Evaluation of network performance of Microsoft operating systems

Yi SHI

A dissertation submitted in partial fulfilment of the requirements for the degree of Master of Computing

UNITEC New Zealand, 2009
Abstract

This master’s dissertation investigates the evaluation of network performance of various Microsoft Windows operating systems (Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008 and Windows 7 Beta) for both TCP and UDP protocols, as well as DNS, VoIP and gaming bandwidths. The parameters considered for each of these operating systems are throughput, round trip time and jitter. Results indicate that the newer Microsoft Windows client operating system (Windows Vista) does not bring convincing improvements in network performance compared with its predecessor (Windows XP). However, the newer Windows Server operating system (Windows Server 2008) has much higher network performance than its predecessor: Windows Server 2003.

Acknowledgements

I would like to thank postgraduate programmes director Donald Joyce, my principal supervisor Ranjana Shukla and associate supervisor Shaneel Narayan. They have given me a lot of new ideas and helped me in every step. Without their help, I could not have completed my dissertation.

Finally, I would like to thank my family members who have supported me and given me a chance to finish my master’s degree.
Table of Contents

Abstract ....................................................................................................................................... i
Acknowledgements ..................................................................................................................... i
List of Tables .............................................................................................................................. v
List of Figures ............................................................................................................................ vi
List of Abbreviations ................................................................................................................. ix
1.0 Introduction ......................................................................................................................... 1

2.0 Literature Review ............................................................................................................... 3

2.1 Definition of Network Performance ................................................................................ 3
2.2 Network Performance Issues ........................................................................................... 4
2.3 Measurements and Tools of Network Performance ....................................................... 6
  2.3.1 Network Performance Measurements ..................................................................... 6
  2.3.2 Network Performance Measuring Tools ................................................................. 10
2.4 Network Performance Enhancement ............................................................................ 14
2.5 Chapter Summary .......................................................................................................... 17

3.0 Research Methodology ..................................................................................................... 18

3.1 Research Questions ....................................................................................................... 18
3.2 Method of Study ............................................................................................................ 18
3.3 Experimental Design ...................................................................................................... 19
  3.3.1 Testbed Design and Experimental Tasks ................................................................. 19
  3.3.2 Packet Sizes ............................................................................................................. 21
  3.3.3 Task Duration and Result Objectivity ...................................................................... 21
  3.3.4 Network Performance Measuring Tool ................................................................. 21
  3.3.4 Packet Rate ............................................................................................................. 25
  3.3.6 Measurements ........................................................................................................ 27
  3.3.7 IPv6 Configuration ................................................................................................... 29
  3.3.8 Multi Boot Manager .............................................................................................. 32
  3.3.9 Backup Management .............................................................................................. 33
3.4 Chapter Summary .......................................................................................................... 36

4.0 Data Collection .................................................................................................................. 38

4.1 Existing Literature Gathering ......................................................................................... 38
4.2 Primary Data Gathering ................................................................................................. 39
  4.2.1 Data Entry ............................................................................................................... 39
  4.2.2 Generating Charts ................................................................................................. 40
5.0 Data Analysis ............................................................................................................... 43

5.1 TCP Analysis .................................................................................................................. 43
  5.1.1 TCP Throughput ...................................................................................................... 43
  5.1.2 TCP Round Trip Time ............................................................................................ 46
  5.1.3 TCP Jitter .............................................................................................................. 48

5.2 UDP Analysis .................................................................................................................. 49
  5.2.1 UDP Throughput ..................................................................................................... 49
  5.2.2 UDP Round Trip Time ............................................................................................ 52
  5.2.3 UDP Jitter ................................................................................................................ 54

5.3 DNS Analysis .................................................................................................................. 55
  5.3.1 TCP Throughput of DNS ....................................................................................... 55
  5.3.2 TCP Round Trip Time of DNS ................................................................................ 56
  5.3.3 TCP Jitter of DNS .................................................................................................. 57
  5.3.4 UDP Throughput of DNS ....................................................................................... 58
  5.3.5 UDP Round Trip Time of DNS ................................................................................ 59
  5.3.6 UDP Jitter of DNS ................................................................................................ 60

5.4 Gaming Analysis ............................................................................................................. 61
  5.4.1 Throughput of Counter Strike .............................................................................. 61
  5.4.2 Round Trip Time of Counter Strike .................................................................... 62
  5.4.3 Jitter of Counter Strike ......................................................................................... 63
  5.4.4 Throughput of Quake 3 ....................................................................................... 64
  5.4.5 Round Trip Time of Quake 3 ............................................................................... 65
  5.4.6 Jitter of Quake 3 .................................................................................................. 66

5.5 VoIP Analysis .................................................................................................................. 67
  5.5.1 Throughput of G.711.1 .......................................................................................... 67
  5.5.2 Round Trip Time of G.711.1 ................................................................................ 68
  5.5.3 Jitter of G.711.1 .................................................................................................... 69
  5.5.4 Throughput of G.711.2 ........................................................................................ 70
  5.5.5 Round Trip Time of G.711.2 ................................................................................ 71
  5.5.6 Jitter of G.711.2 .................................................................................................... 72
  5.5.7 Throughput of G.723.1 ........................................................................................ 73
  5.5.8 Round Trip Time of G.723.1 ................................................................................ 74
  5.5.9 Jitter of G.723.1 .................................................................................................... 75
List of Tables

Table 2-1: Comparison of network performance tools ........................................................... 13
Table 3-1: Experimental Hardware Specification ................................................................. 19
Table 3-1: Result of Packet Rate ......................................................................................... 26
Table A-1: IPv4 & IPv6 TCP Throughput of Microsoft Windows ......................................... 95
Table A-2: IPv4 & IPv6 Round Trip Time of Microsoft Windows ........................................ 96
Table A-3: IPv4 & IPv6 TCP Jitter of Microsoft Windows ...................................................... 97
Table A-4: IPv4 & IPv6 UDP Throughput of Microsoft Windows ........................................ 98
Table A-5: IPv4 & IPv6 UDP Round Trip Time of Microsoft Windows ................................ 99
Table A-6: IPv4 & IPv6 UDP Jitter of Microsoft Windows .................................................... 100
Table A-7: IPv4 & IPv6 TCP throughput of DNS of Microsoft Windows .............................. 101
Table A-8: IPv4 & IPv6 TCP Round Trip Time of DNS of Microsoft Windows .................. 101
Table A-9: IPv4 & IPv6 DNS TCP Jitter of Microsoft Windows ........................................... 101
Table A-10: IPv4 & IPv6 UDP Throughput of DNS of Microsoft Windows ........................ 101
Table A-11: IPv4 & IPv6 UDP Round Trip Time of DNS of Microsoft Windows ................. 102
Table A-12: IPv4 & IPv6 UDP Jitter of DNS of Microsoft Windows ..................................... 102
Table A-13: IPv4 & IPv6 Gaming Throughput of Microsoft Windows ............................... 103
Table A-14: IPv4 & IPv6 Gaming Round Trip Time of Microsoft Windows ....................... 103
Table A-15: IPv4 & IPv6 Gaming Jitter of Microsoft Windows ............................................ 103
Table A-16: IPv4 & IPv6 VoIP Throughput of Microsoft Windows ..................................... 104
Table A-17: IPv4 & IPv6 VoIP Round Trip Time of Microsoft Windows ............................ 104
Table A-18: IPv4 & IPv6 VoIP Jitter of Microsoft Windows ............................................... 105
List of Figures

Figure 2-1: Network Topology ................................................................................................... 4
Figure 2-2: Network Bottleneck ................................................................................................. 9
Figure 2-3: D-ITG Architecture ............................................................................................... 11
Figure 3-1: Network Design of Experiment .............................................................................. 20
Figure 3-2: D-ITG Codecs for VoIP packet generator ............................................................... 23
Figure 3-3: Test Different Packet Rates ................................................................................... 27
Figure 3-4: Jitter Formula (Avallone, Botta etc., 2008) ............................................................ 28
Figure 3-5: Round Trip Time equation (Avallone, Botta etc., 2008) ........................................ 28
Figure 3-6: Windows XP and 2003 IPv6 Installation ................................................................ 29
Figure 3-7: Assign IPv6 address ............................................................................................... 30
Figure 3-8: IPv6 information .................................................................................................... 30
Figure 3-9: IPv6 Component in Windows Vista ........................................................................ 31
Figure 3-10: Obtain Properties of Local Area Connection ....................................................... 31
Figure 3-11: IPv6 Properties in Windows Vista ....................................................................... 32
Figure 3-12: Windows XP and Windows Server 2003 Multi Boot Menu ................................. 33
Figure 3-13: All Experimental Microsoft Windows Boot Menu ............................................ 33
Figure 3-14: Symantec Ghost 11.5 ........................................................................................... 34
Figure 3-15: Backup Disk to Image File ................................................................................... 35
Figure 3-16: Backup Disk – Select Source Drive .................................................................... 35
Figure 3-17: Backup Disk – Enter Image Name ....................................................................... 35
Figure 3-18: Restore Disk from Image File ............................................................................. 36
Figure 3-19: Restore Disk – Select Image File ........................................................................ 36
Figure 4-1: The Sample of Log File ........................................................................................ 39
Figure 4-2: Excel Template for TCP and UDP ........................................................................ 40
Figure 4-3: Excel Template for Other Tasks .......................................................................... 40
Figure 4-4: Example of Line Chart ........................................................................................ 41
Figure 4-5: Example of Bar Chart .......................................................................................... 41
Figure 5-1: TCP Throughput of Five Microsoft Windows ....................................................... 43
Figure 5-2: TCP Throughput of Three Microsoft Windows (XP, Vista, 7) ............................. 44
Figure 5-3: TCP Throughput of Three Microsoft Windows Server (2003, 2008) ............... 45
Figure 5-4: TCP Round Trip Time of Three Microsoft Windows (XP, Vista, 7) .................... 46
Figure 5-5: TCP Round Trip Time of Two Microsoft Windows Server (2003, 2008) ......... 47
Figure 5-6: TCP Jitter of Five Microsoft Windows .................................................................. 48
Figure 5-7: UDP Throughput of Five Microsoft Windows ......................................................... 49
Figure 5-8: UDP Throughput of Three Microsoft Windows (XP, Vista, 7) ................................. 50
Figure 5-9: UDP Throughput of Two Microsoft Windows Server (2003, 2008) ....................... 51
Figure 5-10: UDP Round Trip Time of Three Microsoft Windows (XP, Vista, 7) .................... 52
Figure 5-11: UDP Round Trip Time of Two Microsoft Windows Server (2003, 2008) .......... 53
Figure 5-12: UDP Jitter of Five Microsoft Windows ............................................................ 54
Figure 5-13: TCP Throughput of DNS of Five Microsoft Windows ........................................ 55
Figure 5-14: TCP Round Trip Time of DNS of Five Microsoft Windows ................................... 56
Figure 5-15: TCP Jitter of DNS of Five Microsoft Windows .................................................. 57
Figure 5-16: UDP Round Trip Time of DNS of Five Microsoft Windows .............................. 58
Figure 5-17: UDP Round Trip Time of DNS of Five Microsoft Windows .............................. 59
Figure 5-18: UDP Jitter of DNS of Five Microsoft Windows .................................................. 60
Figure 5-19: Counter Strike (Game) Throughput of Five Microsoft Windows ......................... 61
Figure 5-20: Counter Strike (Game) Round Trip Time of Five Microsoft Windows .................. 62
Figure 5-21: Counter Strike (Game) Jitter of Five Microsoft Windows .................................. 63
Figure 5-22: Quake 3 (Game) Throughput of Five Microsoft Windows .................................. 64
Figure 5-23: Quake 3 (Game) Round Trip Time of Five Microsoft Windows ......................... 65
Figure 5-24: Quake 3 (Game) Jitter of Five Microsoft Windows ............................................. 66
Figure 5-25: VoIP G.711.1 Throughput of Five Microsoft Windows ........................................ 67
Figure 5-26: VoIP G.711.1 Round Trip Time of Five Microsoft Windows ............................... 68
Figure 5-27: VoIP G.711.1 Jitter of Five Microsoft Windows .................................................. 69
Figure 5-28: VoIP G.711.2 Throughput of Five Microsoft Windows ........................................ 70
Figure 5-29: VoIP G.711.2 Round Trip Time of Five Microsoft Windows ............................... 71
Figure 5-30: VoIP G.711.2 Jitter of Five Microsoft Windows .................................................. 72
Figure 5-31: VoIP G.723.1 Throughput of Five Microsoft Windows ........................................ 73
Figure 5-32: VoIP G.723.1 Round Trip Time of Five Microsoft Windows ............................... 74
Figure 5-33: VoIP G.723.1 Jitter of Five Microsoft Windows .................................................. 75
Figure 5-34: VoIP G.729.2 Throughput of Five Microsoft Windows ........................................ 76
Figure 5-35: VoIP G.729.2 Round Trip Time of Five Microsoft Windows ............................... 77
Figure 5-36: VoIP G.729.2 Jitter of Five Microsoft Windows .................................................. 78
Figure 5-37: VoIP G.729.3 Throughput of Five Microsoft Windows ........................................ 79
Figure 5-38: VoIP G.729.3 Round Trip Time of Five Microsoft Windows ............................... 80
Figure 5-39: VoIP G.729.3 Jitter of Five Microsoft Windows .................................................. 81
Figure 6-1: IPv4 and IPv6 Packet Header (Walton, 1999) ......................................................... 86
Figure 6-2: IPv4 and IPv6 Packet Structure (Zhang & Li, 2004) .............................................. 87
List of Abbreviations

