/Total Datagrams
[1928] 591.0-592.0 sec   122 KBytes  1000 Kbits/sec  3.155 ms    0/   85 (0%)
[1928] 592.0-593.0 sec   122 KBytes  1000 Kbits/sec  0.254 ms    0/   85 (0%)
[1928] 593.0-594.0 sec   122 KBytes  1000 Kbits/sec  0.022 ms    0/   85 (0%)
[1928] 594.0-595.0 sec   122 KBytes  1000 Kbits/sec  2.773 ms    0/   85 (0%)
[1928] 595.0-596.0 sec   122 KBytes  1000 Kbits/sec  0.450 ms    0/   85 (0%)
[1928] 596.0-597.0 sec   122 KBytes  1000 Kbits/sec  0.011 ms    0/   85 (0%)
[1928] 597.0-598.0 sec   122 KBytes  1000 Kbits/sec  0.879 ms    0/   85 (0%)
[1928] 598.0-599.0 sec   122 KBytes  1000 Kbits/sec  2.891 ms    0/   85 (0%)
[1928] 599.0-600.0 sec   122 KBytes  1000 Kbits/sec  1.794 ms    0/   85 (0%)
[ ID] Interval       Transfer     Bandwidth       Jitter   Lost/Total Datagrams
[1928]  0.0-600.0 sec  73207 KBytes   999 Kbits/sec  1.682 ms    2/50998 (0.0039%)
```

Table 7: 10-second jperf output from IPv6 multicast receiver in the same subnet as the source in a dual-stack network

9.3.2 Protocol Overheads

As in the case of the previous scenarios, the routing protocol PIM does not contribute to any significant router traffic as can be seen from the capture below. Every 30 seconds, hello packets are exchanged and it can be seen from this Wireshark capture for both IPv4 and IPv6 multicast.



No.	Time	Source	Destination	Protocol	Info
41	20.994324	fe80::21c:58ff:fef6:b	ff02::d	PIMv2	Hello
44	21.170146	10.10.20.1	224.0.0.13	PIMv2	Hello
93	50.706354	10.10.20.1	224.0.0.13	PIMv2	Hello
94	51.194460	fe80::21c:58ff:fef6:b	ff02::d	PIMv2	Hello
2143	80.454570	10.10.20.1	224.0.0.13	PIMv2	Hello
2330	81.194686	fe80::21c:58ff:fef6:b	ff02::d	PIMv2	Hello
9495	110.106749	10.10.20.1	224.0.0.13	PIMv2	Hello
9764	111.195071	fe80::21c:58ff:fef6:b	ff02::d	PIMv2	Hello
16940	140.106950	10.10.20.1	224.0.0.13	PIMv2	Hello
17061	140.595077	fe80::21c:58ff:fef6:b	ff02::d	PIMv2	Hello
24337	169.815151	10.10.20.1	224.0.0.13	PIMv2	Hello
24480	170.395362	fe80::21c:58ff:fef6:b	ff02::d	PIMv2	Hello
31694	199.491335	10.10.20.1	224.0.0.13	PIMv2	Hello
31820	199.995487	fe80::21c:58ff:fef6:b	ff02::d	PIMv2	Hello

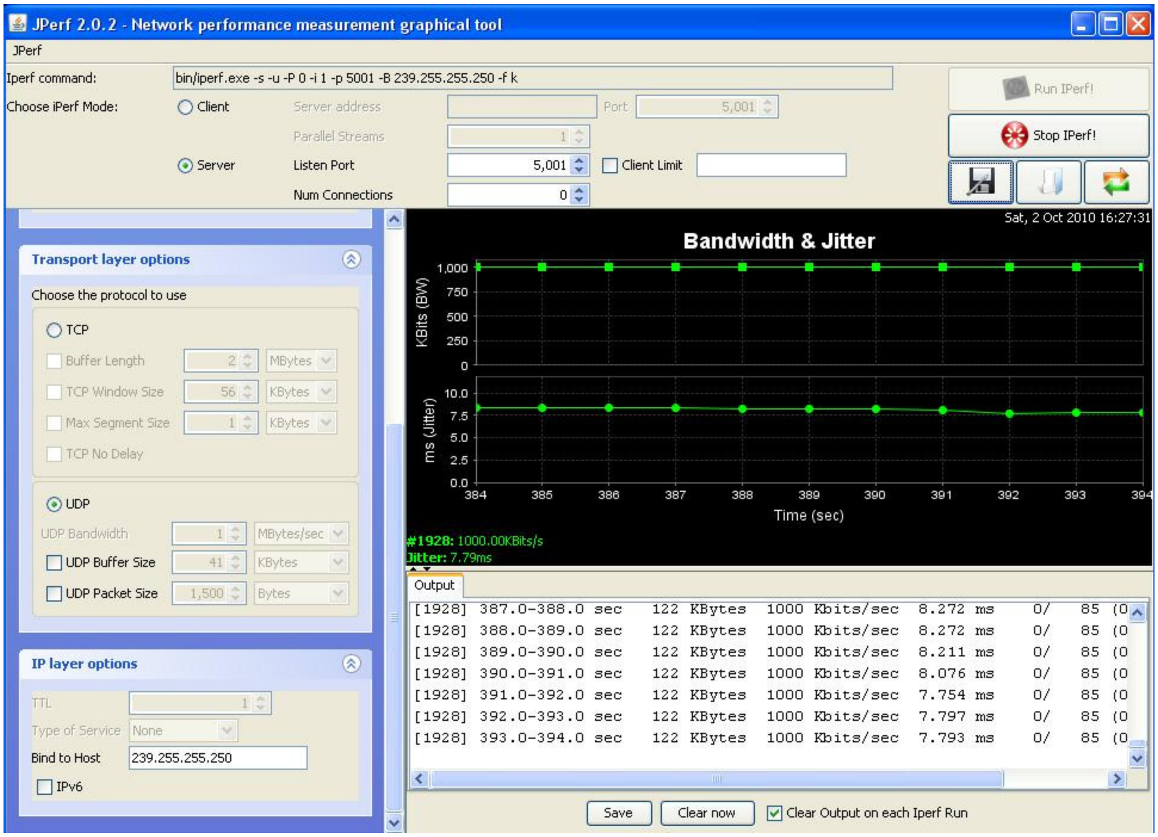
Figure 16: PIM hello packets for IPv4-IPv6 dual-stack multicast

9.4 IPv4-IPv6 network – GRE Tunneling

This scenario is one that is most likely to occur during the interim period when the transition from an IPv4 only network to an IPv6 only network takes place. While ISPs may start the migration, end users may not make the transition at the same pace. The GRE tunnel was configured to route the IPv4 multicast traffic across an IPv6 backbone. Refer to Appendix for sample router configuration.

9.4.1 Throughput and Jitter

Similar to an IPv4 or IPv6 only network, this network also did not have much jitter and had no packet loss during all the tests, which can be seen from the jperf screenshot and outputs below:



Iperf command: bin/iperf.exe -s -u -P 0 -i 1 -p 5001 -B 239.255.255.250 -f k

Choose iPerf Mode: ○ Client ● Server

Server address:

Port: 5,001

Parallel Streams: 1

Listen Port: 5,001

☐ Client Limit

Num Connections: 0

Bind to Host: 239.255.255.250

