ABSTRACT

STUDY OF THE PERFORMANCE OF INTERNET PROTOCOL VERSION 4 AND INTERNET PROTOCOL VERSION 6 IN A CLIENT SERVER MODEL

By

Manu Sharma

December 2013

The ever increasing demand for Internet connected devices has lead to the situation where we are running out of the Internet Protocol (IP) addresses. The proposed solution to this problem is an advance from IPv4 to the IPv6 Internet Protocol.

Several studies are in place, which predict that with IPv6 in place, every device on our planet will get an IP address, but the overhead caused by IPv6's larger packet size and larger IP addresses, will lead to an extra burden on the intermediate devices between client and server. This may result in increased network congestion and hence may lead to poor overall network performance.

In this thesis, we investigate the performance difference between IPv4 and IPv6 in a realistic, small-scale client server network model. A network traffic generator is used to generate realistic traffic across the network, and the attention is given to maintaining consistent parameters across all the scenarios.

STUDY OF THE PERFORMANCE OF INTERNET PROTOCOL VERSION 4 AND INTERNET PROTOCOL VERSION 6 IN A CLIENT SERVER MODEL

A THESIS

Presented to the Department of Computer Engineering and Computer Science

California State University, Long Beach

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Computer Science

Option in Computer Science

Committee Members:

Tracy Bradley Maples, Ph.D. (Chair)
Burkhard Englert, Ph.D.
Min He, Ph.D.

College Designee:

Antonella Sciortino, Ph.D.

By Manu Sharma

B.E., 2006, Acharya Institute of Technology, Bangalore

December 2013

ACKNOWLEDGMENTS

I would like to thank my parents, Late Dr. Ajit Kumar Sharma and Shashi Sharma, my sister Shilpa Mudgal and my brother-in-law Anupam Mudgal, who have patiently supported me throughout graduate school and whose love and support have encouraged me while working on this research work. Without their support and encouragement, I would have never fulfilled my dream of a higher education in the United States.

I would like to thank Dr. Tracy Bradly Maples, my Thesis Advisor. Her breadth of experience has helped me to achieve my goal. Dr. Maples has also been my mentor throughout the master's program.

I also want to thank my thesis committee, Dr. Burkhard Englert and Dr. Min He, for providing me their time and valuable suggestions.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

LIST OF TABLES

TABLE Page

LIST OF FIGURES

FIGURE Page

FIGURE Page

CHAPTER 1

INTRODUCTION

A collection of computers connected through copper wire, fiber optics, microwaves, infrared or some other medium that enables them to share information is known as a computer network. These computers are devices capable of communicating with each other over a network. Some examples of such devices are: desktop computers, laptops, tablets, smart phones, etc. Nowadays computer networks can be found anywhere—in homes, offices, coffee shops, universities, etc. With the need for computer communication constantly increasing, the demand for these networks is also growing rapidly.

To enable computers to communicate with each other we must follow a set of predefined rules known as protocols. Among these protocols, the one that glues the whole Internet together is Internet Protocol (IP). The job of IP is to provide a best-effort method to transport datagrams from source to destination without regard to whether these machines are on the same network or are connected with the help of other networks.

There are several existing versions of the Internet Protocol. The one that is most widely used across the world is Internet Protocol Version 4 (IPv4). IPv4 is the backbone of the Internet infrastructure. Due to the increasing number of online users over the years, the maximum number of possible IP addresses that can be assigned to the users has been reached. The available IP addresses may be exhausted at any moment in the near future.

The years 2011 to 2012 were predicted to be when IP addresses would be finally unavailable for further assignment [1]. The cause for this shortfall is the limited address space (i.e., 32 bits, equal to 2^{32} addresses), that is, approximately 4.3 billion addresses that are administered globally. As a solution for this problem, the Internet Engineering Task Force's (IETF) has proposed a new version for the Internet Protocols: IPv6. This new version promises many improvements including increased address space from 2^{32} to 2^{128}, enhanced security, enhanced end-user benefits, mobility support and integrated Quality of Service (QoS).

Theoretically, some features in IPv6 may lead to higher performance costs. The header size in IPv6 is twice that in IPv4, which implies huge overhead and performance degradation. This increase in overhead is proportional to data payload size. For example, with no fragmentation of datagrams, a payload of 128 bytes has performance difference of 10.3%, and for payload of 1408 bytes the difference is approximately 1.3%. Another important detail to consider here is that the performance difference between IPv4 and IPv6 also depends on the operating system being used and the application generating traffic.

Most of the research relating to IPv4 and IPv6 compares the versions using various application layer protocols on 32-bit versions of Windows [2]. The 64-bit version of Windows performs better than 32-bit version because of the newly available 64-bit addressing scheme. With the 64-bit addressing scheme in place, computers are not restricted to 4 Gigabytes (GB) of random-access memory (RAM); instead, they can address up to 192 GB of RAM. And with 64-bit registers available to processors, 64-bit Windows performs better on 64-bit processors. Therefore, the latest 64-bit version of

Microsoft Windows 7 Professional Edition is selected to conduct all the experiments in this study.

Objective of the Thesis

The primary objective of this thesis is to study the impact on performance that will occur should be expected when Internet Protocol version 4 is replaced by the Internet Protocol version 6. This goal will be achieved by connecting a set of computers using a router using a client server model. These computers will be configured in such a way as to keep the test-bed parameters consistent throughout the experiments. Traffic will be generated using the Distributed Internet Traffic Generator (D-ITG) tool to match the network traffic normally generated by the software we use in day-to-day life.

CHAPTER 2

NETWORKS AND INTERNET PROTOCOLS

Introduction

Networks allow multiple computers to communicate together over a shared communication channel, and the Internet Protocol regulates the format of data transmitted between computers. This chapter examines the networks and protocols in detail.

Types of Networks

Networks can be classified into different types depending on the geographical area they span. The common types of networks are discussed here.

Local Area Network (LAN)

A Local Area Network normally exists within a single organization. An organization typically lets their computers communicate on the same LAN. A single LAN can also exist on a floor or in an entire building—primarily, it all depends on the total number of computers connected together in a LAN, how frequently they communicate with each other and the kind of the performance is expected from the network. LANs are typically connected via Layer 2 switches.

Metropolitan Area Network (MAN)

A MAN can normally span a city and is typically owned by an organization or a government body. A typical scenario can be a government body where large number of computers are required to communicate with each other but not with any computer outside the network.

Wide Area Network (WAN)

A WAN is a geographically dispersed collection of special devices known as routers. Routers are required to allow the computers to communicate over similar or different types of networks. The Internet Backbone Networks is the largest WAN, spanning the entire Earth.

Network Performance

Measuring network performance is an extremely complex task. It is complex because of the variety of parameters that are required to evaluate the network performance. Other factors that can affect the network performance are network load, CPU utilization on the communicating computers, hardware performance and software performance.

The following measurable elements are used to conduct this research.

Availability

The initial step required is to measure the network performance to check whether the computers in the network are reachable or not. This can easily be verified using the ping command on any computer (Example – C:\> ping computer name). The output of this command contains the IP address of the destination host, data (in bytes) sent to the destination per request, the time it took (in milliseconds) and the time to its being live.

Response Time

The time it takes for data packets to travel on a network to and from a host is known as response time. Several factors can affect the response time of a host. They include: an overloaded network, faulty network devices, overloaded hosts, network errors and faulty communication mediums. The occurrence of any of these factors mentioned can contribute to high response time.

Network Utilization

Calculating the network utilization between two hosts can be quite complicated in real time, and special software is used to get the most accurate reading for this factor.

Network Throughput

Network throughput is the amount of network bandwidth available at any given moment to a network application. This factor can help network administrators find the bottlenecks in their networks. Example: Assume two client computers with gigabit network interfaces are connected to a gigabit router. This router is now connected to the Internet through an Internet Service Provider that limits the bandwidth to 10 Mbps. Now when clients try to access the remote computer over the Internet, the 10 Mbps Internet line will become the bottleneck in this scenario.

Bandwidth Capacity

Bandwidth capacity is the total amount of bandwidth available between two network hosts. The Transmission Control Protocol (TCP) and Internet Protocol (IP) are the commonly used protocols in networks such as in the World Wide Web (WWW), Local Area Networks (LANs) and so on. The performance of the network should therefore be gauged by focusing on the TCP/IP suite.

Internet Protocols

Internet Protocol (IP) is a protocol that defines the way data is sent among computers over the Internet. The data sent over a network is first broken down into packets, which are later forwarded to the gateways; eventually, these packets arrive at the destination computer. IP delivers the packets in any order to the destination, as IP is a connectionless protocol. The reason packets get delivered in order to the destination in the real world is due to the Transmission Control Protocol (TCP), which puts packets back in the correct order. The most widely used IPs today are IPv4 and IPv6. IPv6 provides 64-bit addresses and therefore supports a significant amount more Internet users in comparison to IPv4.

<u>IPv4 versus IPv6</u>

Explanations from the literature indicate that IPv4 networks have better performance than IPv6 networks. The main reason for this lies with the differences between the structures of packets of these two protocols. IPv6 design was not an idea made from the very beginning; instead, it was made with the idea to add major capabilities to it over IPv4. IPv6 eliminates or makes optional some of the IPv4 header fields to reduce the packet-handling overhead, which provides some compensation for the larger address [3]. Although IPv6 reduces some fields in the packet header, the addresses used by IPv6 are four times longer than the addresses used by IPv4—making IPv6 packet headers twice the size of IPv4 packet headers (see Figure 1). Figure 1 shows the details of the IPv4 and IPv6 packet header structures.

Unlike IPv4, IPv6 headers do not contain any options field (see Figure 2). The capabilities that the variable-sized option field offered in IPv4 are now deployed by a chain of extension headers that follow IPv6 basic header [3]. Each IPv6 packet is made up of a packet header, one or more extension headers and data. Each extension header is identified by the 'next header' field of the preceding header, and this has a fixed length

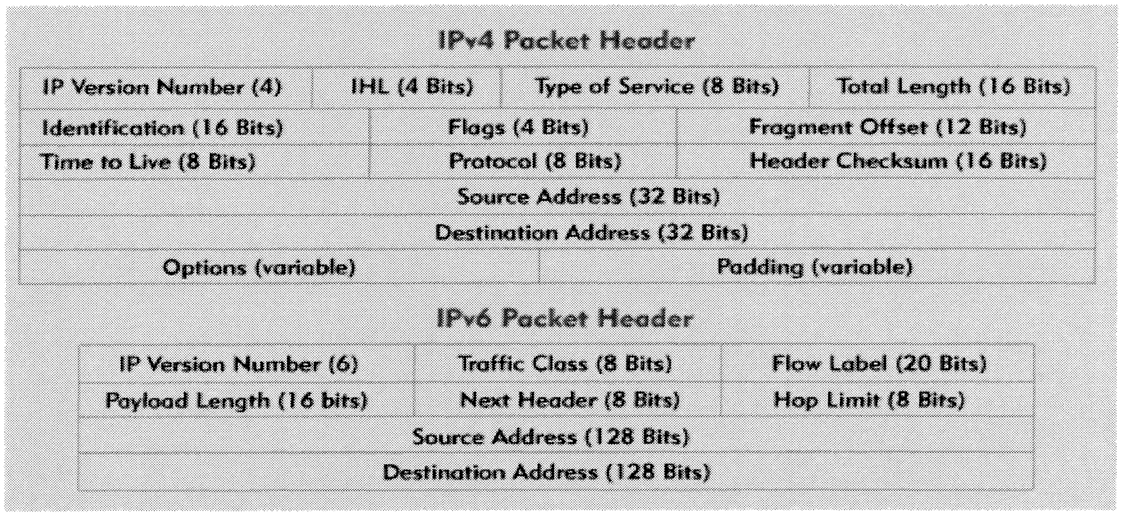

FIGURE 1. IPv4 and IPv6 packet header [4].

and particular capability. On the other hand, each IPv4 packet is made up of a packet header, options and data. The figure above shows the details of IPv4 and IPv6 packet structures. IPv4 packets have a fixed structure that is the basic header plus variable length data. IPv6 packets have countless types of structures that include the basic header plus one or more extension headers and then the addition of variable length data. In IPv6 packet format, each extension header may be one of ten possibilities and the number of the extension header is variable, therefore the number of IPv6 packet structure is enormous [3].

In IPv6 networks, because of the characteristics of IPv6 packets, the packets need more time to transmit through the network. For that reason, Zeadally and Raicu (2003) indicate that IPv6 might solve several of IPv4's shortcomings, but the longer headers and address space add overhead that affects a range of performance metrics for both TCP and UDP [5]. Furthermore, Visoottiviseth and Bureenok (2008) also point out that IPv4

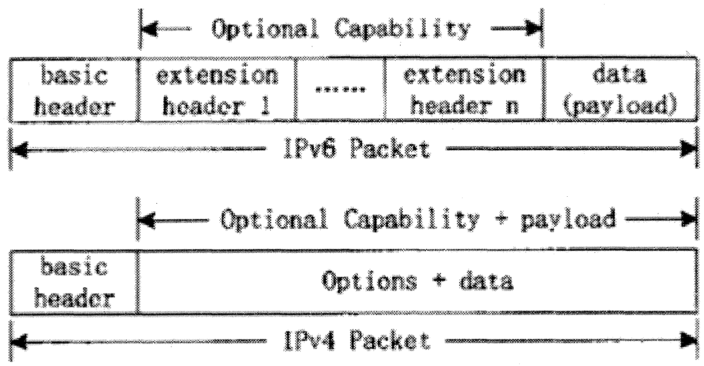

FIGURE 2. IPv4 and IPv6 packet structure [9].

yields the highest data throughput in both transmissions via TCP with no-delay option and UDP, followed by IPv6 [6].