AFD  Ancillary Function Driver
API  Application Programming Interface
BIOS Basic Input Output System
CSMA/CA Carrier Sense Multiple Access with Collision Avoidance
CSMA/CD Carrier Sense Multiple Access with Collision Detection
D-ITG Distributed Internet Traffic Generator
DCCP Datagram Congestion Control Protocol
DDoS Distributed Denial of Service
DNS Domain Name System
DoS Denial of Service
E-commerce Electronic commerce
E-mail Electronic mail
ECN Explicit Congestion Notification
GUI Graphical User Interface
ICMP Internet Control Message Protocol
IDT Inter Departure Time
IEEE Institute of Electrical and Electronics Engineers
IP Internet Protocol
IPSec Internet Protocol Security
IPv4 Internet Protocol version 4
IPv6 Internet Protocol version 6
LAN Local Area Network
MAC Media Access Control
MAN Metropolitan Area Network
MTU Maximum Transmission Unit
NDIS Network Driver Interface Specification
NPME Network Performance Measurement Environment
OWD One Way Delay
P2P Peer-to-Peer
PPP Point-to-Point Protocol
QoS Quality of Service
RTT Round Trip Time
SCTP Stream Control Transmission Protocol
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>Transmission Control Protocol</td>
</tr>
<tr>
<td>TDI</td>
<td>Transport Driver Interface</td>
</tr>
<tr>
<td>TDX</td>
<td>TDX is a translation layer between TDI and the Next Generation TCP/IP stack</td>
</tr>
<tr>
<td>UDP</td>
<td>User Datagram Protocol</td>
</tr>
<tr>
<td>VLAN</td>
<td>Virtual Local Area Network</td>
</tr>
<tr>
<td>VoIP</td>
<td>Voice over Internet Protocol</td>
</tr>
<tr>
<td>WAN</td>
<td>Wide Area Network</td>
</tr>
<tr>
<td>WFP</td>
<td>Windows Filtering Platform</td>
</tr>
<tr>
<td>WLAN</td>
<td>Wireless Local Area Network</td>
</tr>
<tr>
<td>WSK</td>
<td>Winsock Kernel</td>
</tr>
<tr>
<td>WWW</td>
<td>World Wide Web</td>
</tr>
</tbody>
</table>
1.0 Introduction

Nowadays, networks are becoming indispensable as nearly every computer connects to a network, from the small home Ethernet to the large enterprise network. On these networked computers, Garnham (2008) indicates that Microsoft operating systems represent nearly 90% of the market share worldwide: the same result for operating system market share presented by NetApplication.com on 3rd September 2008 (NetApplications.com, 2008). Therefore, evaluating the network performance of currently used Microsoft operating systems is necessary for anyone with whom network performance is a high priority, such as those concerned with network maintenance tasks. Different operating systems have different influences on network performance; improving network performance results in enhanced efficiency and cuts down the time spent on system maintenance tasks, which allows staff and users to spend more time on other tasks. In this research, the relative performance of Microsoft’s network operating systems will be evaluated. The purpose of this research is to provide an understanding of the variable network performance offered by different Microsoft operating systems, and to evaluate which Microsoft operating system offers a standard of network performance that is superior with regards to significant network tasks. This research also enables network administrators to select an operating system based not only on its release date, but on the performance of the operating system relative to the particular tasks that are most important to the individual network in question.

Will different operating systems influence network performance? The answer is affirmative. “Lake Partners’ research shows that for maintenance tasks, different operating systems had varied impact on network efficiency” (JuniperNetworks, 2008). Nowadays, the most common operating systems used both in homes and enterprises are Microsoft operating systems: most commonly Windows XP, Windows Server 2003, Windows Vista and Windows Server 2008. Among these operating systems differences in a range of network variables can be found. In the TCP/IP stack for example, Microsoft adopted the Next Generation TCP/IP stack with Windows Vista, and this brings with it a lots of new capabilities (Davies, 2005a); will this change bring a significant impact to the network performance of the operating system? On the other hand, researchers found that people prefer using Windows XP rather than the newer operating system Windows Vista for some network tasks.
(Schlaffer, 2007). Is this trend due to superior network performance in Windows XP or is it simply a case of a general reluctance to change? The answers to these and other questions will be discussed in more depth in the following report.
2.0 Literature Review

This chapter will begin with defining network performance and then go on to review the literature related to the network performance issues, network performance measurements, network performance tools and network performance enhancements.

2.1 Definition of Network Performance

Before diving into detailed discussions of network performance, it is a good idea to first understand what a network is. This section defines network, describes various types of networks and goes on to discuss network performance.

What is a network?

Microsoft (2008) defines network as “a configuration of data processing devices and software connected for information interchange.” Few other perspectives of network can be understood as “two or more computers connected together so that they can exchange messages, files, or other means of communication. A network is part hardware, usually cables and communications devices such as modems, and part software” (Arizona Board of Regents, 2003) or “A system of connected computers exchanging information with each other” (SFCN, 2002). In summary, a network can be understood to be a link to connect isolated computing devices or workstations together, in order to reach the goal of resources sharing and communication.

Network classifications

Harbeck (2006) point out that network can be classified in terms of spatial distance as LAN (Local Area Network), MAN (Metropolitan Area Network) and WAN (Wide Area Network). The computers of LAN are geographically close together such as in a house or a building. MAN is design for a town or a city and the computers of WAN are farther apart and usually connected by dedicated line such as telephone or radio waves. In terms of topology, network can be commonly classified as Star, Ring and Bus.
Network performance

The network performance can be understood in terms of the network’s actual output and the quality of work performed. “The main purpose of network performance studies is to help the design of the networks so that high efficiency and low cost can be achieved. To conduct performance studies, three approaches are often used: analytical modelling, queuing theory and Petri net theories are most commonly used.” (Tripathi, Huang & Jajodia, 1987). Blum (2003) indicates that network performance is a complex issue, with lots of independent variables that affect how clients access servers across a network. Usually five elements can be used to measure network performance; they are availability, response time, network utilization, network throughput and network bandwidth capacity.

2.2 Network Performance Issues

When network users feel something is wrong with the network, first time they always say “Why is the network so slow?” this is the significant network performance issue which users can experience. However, user’s computer or operating system bottlenecks also will cause this latency issue, as Santos (1994) points out that “system latency will have a large effect on the transition response time.” Many other possibilities also will cause this significant latency issue such as: network adopting latency network architecture, a mass of users utilising the same link to transmit data, existence of faulty equipment in the network, and network having restrictions such as firewalls and filters. “Solutions to the latency issues cannot be fully addressed until
the sources of latency are understood for a particular network system” (Santos, 1994). Carlson (2007) indicated three categories will affect network performance, these are network infrastructure, host computer and application design. The performance study in LAN infrastructure mainly focus on the “network topology, network traffic, time delay of the physical network elements, channel capacity, noise effect, network traffic, and techniques used in data communication” (Tripathi, Huang & Jajodia, 1987). Most LAN performance studies are related to the MAC (Media Access Control) protocols, the understanding of these protocols can grant valuable insight into the overall performance of the network. Yuang & Hsu (1994) pointed out “Medium Access Control (MAC) protocols for Local Area Networks (LANs) yielding minimal delay and maximal throughput.” However Tripathi, Huang & Jajodia (1987) said “Medium access control protocols introduce extra delay in data transmission.”

WLAN (Wireless Local Area Network), Ethernet and Token Ring are the most common LANs which are adopted widely at the moment, these LANs use CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance), CSMA/CD (Carrier Sense Multiple Access with Collision Detection) and Token Passing as their basic protocols, these protocols have been reported that may introduce extra delay by many researchers. “One disadvantage of CSMA type networks is that as the total message traffic increases, the amount of message collisions also increases resulting in a degradation of network performance.” (Wang & Hong, 1997) and Peng & Cheng (2006) also indicate that there are two disadvantages of network performance in CSMA/CA, “one is that they have limited effectiveness in dealing with hidden terminals because not every hidden terminal may be able to correctly receive them. The second disadvantage of in-band control frames is that they introduce significant overhead into the network due to their relatively long transmission times.” Token passing also has greater delay, as Abeyesundara & Kamal (1991) state that “in IEEE 802.5 Token Ring the overhead associated with a medium access protocol increases with the propagation delay.” As same as Stallings (1984) and Tripathi, Huang & Jajodia (1987) also point out that CSMA/CD is time consuming on collisions and Token Passing is time consuming on token transmission. Comparing CSMA/CD and Token Passing, “Token Passing has greater delay than CSMA/CD in light load environment, however it has less delay and stable throughput at heavy loads” (Stallings, 1984).
2.3 Measurements and Tools of Network Performance

This section will cover five common measurements for network performance; they are availability, response time, network utilization, network throughput and network bandwidth capacity. Besides that, three common measuring tools of network performance are introduced.