```
[1928]  387.0-388.0 sec    122 KBytes   1000 Kbits/sec  8.272 ms     0/    85 (0
[1928]  388.0-389.0 sec    122 KBytes   1000 Kbits/sec  8.272 ms     0/    85 (0
[1928]  389.0-390.0 sec    122 KBytes   1000 Kbits/sec  8.211 ms     0/    85 (0
[1928]  390.0-391.0 sec    122 KBytes   1000 Kbits/sec  8.076 ms     0/    85 (0
[1928]  391.0-392.0 sec    122 KBytes   1000 Kbits/sec  7.754 ms     0/    85 (0
[1928]  392.0-393.0 sec    122 KBytes   1000 Kbits/sec  7.797 ms     0/    85 (0
[1928]  393.0-394.0 sec    122 KBytes   1000 Kbits/sec  7.793 ms     0/    85 (0
```

Figure 17: Sample jperf screenshot from IPv4 multicast receiver across GRE tunnel

The last ten seconds of the jperf output was as below:

```
[ ID] Interval       Transfer     Bandwidth       Jitter   Lost/Total Datagrams
[1928] 591.0-592.0 sec   122 KBytes  1000 Kbits/sec  0.002 ms    0/   85 (0%)
```

```
[1928] 592.0-593.0 sec   122 KBytes  1000 Kbits/sec  0.000 ms    0/   85 (0%)
[1928] 593.0-594.0 sec   122 KBytes  1000 Kbits/sec  0.000 ms    0/   85 (0%)
[1928] 594.0-595.0 sec   122 KBytes  1000 Kbits/sec  0.000 ms    0/   85 (0%)
[1928] 595.0-596.0 sec   122 KBytes  1000 Kbits/sec  0.000 ms    0/   85 (0%)
[1928] 596.0-597.0 sec   122 KBytes  1000 Kbits/sec  0.000 ms    0/   85 (0%)
[1928] 597.0-598.0 sec   122 KBytes  1000 Kbits/sec  1.078 ms    0/   85 (0%)
[1928] 598.0-599.0 sec   122 KBytes  1000 Kbits/sec  0.004 ms    0/   85 (0%)
[1928] 599.0-600.0 sec   122 KBytes  1000 Kbits/sec  0.118 ms    0/   85 (0%)
[ ID] Interval      Transfer     Bandwidth       Jitter   Lost/Total Datagrams
[1928]  0.0-600.0 sec  73244 KBytes  1000 Kbits/sec  0.111 ms    0/51022 (0%)
```

Table 8: 10-second jperf output from IPv4 multicast receiver across GRE tunnel

In this sample, the jitter was 0.111 ms with no packet loss. During all the sample tests the jitter ranged from 0 ms to 7.792 ms.

9.4.2 Protocol Overhead

As in the case of all the scenarios, the only traffic that PIM generated was the hello packets at 30-second intervals. This can be seen from the Wireshark capture below, where only the PIM traffic has been filtered out.

Filter: pim

No.	Time	Source	Destination	Protocol	Info
73	21.028071	10.10.20.1	224.0.0.13	PIMv2	Hello
3743	50.296191	10.10.20.1	224.0.0.13	PIMv2	Hello
8880	80.176296	10.10.20.1	224.0.0.13	PIMv2	Hello
14029	110.080397	10.10.20.1	224.0.0.13	PIMv2	Hello