In conclusion, because of the characteristics of the IPv6 packet, IPv4 shows better network performance than IPv6; however, IPv6 brings a lot of new features which do not exist in IPv4.

CHAPTER 3

EXPERIMENTAL SETUP

In this section, we explain the details of the network test-bed used in this thesis which includes hardware and software specifications, network design, the network layer and experimental tasks.

A simple experimental setup to observe and compare the performance differences between IPv4 and IPv6 in the real world is a client computer connected to a server over the Internet as shown in Figure 3. However, the computers in this setup—client and server—are connected to the Internet though ISPs. The ISPs regulate the bandwidth between a computer and the Internet depending on a typical data plan selected by a user.

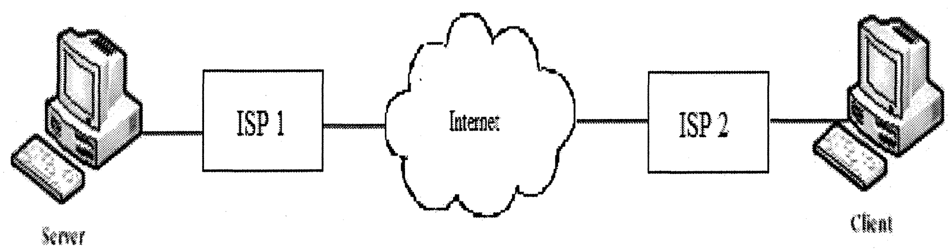

FIGURE 3. Possible real world connection between client and server.

The primary goal in this experiment is to observe the network performance for both protocols, IPv4 and IPv6, which can easily be affected by the bandwidth assigned by the ISP to connecting the computers to the Internet. The graph in Figure 4, shows the

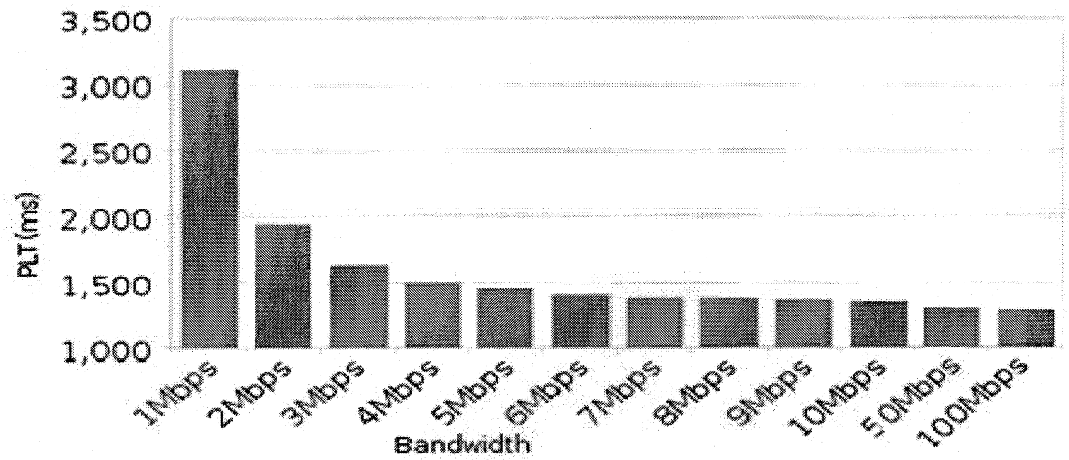

FIGURE 4. Latency per bandwidth [7].

affect of varying bandwidth on the page load time (PLT) [7]. From this graph one can observe that upgrading bandwidth from 1 Mbps to 10 Mbps in increments of 1 Mbps of speed at a time inversely affects the PLT. From the same graph we can assume that at high allotted bandwidth (>= 100 Mbps) latency becomes almost constant irrespective of the increasing bandwidth, as the difference among latencies almost diminishes for high bandwidth values. As client and server can connect to the Internet through the same or different ISPs, the performance parameters for the overall network can also vary depending on the bandwidth assigned by ISPs. Businesses in the United States have access to Internet speed >= 100 Mbps while the average Internet speed for home users average about 6~7 Mbps [7]. Therefore, in order to evaluate the real performance difference between IPv4 and IPv6, bandwidth is set to the constant value of 100 Mbps for this experiment.

Hardware Specification

In this experimental design, two computers of the hardware configuration as shown in Table 1 are connected using a Linksys EA 4500 router. The Linksys router operates at 100 Mbps in this experiment. This router belongs to the new generation of routers made by Linksys, which supports both, IPv4 and IPv6 configurations. A unique feature in this router is that it can be configured for Quality of Service, that is, traffic prioritization for music, voice and video.

Software Specification

Both the computers are installed with Windows 7, 64-bit Professional Edition, and the required device drivers. Once the computers were prepared, they were connected to the Internet to receive all updates released by Microsoft and hence kept current.

TABLE 1. Experimental Hardware Specification

Hardware	Detail
CPU	Intel Core 2 Duo @ 2. 0 GHz 3MB cache
Hard Drive	250 Gb SATA 7200 rpm
Memory	4GB DDR2 800Mhz
Network Card	PCI Intel Pro 1000
Router	Linksys EA 4500

Network Design

The network used in this experiment contains two computers connected by a Linksys EA 4500 router. The job of server is acting as sender, responding to sent data, decoding log files and recording results while the client acts as a receiver, responding to received data. Figure 5 shows the network design of this experiment.

The computers are then configured first for IPv4 and data is collected. Later this setup is repeated with IPv6, ensuring the test-bed parameters will remain the same. The traffic is generated using TCP, UDP, Games (Counter-Strike and Quake 3), VoIP using following codec: G.711.1, G.711.2, G.723.1, G.729.2 and G.729.3.

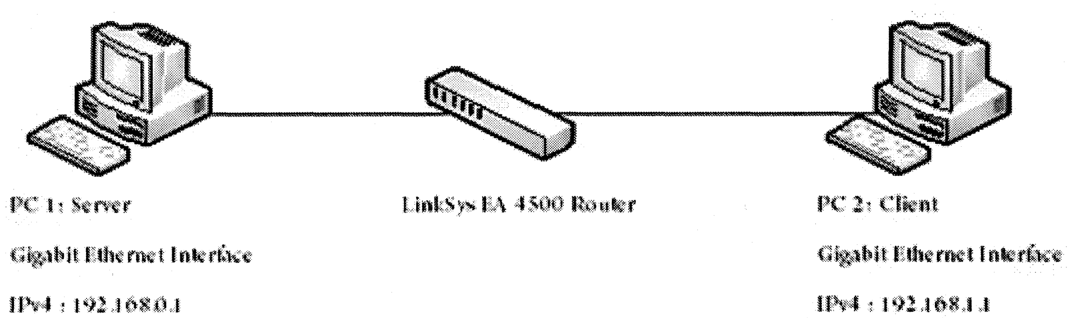

FIGURE 5. Network design.

Packet Sizes

In order to gain a wide range of data for Windows network performance in TCP and UDP, 13 different packet sizes were chosen for measurement. These packet sizes are represented in bytes as follows: 64, 128, 256, 384, 512, 640, 768, 896, 1024, 1152, 1280, 1408 and 1536.

Test Duration and Data Gathering

The duration of each task is set at 60 seconds. In order to maintain objectivity in the results and to increase accuracy in the findings, each task is run 15 times. Results are taken as the average of these 15 results.

Network Performance Measuring Tool

A network performance measuring tool called D-ITG (Distributed Internet Traffic Generator) is adopted for the experiment. D-ITG is a platform capable of producing traffic at packet level accurately, and replicating appropriate stochastic processes for both IDT (Inter Departure Time) and PS (Packet Size) random variables (exponential, uniform, Cauchy, normal, Pareto, etc.). D-ITG supports both IPv4 and IPv6 traffic generation and it is capable of generating traffic at network, transport, and application layer [8].

The latest stable version of D-ITG is 2.8.1. This version of D-ITG is used in the experiments. The multiplatform source code and binary files of D-ITG for Windows can be downloaded from the official website. Normally, D-ITG runs under command line mode in Windows, and the software contains five executable files; they are ITGSend, ITGRecv, ITGLog, ITGDec and ITGManager.

In the experiment, ITGSend, ITGRecv, ITGLog and ITGDec commands are mainly used to obtain raw data. The following section explains the details of these commands that how they are used in the experiment.

ITGSend

ITGSend is a sender component of the D-ITG platform. The script mode enables ITGSend to simultaneously generate several flows. Each flow is managed by a single thread, with a separate thread acting as a master and coordinating the other threads. To generate n flows, the script file has to contain n lines, each of which is used to specify the characteristics of one flow. Each line can contain the options, but those regarding the logging process (-l, -L, -X, -x). Such options can be specified at the command line and refer to all the flows [9].

ITGSend is used to generate and send specific traffic to the destination; the following options in ITGSend component are used in this experiment:

1. -a: Set the destination IP address, for example: 192.168.1.1.

2. -C: Constant inter-departure time (IDT). Set the number of packets sent per second. In this experiment, IDT with 30000 packets per second is assumed.

3. -c: Constant payload size. In this experiment 13 different packet sizes are chosen (64, 128, 256, 384, 512, 640, 768, 896, 1024, 1152, 1280, 1408, and 1536).

4. -m: Set the type of meter, two values are allowed: owdm (one way delay meter) and rttm (round trip time meter). In this experiment, round trip time meter is used in all time measurements.

5. -T: Set the protocol type. Valid values are UDP, TCP, ICMP, SCTP and DCCP. In this experiment, TCP and UDP are adopted.

6. -t: Set the generation duration. It is expressed in milliseconds (ms). In this experiment, the duration is 60000 ms.

7. VoIP: Generate traffic with VoIP traffic characteristics. In this experiment, five different codec types of VoIP are chosen (G.711.1, G.711.2, G.723.1, G.729.2, and G.729.3), with option -x is able to set the codec type and all VoIP traffic with real time protocol type is assumed. Codecs G.711, G.723 and G.729 are selected in particular to be used in this experiment as they are the most popular codecs in use. G.711 offers bit rate of 64 Kbps and does not compromise on the voice quality. The downside of this codec is that it consumes most bandwidth. G.723 has a bit rate of 6.3 Kbps. It uses Algebraic Code Excited Linear Prediction encoding technique for converting the voice signal from an analog to a digital signal. It uses most time to process the signal. G.729 has a bit rate

of 8 Kbps. This is most widely used codec in VoIP networks. It offers the best of the two codecs mentioned above, that is, it uses only the optimal amount of bandwidth and yet offers a good voice quality [10] (see Figure 6).

8. CSa: Generate gaming traffic with Counter-Strike traffic characteristics related to the active phase of the game.

9. Quake3: Generate gaming traffic with Quake III Arena traffic characteristics.

<u>ITGRecv</u>

ITGRecv is a receiver component of the D-ITG platform. It can receive data flows from different senders [8]. In the experiment, ITGRecv runs on the destination computer to receive data flow from the sender.

<u>ITGLog</u>

ITGLog is the log server of the D-ITG platform. It receives log information from ITGSend sender and the ITGRecv receiver. It listens on ports dynamically allocated in the range [9003-10003] [8]. In the experiment, ITGLog generates a log file after each test run that contains information pertaining to the experiment.

<u>ITGDec</u>

The ITGDec decoder is the utility to analyze the results of the experiments conducted by using the D-ITG generation platform. ITGDec parses the log files generated by ITGSend and ITGRecv and calculates the average values of bit rate, delay and jitter either on the whole duration of the experiment or on variable-sized time intervals. You can analyze the binary log file only on the operating system used to create that file. You can use another operating system if the log file is in text receiving time of last and first packet [8].

D-ITG Codecs for VoIP packet generator

Codec	samples	framesize	packets (per sec)
G.711.1	1	80	100
G.711.2	2	80	50
G.729.2	2	10	50
G.729.3	3	10	33
G.723.1	1	30	26

Packetsize is calculated according to the formel:

Packetsize = Random(VAD * framesize * samples * RV * header)

where:
VAD = 1; without Voice Activation Detection (VAD)
VAD = 0,65; with VAD flag set (VAD activated)
RV is random variable that is constant (1) for VoIP packet generator
header = 12; per default,
header = 8; for RTP, and
header = 2; for CRTP (compressed RTP)

Packets interarrival is calculated according to the formel:

IntArriv = Random(1000/packets * RV)

FIGURE 6. D-ITG codecs for VoIP packet generator [11].

In the experiment, ITGDec runs on sender computer to decode the log files that were generated by ITGLog. All raw data are decoded by ITGDec into the readable text files.

The sample commands of D-ITG used in this experiment are available in Appendix A.

Packet Rate

The packet rate is the number of packets that the sender is able to send per second. In the case of local implementation, we observe a negligible error rate for D=30 and required packet rate close to 28000 pkt/s. The error rate is about 5% for a packet rate close to 30000 pkt/s. We can therefore consider an optimal value of 30, which is related to a maximum achieved packet rate of 28000 pkt/s. In the case of the other implementations, it is easy to draw an optimal D value of 40, in correspondence of a maximum achieved packet rate of 30000 pkt/s [12]. Therefore, in this experiment the packet rate is 30,000. Thus, in each run, ITGSend sends 30,000 packets per second. In order to ensure that the packet rate is correct, a test is made before the experiment starts.