2.3.1 Network Performance Measurements

To evaluate a network performance is a complex work, as various situations exist in each different network. Liu, Han, Zhang, & Nie (2004) point out that the main network performance measurements include usability, response time, precision and utilization. Other article point out that “Performance measurements are taken of the system as well as the workload, using both software and hardware monitors. The parameters which are most interesting are throughput, utilization and response time” (Tripathi, Huang & Jajodia, 1987). And in Windows operating systems, Rindos, Loeb, etc. (1999) indicated two parameters: latency and data throughput, they are very important measurement in evaluating the network performance.

Abeysundara & Kamal (1991) also point out that a variety of measures have been utilized to evaluate performance of LANs and the most common measures are information throughput, channel utilization and delay.

- **Delay** - in several forms, depending on the time instants considered in the measurement of delay.
- **Information throughput** - the total number of information bits transmitted per unit time.
- **Channel utilization** - the fraction of time spent in transmitting information bits compared to the total time spent in transmitting information and overhead bits.

To conclude, five elements are able to measure the network performance, they are:

- **Availability**
- **Response time**
- **Network utilization**
- **Network throughput**
- **Network bandwidth capacity**
Therefore these elements should guide the research to complete the experiment of network performance.

**Availability**

The first step of evaluating network performance is to ensure if the network works properly. If data can’t be transmitted through the network and the network probably has a bigger problem than just network performance issues. The simplest way for network availability test is use `ping` command in Windows operating system. The ping program sends ICMP (Internet Control Message Protocol) echo request packet to the remote host from local host and when local host receives ICMP echo reply packet then you are able to determine the state of the network.

Ping program has many functions that can be used to perform advanced testing, such as parameter `-c` is able to assign the number of echo request, `-s` is able to assign the packet size for each sending. “Many network devices handle packets with multiple packet buffers, based on average packet sizes. Different buffer pools handle different sized packets” (Blum, 2003). For example, normally switches have three kinds of packet buffers: one aimed at small class of packets, one aimed at medium class of packets, and one aimed at large class of packets. In order to measure performance in these kinds of network devices accurately, the measuring tools must have the ability to send different class of packets with the Ping `-s` command can be use in this situation.

**Response time**

In order to evaluate the network performance more accurately, network administrator must observe how long it takes for packets to transmit through the network. “The time that it takes a packet to travel between two points on the network is called the response time” (Blum, 2003). Xie, Wu & Liu etc. (2000) defined response time or latency as “the time required for a block of data to be transmitted across a network connection from the sending host to the receiving host.” For example, the consume time of echo request/reply packet of Ping command is the response time. There are many reasons that can influence response times between two hosts, they include:

- **Overloaded network segments**
• Faulty network wiring
• Network errors
• Broadcast storms
• Faulty network devices
• Overloaded network hosts

“Any one or combination of these factors can contribute to slow network response time” (Blum, 2003). When network works normally, network administrator can record the normal response times, and compare with the response times when users complain network runs slowly. If the response times have big differences, then it means there are troubles exist in network devices.

**Network utilization**

“The network utilization represents the percent of time that the network is in use over a given period” (Blum, 2003). For example, although Ethernet is a shared network, only one packet transmits at a time. So for any given time, the Ethernet is either at 100% utilization or at 0% utilization.

To calculate network utilization on a network segment could be easy. However, to determine the network utilization between two separate endpoints on the network can be more complex. Therefore, most network performance measuring tools use network throughput and network bandwidth to decide network performance between two remote endpoints.

**Network throughput**

“The throughput of network represents the amount of network bandwidth available for a network application at any given moment, across the network links” (Blum, 2003).

For TCP (Transmission Control Protocol) throughput, there are several aspects should be taken into consideration. These are socket buffer size, TCP send performance, TCP receive performance and TCP peak performance. (Borriss, Dannowski, & Hartig, 1998)

Network throughput can help a network administrator find the bottlenecks of network
path. For example (Figure 2-2), even though a client computer and a server are each connected with a 100 Megabit Ethernet, if these two 100 Megabit Ethernet are connected by a 10 Megabit Ethernet, then this intermediate 10 Megabit Ethernet is the bottleneck of the network.

![Network Bottleneck Diagram](image)

Figure 2-2: Network Bottleneck

Network throughput is extremely dependent on the network load at any given time. Therefore, in order to obtain the correct network throughput, the best way to test network throughput is test it at different time of the day and different day of the week, so the network throughput at any network situation are able to be collected.

**Bandwidth capacity**

Bandwidth capacity is different from network throughput; it represents “the total amount of bandwidth available between two network endpoints can greatly affect the performance of a network” (Blum, 2003). Bandwidth capacity is determined by the equipments which make up the network.

Some articles state that TCP/IP protocol suit are value to network performance evaluation. The Transmission Control Protocol (TCP) and Internet Protocol (IP) are the most common protocol in networking and widely use worldwide, such as World Wide Web (WWW), Local Area Network (LAN), E-mail, E-commerce, etc. they all run over the TCP/IP protocol suit. Therefore the evaluation of network performance should focus on TCP/IP protocol suit (Gotsis, Goudos & Sahalos, 2005).
Furthermore TCP is relating to the network performance enhancement. “The TCP is the primary protocol for congestion control in the Internet.” (Aweya, Ouellette, Montuno, & Yao, 2000) Therefore TCP/IP protocol suit is important to network performance evaluation.

2.3.2 Network Performance Measuring Tools

Suitable network measurement tools are very important for network performance analysis. “The network performance tools can help the network administrator determine the status of the network, and identify the areas of the network that could be improved to increase performance. Often, network bottlenecks can be found, and simply reallocating the resources on a network can greatly improve performance, without the addition of expensive new network equipment.” (Blum, 2003) and “such tools should include a flexible workload generator, a performance measurement subsystem, and a graphical user interface.” (Wang, Dujmovic & Nathews, 2000)

Netperf is one of measurement tool which is able to analyse various aspects of network performance and runs over Microsoft Windows. “It focuses on bulk data transfer and on request/response performance for TCP” (Gotsis, Goudos & Sahalos, 2005) and Netperf measures the throughput and latency of varied types of network (Wang, Dujmovic & Nathews, 2000). And the other useful tool called Network Performance Measurement Environment (NPME) which is used in Microsoft Windows LANs is presented by (Wang, Dujmovic & Nathews, 2000); it has four predefined scalable workload types and integrated with workload generator.

The following paragraphs introduce three common open source measuring tools that can be used to help in the analysis of network performance in the search. They are D-ITG, Netperf and Iperf.

D-ITG (Distributed Internet Traffic Generator)

“D-ITG is a platform capable to produce traffic at packet level accurately 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 to generate traffic at network, transport, and application layer” (D-ITG, 2008).
D-ITG is able to perform one-way-delay (OWD) and round-trip-time (RTT) measurements, as well as packet loss evaluation, delay, jitter and throughput measurements (Avallone, etc., 2004). Some basic features include running on multi-platform, such as Linux, Microsoft Windows and Linux familiar platforms; generate multiple flows; reproduce realistic traffic patterns, such as TCP, UDP(User Datagram Protocol), ICMP, VoIP(Voice over Internet Protocol), Telnet and DNS(Domain Name System). The D-ITG architecture showed below:

![D-ITG Architecture Diagram](image)

- ITGSend – Sending processes
- ITGRecv – Receiving processes
- ITGLog – Storage server
- ITGManager – Manager for the remote control
- ITGDec – Results analysis, include packet loss, throughput, jitter, delay
  (D-ITG, 2008)

**Netperf**

“Netperf is a benchmark that can be used to measure the performance of many different types of networking. It provides tests for both unidirectional throughput, and end-to-end latency” (Netperf, 2007). The Netperf program works as a client/server application, server side command called netserver which is a server program that listens for connections from remote hosts, and in client side command called netperf which is a client program that is used to initiate the network tests with the server (Blum, 2003).
As TCP protocol is widely used to transfer streams of data, Netperf can simulate three types of TCP traffics:

- A single TCP connection used to bulk transfer a large quantity of data
- A single TCP connection used to transfer client requests and server responses
- Multiple request/response pairs, each one a separate TCP connection (Blum, 2003)

Netperf sends blocks of data from the client to the server, measuring how fast the data is sent and received by the hosts.

UDP provide faster speed in transfer of data, Netperf can perform two types of UDP packet tests:

- A unidirectional bulk data transfer from the client to the server
- A request/response session using UDP (Blum, 2003)

In summary, Netperf measures throughput and response time between network endpoints, using both TCP and UDP packets, and can be configured from the command-line option.

**Iperf**

Iperf is a network performance tool which has some similarities to Netperf. It provides several types of TCP and UDP communication tests between two network endpoints, and is able to perform these tests in UNIX, Linux and Microsoft Windows. “The Iperf application was designed to work as a simple, interactive application that allows network and system administrators to see how TCP socket parameters used in applications and in host configurations can affect network performance” (Blum, 2003).

Iperf can be used to evaluate the following TCP network characteristics:

- Total bandwidth of the test connection
- Stream bandwidth assigned to multiple test connections
- Default TCP windows size value used by the test host
• Default Path MTU Discovery value used by the test host
• Router handling of IP Type-of-Service (TOS) packets  
  (Blum, 2003)

Beside the TCP tests, Iperf can also be used to evaluate UDP network characteristics:

• UDP performance at a specified bandwidth
• UDP packet loss in a stream of packets
• UDP delay jitter in a stream of packets
• UDP multicast packet performance  
  (Blum, 2003)

The following table is the brief comparison of these three performance tools:

<table>
<thead>
<tr>
<th></th>
<th>Netperf 2.1</th>
<th>Iperf 2.0.2</th>
<th>D-ITG 2.61</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Measurement</strong></td>
<td>TCP, UDP, SCTP</td>
<td>TCP, UDP</td>
<td>TCP, UDP, ICMP, DCCP, SCTP, VoIP, Telnet and DNS</td>
</tr>
<tr>
<td><strong>Protocol</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Measurements</strong></td>
<td>Throughput, response time</td>
<td>Bandwidth, Jitter,</td>
<td>Throughput, Jitter, Delay (OWD and RTT), Packet loss rate and able to generate multi flow</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Packet loss rate</td>
<td></td>
</tr>
<tr>
<td><strong>Interface</strong></td>
<td>Command line</td>
<td>Command line</td>
<td>Command line &amp; Graphic User Interface</td>
</tr>
<tr>
<td><strong>Topology</strong></td>
<td>client - server</td>
<td>client - server</td>
<td>client - server</td>
</tr>
<tr>
<td><strong>Operating System</strong></td>
<td>Windows, Unix and Linux</td>
<td>Windows, Unix and Linux</td>
<td>Windows and Linux</td>
</tr>
</tbody>
</table>

Table 2-1: Comparison of network performance tools
2.4 Network Performance Enhancement

Nowadays, network is almost utilised in each enterprise and school, it is everywhere. Utilising bandwidth efficiently to avoid unnecessary bandwidth cost and achieving optimization of whole network performance is a significant issue. Some elements impact or enhance network performance and will be discussed in the following section.