19036	139.229038	10.10.20.1	224.0.0.13	PIMv2	Hello
24170	169.001467	10.10.20.1	224.0.0.13	PIMv2	Hello
29236	198.468744	10.10.20.1	224.0.0.13	PIMv2	Hello
34335	228.165264	10.10.20.1	224.0.0.13	PIMv2	Hello
39421	257.740978	10.10.20.1	224.0.0.13	PIMv2	Hello
44458	287.073100	10.10.20.1	224.0.0.13	PIMv2	Hello
49590	316.973232	10.10.20.1	224.0.0.13	PIMv2	Hello
54729	346.801343	10.10.20.1	224.0.0.13	PIMv2	Hello
59871	376.757436	10.10.20.1	224.0.0.13	PIMv2	Hello
64937	406.237542	10.10.20.1	224.0.0.13	PIMv2	Hello
69996	435.605980	10.10.20.1	224.0.0.13	PIMv2	Hello
75138	465.569793	10.10.20.1	224.0.0.13	PIMv2	Hello
80182	494.910472	10.10.20.1	224.0.0.13	PIMv2	Hello
85323	524.818020	10.10.20.1	224.0.0.13	PIMv2	Hello

Figure 18: PIM hello packets for IPv4-IPv6 GRE tunneled network multicast

9.5 Graphical representation of results

The 10-minute and 1-hour test results collected from the different scenarios were plotted in graph charts.

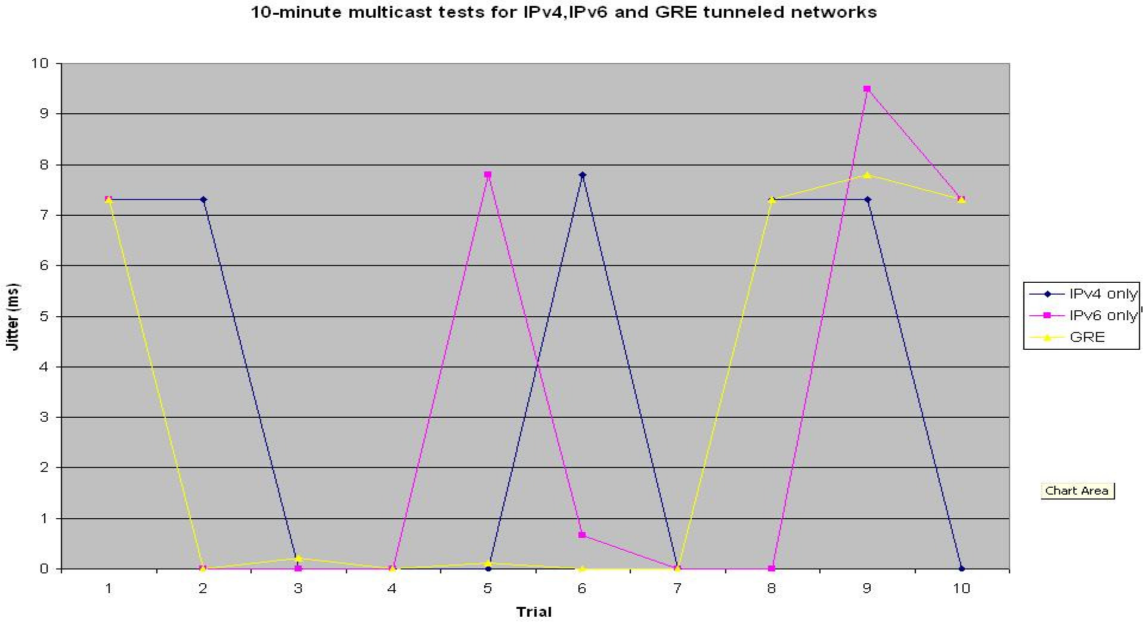


Figure 19: 10-minute multicast tests for IPv4, IPv6 and GRE tunneled networks

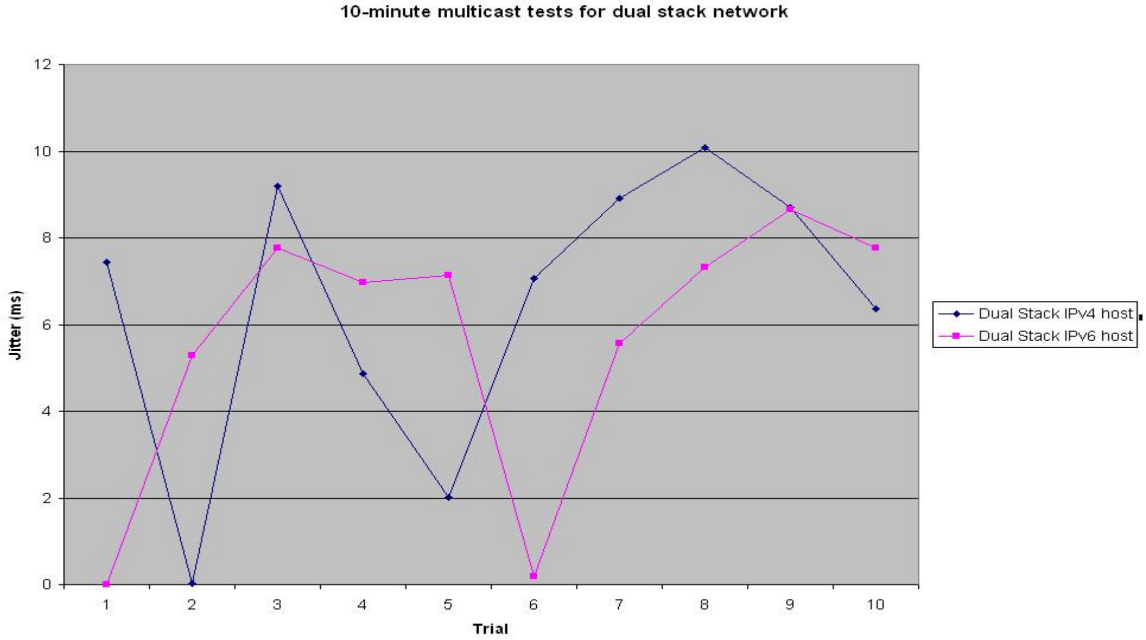


Figure 20: 10-minute multicast tests for dual-stack network

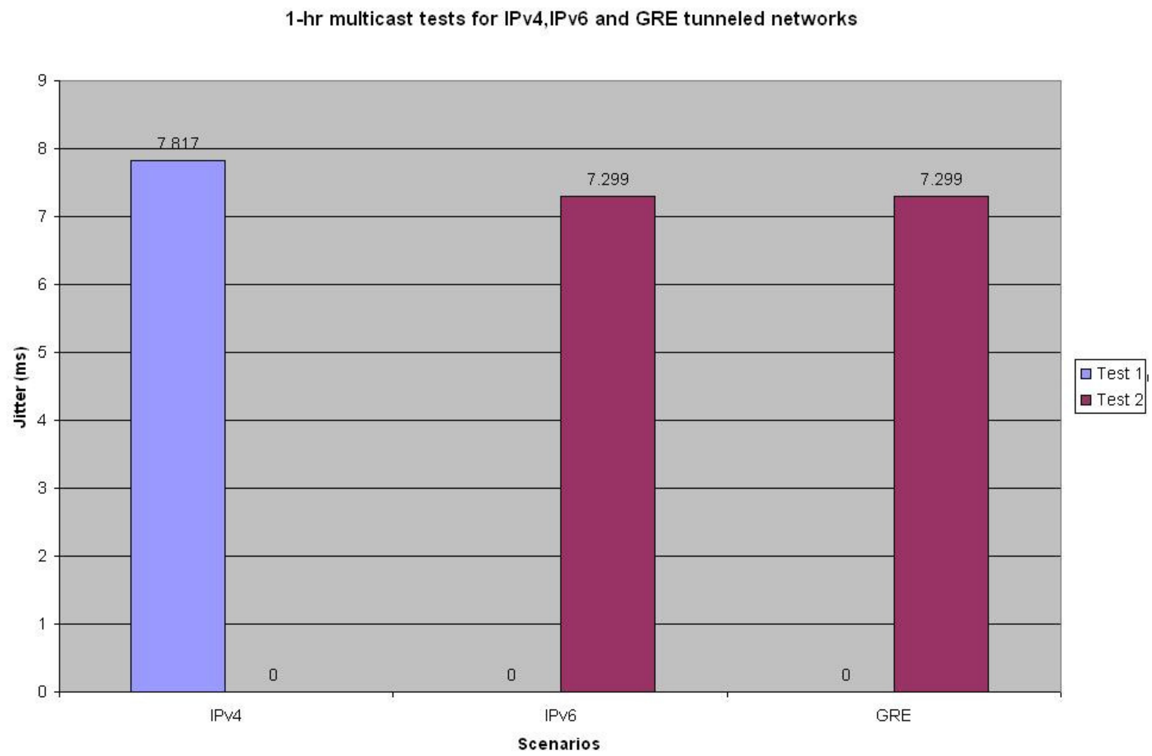


Figure 21: 1-hr multicast tests for IPv4, IPv6 and GRE tunneled networks
Note: The values on the graphs indicate the jitter was 0 ms in some of the tests

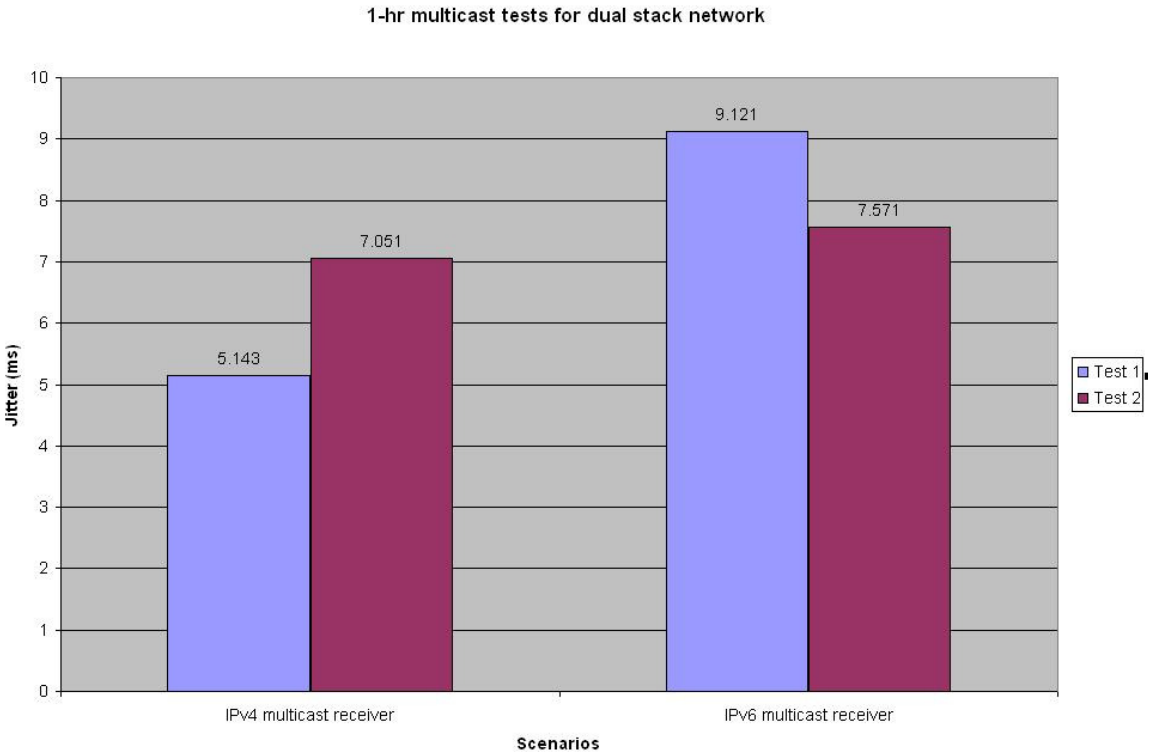


Figure 22: 1-hr multicast tests for dual-stack network

10 Conclusions from experimental results

The same set of outputs was gathered from an all IPv4 and an all IPv6 network. While the difference in the results is not significantly different, the results disprove the hypothesis of this thesis that the protocol overhead, jitter and throughput in an IPv6 network would be significantly larger than an IPv4 network, due to its larger address space. The protocol overheads in both the networks remained the same.

In the experiments conducted in this thesis, the payload in the case of IPv4 and IPv6 was kept constant. The interface Maximum Transfer Units (MTUs) were kept at their default values - PC Network Interface Cards (NICs) had the default MTU of 1500 and the Cisco routers were also left at the default value of 1500. In the case of IPv4, there was no fragmentation, whereas in IPv6 the fragmentation was handled by the host. Even with the additional task of fragmentation, there was no deterioration in the performance of the IPv6 network, which proves that IPv6 handled the fragmentation efficiently. A future study could be conducted with varying MTUs/packet sizes across the network and see how it affects the performance.