This test runs under IPv4 and uses 64 bytes as its packet size to test UDP performance. Five packet rates, 10,000; 20,000; 30,000; 40,000 and 50,000, are tested to determine the throughput and packet loss rates. Each task runs for 60 seconds. The following commands are used in the sender computer:

1. Itgsend -a 192. 168. 1. 1 -m rttm -T UDP -C 10000 -c 64 -t 60000
2. Itgsend -a 192. 168. 1. 1 -m rttm -T UDP -C 20000 -c 64 -t 60000
3. Itgsend -a 192. 168. 1. 1 -m rttm -T UDP -C 30000 -c 64 -t 60000
4. Itgsend -a 192. 168. 1. 1 -m rttm -T UDP -C 40000 -c 64 -t 60000
5. Itgsend -a 192. 168. 1. 1 -m rttm -T UDP -C 50000 -c 64 -t 60000

The results are shown below:

TABLE 2. Result of Packet Rate

Packet Rate (pkt/s)	10000	20000	30000	40000	50000
Throughput (Mbps)	5. 000019	9. 998891	14. 883058	14. 225140	13. 220129
Packet Loss (%)	0. 00	0. 01	0. 78	27. 88	42. 54

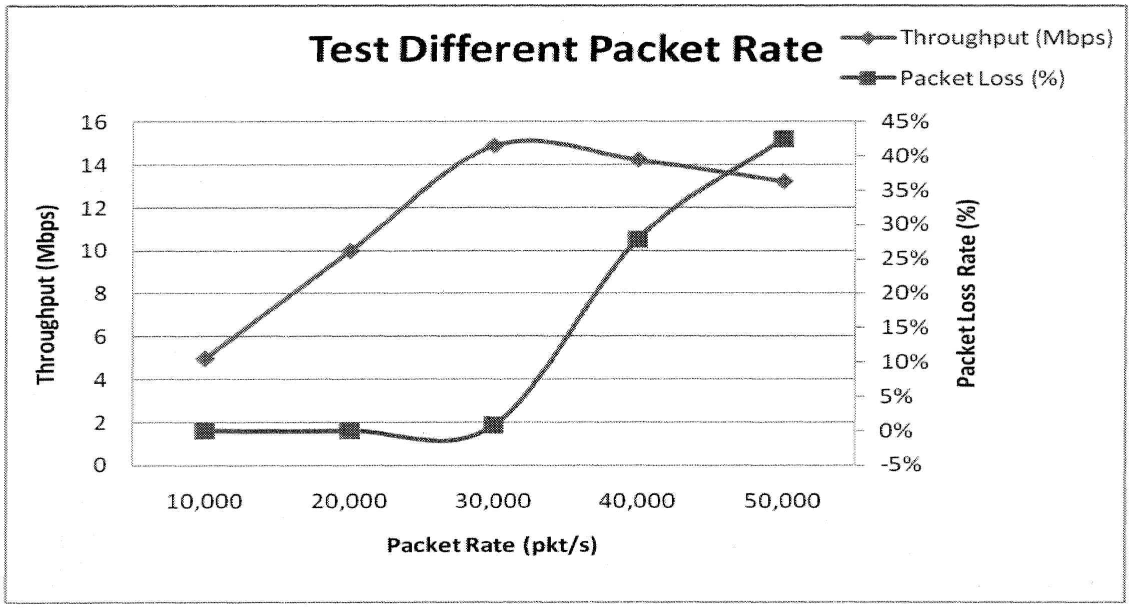

FIGURE 7. Test of different packet rates for UDP data traffic.

Figure 7 indicates that the respective packet loss of packet rates 10,000; 20,000 and 30,000 are close to zero. But the loss rate increases sharply to nearly 28% at packet rate 40,000, and it reaches 43% in packet rate 50,000. These high packet loss rates render the results for the two top packet rates useless, because transfer at these packet rates is not able to reach maximum throughput. Thus, although the packet rates 40,000 and 50,000 send 33% more and 66% more packets per second respectively than packet rate 30,000, their maximum throughputs are less than that of the packet rate 30,000. On the other hand, although packet rates 10,000 and 20,000 have nearly zero packet loss rates, their maximum throughput values are less than that of the packet rate 30,000. Therefore, considering the balance of throughput and packet loss rate, the packet rate 30,000 is the best choice in this experiment. The researcher notes that this rate is similar to the rate that Emma, Pescape and Ventre (2004) mentioned in their study [12].

Measurements

In order to evaluate network performance, the parameters throughput, jitter and round trip time are used in this experiment. The following section describes the details of these measurements.

Throughput

The throughput of a network represents the amount of network bandwidth available for an application at any given moment across the network links [13]. In this experiment, the unit of throughput is represented in megabits per second (Mbps) and kilobits per second (Kbps).

Jitter

Jitter represents how variable latency is in a network. It is the variation in the time between packets arriving. Jitter is typically caused by network congestion, timing drift, or route changes [14]. Perez, Zarate, Montes and Garcia (2006) mentions that higher jitter can result in both increased latency and packet loss, and it is recommended that jitter should not exceed 50 milliseconds [15]. Jitter is always expressed as a positive number, so the absolute value of the difference is always used. Average jitter is the average value of the jitter of consecutive packet pairs as shown in Figure 8. In this experiment the unit of jitter is represented in millisecond (ms). D-ITG calculates jitter according to the formula in Figure 8.

Round Trip Time (RTT)

On the network, the round trip time is the time that a packet takes to travel between the source node and destination node. In this experiment, the unit of round trip time is represented in milliseconds (ms). D-ITG calculates the standard deviation for

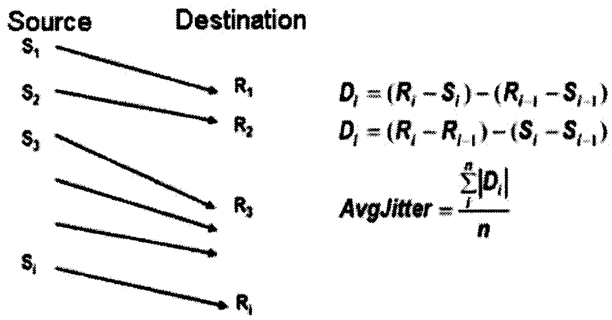

FIGURE 8. Jitter formula [9].

delay according to the following equation. N is the number of packets considered, d_i is the delay of packet i, and $\hat{d}$ is the average delay of packets.

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left(d_i - \hat{d}\right)^2}$$

FIGURE 9. Round trip time equation [9].

IPv6 Configuration

This section shows how IPv6 was installed and configured in Microsoft Windows network applications used in the experiment. To set up an IPv6 test environment, three important things are needed: router, an IPv6-supported operating system and a IPv6 supported network measuring tool.

In this experiment, the IPv6-supported network measuring tool D-ITG was downloaded from the official website.

In Windows 7, IPv6 appears as the Internet Protocol Version 6 (TCP/IPv6) component on the Networking tab when viewing the properties of "Local Area Connection" in the Network Connections folder (available from the Network and Sharing Center). The next step is to select the Internet Protocol Version 6 (TCP/IP), clicking properties (see Figure 10). Now IPv6 address is able to be manually placed into address field when the heading of "use the following IPv6 address" is selected (see Figure 11).

After entering the IPv6 address and clicking OK, the IPv6 address will have been assigned to the operating system (see Figure 12).

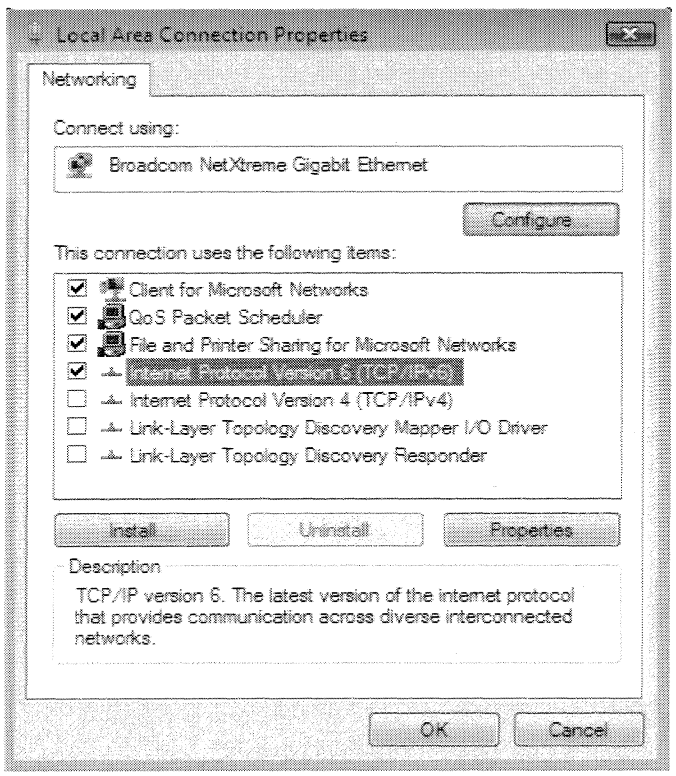

FIGURE 10. IPv6 component in Windows 7.

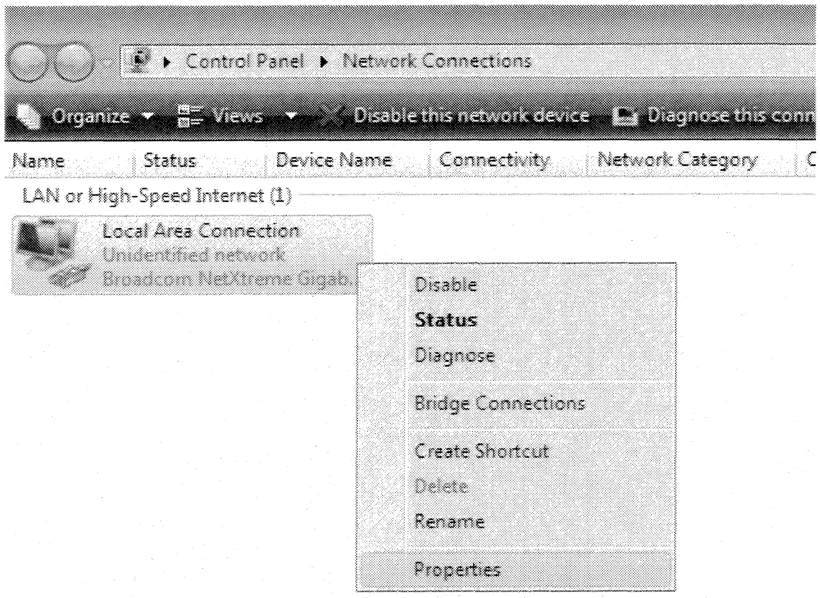

FIGURE 11. Obtaining properties of Local Area Connection.

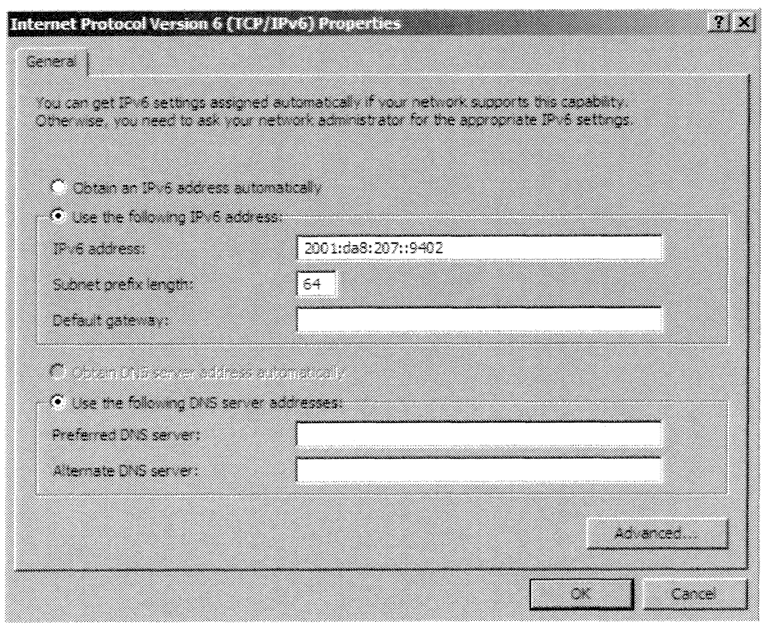

FIGURE 12. IPv6 properties in Windows 7.

CHAPTER 4

EXPERIMENTAL RESULTS

The results obtained from all the performed experiments are available in Appendix B. These results are used to make line charts and bar charts used in this chapter.

TCP Analysis

This section analyses TCP performance in the Microsoft Windows 7 operating system, which includes throughput, round trip time and jitter.

TCP Throughput

The line chart in Figure 13 shows the TCP throughput with different packet sizes in both IPv4 and IPv6 networks.

From this line chart, the following observations can be made:

1. At packet sizes between 64 bytes and 384 bytes, the difference in throughput is less than 0.2%. Therefore, between packet sizes of 64 bytes and 256 bytes, the TCP throughput is similar in both IPv4 and IPv6 networks.

2. At packet sizes of 512 bytes and greater, TCP throughput is higher in the IPv4 network than it is in the IPv6 network.

TCP Round Trip Time

The line chart in Figure 14 shows the TCP round trip time with different packet sizes in both the IPv4 and the IPv6 networks.