Network design determines the network speed and impacts network performance enhancement, as Ravikumar, Pandit & Mishra (1998) point out that “the number of users may grow way beyond the number for which the network was originally designed, resulting in a performance degradation.” A good network design plan not only satisfies the requirements, but also should achieve network further growth at a less cost. Normally, a good network design should contain requirements as shown below.

- Reliability - The network must work properly, “in order to design a network that performs well even in the presence of failures, and which delivers the best average performance over a period of time, performance and reliability have to be considered together” (Sesmun & Turner, 2000). The network should be able to meet requirements for user’s everyday working, at reasonable network speed and reliability to provide user’s point-to-point connections.

- Scalability and Extensibility – “Scalability and extensibility are the hallmarks of a good network design” (Raza & Turner, 2002). The network should be scalable and extensible; the original design is able to accommodate network growth without having major change to the whole network.

- Adaptability – Long term perspective should be adopted when designing a network and it should be taken into consideration during the further development. Work related to network performance has focused on raw bandwidth performance and adaptability (Evans & Hood, 2005).

- Easy to manage – the network should support monitored and managed, in order to keep the network run stably.

To determine a network is stable or not depends on some key servers and services if run stably. “Keep the system as simple and clean as possible and only give the user what they need to do the job and then you will have a well running and stable
network" (Shields, 2006). In order to keep the whole network runs smoothly, a stable network system is needed.

QoS (Quality of Service) is able to ensure network key service running stably; it reserves a fixed bandwidth for key data packets, so the performances of key network services are guaranteed. However, adopting QoS service means about 20% bandwidth wasted; therefore, shutting down the QoS service will enhance the network performance. “Quality of service (QoS) refers to the network’s policies that give preferential treatment to certain classes of traffic, and is required whenever delay-sensitive, business-critical applications transit the LAN. Preferential treatment could mean limiting the bandwidth used by certain applications, such as e-mail, or ensuring fairness for all the users of an important application, such as SAP. Whether the network supports automated deployment of QoS or it is done through the use of management tools, QoS is an important part of the overall equation.” (Sage, 2006)

The configuration of a switch is one of the key elements to improve network performance. VLAN (Virtual Local Area Network) is a common method to enhance network performance with switch configuration. VLAN is a technique based on the segmentation of any switched LAN into multiple logical LANs (Mahmood & Mahmood, 2008). The obvious benefit of VLAN is to prevent the broadcast storm. “Multicasting/VLAN techniques are used to enhance the real time performance of industrial Ethernet by modifying packet generation process” (Mahmood & Mahmood, 2008). Normally, network performance degradation because of broadcast packets is more than 30% in a network, thus separating computers into VLANs will enhance network performance. As Guo & Zhuang (1997) point out that using VLAN technology is able to reduce the network traffic and improve network performance. Other method of switch configuration to enhance network performance is adopting the full duplex mode for the network card and switch.

The network administrator should well understand what exactly is transmitting on the network in order to control the network performance. There is a great impact to the network performance when some BitTorrent-like Peer-to-Peer applications are running over the network. Qiu & Srikant (2004) stated that “in P2P file sharing, the number of peers in the system is an important factor in determining network performance.” If the BitTorrent-like Peer-to-Peer applications are not necessary for
business, restricting these kinds of applications running over the network would enhance network performance.

In the security area, some activities will have great impact to the network performance, such as DoS (Denial of Service) and DDoS (Distributed Denial of Service) attacks from external network and port scans. The enhancement of network performance was hampered by these activities. So to setup a firewall is able to solve these kinds of network performance issues. On the other hand, the internal computers are needed to be scanned to ensure all backdoors are closed, and the log files should be checked regularly. The most important thing is to pay close attention to the information of system upgrade; many bug-fixes from the system upgrade will avoid potential security issues.

In a network, operating system is the essential application for end user, as the Microsoft Windows has about 90% market share (NetApplications.com, 2008), so to enhance Microsoft Windows network performance is one of useful way to increase whole network performance. Moulton (2003) in his article pointed out that “to improve Windows Network performance, all Windows networks should use TCP/IP. There is no need for NETBEUI. The NETBIOS information can be configured to run over TCP/IP.” The Windows operating system can only use the TCP/IP protocol in network connection for normal usage and uninstalling other unnecessary protocols will enhance network performance. Also Sheesley (2004) indicates that one way to increase Windows 98 network performance is to reduce the number of protocols, “One of the most common reasons why network performance suffers in a Windows environment is from running too many, and often unnecessary, protocols on the workstation.” The second method from Sheesley is stopping browser elections: “a list of available network resources is controlled by one of the computers on the network, which is called the Master Browser,” usually the primary domain controller is supposed to be the master browser automatically, and the Windows 98 could flood a large network with election traffic, so “you can reduce or eliminate this problem by stopping the Windows 98 workstation from participating in elections and allowing itself to be eligible to be a master browser.” The third method is to Modify MTU size, “One way you can increase performance on a Windows 98 workstation is by modifying how it uses TCP/IP. One easy way to do so is by tweaking the Maximum Transmission Unit (MTU) size. MTU size reflects the maximum packet size (in bytes)
that TCP/IP will transmit over your network.” and Sheesley points out that the most proper MTU size for Windows 98 is 1454.

There is a gap in the research reporting about the network performance of Windows operating systems in different network situations, and there is a need to research the network performance of Windows operating systems under different network tasks.

2.5 Chapter Summary

In this chapter, the literature of network performance has been reviewed, including what makes network performance issues, five network performance measurements, three network performance measuring tools and network performance enhancement.

In the next chapter, research methodology will be introduced.
3.0 Research Methodology

This chapter starts with the research question and method of study, and describes the experiment design adopted for the data gathering of the research.

3.1 Research Questions

The primary research question is: Does a newer Microsoft operating system give better network performance?

And the secondary research questions are:

- What are the differences between the network performance of different Windows operating systems?
- Which Windows operating system has the best network performance?
- What suggestions may improve the network performance of Windows operating systems?

3.2 Method of Study

The main methodology used in this research is the quantitative approach. The focus of quantitative method is to utilize numbers to obtain information (Neill, 2003). Quantitative data are important and essential for this research, where the data are obtained from experiments and are used to analyse the results of the tests. This research study intends to analyse the collected data from experiments, in order to evaluate the network performance of Windows operating systems.

Hopkins (2000) has stated there are two main types of quantitative study: descriptive and experimental. In this quantitative research, the experimental approach has been selected. To evaluate the network performance of Microsoft Windows operating systems is the main subject of this experimental study which will involve repeated measurements of network performance. Descriptive statistics are going to be utilized in the data analysis.

This experiment will be used to deliver initial data, which other researchers could later build on.
3.3 Experimental Design

This section explains the details of the experiment. The goal of the experiment is to evaluate network performance of Microsoft Windows operating systems. These operating systems include Windows XP Professional, Windows Server 2003 Enterprise, Windows Vista Business, Windows Server 2008, and Windows 7 Beta.

3.3.1 Testbed Design and Experimental Tasks

This section will explain the details of the testbed which includes hardware and software specifications, network design, network layer and experimental tasks.

Hardware Specification

In testbed design, in order to minimize the impact of the computer capacity on measured network performance, experimental computers should have the same hardware configuration, as well as use the same device drivers in this experiment (Rindos, Loeb, Hirasawa, Woollet, & Zaghloul, 1999). Therefore the hardware specification of sender and receiver are the same, the following table presents the hardware detail:

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>Intel Core 2 Duo E6300 @ 1.866 GHz</td>
</tr>
<tr>
<td>Hard Drive</td>
<td>Hitachi HDS721616PLA380 (160 GB, 7200 RPM, SATA-II)</td>
</tr>
<tr>
<td>Memory</td>
<td>2GB, Samsung M3 1GB DDR2-667 PC2-5300(333MHz) x 2</td>
</tr>
<tr>
<td>Motherboard</td>
<td>ThinkCentre M55</td>
</tr>
<tr>
<td>Motherboard Chipset</td>
<td>Intel Broadwater Q965 Rev.C1 82801HB/HR (ICH8/R)</td>
</tr>
<tr>
<td>BIOS</td>
<td>Brand: Lenovo&lt;br&gt;Version: 2JKT39AUS&lt;br&gt;Date: 10/29/2007</td>
</tr>
<tr>
<td>Network Card</td>
<td>Broadcom NetXtreme Gigabit Ethernet</td>
</tr>
<tr>
<td>Switch</td>
<td>100 Megabit switch</td>
</tr>
</tbody>
</table>

Table 3-1: Experimental Hardware Specification

Software Specification

There are five Microsoft Windows products that are tested in this experiment. In order to keep all Windows operating systems up to date, all operating systems are installed with latest Service Pack (SP), these Windows operating systems are
showed below:

- **Windows XP Professional SP3 version 5.1.2600**
  Windows XP was released on 25th October, 2001. The latest Service Pack 3 was released on 21st April, 2008.

- **Windows Server 2003 Enterprise SP2 version 5.2.3790**
  Windows Server 2003 was released on 24th April, 2003. The latest Service Pack 2 was released on 13th March 2007.

- **Windows Vista Business SP1 version 6.0.6001**
  Windows Vista was released on 30th January, 2007. The latest Service Pack 1 was released on 25th April, 2008.

- **Windows Server 2008 Enterprise SP1 version 6.0.6001**
  Windows Server 2008 was released on 27th February, 2008. The latest Service Pack 1 was released on 17th September, 2008.

- **Windows 7 Beta version 6.1.7000**
  Windows 7 Beta is the preview version of Windows 7; it was released on 17th January, 2009.

**Network Design**

The network used in the experiment contains two computers connected by a 100 Megabits switch. One computer is acting as sender, responding to sent data, decoding log files and recording results; the other computer acts as a receiver, responding to received data. The following figure shows the network design of this experiment:

![Network Design of Experiment](image)

**Network Layer**

This experiment runs under Internet Protocol version 4 (IPv4) and Internet Protocol version 6 (IPv6). When the experiment is testing under IPv4, all IPv6 protocols are disabled; conversely, all IPv4 protocols are disabled when the experiment is testing...
under IPv6.

**Experimental Tasks**

Some tasks are chosen to measure network performance of Windows operating systems. In tests regarding transport layer, Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) have been chosen to measure the network performance. In addition, some types of network traffic in the application layer are adopted for measuring: these include Domain Name System (DNS), Voice over Internet Protocol (VoIP), and two popular multiplayer games (Counter Strike and Quake III). During the test, the same Windows operating system is used in both the sender and the receiver in order to test the operating system’s influence on network performance.