Moreover, since IPv6 was designed as a replacement for IPv4, it was designed to be better than IPv4. The IPv6 header is simpler than an IPv4 header. For instance, the options field, which is included in the IPv4 header, is an extension in the IPv6 header. So without any options, the IPv6 header is not as complex as an IPv4 header. Checksum, for error detection in IPv4, is eliminated in IPv6 (other layers take care of error detection). More information on IPv6 design can be obtained from its RFC [20].

In the case of the interim period during the transition from IPv4 to IPv6, both protocols would co-exist. From the experimental results, it can be seen that an end-to-end dual-stack network experiences more packet loss as compared to tunneling. It has to be noted though, that in the case of the dual-stack network, there were two multicast streams – one for IPv4 and one for IPv6. Running multiple streams of multicast traffic on the different scenarios has been proposed as a future work. This would help in understanding if tunneling is a better option thank dual-stack in the case of mixed IPv6-IPv6 networks.

11 Future Work

- All the tests were conducted in a lab environment, with no other traffic, except for that generated for experimental purposes. As a next step, other traffic can be introduced into the network, to study the performance in a closer to real-world setup.

- Further complexity can be introduced into the network, by adding more multicast groups and receivers being members of more than one multicast group. This would help in understanding what latency/jitter the router introduces when it has to process more multicast traffic and multicast routing decisions.

- A similar study can be conducted to test other unicast and multicast routing protocols that are used. This would help in understanding how different protocols perform and aid in deciding a protocol that would best suit a network.

- The impact of scalability can be studied by increasing the number of multicast sources and receivers either in an experimental setup where feasible or using a simulation tool.

12 Appendix

12.1 Sample IPv4 Router configuration