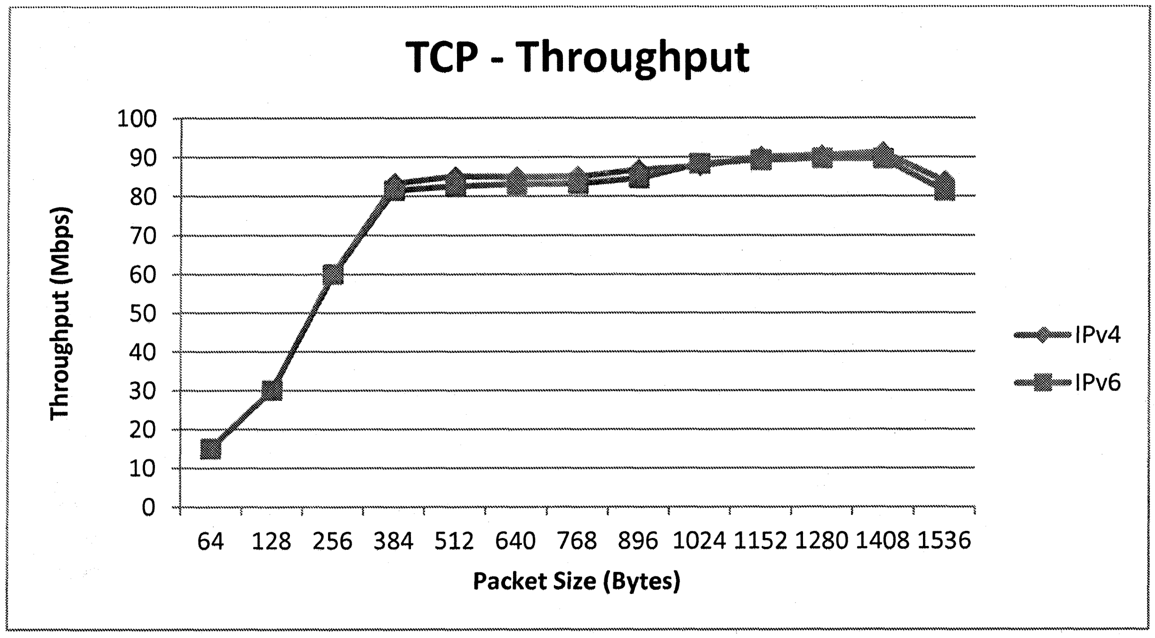

FIGURE 13. TCP throughput for different packet sizes in IPv4 and IPv6 Networks.

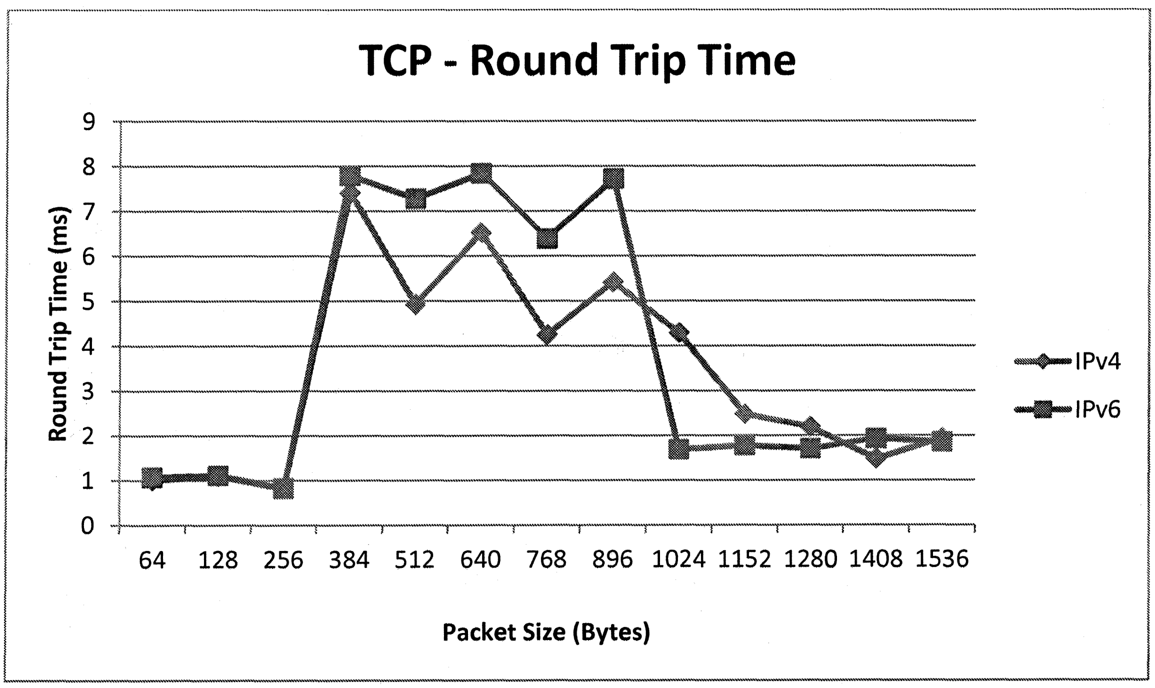

FIGURE 14. TCP round trip time for different packet sizes in IPv4 and IPv6 networks.

From the line chart in Figure 14, the following observations can be made:

1. At packet sizes of 64 bytes to 384 bytes, IPv4 and IPv6 networks have similar TCP round trip time.

2. From packet sizes of 384 bytes to 1024 bytes and 1408 bytes to 1536 bytes, TCP round trip time is higher in the IPv6 Network than it is in the IPv4.

TCP Jitter

The following line chart shows the TCP jitter of different packet sizes in both IPv4 and IPv6 networks.

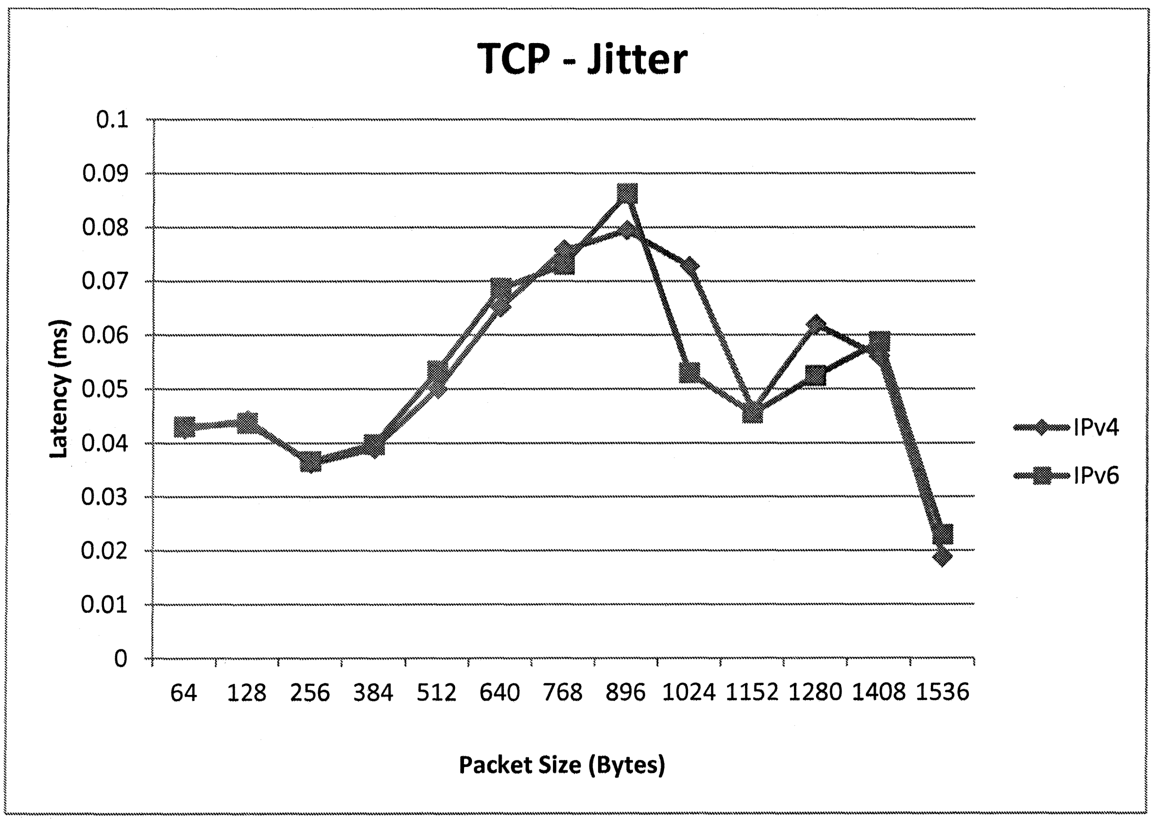

FIGURE 15. TCP jitter for different packet sizes in IPv4 and IPv6 networks.

From the line chart in Figure 15, the following observations can be made:

1. From packet sizes of 64 bytes to 768 bytes and 1408 bytes to 1536 bytes, IPv4 and IPv6 networks have similar TCP jitter.

2. From packet sizes of 896 bytes to 1408 bytes, the IPv6 network has lower jitter than the IPv4.

UDP Analysis

This section analyses UDP performance in Microsoft Windows 7 operating system, which includes throughput, round trip time and jitter.

UDP Throughput

The following line chart shows the UDP throughput with different packet sizes in both IPv4 and IPv6 networks.

From line chart in Figure 16, the following observations can be made:

1. From packet sizes of 64 bytes to 384 bytes, UDP throughput is similar in both IPv4 and IPv6 networks.

2. From packet sizes of 384 bytes to 1536 bytes, the IPv4 network has slightly higher throughput than that of the IPv6 network.

UDP Round Trip Time

The line chart in Figure 17 shows the UDP round trip time with different packet sizes in both IPv4 and IPv6 networks.

From the line chart in Figure 17, the following observations can be made:

1. From packet sizes of 64 bytes to 128 bytes, the IPv6 network has slightly higher UDP round trip time than does the IPv4 network.

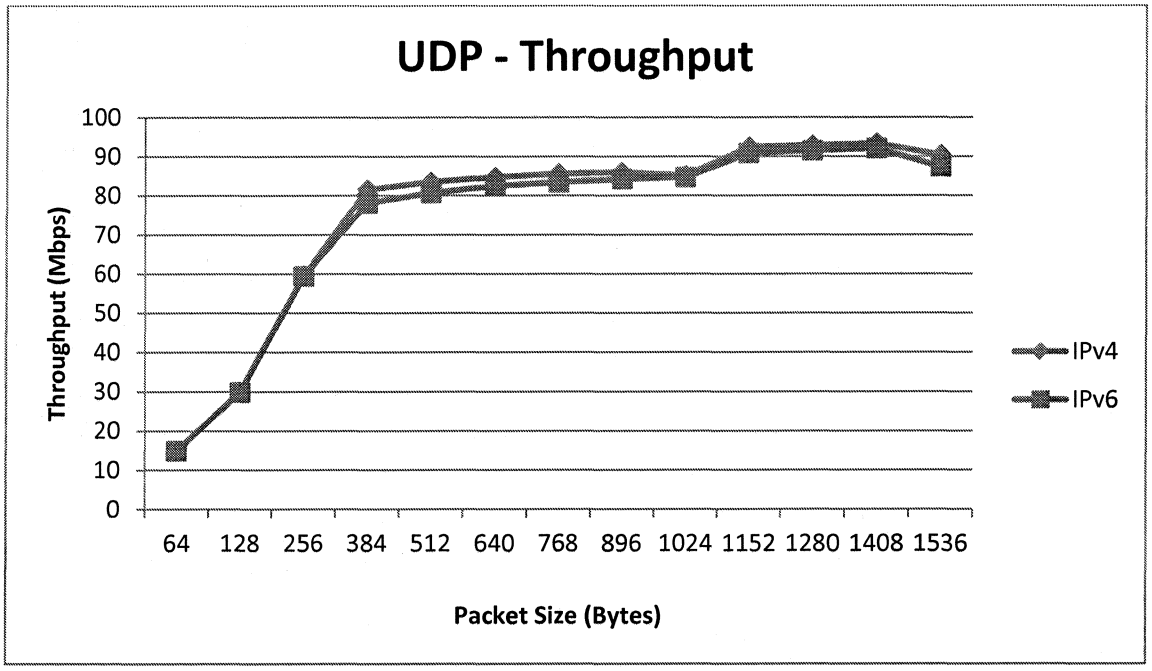

FIGURE 16. UDP throughput for different packet sizes in IPv4 and IPv6 networks.

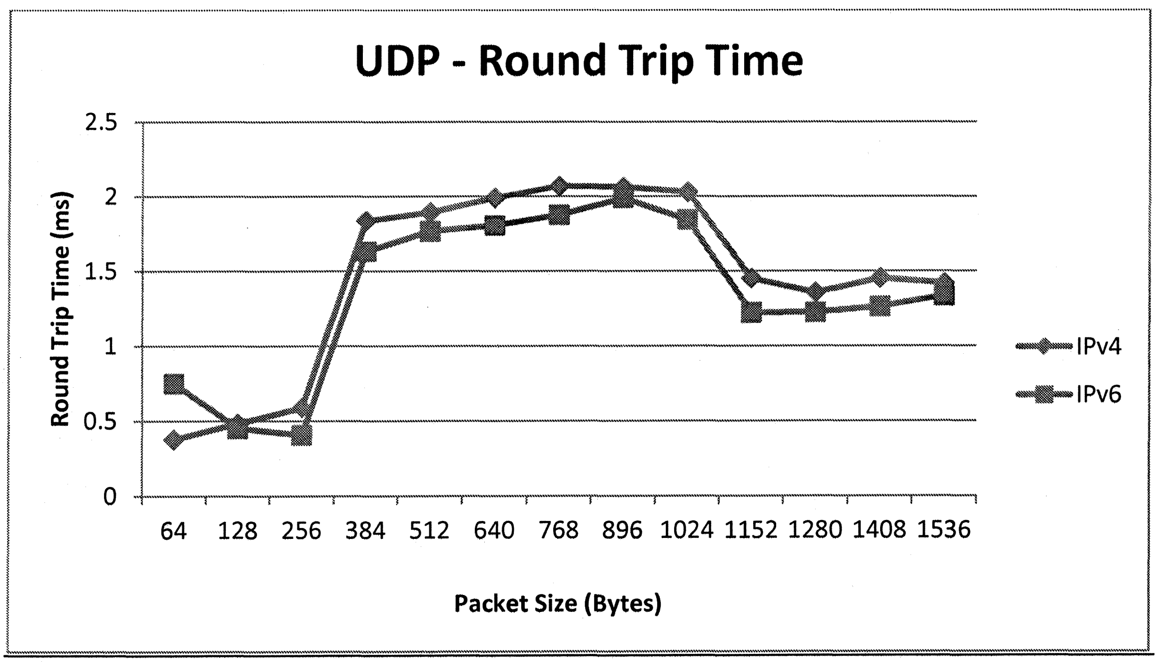

FIGURE 17. UDP round trip time for different packet sizes in IPv4 and IPv6 networks.