**3.3.2 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: 64, 128, 256, 384, 512, 640, 768, 896, 1024, 1152, 1280, 1408 and 1536.

**3.3.3 Task Duration and Result Objectivity**

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 10 times. Results are taken as the average of the 10 results.

**3.3.4 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” (D-ITG, 2008).
The latest stable version of D-ITG is 2.6.1d, this version of D-ITG is used in the experiments. The multiplatform source code and binary files for Windows of D-ITG 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.*” (Avallone, Botta etc., 2008)

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

- **-a**
  Set the destination IP address, for example: 192.168.1.2

- **-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.

- **-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).

- **-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.

- **-T**
  Set the protocol type. Valid values are UDP, TCP, ICMP, SCTP, and DCCP. In this experiment, TCP and UDP are adopted.
- **-t** Set the generation duration. It is expressed in milliseconds (ms). In this experiment, the duration is 60000 ms.
- **DNS** Generate traffic with DNS traffic characteristics. In this experiment, DNS traffic is tested with both TCP and UDP layer protocols.
- **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.

![D-ITG Codecs for VoIP packet generator](image)

**Figure 3-2: D-ITG Codecs for VoIP packet generator**

(Cuturic & Lozanovski, 2006)

- **CSa** Generate gaming traffic with Counter Strike traffic characteristics related to the active phase of the game.
- **Quake3** Generate gaming traffic with Quake III Arena traffic characteristics.

**ITGRecv**

ITGRecv is a “receiver component of the D-ITG platform. It can receive flows from different senders” (Avallone, Botta etc., 2008). In the experiment, ITGRecv runs on
the destination computer to receive data flow from the sender.

ITGLog
ITGLog is a “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]” (Avallone, Botta etc., 2008). In the experiment, ITGLog generates log file after each test run which contains experimental information.

ITGDec
“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 format. The Total time of the experiment is calculated as the difference between receiving time of last and first packet” (Avallone, Botta etc., 2008). In the experiment, ITGDec runs on sender computer to decode the log files which generated by ITGLog. All raw data are decoded by ITGDec into the readable text files.

The sample commands of D-ITG that are used in this experiment:

1. start the receiver on the destination host (192.168.1.2)
   
   ./ITGRecv

2. start the sender on the source host (192.168.1.1)
   
   ./ITGSend –a 192.168.1.2 -m rttm -T TCP -C 30000 -c 64 -t 60000
   ./ITGDec ITGSend.log >> IPv4_64_TCP_30000.txt

The resulting flow from 192.168.1.1 to 192.168.1.2 has the following characteristic:

- 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
This is the content of IPv4_64_TCP_30000.txt

----------------------------------------------------------
Flow number: 1
From 192.168.1.1:1451
To 192.168.1.2:8999
----------------------------------------------------------
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

3.3.4 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” (Emma, Pescape & Ventre, 2004).
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 find out the throughput and packet loss rates. Each task runs for 60 seconds. The following commands are used in the sender computer:

```
ITgsend -a 192.168.1.2 -m rttm -T UDP -c 10000 -t 60000
ITgsend -a 192.168.1.2 -m rttm -T UDP -c 20000 -t 60000
ITgsend -a 192.168.1.2 -m rttm -T UDP -c 30000 -t 60000
ITgsend -a 192.168.1.2 -m rttm -T UDP -c 40000 -t 60000
ITgsend -a 192.168.1.2 -m rttm -T UDP -c 50000 -t 60000
```

The result showed below:

<table>
<thead>
<tr>
<th>Packet Rate (pkt/s)</th>
<th>10000</th>
<th>20000</th>
<th>30000</th>
<th>40000</th>
<th>50000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packet Loss (%)</td>
<td>0</td>
<td>0.01</td>
<td>0.78</td>
<td>27.88</td>
<td>42.54</td>
</tr>
</tbody>
</table>

Table 3-1: Result of Packet Rate

The following figure gives a clear view of the result:
The above figure indicates that the respective packet loss rates of packet rates 10,000; 20,000 and 30,000 are close to zero. But the loss rate increases sharply to nearly 28% in packet rate 40,000, and it even 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, in figure 3-2, although the packet rates 40,000 and 50,000 send 33% more and 66% more packets per second 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 which Emma, Pescape & Ventre (2004) mentioned in their study.

3.3.6 Measurements

In order to evaluate network performance for Windows operating system, parameters throughput, jitter and round trip time are adopted in this experiment. The following section describes the details of these measurements.
**Throughput**

“The throughput of network represents the amount of network bandwidth available for a network application at any given moment, across the network links” (Blum, 2003). 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, caused by network congestion, timing drift, or route changes (Santkuyl, 2008). 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 millisecond. In this experiment the unit of jitter is represented in millisecond (ms). D-ITG calculates jitter according to the following formula:

![Jitter Formula](image)

*Figure 3-4: Jitter Formula (Avalone, Botta etc., 2008)*

**Round Trip Time (RTT)**

On the network, the Round Trip Time is the time that a packet takes to travel round between the source node and destination node. It also recognized Response Time. In this experiment, the unit of Round Trip Time is represented in millisecond (ms). D-ITG calculates the delay standard deviation according to the following equation:

\[ \sigma = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (d_i - \hat{d})^2} \]

*Figure 3-5: Round Trip Time equation (Avalone, Botta etc., 2008)*

Where \( N \) is the number of packets considered, \( d_i \) is the delay of packet \( i \), and \( \hat{d} \) is
3.3.7 IPv6 Configuration

This section shows how IPv6 was installed and configured in all Microsoft Windows network applications used in the experiment. To setup an IPv6 test environment, three important things are needed:

- Switch,
- IPv6 supported operating system and,
- IPv6 supported network measuring tool.

In this experiment, the IPv6 supported network measuring tool D-ITG was downloaded from the official website. However, not all Microsoft Windows operating systems come with IPv6 supported; Windows XP and Windows Server 2003, both require the installation of a recent Service Pack to implement IPv6. Microsoft Windows Vista, Windows Server 2008 and Windows 7 on the other hand all include both IPv4 and IPv6 protocol stacks that are installed and enabled by default, and they have Graphical User Interface (GUI)-based configurations.

To install IPv6 in Windows XP SP3 and Windows Server 2003 SP2, the first step is to start a command prompt session by clicking Start, pointing to Programs then to Accessories, and clicking Command Prompt. Next the following command should be typed into the command prompt:

`ipv6 install`

A message “Installing…” comes up, followed by “Succeeded”. The final view in this process is shown in figure 3-6 below:

![Figure 3-6: Windows XP and 2003 IPv6 Installation](image)

The next step is to use the command `netsh` to configure the IPv6 address according
To configure the IPv6 address, type the following:

1. `netsh`
2. `interface ipv6`
3. `add address "Local Area Connection" 2001:da8:207::9402`

When a message “Ok” shows, this IPv6 address is assigned to the operating system.

Similar to figure 3-7:

![Figure 3-7: Assign IPv6 address](image1)

The command “ipv6 if” causes the computer to check the address information after assigning the IPv6 address. To run this check, type the following in the **Command Prompt** session:

`ipv6 if`

The IPv6 addresses found will be shown under the heading “Interface: Local Area Connection” as in figure 3-8:

![Figure 3-8: IPv6 information](image2)

Configuration of IPv6 is easier in Windows Vista, Windows Server 2008 and Windows 7. The IPv6 protocol for Windows Vista, Windows Server 2008 and Windows 7 is installed and enabled by default. Thus, to take Windows Vista as an example, the operating system uses GUI-based configuration to input IPv6 addresses for the system and a similar process is inherent in Windows Server 2008.
and Windows 7.

In Windows Vista, IPv6 appears as the Internet Protocol Version 6 (TCP/IPv6) component (figure 3-9) on the Networking tab when viewing the properties of “Local Area Connection” in the Network Connections folder (available from the Network and Sharing Centre: figure 3-10)

Figure 3-9: IPv6 Component in Windows Vista

Figure 3-10: Obtain Properties of Local Area Connection

The next step is to select the Internet Protocol Version 6 (TCP/IP), clicking properties. Now IPv6 address is able to manually put into address field, when the
heading of “use the following IPv6 address” is selected. Same as figure 3-11:

![IPv6 Properties in Windows Vista](image)

Figure 3-11: IPv6 Properties in Windows Vista

After enter the IPv6 address and clicking **OK**, the IPv6 address has been assigned to the operating system.

### 3.3.8 Multi Boot Manager

In this experiment, five Microsoft Windows operating systems are tested. For each operating system test installation of a different operating system is required. In order to avoid repeating a series of time consuming installations, these five operating systems are installed onto one hard drive. For this reason a multi boot menu is needed before a specified operating system can be selected on boot-up.

In the researcher's experience of multiple operating system environments, a multi boot menu is created automatically when an older Microsoft Windows operating system is installed first before newer operating systems. In this experiment, the older operating system was installed first, therefore the installation sequence was as follows: Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008 and finally Windows 7. To test the multi boot menu, Windows XP and Windows Server 2003 are installed first, and then a multi boot menu was displayed as in figure 3-12:
Next, Windows Vista is installed. For managing boot selection for multiple operating systems, Windows Vista uses new technology called Boot Configuration Data. The presence of this new boot configuration data in Windows Vista makes the chronological sequence of operating systems installation essential: otherwise the operating systems would not start. After all of the operating systems are installed the booting menu shows as in figure 3-13:

![Figure 3-13: All Experimental Microsoft Windows Boot Menu](image)

Figure 3-13 indicates that all of the Microsoft Windows operating systems required for the experiment have been installed on the computer’s hard drive. In the Windows Boot Manager menu, each operating system can be selected and all operating systems are available on boot-up. In this menu, Windows XP and Windows Server 2003 are not visible; however, when the field “Earlier Version of Windows” is selected, the older style of menu shown in figure 3-12 will be displayed with Windows XP and Windows Server 2003 selectable.

### 3.3.9 Backup Management

In this experiment, five Microsoft Windows operating systems are installed onto one hard drive. If some significant issues occur during the experiment, then all Windows
operating systems can be reinstalled again in the same order. To avoid repeating the complex installation processes in this eventuality, a backup solution must be applied. Because each operating system is installed on its own unique partition, it is practicable to back up the whole disk as a solution to backing up each individual operating system simultaneously. For this reason, a clone program called Symantec Ghost is adopted.

Symantec Ghost is a disk cloning program, originally produced by Binary Research Ltd. (a New Zealand company), but acquired by Symantec in 1998. The software is able to clone whole disk to an image file from which the disk can then be restored as required. To backup the entire disk two tools are needed:

- A bootable CD containing Symantec Ghost software. In this experiment, Symantec Ghost version 11.5 is adopted.
- Another hard drive: this can be a removable hard drive. This additional hard drive is used to store the image file which was created by Symantec Ghost. Because it is not possible to store the image file on the same disk that the image file backs up.

To start the process of backing up the entire disk with Symantec Ghost 11.5 we first reach the interface shown in figure 3-14,

![Figure 3-14: Symantec Ghost 11.5](image)

Then the path **Local, Disk** and then **To Image**, is selected, leading to figure 3-15.
We then select the whole disk that is going to be backed up. After clicking **OK**, figure 3-17 is shown.