```
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R4v4
!
boot-start-marker
boot-end-marker
!
enable password cisco
!
no aaa new-model
no network-clock-participate wic 2
no network-clock-participate wic 3
!
!
ip cef
!
!
ip multicast-routing
ip auth-proxy max-nodata-conns 3
ip admission max-nodata-conns 3
!
!
voice-card 0
 no dspfarm
!
!
 controller T1 0/2/0
 framing esf
 linecode b8zs
!
controller T1 0/3/0
 framing esf
 linecode b8zs
!
!
interface FastEthernet0/0
 ip address 10.10.20.1 255.255.255.0
 ip pim sparse-mode
```

```
 ip igmp static-group 239.255.255.250
 duplex auto
 speed auto
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet0/1/0
!
interface FastEthernet0/1/1
!
interface FastEthernet0/1/2
!
interface FastEthernet0/1/3
!
interface Serial0/0/0
 no ip address
 shutdown
 no fair-queue
 clock rate 2000000
!
interface Serial0/0/1
 ip address 192.168.3.2 255.255.255.252
 ip pim sparse-mode
 ip igmp static-group 239.255.255.250
!
interface Vlan1
 no ip address
!
router ospf 4
 log-adjacency-changes
 network 10.10.0.0 0.0.255.255 area 0
 network 192.168.0.0 0.0.255.255 area 0
!
ip forward-protocol nd
!
!
ip http server
no ip http secure-server
ip pim rp-address 10.10.10.1
!
control-plane
!
```