2. From packet sizes of 384 bytes to 1536 bytes, UDP round trip time is slightly higher in the IPv4 network than it is in the IPv6 network

UDP Jitter

The following line chart shows the UDP jitter of with different packet sizes in both IPv4 and IPv6 networks.

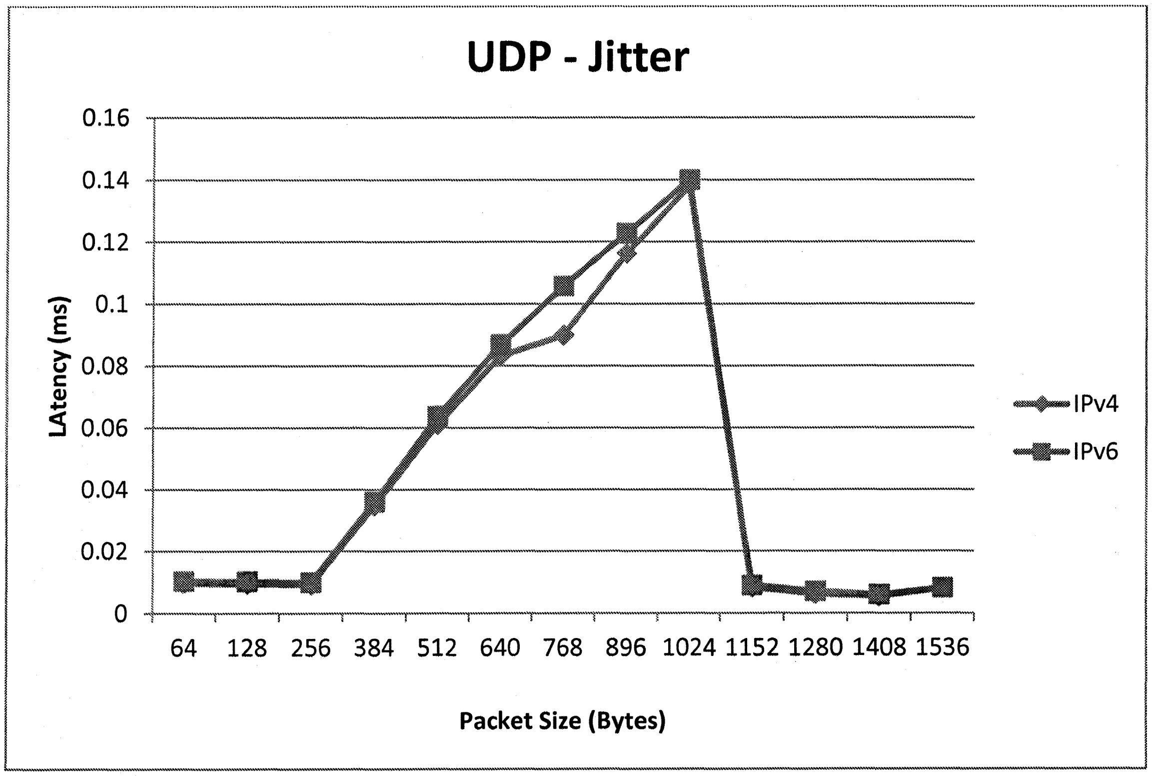

FIGURE 18. UDP jitter for different packet sizes in IPv4 and IPv6 networks.

From line chart in Figure 18, the following observations can be made:

1. UDP jitter is similar in both IPv4 and IPv6 networks.

2. From packet sizes of 1068 bytes to 1152 bytes, the IPv6 network has slightly higher UDP jitter than does the IPv4 networks.

Analysis of Gaming Data

This section analyses network performance of two games in regard to factors including throughput, round trip time and jitter.

Counter Strike Throughput

The following bar chart shows the Counter Strike (game) throughput in both IPv4 and IPv6 networks.

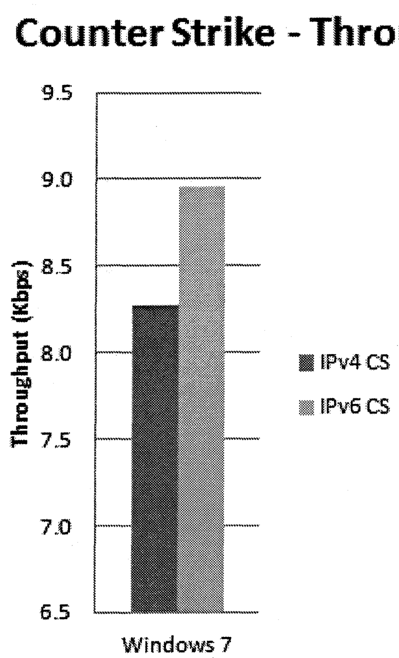

FIGURE 19. Counter Strike throughput.

From the bar chart in Figure 19, the following observation can be made:

1. In the game Counter-Strike, throughput is higher in the case of the IPv6 network in comparison to the IPv4 network.

Counter Strike Round Trip Time

The following bar chart shows the Counter-Strike (game) round trip time in both IPv4 and IPv6 networks.

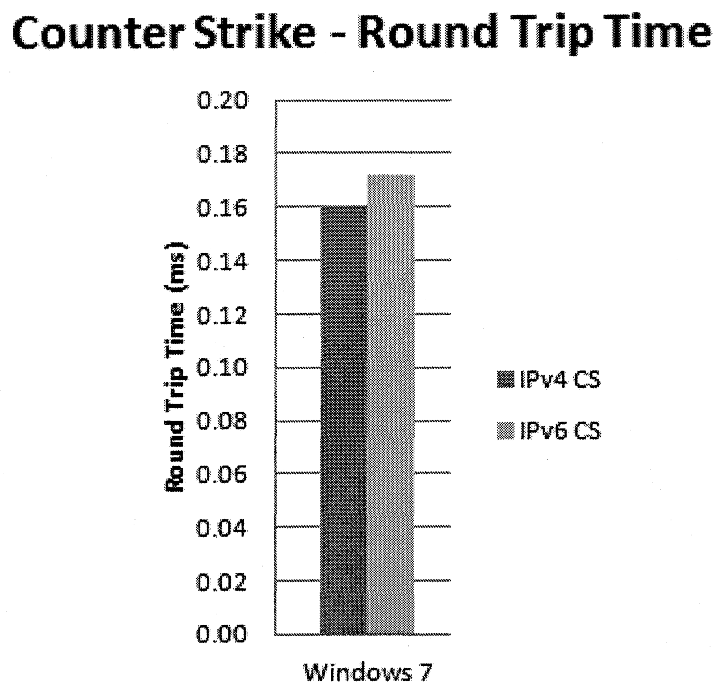

FIGURE 20. Counter Strike round trip time.

From the preceding bar chart, the following observation can be made:

1. Round trip time in the game Counter-Strike is close to same in both networks (difference ~ 0. 01ms).

Counter-Strike Jitter

The following bar chart shows the game Counter-Strike's jitter in both IPv4 and IPv6 networks.

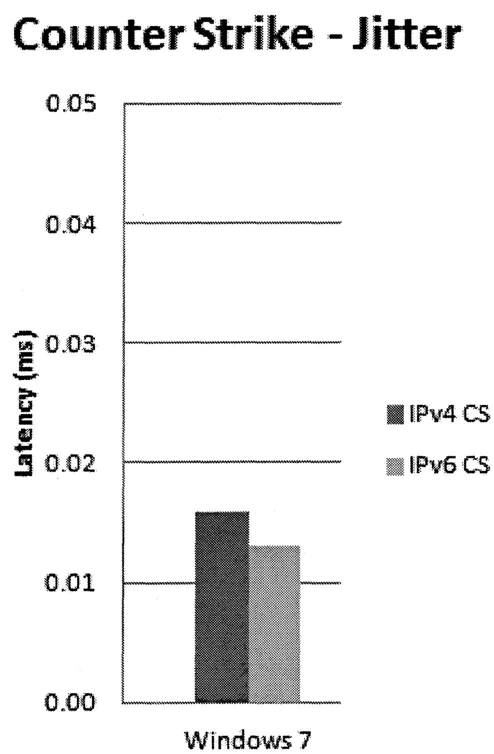

FIGURE 21. Counter Strike jitter.

From the preceding bar chart, the following observation can be made:

1. Jitter is higher in the IPv4 network than it is in the IPv6 network only by a negligible amount (<0. 005ms).

Quake 3 Throughput

The bar chart in Figure 22 shows the game Quake 3's throughput in both IPv4 and IPv6 networks .

From the bar chart in Figure 22, the following observation can be made:

1. In the game Quake 3, throughput in the IPv4 network is higher than that in the IPv6 network.

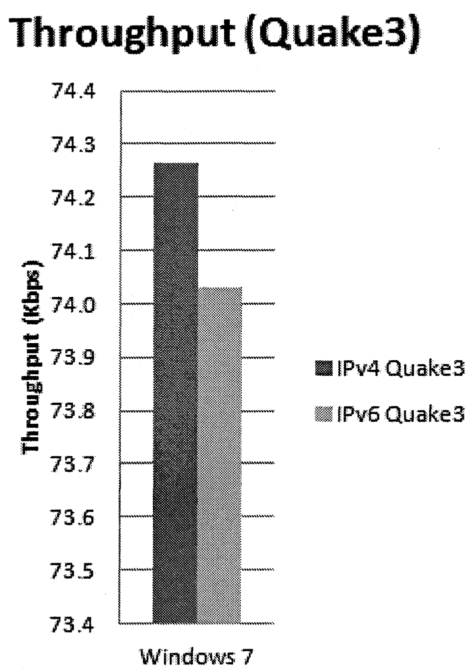

FIGURE 22. Quake 3 throughput.

Quake 3 Round Trip Time

The following bar chart in Figure 23 shows the round trip time for the game Quake 3 in both IPv4 and IPv6 networks.

From the bar chart in Figure 23, the following observation can be made:

1. Round trip time in game Quake 3 is higher in IPv6 network than that of IPv4 network.

Quake 3 Jitter

The following bar chart in Figure 24 shows the Quake 3 (game) jitter in both IPv4 and IPv6 networks.

From this bar chart, the following observation can be made:

1. Jitter in game Quake 3 is similar for both IPv4 and IPv6 networks.

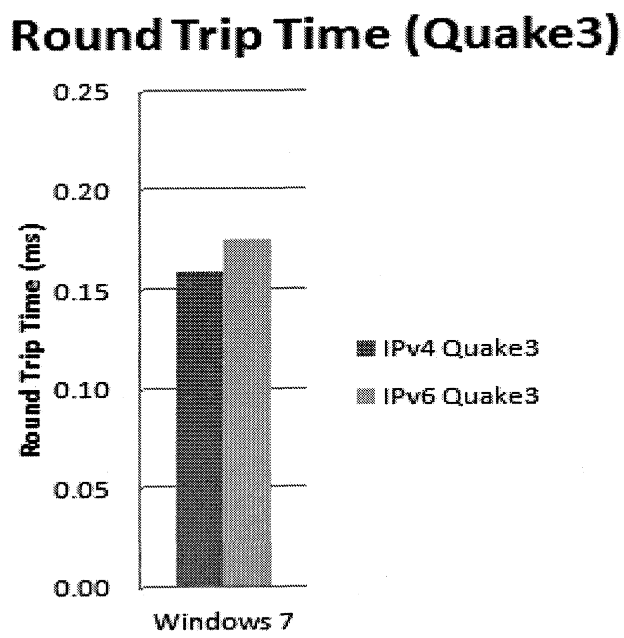

FIGURE 23. Quake 3 round trip time.

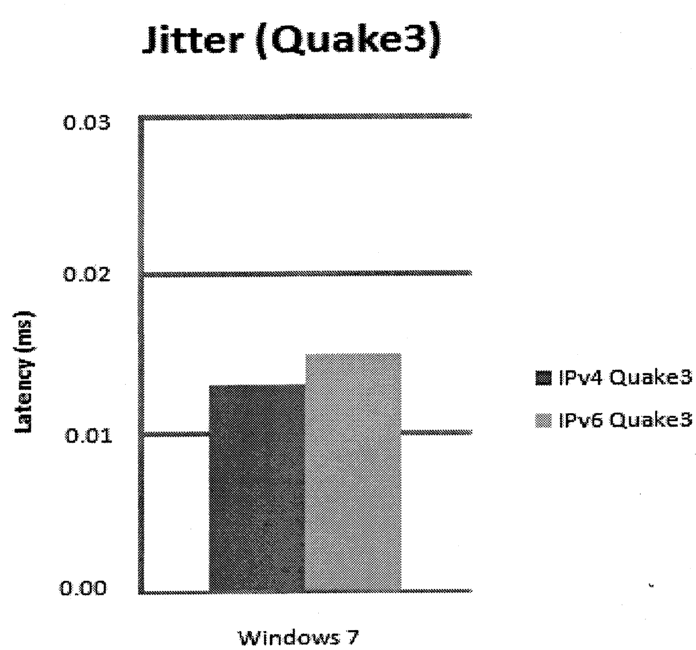

FIGURE 24. Quake 3 jitter.