A different drive must be selected to hold the image file, and the intended file name of the image is entered before clicking **Save**: upon which the backup process will start. Before pressing **Save**, it is important to make sure the recipient drive has enough space to store the image file. Since the image file will contain five Microsoft Windows operating systems, it will be very large. In this experiment, the image file is about 11 Gigabytes.
When some serious issues occur during the experiment, a disk restore process is needed. After the restore process has finished, all data that existed on the target disk will be removed, and all five Microsoft Windows operating systems are reinstalled onto the disk. For this reason it is important to backup any data existing on the target disk before the process is started. Upon starting Symantec Ghost 11.5 we reach an interface as in figure 3-14. We then select Local then Disk and then click From Image, at which point figure 3-19 is shown.

![Figure 3-18: Restore Disk from Image File](image)

In figure 3-19, we must first select the drive on which the image file is held. Then we will select the image file, and click Open to confirm our selection. The menu shown in figure 3-16 will display. We select the target disk and click OK to start the restore process. The estimate of the expected duration of this process is about half an hour.

![Figure 3-19: Restore Disk – Select Image File](image)

In this chapter, research methodology has been introduced; one main research...
question and three sub questions are leading this research. Quantitative approach has been adopted as the main research methodology to obtain data. In experimental design, network design and testbed have been introduced in depth. Five Microsoft Windows operating systems are tested under IPv4 and IPv6; they are Windows XP, Windows Server 2003, Windows Vista, Windows 2008 and Windows 7. The experiment evaluates network performance for both TCP and UDP protocols, as well as DNS, VoIP and gaming bandwidth. The test details have been defined which includes packet size, task duration, packet rate, network performance measuring tool, and measurements. Finally, some related works of experiment have been introduced, which include IPv6 configuration in Microsoft Windows, multi boot management, and backup system. The next chapter discusses data collection of this research.
4.0 Data Collection

In this chapter the methodology of data collection is described. Creswell (1994) has stated that, "the data collection steps involve (1) setting the boundaries for the study, (2) collecting information through observations, interviews, documents, and visual materials, and (3) establishing the protocol for recording information."

Initial data will be obtained through two methods, one is from literature such as books, journals, reports, and other related documents; another is from experiment conducted at UNITEC. During the literature search, key results and additional information will be recorded along with the researcher’s notes, and all useful data also will be presented in the research report.

Experimentation is one of important activities for data collection in this research; it gives an opportunity for researcher to have a direct view into the data. The data from the experiments will help the researcher to answer the research questions. In this research, the network performance of Microsoft operating systems are tested, the results determine which operating system has better network performance for specific network tasks. The operating systems are provided by UNITEC which include Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008 and Windows 7 Beta. In order to achieve objectivity repeated tests on each operating system are required. The results of the tests will be recorded in detail each time and the average values are used to minimise errors. After all the data are collected, pie charts and tables will be constructed for these data to make the results easier to read and understand.

4.1 Existing Literature Gathering

To obtain data from existing literature is an important part of data collection process. The literature contains concepts and provides the known facts in the same area of the research for the reader.

The literature included journals, reports, conference proceedings, books and online sources. The sources for the literature are shown in the following:

- Electronic Database, such as IEEE, Association for Computing Machinery
Digital Library and EbscoHost.

- Libraries, such as UNITEC library.
- Online search engines, such as Google.

## 4.2 Primary Data Gathering

The major data source for the research is the experiment. The data collected from the decoded log files which are created by network measuring tool called D-ITG.

### 4.2.1 Data Entry

The sample log file is shown in figure 4-1. There are three items recorded: average delay, average jitter and average bit rate; they are representing Round Trip Time, Jitter and Throughput.

<table>
<thead>
<tr>
<th>Flow number: 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>From 192.168.1.1:4256</td>
</tr>
<tr>
<td>To 192.168.1.2:9999</td>
</tr>
<tr>
<td><strong>Total time</strong></td>
</tr>
<tr>
<td><strong>Total packets</strong></td>
</tr>
<tr>
<td><strong>Minimum delay</strong></td>
</tr>
<tr>
<td><strong>Maximum delay</strong></td>
</tr>
<tr>
<td><strong>Average delay</strong></td>
</tr>
<tr>
<td><strong>Average jitter</strong></td>
</tr>
<tr>
<td><strong>Delay standard deviation</strong></td>
</tr>
<tr>
<td><strong>Bytes received</strong></td>
</tr>
<tr>
<td><strong>Average bitrate</strong></td>
</tr>
<tr>
<td><strong>Average packet rate</strong></td>
</tr>
<tr>
<td><strong>Packets dropped</strong></td>
</tr>
</tbody>
</table>

---

**Figure 4-1: The Sample of Log File**

The collected data are entered into Excel spreadsheets using templates with different tabs. The Excel templates appear as in figure 4-2, and there are three tabs at the bottom.
The template in figure 4-2 is used for TCP and UDP tasks. For other tasks such as DNS, VoIP and gaming, the template shown in figure 4-3 is adopted.

In order to improve the accuracy of data collected, the same measurement is repeated 10 times, therefore each measurement involves 10 results that need to be entered into the form. When the entire form has been filled with data, average results are obtained.

**4.2.2 Generating Charts**

After the average records are obtained, these average records are put into charts. Two types of charts are used by the researcher to analyse results. One is a line chart, that is able to show the trends for the data, researcher is able to observe changes in each stage, such as in figure 4-4.
Another type of chart to be used is the bar chart, it shows the differences in the data, through which the researcher is able to make comparison easily. Figure 4-5 shows the example of bar chart.

Line charts are used for TCP and UDP results, and bar charts are used for DNS, VoIP and gaming results.

4.3 Participants

All data are collected by researcher individually.
4.4 Chapter Summary

In this chapter, the methodology of data collection has been introduced. Two major approaches are leading data gathering, they are literature review and experiment. There are three ways to obtain existing literature: electronic databases, libraries and online search engines. Experiment has been adopted for collecting primary data. In the data entry of experiment, two excel templates are used to hold the records and calculate the average results. After all average records are generated, these data are put into line charts or bar charts for analysis.
5.0 Data Analysis

This chapter analyses experimental results, all results of experiment are shown in the appendices with tables. These results have been put into charts in this chapter. There are five sections in this chapter, TCP analysis, UDP analysis, DNS analysis, gaming analysis and VoIP analysis.

5.1 TCP Analysis

This section analyses TCP performance in five Microsoft Windows operating systems which includes throughput, round trip time and jitter.

5.1.1 TCP Throughput

The following line chart shows the TCP Throughput of five Windows operating systems with different packet sizes in both IPv4 and IPv6 networks.

![TCP - Throughput](image)

Figure 5-1: TCP Throughput of Five Microsoft Windows

From the above line chart, the following conclusions can be made:

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

2. From packet sizes of 384 bytes to that of 1536 bytes, Windows XP has higher TCP throughput than the other Windows operating systems in IPv4 network (with the exception of 768 bytes, for which only Windows 7 has higher throughput).

3. From packet sizes of 1024 bytes to 1536 bytes, Windows 2003 has lower TCP throughput than other Windows operating systems in both IPv4 and IPv6 networks.

The following line chart is an enlargement of figure 5-1, showing the TCP throughput of three Windows client operating systems with packet size of 384 bytes to 1536 bytes in both IPv4 and IPv6 networks.

![TCP - Throughput](image)

**Figure 5-2: TCP Throughput of Three Microsoft Windows (XP, Vista, 7)**

From above line chart, the following conclusions can be made:

1. At packet size of 768 bytes, Windows 7 has higher TCP throughput than other Windows operating systems in IPv4 network.

2. From packet sizes of 384 bytes to 1536 bytes (with the exception 768 bytes as in point 1), Windows XP has higher TCP throughput than other Windows operating systems in IPv4.

3. In most packet sizes, Windows Vista has lower TCP throughput than other operating systems in both IPv4 and IPv6 networks.

4. The majority of systems using IPv4 network have higher TCP throughput than those using IPv6 network.
5. The TCP throughput of Windows 7 is between Windows XP and Windows Vista in IPv4 network.

The following line chart is the an enlargement of figure 5-1, showing TCP throughput of two Windows Server operating systems with packet size of 384 bytes to 1536 bytes in both IPv4 and IPv6 networks.

![TCP Throughput](image)

Figure 5-3: TCP Throughput of Three Microsoft Windows Server (2003, 2008)

From above line chart, the following conclusions can be made:

1. From packet sizes of 384 bytes to 1536 bytes, Windows Server 2008 has higher TCP throughput than Windows Server 2003 in IPv4 network (with the exception of packet size of 896 at which throughput for Windows Server 2003 spikes).
2. From packet sizes of 1024 bytes to 1536 bytes, Windows Server 2008 has higher TCP throughput than Windows Server 2003 in both IPv4 and IPv6 networks.
3. The majority of systems using IPv4 network have higher TCP throughput than those using IPv6 network.
5.1.2 TCP Round Trip Time

The following line chart shows the TCP round trip time of three Windows client operating systems with different packet sizes in both IPv4 and IPv6 networks.

![TCP - Round Trip Time](image)

**Figure 5-4**: TCP Round Trip Time of Three Microsoft Windows (XP, Vista, 7)

From above line chart, the following conclusions can be made:

1. At packet sizes of 64 bytes to 256 bytes and 1152 bytes to 1536 bytes, these three Windows operating systems perform similar TCP round trip time.
2. From packet sizes of 384 bytes to 1024 bytes, Windows 7 has higher TCP round trip time than other Windows operating systems in IPv6 network.
3. Windows XP has lower TCP round trip time than other Windows operating system in IPv6 network.
The following line chart shows the TCP round trip time of two Windows Server operating systems with different packet sizes in both IPv4 and IPv6 networks.

From above line chart, the following conclusions can be made:
1. From packet sizes of 64 bytes to 256 bytes, the TCP round trip time of two Windows Server operating systems are similar.
2. From packet sizes of 512 bytes to 896 bytes, the TCP round trip time of Windows Server 2003 is lower than Windows Server 2008 in both IPv4 and IPv6 networks.
3. From packet sizes of 1280 bytes to 1536 bytes, the TCP round trip time of Windows Server 2008 is lower than Windows Server 2003 in both IPv4 and IPv6 networks.
4. At packet size of 384 bytes, two Windows Server operating systems almost reach the peak of round trip time in both IPv4 and IPv6 networks.

Figure 5-5: TCP Round Trip Time of Two Microsoft Windows Server (2003, 2008)
5.1.3 TCP Jitter

The following line chart shows the TCP jitter of five Windows operating systems with different packet sizes in both IPv4 and IPv6 networks.

![TCP Jitter Chart](image.png)

Figure 5-6: TCP Jitter of Five Microsoft Windows

From above line chart, the following conclusions can be made:

1. From packet sizes of 64 bytes to 768 bytes, these five Windows operating systems have similar TCP jitter in both IPv4 and IPv6 networks.