```
!line con 0
line aux 0
line vty 0 4
 login
!
scheduler allocate 20000 1000
!
End
```

12.2 Sample IPv6 router configuration:

```
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R4v6
!
boot-start-marker
boot-end-marker
!
enable password cisco
!
no aaa new-model
no network-clock-participate wic 2
no network-clock-participate wic 3
!
!
ip cef
!
!
 --More—
ip auth-proxy max-nodata-conns 3
ip admission max-nodata-conns 3
!
ipv6 unicast-routing
ipv6 multicast-routing
!
voice-card 0
 no dspfarm
!
!
!
 --More--
```

```
controller T1 0/2/0
 framing esf
 linecode b8zs
!
controller T1 0/3/0
 framing esf
 linecode b8zs
!
!
 interface FastEthernet0/0
 no ip address
 duplex auto
 speed auto
 ipv6 address 2001:179::1/64
 ipv6 mld static-group FF06::6
 ipv6 ospf 4 area 0
!
interface FastEthernet0/1
 --More—
no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet0/1/0
!
interface FastEthernet0/1/1
!
interface FastEthernet0/1/2
!
interface FastEthernet0/1/3
!
interface Serial0/0/0
 no ip address
 shutdown
 clock rate 2000000
!
interface Serial0/0/1
 no ip address
 encapsulation ppp
 ipv6 address 2001:178::2/64
 ipv6 mld static-group FF06::6
 --More—
ipv6 ospf 4 area 0
 clock rate 2000000
!
```

```
interface Vlan1
 no ip address
!
ip forward-protocol nd
!
!
ip http server
no ip http secure-server
!
snmp-server community public RW
ipv6 router ospf 4
 router-id 10.10.40.1
 log-adjacency-changes
!
ipv6 pim rp-address 2001:175::1
!
!
control-plane
 --More--
!
!
line con 0
line aux 0
line vty 0 4
 login
!
scheduler allocate 20000 1000
!
end
```

Notes:

- There are small variations in the configuration of OSPF for IPv4 and IPv6. In OSPF for IPv6, every interface has to be explicitly configured to run OSPF. In IPv4, once OSPF is configured, the interfaces indirectly start participating in the OSPF process.
- The router ID that is configured on the routers for neighbor discovery is a 32-bit ID. For IPv4, since the address is a 32-bit address, it was used as the router ID. In IPv6, any 32-bit ID has to be set as the router ID (I chose to use an IPv4 address as the 32-bit router ID).

Reference
http://www.cisco.com/en/US/docs/ios/ipv6/configuration/guide/ip6-ospf.html

12.3 Sample IPv4-IPv6 dual-stack configuration:

```
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R4v4v6ds
!
boot-start-marker
boot-end-marker
!
no aaa new-model
no network-clock-participate wic 2
no network-clock-participate wic 3
!
!
ip cef
!
!
ip multicast-routing
ip auth-proxy max-nodata-conns 3
ip admission max-nodata-conns 3
!
ipv6 unicast-routing
ipv6 multicast-routing
!
voice-card 0
 no dspfarm
!
!
controller T1 0/2/0
 framing esf
 linecode b8zs
!
controller T1 0/3/0
 framing esf
 linecode b8zs
!
!
interface FastEthernet0/0
 ip address 10.10.20.1 255.255.255.0
 ip pim sparse-mode
 ip igmp static-group 239.255.255.250
 ip ospf 4 area 0
 duplex auto
```

```
 speed auto
 ipv6 address 2001:179::1/64
 ipv6 mld static-group FF06::6
 ipv6 ospf 4 area 0
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet0/1/0
!
interface FastEthernet0/1/1
!
interface FastEthernet0/1/2
!
interface FastEthernet0/1/3
!
interface Serial0/0/0
 no ip address
 shutdown
 no fair-queue
 clock rate 2000000
!
interface Serial0/0/1
 ip address 192.168.3.2 255.255.255.252
 ip pim sparse-mode
 encapsulation ppp
 ip igmp static-group 239.255.255.250
 ip ospf 4 area 0
 ipv6 address 2001:178::2/64
 ipv6 mld static-group FF06::6
 ipv6 ospf 4 area 0
!
interface Vlan1
 no ip address
!
router ospf 4
 log-adjacency-changes
!
ip forward-protocol nd
!
!
ip http server
no ip http secure-server
```

```
ip pim rp-address 10.10.10.1
!
ipv6 router ospf 4
 router-id 4.4.4.4
 log-adjacency-changes
!
ipv6 pim rp-address 2001:175::1
!
control-plane
!
!
line con 0
line aux 0
line vty 0 4
 login
scheduler allocate 20000 1000
!
End
```

12.4 Sample IPv4-IPv6 GRE tunnel configuration:

12.4.a Edge router sample configuration:

```
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R4v4v6gre
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
no network-clock-participate wic 2
no network-clock-participate wic 3
!
ip cef
!
ip multicast-routing
ip auth-proxy max-nodata-conns 3
ip admission max-nodata-conns 3
!
ipv6 unicast-routing
```

```
!
voice-card 0
 no dspfarm
!
!
controller T1 0/2/0
 framing esf
 linecode b8zs
!
controller T1 0/3/0
 framing esf
 linecode b8zs
!
!
interface Tunnel10
 ip address 192.168.1.1 255.255.255.252
 ip pim sparse-mode
 ip igmp static-group 239.255.255.250
 ip ospf 4 area 1
 ipv6 address 2001:180::1/64
 tunnel source Serial0/0/1
 tunnel destination 2001:176::1
 tunnel mode gre ipv6
!
 interface FastEthernet0/0
 ip address 10.10.20.1 255.255.255.0
 ip pim sparse-mode
 ip igmp static-group 239.255.255.250
 ip ospf 4 area 1
 duplex auto
 speed auto
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet0/1/0
!
interface FastEthernet0/1/1
!
interface FastEthernet0/1/2
!
interface FastEthernet0/1/3
!
```

```
interface Serial0/0/0
no ip address
 shutdown
 no fair-queue
 clock rate 2000000
!
interface Serial0/0/1
 no ip address
 encapsulation ppp
 ipv6 address 2001:178::2/64
 ipv6 ospf 4 area 0
 clock rate 2000000
!
interface Vlan1
 no ip address
!
router ospf 4
 log-adjacency-changes
!
ip forward-protocol nd
!
!
ip http server
no ip http secure-server
 ip pim rp-address 10.10.10.1
!
ipv6 router ospf 4
 router-id 4.4.4.4
 log-adjacency-changes
!
!
control-plane
!
!
line con 0
line aux 0
 line vty 0 4
 login
!
scheduler allocate 20000 1000
!
End
```

12.4.b Core IPv6 router sample configuration:

```
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R3v4v6gre
boot-start-marker
boot-end-marker
!
no aaa new-model
!
!
ip cef
!
!
ip auth-proxy max-nodata-conns 3
ip admission max-nodata-conns 3
!
 ipv6 unicast-routing
!
voice-card 0
 no dspfarm
!
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet0/1/0
!
interface FastEthernet0/1/1
!
interface FastEthernet0/1/2
!
interface FastEthernet0/1/3
!
```