VoIP Analysis

This section analyses network performance of VoIP in regard to factors including throughput, round trip time and jitter.

VoIP G.711.1 Throughput

The following bar chart shows the throughput of VoIP G.711.1 codec with 1 sample per packet in both IPv4 and IPv6 networks.

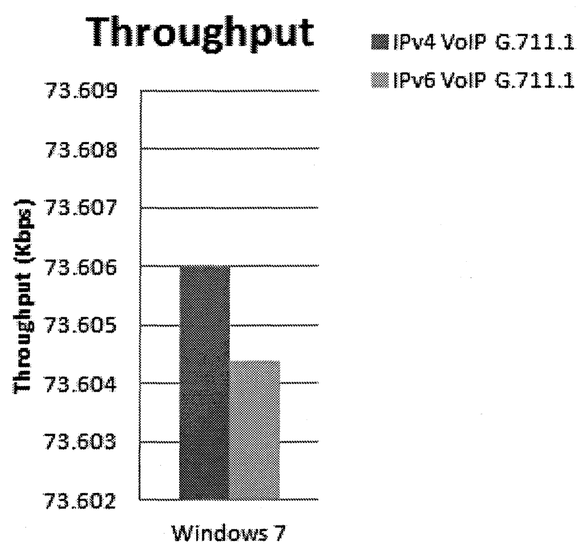

FIGURE 25. VoIP G.711.1 throughput.

From above bar chart, the following observation can be made:

1. The IPv4 network has higher VoIP G.711.1 throughput than does the IPv6 network.

VoIP G.711.1 Round Trip Time

The bar chart in Figure 26 shows the round trip time of VoIP G.711.1 codec with 1 sample per packet in both IPv4 and IPv6 networks.

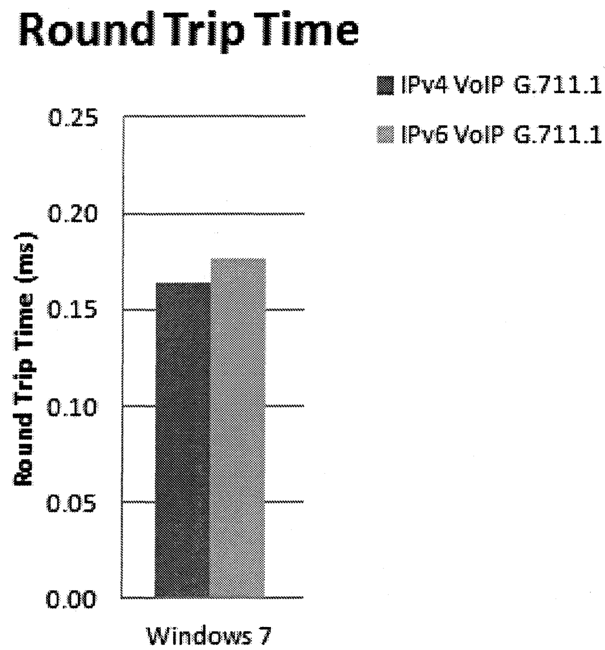

FIGURE 26. VoIP G.711.1 round trip time.

From the above bar chart, the following observation can be made:

1. Round trip time is similar for VoIP G.711.1 in both IPv4 and IPv6 networks.

VoIP G.711.1 Jitter

The bar chart in Figure 27 shows the jitter of VoIP G.711.1 codec with 1 sample per packet in both IPv4 and IPv6 networks.

From bar chart in Figure 27, the following observation can be made:

1. IPv4 and IPv6 networks display similar latency for VoIP G.711.1 traffic.

VoIP G.711.2 Throughput

The following bar chart in Figure 28 shows the throughput of VoIP G.711.2 codec with 2 samples per packet in both IPv4 and IPv6 networks (see Figure 28).

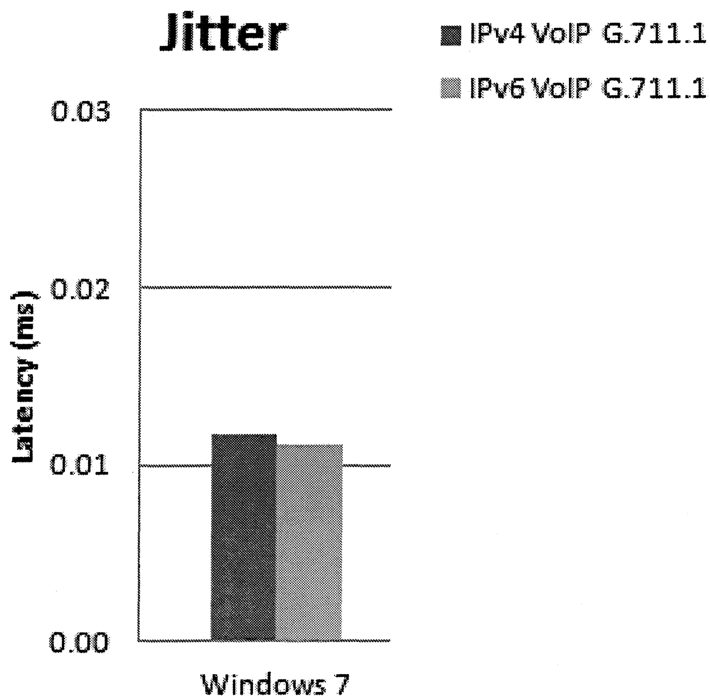

FIGURE 27. VoIP G.711.1 jitter.

From bar chart in Figure 28, the following observation can be made:

1. Throughput is the same in the system in VoIP G.711.2 in both IPv4 and IPv6 networks.

VoIP G.711.2 Round Trip Time

The bar chart in Figure 29 shows the round trip time of VoIP G.711.2 codec with 2 samples per packet in both IPv4 and IPv6 networks (see Figure 29).

From the bar chart in Figure 29, the following observation can be made:

1. The IPv6 network has a higher round trip time than that of the IPv4 network in VoIP G.711.2.

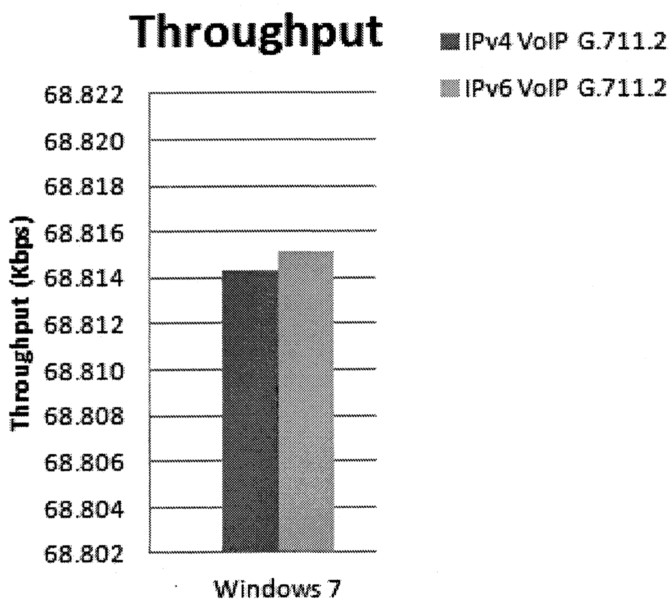

FIGURE 28. VoIP G.711.2 throughput.

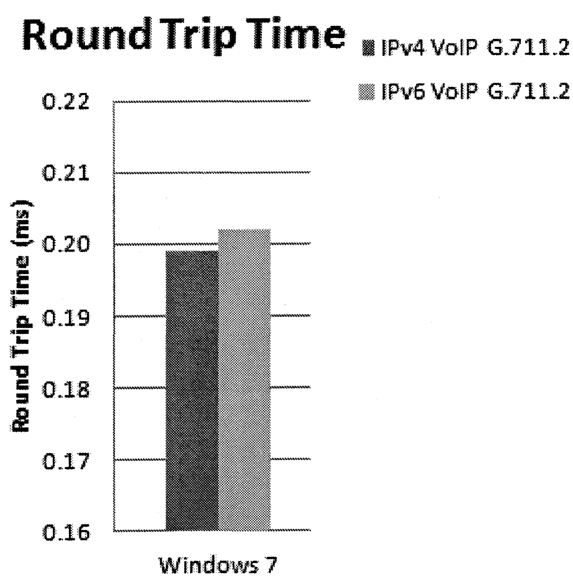

FIGURE 29. VoIP G.711.2 round trip time.

VoIP G.711.2 Jitter

The following bar chart shows the jitter of VoIP G.711.2 codec with 2 samples per packet in five both IPv4 and IPv6 networks.

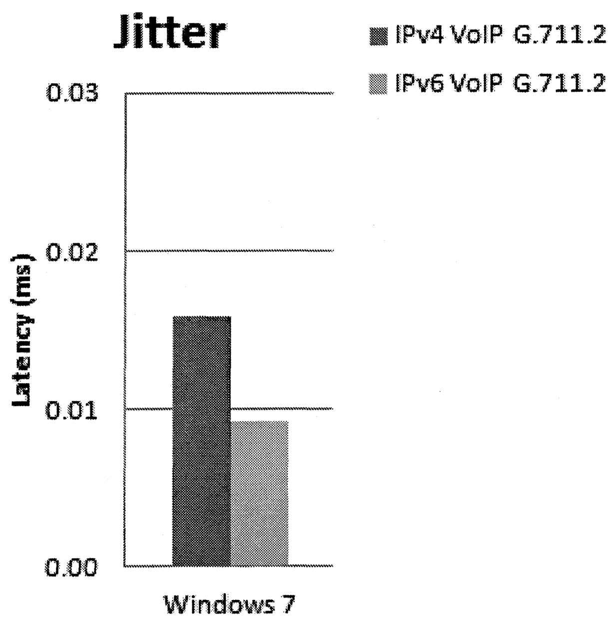

FIGURE 30. VoIP G.711.2 jitter.

From the above bar chart, the following observation can be made:

1. VoIP G.711.2 jitter in the IPv4 network is almost twice that in the IPv6 network.

VoIP G.723.1 Throughput

The bar chart in Figure 31 shows the throughput of VoIP G.723.1 codec in both the IPv4 and IPv6 networks.

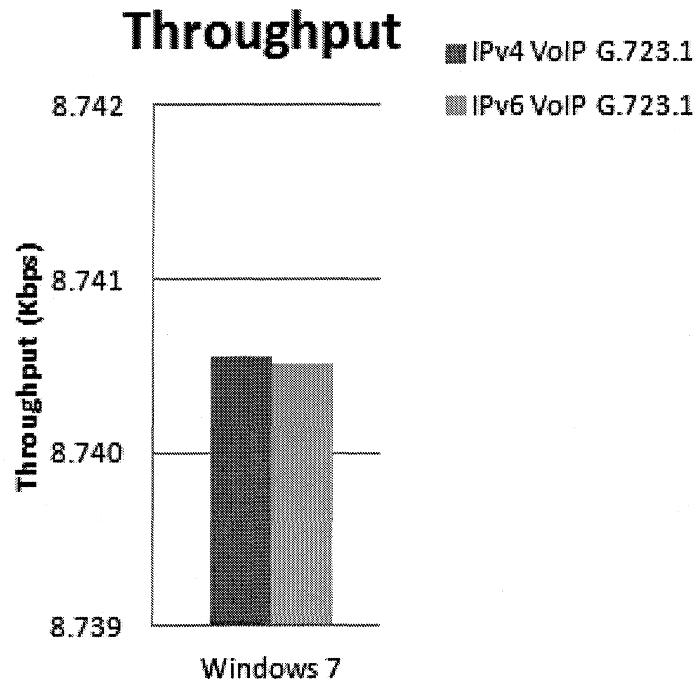

FIGURE 31. VoIP G.723.1 throughput.

From the above bar chart, the following observation can be made:

1. Throughput in VoIP G.723.1 is similar in both the networks.

<u>VoIP G.723.1 Round Trip Time</u>

The bar chart in Figure 32 shows the round trip time of VoIP G.723.1 codec in both the IPv4 and IPv6 networks.

From this bar chart, the following observation can be made:

1. Round trip time is similar in VoIP G.723.1 in both the IPv4 and IPv6 networks.

<u>VoIP G.723.1 Jitter</u>

The bar chart in Figure 33 shows the jitter of VoIP G.723.1 codec in both IPv4 and IPv6 networks.

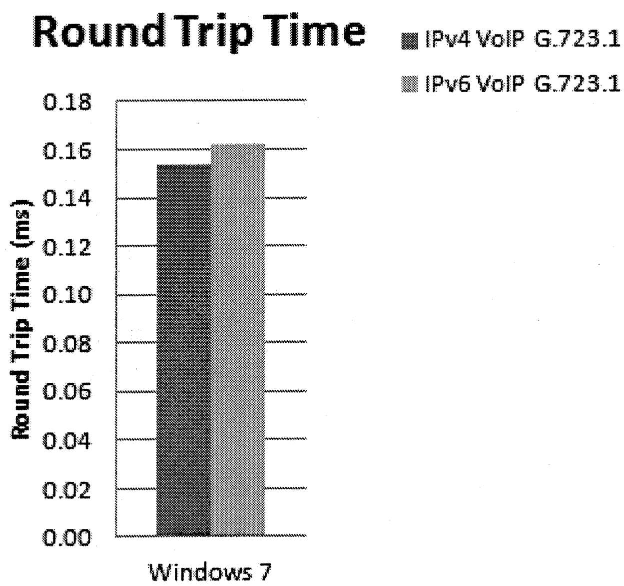

FIGURE 32. VoIP G.723.1 round trip time.