2. From packet sizes of 1024 bytes to 1536 bytes,
   - Windows XP and Windows 7 have similar TCP jitter and lower than other Windows operating systems in both IPv4 and IPv6 networks.
   - Windows Vista and Windows Server 2008 have similar TCP jitter in both IPv4 and IPv6 networks, but perform higher than Windows XP and Windows 7.
   - Windows Server 2003 has much higher TCP jitter than other Windows operating systems.
5.2 UDP Analysis

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

5.2.1 UDP Throughput

The following line chart shows the UDP throughput of five Windows operating systems with different packet sizes in both IPv4 and IPv6 networks.

From above line chart, the following conclusions can be made:

1. From packet sizes of 64 bytes to 256 bytes, five Windows operating systems have similar UDP throughput in both IPv4 and IPv6 networks.

2. From packet sizes of 1152 bytes to 1536 bytes, five Windows operating systems have similar UDP throughput in both IPv4 and IPv6 networks.

3. From packet sizes of 384 bytes to 1024 bytes, Windows Server 2003 has lower UDP throughput than other Windows operating systems in both IPv4 and IPv6 networks.
The following line chart is the enlargement of figure 5-7, shows UDP throughput of three Windows client operating systems with packet sizes of 384 bytes to 1536 bytes in both IPv4 and IPv6 networks.

From above line chart, the following conclusions can be made:
1. Windows XP has little higher UDP throughput than other Windows operating systems in IPv4.
2. From packet sizes of 768 bytes to 1024 bytes, Windows 7 has higher UDP throughput than Windows Vista, but lower than Windows XP in IPv4 network.
3. These three Windows operating systems have similar UDP throughput in IPv6 network.
4. The majority of systems using IPv4 network have higher UDP throughput than those using IPv6 network.
The following line chart shows UDP throughput of two Windows Server operating systems with different packet sizes in both IPv4 and IPv6 networks.

From above line chart, the following conclusions can be made:

1. From packet sizes of 64 bytes to 1024 bytes, Windows Server 2008 has higher UDP throughput than Windows Server 2003 in both IPv4 and IPv6 networks.

2. From packet sizes of 1152 bytes to 1536 bytes, two Windows Server operating systems have similar UDP throughput in both IPv4 and IPv6 networks.

3. From packet sizes of 384 bytes to 1024 bytes, Windows Server 2008 has on average 36% more UDP throughput than Windows Server 2003 in both IPv4 and IPv6 networks.
5.2.2 UDP Round Trip Time

The following line chart shows the UDP round trip time of three Windows client operating systems with different packet sizes in both IPv4 and IPv6 networks.

From above line chart, the following conclusions can be made:

1. From packet sizes of 64 bytes to 128 bytes, Windows Vista has higher UDP round trip time than other Windows operating systems in both IPv4 and IPv6 network.

2. From packet sizes of 384 bytes to 1536 bytes, three Windows operating systems have similar UDP round trip time in both IPv4 and IPv6 networks, however, Windows Vista shows a little bit higher than other in IPv4 network.

3. From packet sizes of 64 bytes to 256 bytes, Windows XP has lower UDP round trip time than other Windows operating systems in IPv4 network.
The following line chart shows the UDP round trip time of two Windows Server operating systems with different packet sizes in both IPv4 and IPv6 networks.

![UDP Round Trip Time](image)

Figure 5-11: UDP Round Trip Time of Two Microsoft Windows Server (2003, 2008)

From above line chart, the following conclusions can be made:

1. The UDP round trip time of Windows Server 20008 is lower than Windows Server 2003 in both IPv4 and IPv6 networks.
2. Windows Server 2003 reaches the highest UDP round trip time at packet size of 384 bytes in both IPv4 and IPv6 networks.
4. From packet sizes of 384 bytes to 1536 bytes, the UDP round trip time of Windows Server 2003 is on average 37% higher than Windows Server 2008 in both IPv4 and IPv6 networks.
5.2.3 UDP Jitter

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

![UDP Jitter Chart](image)

**Figure 5-12: UDP Jitter of Five Microsoft Windows**

From above line chart, the following conclusions can be made:

1. At packet size of 896 bytes, Windows Vista has lowest UDP jitter than other Windows operating systems in IPv4 network.
2. From packet sizes of 640 bytes to 1024 bytes, Windows Server 2003 has 40% higher UDP jitter than other Windows operating systems in both IPv4 and IPv6 networks.
3. From packet sizes of 384 bytes and 640 bytes, Windows Server 2003 has lower UDP jitter than other Windows operating systems in both IPv4 and IPv6 networks.
4. Except Windows Server 2003, other Windows operating systems have similar UDP jitter in both IPv4 and IPv6 networks.
5.3 DNS Analysis

This section analyses DNS performance in five Microsoft Windows operating systems which includes throughput, round trip time and jitter.

5.3.1 TCP Throughput of DNS

The following bar chart shows the TCP throughput of DNS of five Windows operating systems in both IPv4 and IPv6 networks.

From above bar chart, the following conclusions can be made:
3. Windows 7 has lower TCP throughput of DNS than Windows Vista in IPv4 network.
4. Windows Server 2008 and Windows 7 have higher TCP throughput of DNS than other Windows operating systems in IPv6 network.
5. Windows XP has higher TCP throughput of DNS than Windows Vista in IPv6 network.
6. Windows Vista has lower TCP throughput of DNS than other Windows operating systems in both IPv4 and IPv6 networks.
7. Except Windows Server 2003, other operating systems using IPv6 network have
higher TCP throughput of DNS than those using IPv4 network.

5.3.2 TCP Round Trip Time of DNS

The following bar chart shows the TCP round trip time of DNS of five Windows operating systems in both IPv4 and IPv6 networks.

![TCP Round Trip Time Chart](chart.png)

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

1. TCP round trip time of DNS of all five Windows operating systems is similar.
2. Windows XP has lower TCP round trip time of DNS than other Windows operating systems in both IPv4 and IPv6 networks.
3. Windows 7 has higher TCP round trip time of DNS than other Windows operating systems in both IPv4 and IPv6 networks.
4. All five Windows operating systems using IPv4 network have lower TCP round trip time of DNS than those using IPv6 network.
5.3.3 TCP Jitter of DNS

The following bar chart shows the TCP jitter of DNS of five Windows operating systems in both IPv4 and IPv6 networks.

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

1. Five Windows operating systems have similar TCP jitter of DNS in IPv4 network.
2. Windows XP has lowest TCP jitter of DNS than other Windows operating systems in IPv6 network.
3. Windows 2003 has highest TCP jitter of DNS than other Windows operating systems in IPv6 network.
5.3.4 UDP Throughput of DNS

The following bar chart shows the UDP throughput of DNS of five Windows operating systems in both IPv4 and IPv6 networks.

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

1. Two Windows Server operating systems have higher UDP throughput of DNS than other Windows operating systems in IPv4 network.
2. Windows Vista has lower UDP throughput of DNS than other Windows operating systems in IPv4 network.
3. Windows 7 has lower UDP throughput of DNS than Windows XP in both IPv4 and IPv6 networks.
4. Windows Server 2008 has higher UDP throughput of DNS than other Windows operating systems in IPv6 network.
5.3.5 UDP Round Trip Time of DNS

The following bar chart shows the UDP round trip time of DNS of five Windows operating systems in both IPv4 and IPv6 networks.

![UDP - Round Trip Time](data:image/png;base64,base64encodedimage)

Figure 5-17: UDP Round Trip Time of DNS of Five Microsoft Windows

From above bar chart, the following conclusions can be made:
1. UDP round trip time of DNS of all five Windows operating systems is similar.
2. Windows XP has lower UDP round trip time of DNS than other Windows operating systems in both IPv4 and IPv6 networks.
3. Windows 7 has higher UDP round trip time of DNS than other Windows operating systems in both IPv4 and IPv6 networks.
4. IPv4 has lower UDP round trip time of DNS than IPv6 in all five Windows operating systems.
5.3.6 UDP Jitter of DNS

The following bar chart shows the UDP jitter of DNS of five Windows operating systems in both IPv4 and IPv6 networks.

From above bar chart, the following conclusions can be made:
1. UDP jitter of DNS of all five Windows operating systems is similar.
2. Windows XP has higher UDP jitter of DNS than other Windows operating systems in IPv4 network.
3. Windows Vista has higher UDP jitter of DNS than other Windows operating systems in IPv6 network; however it has the lowest UDP jitter in IPv4 network.
5.4 Gaming Analysis

This section analyses network performance of two games in five Microsoft Windows operating systems which includes throughput, round trip time and jitter.

5.4.1 Throughput of Counter Strike

The following bar chart shows the Counter Strike (game) throughput of five Windows operating systems in both IPv4 and IPv6 networks.

![Counter Strike - Throughput](image)

Figure 5-19: Counter Strike (Game) Throughput of Five Microsoft Windows

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

1. Windows XP and Windows Server 2003 have higher throughput than other Windows operating systems in game Counter Strike in IPv4 network.
3. Windows Vista has lower throughput than other Windows operating systems in game Counter Strike in IPv6 network.
4. Windows 7 has higher throughput than other Windows operating systems in game Counter Strike in IPv6 network.
5.4.2 Round Trip Time of Counter Strike

The following bar chart shows the Counter Strike (game) round trip time of five Windows operating systems in both IPv4 and IPv6 networks.

![Counter Strike - Round Trip Time](image)

Figure 5-20: Counter Strike (Game) Round Trip Time of Five Microsoft Windows

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

1. All five Windows operating systems using IPv4 network have lower round trip time than those using IPv6 network in the game Counter Strike.

2. Windows XP has lower round trip time than other Windows operating systems in game Counter Strike in both IPv4 and IPv6 networks.

3. Windows Server 2003 has higher round trip time than other Windows operating systems in game Counter Strike in IPv6 network.
5.4.3 Jitter of Counter Strike

The following bar chart shows the Counter Strike (game) jitter of five Windows operating systems in both IPv4 and IPv6 networks.

![Counter Strike - Jitter](image)

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

1. Windows Server 2003 has higher jitter than other Windows operating systems in game Counter Strike in IPv6 network.
2. Windows XP has lower jitter than other Windows operating systems in game Counter Strike in IPv6 network.
3. All five Windows operating systems have similar jitter in game Counter Strike in IPv4 network.
5.4.4 Throughput of Quake 3

The following bar chart shows the Quake 3 (game) throughput of five Windows operating systems in both IPv4 and IPv6 networks.

![Throughput (Quake3)](image_url)

Figure 5-22: Quake 3 (Game) Throughput of Five Microsoft Windows

From above bar chart, the following conclusions can be made:
1. Windows Server 2003 and Windows 7 have higher throughput than other Windows operating systems in game Quake 3 in IPv4 network.
2. Windows XP has higher throughput than other Windows operating systems in game Quake 3 in IPv6 network.
3. Windows Vista has lower throughput than other Windows operating systems in game Quake 3 in IPv6 network.
5.4.5 Round Trip Time of Quake 3

The following bar chart shows the Quake 3 (game) round trip time of five Windows operating systems in both IPv4 and IPv6 networks.