```
interface Serial0/0/0
no ip address
 encapsulation ppp
 ipv6 address 2001:178::1/64
 ipv6 ospf 3 area 0
 no fair-queue
!
interface Serial0/0/1
 no ip address
 encapsulation ppp
 ipv6 address 2001:177::2/64
 ipv6 ospf 3 area 0
 clock rate 2000000
!
interface Vlan1
 no ip address
!
ip forward-protocol nd
!
!
ip http server
no ip http secure-server
!
ipv6 router ospf 3
router-id 3.3.3.3
 log-adjacency-changes
!
!
control-plane
!
!
line con 0
line aux 0
line vty 0 4
 login
!
scheduler allocate 20000 1000
!
end
```

12.5 Steps to configure manual IPv6 address on Windows XP:

From command prompt, execute the following commands:

Step 1: Ipv6 install

Once IPv6 is installed successfully, proceed to the next step.

Step 2: ipv6 if
This command will display all the interfaces on the machine along with the interface indices.

Step 3: ipv6 adu <interface index>/<IPv6 address>

Where Interface index, is the index of the interface for which the IPv6 address is to be configured (obtained from Step 2).

To uninstall IPv6

From the command prompt, run the following command:

netsh int ipv6 uninstall

Reference:
http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/sag_ip_v6_pro_inst.mspx?mfr=true

Bibliography

1. S.E. Deering, *Multicast routing in internetworks and extended LANs*. Stanford, CA: ACM, 1988, pp. 55-64
2. Iperf
   http://iperf.sourceforge.net/
3. Iperf – The Easy Tutorial
   http://openmaniak.com/iperf.php
4. RFC 2236 Internet Group Management Protocol, Version 2 , November 1997
   http://www.ietf.org/rfc/rfc2236.txt
5. RFC 3376 Internet Group Management Protocol, Version 3, October 2002
   http://tools.ietf.org/rfc/rfc3376.txt
6. RFC 2710 Multicast Listener Discovery (MLD) for IPv6, October 1999
   http://tools.ietf.org/rfc/rfc2710.txt
7. 7. RFC 1112 Host extensions for IP multicasting, August 1989
   http://www.ietf.org/rfc/rfc1112.txt
8. RFC 1584 Multicast extensions to OSPF, March 1994
   http://www.rfc-editor.org/rfc/rfc1584.txt
9. Deborah Estrin, Ahmed Helmy and David Thaler, "PIM Multicast Border Router (PMBR) specification for connecting PIM – SM domains to a DVMRP backbone," 1997
   http://www.cise.ufl.edu/~helmy/papers/PMBR-spec.pdf
10. Li-wei H. Lehman, Stephen J. Garland and David L. Tennenhouse, "Active Reliable Multicast"
    http://web.mit.edu/lilehman/www/paper/arm.pdf
11. Yuji IMAI, Hiro KISHIM0TO, Myung-Ki SHIN and Young-Han KIM, "XCAST6: eXplicit Multicast on IPv6," in *Proceedings of the 2003 Symposium On Applications and the Internet Workshops,* 2003, pp. 238
12. RFC 4966 Reasons to Move the Network Address Translator – Protocol Translator (NAT-PT) to Historic Status
    http://www.rfc-editor.org/rfc/rfc4966.txt
13. Kazuaki Tsuchiya, "An IPv6/IPv4 Multicast Translator based on IGMP/MLD Proxying (MTP)," presented at *52nd IETF MAGMA Meeting, Salt Lake City, Utah, December 2001.*
    http://www.ietf.org/proceedings/52/slides/magma-4.pdf
14. S. Venaas, "An IPv4 – IPv6 Multicast Gateway , *IETF Draft, February 2003.*
    http://www.6net.org/publications/standards/draft-venaas-mboned-v4v6mcastgw-00.txt
15. *Performance-Comparison Testing of IPv4 and IP v6 Throughput and Latency on Key Cisco Router Platforms,* A Cisco White Paper
    http://www.cisco.com/web/strategy/docs/gov/IPv6perf_wp1f.pdf
16. Narayan, S., Kolahi, S.S., Sunarto, Y., Nguyen, D. and Mani, P, "Performance comparison of IPv4 and IPv6 on various Windows Operating Systems," in *Computer and Information Technology, 11th International Conference*, 2008, pp. 663-668
17. RFC 3973 Protocol Independent Multicast – Dense Mode (PIM – DM): Protocol Specification (Revised), January 2005
    http://www.ietf.org/rfc/rfc3973.txt
18. RFC 4601 Protocol Independent Multicast - Sparse Mode (PIM – SM): Protocol Specification (Revised), August 2006

http://www.rfc-editor.org/rfc/rfc4601.txt

19. PIM – SM Multicast Routing Protocol
    http://technet.microsoft.com/en-us/library/bb742462.aspx
20. RFC 2460 Internet Protocol, Version 6 (IPv6), December 1998
    http://www.ietf.org/rfc/rfc2460.txt