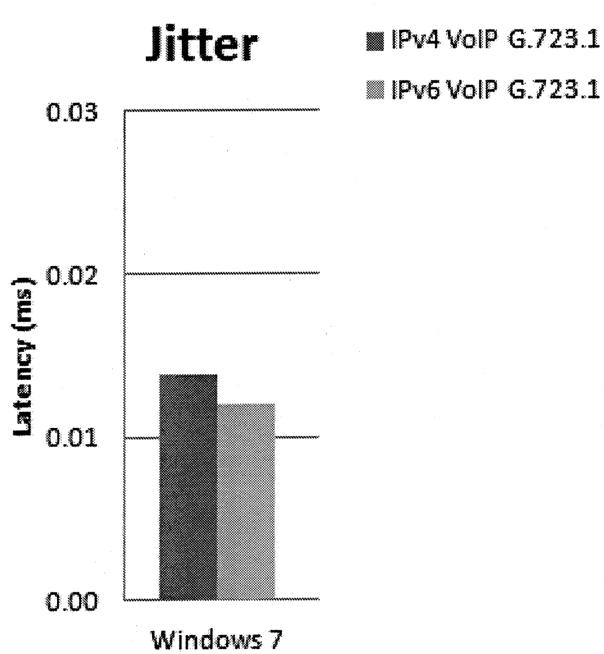

FIGURE 33. VoIP G.723.1 jitter.

From the bar chart in Figure 33, the following observation can be made:

1. Jitter in VoIP G.723.1 in the IPv4 network is similar to the jitter level in the IPv6 network.

VoIP G.729.2 Throughput

The following bar chart shows the throughput of VoIP G.729.2 codec with 2 samples per packet in both IPv4 and IPv6 networks.

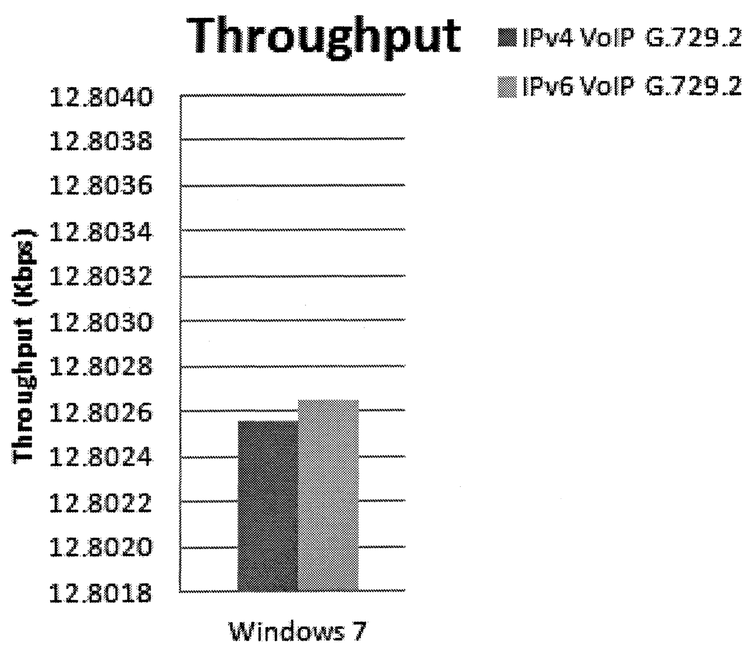

FIGURE 34. VoIP G.729.2 throughput.

From the above bar chart, the following observation can be made:

1. Throughput is slightly higher in VoIP G.729.2 in the IPv6 network than it is in the IPv4 network.

VoIP G.729.2 Round Trip Time

The following bar chart shows the round trip time of VoIP G. 729 codec with 2 samples per packet in both IPv4 and IPv6 networks (see Figure 35).

From the above bar chart, the following observation can be made:

1. Round trip time is higher in VoIP G.729.2 for the IPv6 network than it is for the IPv4 network.

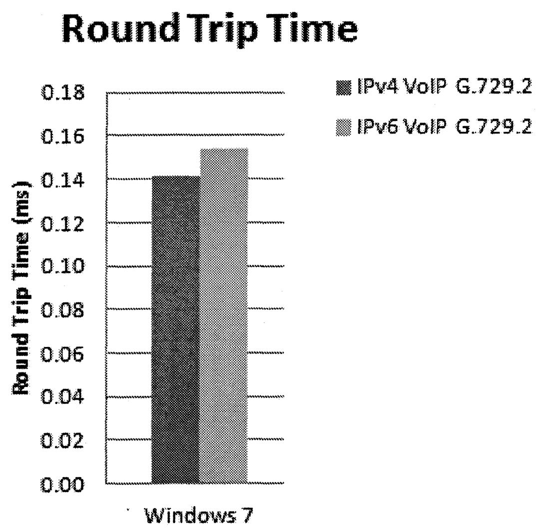

FIGURE 35. VoIP G.729.2 round trip time.

VoIP G.729.2 Jitter

The bar chart in Figure 36 shows the jitter of VoIP G.729.2 codec with 2 samples per packet in both IPv4 and IPv6 networks.

From this bar chart, the following observation can be made:

1. Jitter in the IPv6 network is slightly less (<0.001) than in the IPv4 network.

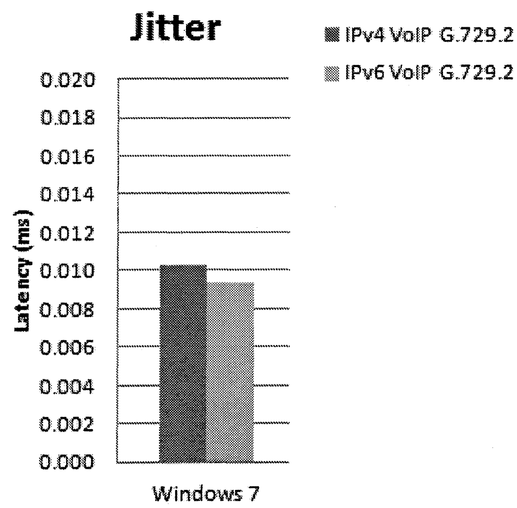

FIGURE 36. VoIP G.729.2 jitter.

<u>VoIP G.729.3 Throughput</u>

The bar chart in Figure 37 shows the throughput of VoIP G.729.3 codec with 3 samples per packet in both IPv4 and IPv6 networks (See Figure 37).

From this bar chart, the following observation can be made:

1. The IPv4 network has higher throughput in VoIP G.729.3 in comparison to the IPv6 network.

<u>VoIP G.729.3 Round Trip Time</u>

The bar chart in Figure 38 shows the round trip time of VoIP G.729.3 codec with 3 samples per packet in both IPv4 and IPv6 networks.

From this bar chart, the following observation can be made:

1. Round Trip Time in the IPv6 network for VoIP G.729.3 traffic is higher than that in the IPv4 network.

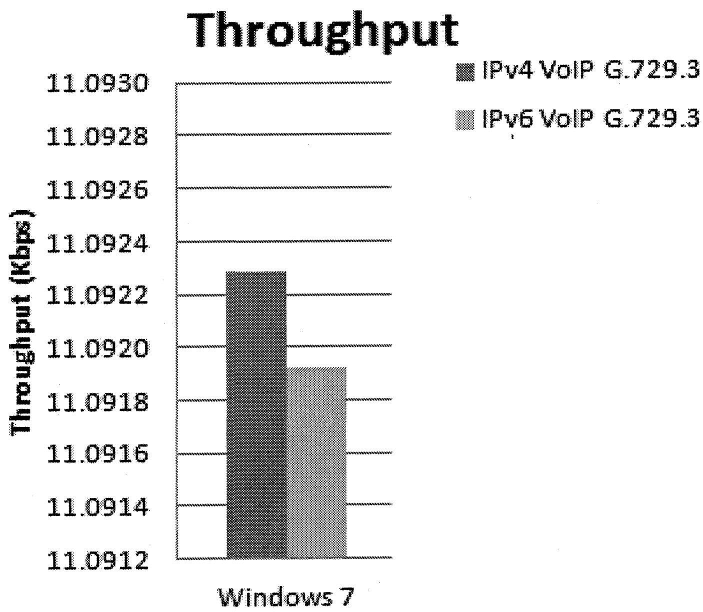

FIGURE 37. VoIP G.729.3 throughput.

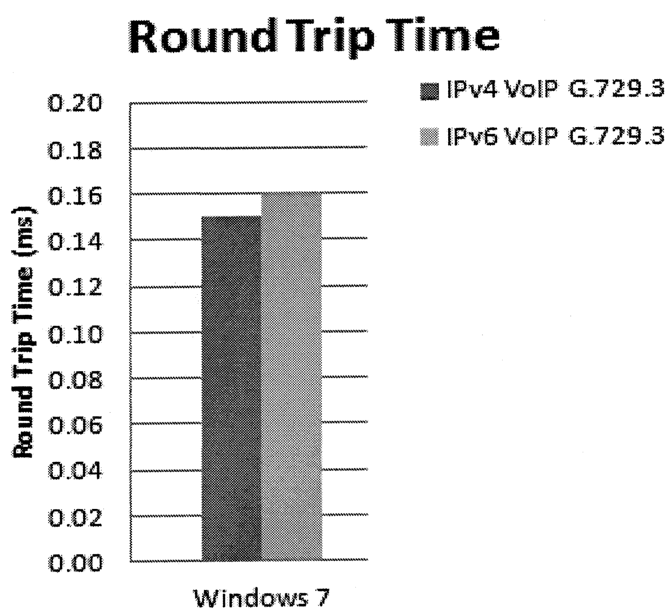

FIGURE 38. VoIP G.729.3 round trip time.

VoIP G.729.3 Jitter

The following bar chart shows the jitter of VoIP G.729.3 codec with 3 samples per packet in both IPv4 and IPv6 networks.

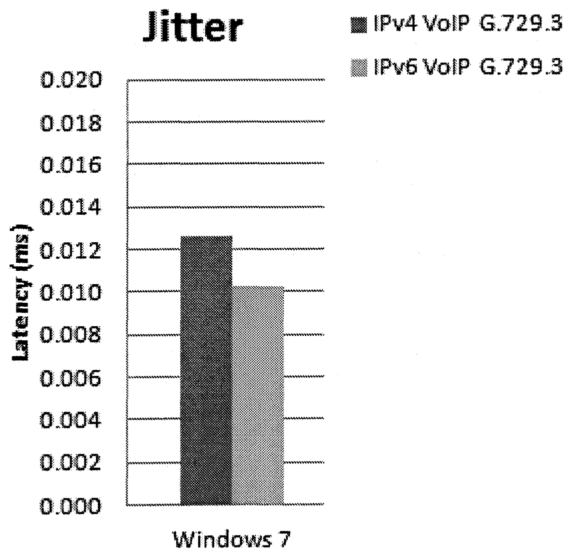

FIGURE 39. VoIP G.729.3 jitter.

From above bar chart, the following observation can be made:

1. The IPv6 network has lower jitter in VoIP G.729.3 than does the IPv4 network.

CHAPTER 5

RESULTS ANALYSIS

This section discusses the results obtained from the experimentation previously discussed and analyses the results in relationship to those given in the literature [9] [11].

Test Results Summary

Microsoft Windows 7 network system was tested with different types of traffic, such as TCP, UDP, VoIP and gaming in both the IPv4 and IPv6 networks. The resulting throughput, round trip time and jitter values were recorded. Among these types of traffic, packet size was gradually increased for TCP and UDP traffic from 64 bytes to 1536 bytes.

TCP Performance

The results of testing of TCP performance indicate that in most cases the IPv4 networks have higher TCP performance than the IPv6 networks.

IPv4 clearly outperformed IPv6 in the following situations:

1. IPv4 has higher throughput for packet size greater than 512 bytes.
2. The round trip time for IPv6 is higher than for IPv4 for packet sizes between 384 bytes and 1024 bytes.
3. Jitter in IPv4 is similar to that found in IPv6 except when the packet size range is 896 bytes to 1408 bytes.

UDP Performance

UDP performance results indicate that the IPv4 network has a higher UDP performance than IPv6. The round trip time is slightly higher on the curve for IPv4, but jitter is same for both the networks.

Gaming Performance

The results of Gaming performance indicate that in gaming, with network traffic generated by Counter-Strike, the IPv6 networks have better performance than IPv4. IPv6 results scored better in all three performance measures—throughput, round trip time and jitter.

However, in gaming with network traffic modeled by Quake 3, IPv4 networks showed slightly better performance than IPv6 networks. IPv4 gives better throughput, round trip time and jitter.

Therefore, from this limited gaming network performance perspective we can conclude that selection and use of IPv4 or IPv6 can affect the performance experienced by the end user. It is interesting to note that the superiority of IPv4 or IPv6 should be made on a case by case basis.

VoIP Performance

The results of VoIP performance indicate that different VoIP codecs have different network performance. Some significant facts that can be drawn from the results are as follows:

1. VoIP in the IPv4 networks is generally faster than in the IPv6 networks.
2. In VoIP G.729.2 codec sample 2, the performance of IPv6 is better than that of the IPv4 network.

From these findings the following can be concluded: the IPv4 networks demonstrate slightly better performance than the IPv6 networks for both TCP and UDP traffic.

CHAPTER 6

CONCLUSION AND FUTURE WORK

Conclusion

This research focuses on the evaluation of network performance in the Microsoft Windows 7 Operating System for Internet Protocol Versions 4 and 6. In order to meet the goals of the research, an experiment was designed to gather data. In this experiment, Windows 7 was tested with different types of traffic, such as TCP, UDP, VoIP and gaming in both IPv4 and IPv6 networks. The experiments were run for a simple network scenario using a client and a server connected via ISPs through a single router. The resulting throughput, round trip time and jitter values were recorded. Packet size was gradually increased for TCP and UDP traffic from 64 bytes to 1536 bytes. The conclusion of the experiments is that for the test network IPv4 networks have a small performance advantage over IPv6 networks for both TCP and UDP traffic. For the two types of gaming traffic we tested, we saw conflicting results. For Quake 3, IPv4 outperformed IPv6. For Counter Strike, IPv6 outperforms IPv4. This variance in results seems to result from the different types of traffic load generated by the two applications. It is clear that the additional features of the protocol brings with it some interesting performance challenges.

Future Work

While this study focuses on comparing the performance of IPv4 and IPv6 on Windows 7 using different traffic types for a simple network scenario, many future directions can be taken with the research.

One interesting area to explore would be to compare the performance of the IPv4 and IPv6 using real time data. A network can be designed like the one in this research and then the multimedia data could send from one computer to another computer using VoIP enabled software.

Another interesting scenario would be a network design in which computers in different Local Area Networks are made to exchange data at same time. Using this scenario, one can experiment, observe and compare the impact of network congestion on the multicasting support by IPv4 and IPv6. A typical example for this scenario might be multiplayer game played being played over LANs.

One might also compare the performance of IPv4 and IPv6 on the latest Operating Systems like Microsoft Windows 8, which were released while this research was in progress. This scenario may be interesting as the Internet protocols are implemented as a part of Operating Systems. These protocols are programmed by the engineers into the Operating System and hence are dependent on the way they are implemented.

Similarly an interesting Internet Protocol comparison can be done by using smart phones and computers in a LAN. This way, one will be able to compare the performance difference between IPv4 and IPv6 with their mobile counter parts (i.e., mobile IPv4 and mobile IPv6). This scenario could be taken by making smart phones and computers communicate with each other over a mobile network like 3G or 4G.

APPENDICES

APPENDIX A

SAMPLE COMMANDS FOR D-ITG

Sample Commands for D-ITG

The sample commands of D-ITG used in the experiments are as follows:

Start the receiver on the destination host (192.168.1.1)

C:\>ITGRecv

Start the sender on the source host (192.168.0.1)

C:\>ITGSend –a 192.168.1.1 -m rttm -T TCP -C 30000 -c 64 -t 60000

C:\>ITGDec ITGSend.log >> SampleData.txt

The resulting flow from 192.168.0.1 to 192.168.1.1 has the following characteristics:

Type of meter is set to Round Trip Time meter

Type of protocol is set to TCP

30000 packets per second are sent

The size of each packet is equal to 64 bytes

The duration of the generation experiment is 60 seconds

At sender side ITGLog cerate log file ITGSend.log

At sender side ITGDec decode log file ITGSend.log into readable text file named SampleData.txt

This is the content of SampleData.txt

Flow number: 1

From 192.168.0.1:1451

To 192.168.1.1:8999

Total time = 60.151123 s

Total packets = 1800000

Minimum delay = 0.000172 s

Maximum delay = 0.151544 s

Average delay = 0.001008 s

Average jitter = 0.000040 s

Delay standard deviation = 0.000384 s

Bytes received = 115200000

Average bitrate = 15321.409710 Kbit/s

Average packet rate = 29924.628340 pkt/s

Packets dropped = 0 (0.00%)

**************** TOTAL RESULTS ******************

Number of flows = 1

Total time = 60.151123 s

Total packets = 1800000

Minimum delay = 0.000172 s

Maximum delay = 0.151544 s

Average delay = 0.001008 s

Average jitter = 0.000040 s

Delay standard deviation = 0.000384 s

Bytes received = 115200000

Average bitrate = 15321.409710 Kbit/s

Average packet rate = 29924.628340 pkt/s

Packets dropped = 0 (0.00%)

Error lines = 0

APPENDIX B

RESULTS OF EXPERIMENTS IN TABULAR FORM

TCP Results

The following table shows the results of TCP throughput. The results are represented in megabits per second.

TABLE 3. IPv4 and IPv6 TCP Throughput

TCP – Throughput (Mbps)		
Packet Size (bytes)	IPv4	IPv6
64	14.947910	14.947729
128	29.988727	29.969039
256	59.999461	59.999463
384	83.278996	81.436111
512	85.054164	82.531848
640	84.931069	83.030494
768	85.003453	83.201779
896	86.814934	84.547541
1024	87.653019	88.307758
1152	90.086541	89.204832
1280	90.487587	89.709317
1480	91.240599	89.650179
1536	83.808512	81.276617

The following table shows the results of TCP round trip time. The results are represented in milliseconds.

TABLE 4. IPv4 and IPv6 Round Trip Time

TCP - Round Trip Time (ms)		
Packet Size (bytes)	IPv4	IPv6
64	0.9919	1.0699
128	1.0882	1.1087
256	0.8103	0.8206
384	7.4129	7.7907
512	4.9223	7.2833
640	6.5169	7.8409
768	4.2420	6.3840
896	5.4141	7.7151
1024	4.2889	1.6830
1152	2.4761	1.7747
1280	2.1957	1.7043
1408	1.4747	1.9279
1536	1.9354	1.8559

The following table shows the results of TCP jitter. The results are represented in milliseconds.

TABLE 5. IPv4 and IPv6 TCP Jitter

TCP - Jitter (ms)		
Packet Size (bytes)	IPv4	IPv6
64	0.0426	0.0430
128	0.0442	0.0437
256	0.0362	0.0366
384	0.0390	0.0397
512	0.0500	0.0533
640	0.0651	0.0686
768	0.0757	0.0730
896	0.0794	0.0862
1024	0.0727	0.0529
1152	0.0459	0.0455
1280	0.0619	0.0524
1408	0.0560	0.0587
1536	0.0187	0.0229

UDP Results

The following table shows the results of UDP throughput. The results are represented in megabits per second.

TABLE 6. IPv4 and IPv6 UDP Throughput

UDP – Throughput (Mbps)		
Packet Size (bytes)	IPv4	IPv6
64	14.932907	14.861681
128	29.825541	29.758537
256	59.495389	59.362781
384	81.424277	78.067455
512	83.446191	80.662258
640	84.633967	82.368051
768	85.496357	83.320411
896	85.925866	84.139388
1024	85.018119	84.636122
1152	92.348707	90.814932
1280	92.848367	91.450994
1408	93.264599	91.971361
1536	90.301409	87.320151

The following table shows the results of UDP round trip time. The results are represented in milliseconds.

TABLE 7. IPv4 and IPv6 UDP Round Trip Time

UDP - Round Trip Time (ms)		
Packet Size (bytes)	IPv4	IPv6
64	0.3759	0.7487
128	0.4823	0.4498
256	0.5870	0.4041
384	1.8342	1.6302
512	1.8907	1.7654
640	1.9887	1.8043
768	2.0673	1.8747
896	2.0611	1.9881
1024	2.0272	1.8428
1152	1.4507	1.2204
1280	1.3552	1.2223
1408	1.4513	1.2627
1536	1.4181	1.3331

The following table shows the results of UDP Jitter. The results are represented in milliseconds.

TABLE 8. IPv4 and IPv6 UDP Jitter

UDP - Jitter (ms)		
Packet Size (bytes)	IPv4	IPv6
64	0.0097	0.0103
128	0.0093	0.0102
256	0.0091	0.0099
384	0.0346	0.0359
512	0.0611	0.0637
640	0.0831	0.0867
768	0.0898	0.1057
896	0.1162	0.1227
1024	0.1381	0.1399
1152	0.0081	0.0089
1280	0.0061	0.0070
1408	0.0053	0.0059
1536	0.0079	0.0081

Gaming Results

Table 9 shows the results of two player game's throughput. The results are represented in Kilobits per second.

TABLE 9. IPv4 and IPv6 Gaming Throughput

	Throughput (Kbps)
IPv4 CS	8.275703
IPv6 CS	8.952823
IPv4 Quake3	74.263874
IPv6 Quake3	74.029725

The following table shows the results of round trip time for two player games. The results are represented in milliseconds.

TABLE 10. IPv4 and IPv6 Gaming Round Trip Time

	Round Trip Time (ms)
IPv4 CS	0.1623
IPv6 CS	0.1747
IPv4 Quake3	0.1581
IPv6 Quake3	0.1741

Table 11 shows the results for jitter in two player games. The results are represented in milliseconds.

TABLE 11. IPv4 and IPv6 Gaming Jitter

	Jitter (ms)
IPv4 CS	0.0151
IPv6 CS	0.0121
IPv4 Quake3	0.0129
IPv6 Quake3	0.0145

VoIP Results

The following table shows the results of VoIP throughput. The results are represented in kilobits per second.

TABLE 12. IPv4 and IPv6 VoIP Throughput

	Throughput (Kbps)
IPv4 VoIP G.711.1	73.605963
IPv6 VoIP G.711.1	73.604348
IPv4 VoIP G.711.2	68.814321
IPv6 VoIP G.711.2	68.815093
IPv4 VoIP G.723.1	8.740547
IPv6 VoIP G.723.1	8.740509
IPv4 VoIP G.729.2	12.802542
IPv6 VoIP G.729.2	12.802633
IPv4 VoIP G.729.3	11.092257
IPv6 VoIP G.729.3	11.091922

The following table shows the results of VoIP round trip time. The results are represented in milliseconds.

TABLE 13. IPv4 and IPv6 VoIP Round Trip Time

	Round Trip Time (ms)
IPv4 VoIP G.711.1	0.1632
IPv6 VoIP G.711.1	0.1721
IPv4 VoIP G.711.2	0.1954
IPv6 VoIP G.711.2	0.2017
IPv4 VoIP G.723.1	0.1539
IPv6 VoIP G.723.1	0.1627
IPv4 VoIP G.729.2	0.1410
IPv6 VoIP G.729.2	0.1549
IPv4 VoIP G.729.3	0.1500
IPv6 VoIP G.729.3	0.1608

The following table shows the results of VoIP jitter. The results are represented in milliseconds.

TABLE 14. IPv4 and IPv6 VoIP Jitter

	Jitter (ms)
IPv4 VoIP G.711.1	0.009
IPv6 VoIP G.711.1	0.0101
IPv4 VoIP G.711.2	0.0179
IPv6 VoIP G.711.2	0.0099
IPv4 VoIP G.723.1	0.0136
IPv6 VoIP G.723.1	0.0113
IPv4 VoIP G.729.2	0.0107
IPv6 VoIP G.729.2	0.0099
IPv4 VoIP G.729.3	0.0129
IPv6 VoIP G.729.3	0.0109

REFERENCES

REFERENCES

[1] K. Perset, "Internet Address Space: Economic Considerations in the Management of IPv4 and in the Deployment of IPv6," *Proc. OECD Ministerial Meeting*, Seoul, Korea, June 2008, pp. 17-18.

[2] S. Narayan, Peng Shang and Na Fan, "Performance Comparison of IPv4 and IPv6 on Windows Vista and Linux Ubuntu," *Proc. International Conference on Networks Security, Wireless Communications and Trusted Computing*, IEEE, 2009, pp. 653-656.

[3] Y. Zhang and Z. Li, "IPv6 Conformance Testing: Theory and Practice," *Proc. Test Conference*, IEEE, 2004, pp. 719-727.

[4] C. Walton, "IPv6: at the Starting Line," 10 Jan. 2009; http://support.novell.com/com/techcenter/articles/nc1999_05a.html.

[5] S. Zeadally and L. Raicu, "Evaluating IPv6 on Windows and Solaris," *Proc. Internet Computing*, IEEE, Mar. 2002, pp. 51-57.

[6] V.Visoottiviseth and N. Bureenok, "Performance Comparison of ISATAP Implementations on FreeBSD, RedHat and Windows 2003," *Proc. Information and Emerging Technologies*, IEEE, Mar. 2008, pp. 547-552.

[7] Natalie Temple, "First Quarter 2013, State of the Internet Report," 23 Jul. 2013; http://www.akamai.com/stateoftheinternet.

[8] S. Avallone, A. Botta, A. Dainotti, W. Donato and A. Pescape, "D-ITG v.2.8.1 Manual," 02 Jul. 2013; http://www.grid.unina.it/software/ITG/download.php

[9] S. Avallone, S. Guadagno, D. Emma, A.Pescape and G. Ventre, " D-ITG Distributed Internet Traffic Generator," *Proc. Quantitative Evaluation of Systems Conference*, IEEE, 2004, pp. 316-317.

[10] Nortel Networks, *Voice Over Internet Protocol (VoIP) Technologies*, Research Triangle Park, NC: Nortel Press, 2008, pp. 544-544.

[11] N. Cuturic and S. Lozanovski, "DCCP vs. UDP for VoIP," master's thesis, Computer Networking Dept., Mid Sweden University, 2006; http://portal.miun.se/~necu9500/dccp/D-ITG_codes.html.

[12] D. Emma, A. Pescape and G. Ventre, " Analysis and Experimentation of an Open Distributed Platform for Synthetic Traffic Generation," *Proc. Distributed Computing Systems Conference*, IEEE, 2004, pp. 719-727.

[13] R. Blum, *Network performance Open Source Toolkit: Using Netperf, Tcptrace, NIST Net and SSFNet*, Indianapolis, Indiana, Wiley, 2003.

[14] M. Santkuyl, "Jitter," 12 Dec. 2008; http://searchunifiedcommunications.techtarget.com/definition/jitter.

[15] J. A. Perez, V. Zarate, A. Montes and C. Garcia, "Quality of Service Analysis of IPSec VPNs for Voice and Video Traffic," *Proc. Computer Engineering and Technology*, IEEE, Feb. 2006, pp. 43-43.