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

1. All five Windows operating systems have similar round trip time in game Quake 3 in both IPv4 and IPv6 networks.
2. Windows XP has lower round trip time than Windows Vista in game Quake 3 in both IPv4 and IPv6 networks.
5.4.6 Jitter of Quake 3

The following bar chart shows the Quake 3 (game) jitter of five Windows operating systems in both IPv4 and IPv6 networks.

From above bar chart, the following conclusions can be made:
1. Except Windows Server 2003, other Windows operating systems have similar jitter in game Quake 3 in both IPv4 and IPv6 networks.
2. Windows Server 2003 has much higher jitter than other Windows operating systems in game Quake 3 in both IPv4 and IPv6 networks.
5.5 VoIP Analysis

This section analyses network performance of VoIP in five Microsoft Windows operating systems which includes throughput, round trip time and jitter.

5.5.1 Throughput of G.711.1

The following bar chart shows the throughput of VoIP G.711 codec with 1 sample per packet in five Windows operating systems in both IPv4 and IPv6 networks.

![Throughput Chart]

**Figure 5-25: VoIP G.711.1 Throughput of Five Microsoft Windows**

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

1. Windows XP has higher VoIP G.711.1 throughput than other Windows operating systems in both IPv4 and IPv6 networks.
2. Windows Vista has lower VoIP G.711.1 throughput than other Windows operating systems in IPv4 network.
5.5.2 Round Trip Time of G.711.1

The following bar chart shows the round trip time of VoIP G.711 codec with 1 sample per packet in five Windows operating systems in both IPv4 and IPv6 networks.

From above bar chart, the following conclusions can be made:
1. All five Windows operating systems have similar round trip time in VoIP G.711.1 in both IPv4 and IPv6 networks.
2. Windows Vista has a little higher round trip time than other Windows operating system in VoIP G.711.1 in both IPv4 and IPv6 networks.
5.5.3 Jitter of G.711.1

The following bar chart shows the jitter of VoIP G.711 codec with 1 sample per packet in five Windows operating systems in both IPv4 and IPv6 networks.

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

1. Windows XP has lower jitter than other Windows operating system in VoIP G.711.1 in both IPv4 and IPv6 networks.

2. Windows Server 2003 has higher jitter than other Windows operating system in VoIP G.711.1 in both IPv4 and IPv6 networks.
5.5.4 Throughput of G.711.2

The following bar chart shows the throughput of VoIP G.711 codec with 2 samples per packet in five Windows operating systems in both IPv4 and IPv6 networks.

![Throughput Chart](image)

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

1. Windows Vista has higher throughput than other Windows operating system in VoIP G.711.2 in both IPv4 and IPv6 networks.
2. Windows XP has lower throughput than other Windows operating system in VoIP G.711.2 in IPv6 network.
5.5.5 Round Trip Time of G.711.2

The following bar chart shows the round trip time of VoIP G.711 codec with 2 samples per packet in five Windows operating systems in both IPv4 and IPv6 networks.

From above bar chart, the following conclusions can be made:
1. All Windows operating systems using IPv6 network have higher round trip time than those using IPv4 network in VoIP G.711.2.
2. Windows Vista has higher round trip time than other Windows operating systems in VoIP G.711.2 in both IPv4 and IPv6 networks.
3. Windows XP has lower round trip time than other Windows operating systems in VoIP G.711.2 in both IPv4 and IPv6 networks.
5.5.6 Jitter of G.711.2

The following bar chart shows the jitter of VoIP G.711 codec with 2 samples per packet in five Windows operating systems in both IPv4 and IPv6 networks.

![Jitter chart](image)

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

2. Windows Server 2003 has higher jitter than other Windows operating systems in VoIP G.711.2 in IPv6 network.
5.5.7 Throughput of G.723.1

The following bar chart shows the throughput of VoIP G.723.1 codec in five Windows operating systems in both IPv4 and IPv6 networks.

From above bar chart, the following conclusions can be made:
1. Except Windows XP, other Windows operating systems have higher throughput in IPv4 network than in IPv6 network.
2. Windows XP has higher throughput than other Windows operating systems for VoIP G.723.1 in both IPv4 and IPv6 networks.
3. Windows 7 has lower throughput than Windows Vista in VoIP G.723.1 in IPv4 network.
5.5.8 Round Trip Time of G.723.1

The following bar chart shows the round trip time of VoIP G.723.1 codec in five Windows operating systems in both IPv4 and IPv6 networks.

![Round Trip Time Chart](chart.png)

**Figure 5-32: VoIP G.723.1 Round Trip Time of Five Microsoft Windows**

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

1. All five Windows operating systems have similar round trip time in VoIP G.723.1 in both IPv4 and IPv6 networks.
2. Windows Vista has a little higher Round Trip Time than other Windows operating systems in VoIP G.723.1 in both IPv4 and IPv6 networks.
5.5.9 Jitter of G.723.1

The following bar chart shows the jitter of VoIP G.723.1 codec in five Windows operating systems in both IPv4 and IPv6 networks.

![Bar chart showing jitter of VoIP G.723.1](image)

From above bar chart, the following conclusions can be made:
1. Windows XP has lower jitter than other Windows operating systems in VoIP G.723.1 in both IPv4 and IPv6 networks.
2. Windows Server 2003 has higher jitter than other Windows operating systems in VoIP G.723.1 in IPv6 network.
3. Except Windows Server 2003, other Windows operating systems have lower jitter in IPv6 network than in IPv4 network.
5.5.10 Throughput of G.729.2

The following bar chart shows the throughput of VoIP G.729 codec with 2 samples per packet in five Windows operating systems in both IPv4 and IPv6 networks.

From above bar chart, the following conclusions can be made:
1. Windows XP and Windows Server 2003 have higher throughput than other Windows operating systems in VoIP G.729.2 in both IPv4 and IPv6 networks.
2. Windows 7 has lower throughput than other Windows operating systems in VoIP G.729.2 in both IPv4 and IPv6 networks.
5.5.11 Round Trip Time of G.729.2

The following bar chart shows the round trip time of VoIP G.729 codec with 2 samples per packet in five Windows operating systems in both IPv4 and IPv6 networks.

![Round Trip Time Chart](image)

Figure 5-35: VoIP G.729.2 Round Trip Time of Five Microsoft Windows

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

1. All five Windows operating systems have similar round trip time in VoIP G.729.2 in both IPv4 and IPv6 networks.
2. Windows Vista has a little higher round trip time than other Windows operating systems in VoIP G.729.2 in both IPv4 and IPv6 networks.
5.5.12 Jitter of G.729.2

The following bar chart shows the jitter of VoIP G.729 codec with 2 samples per packet in five Windows operating systems in both IPv4 and IPv6 networks.

From above bar chart, the following conclusions can be made:
1. Windows XP has lower jitter than other Windows operating systems in VoIP G.729.2 in both IPv4 and IPv6 networks.
2. Windows Server 2003 has higher jitter than other Windows operating systems in VoIP G.729.2 in IPv6 network.
5.5.13 Throughput of G.729.3

The following bar chart shows the throughput of VoIP G.729 codec with 3 samples per packet in five Windows operating systems in both IPv4 and IPv6 networks.

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

1. Windows XP has lower throughput than Windows Vista and Windows 7 in VoIP G.729.3 in IPv4 network; however it has higher throughput than Windows Vista and Windows 7 in IPv6 network.

2. Windows Server 2003 has higher throughput than other Windows operating systems in VoIP G.729.3 in both IPv4 and IPv6 networks.
5.5.14 Round Trip Time of G.729.3

The following bar chart shows the round trip time of VoIP G.729 codec with 3 samples per packet in five Windows operating systems in both IPv4 and IPv6 networks.

![Round Trip Time](image)

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

1. All five Windows operating systems have similar round trip time in VoIP G.729.3 in both IPv4 and IPv6 networks.
2. Windows Vista has a little higher round trip time than other Windows operating systems in VoIP G.729.3 in both IPv4 and IPv6 networks.
3. All five Windows operating system using IPv6 network have higher round trip time than those using IPv4 network in VoIP G.729.3
5.5.15 Jitter of G.729.3

The following bar chart shows the jitter of VoIP G.729 codec with 3 samples per packet in five Windows operating systems in both IPv4 and IPv6 networks.

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

1. Windows XP has lower jitter than other Windows operating systems in VoIP G.729.2 in IPv4 network.

2. Windows Server 2003 has higher jitter than other Windows operating systems in VoIP G.729.2 for both IPv4 and IPv6 networks.

5.6 Chapter Summary

In this chapter, the experimental results have been put into line charts and bar charts for analysis and comparison. These figures illustrate the network performance of various Microsoft Windows operating systems (Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008 and Windows 7) for both TCP and UDP protocols, as well as DNS, VoIP and gaming bandwidths.

In next chapter, discussions of these results are introduced.
6.0 Discussion

The previous chapter of the research compares findings from the results of the experiment. This chapter discusses and summarizes these findings, the analysis of results and relevant literature.

6.1 Summary of Findings

This section presents the findings of this research from the experiment and the review of literature. Five Microsoft Windows network systems were tested with different types of traffic, such as TCP, UDP, DNS, VoIP and gaming in both 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.

6.1.1 TCP Performance

The results of section 5.1 indicate that a majority of IPv4 networks have higher TCP performance than IPv6 networks, with the exception of packet size of 1024 bytes, for which Windows 7 has higher TCP performance in IPv6 than in IPv4. At packet sizes between 64 bytes and 256 bytes, all five Microsoft Windows operating systems have the same TCP performance: their difference being less than 0.2%. For packet sizes of 384 bytes to 1536 bytes, Windows XP has on average 3% higher TCP performance than Windows Vista in both IPv4 and IPv6 networks, and Windows 7 has on average 1% higher TCP performance than Windows Vista in both IPv4 and IPv6 networks. In respect of Windows Server operating systems on the other hand, Windows Server 2008 has on average 5.4% higher TCP performance than Windows Server 2003 under IPv4 network and on average 5.6% higher under IPv6 network.

The above results show that Windows Vista is not able to give better TCP network performance than its predecessor Windows XP whether under IPv4 or IPv6 networks; however Windows 7 Beta shows that it might be able to overcome this shortcoming with the release of the authorised version of this system. On the other hand, Windows Server 2008 gives better TCP performance than its predecessor Windows Server 2003: it increases on average 5.5% on both IPv4 and IPv6 networks.
6.1.2 UDP Performance

The results of section 5.2 indicate that the IPv4 network has a higher UDP performance than IPv6. From packet sizes of 384 bytes to 1536 bytes, Windows XP, Windows Vista and Windows 7 have the same UDP performance, their difference being less than 0.35% in both IPv4 and IPv6 networks. From packet sizes of 64 bytes to 256 bytes, Windows XP has on average 5.06% higher UDP performance than Windows Vista in IPv4 and IPv6, and Windows 7 has on average 5.21% higher UDP performance than Windows Vista in both IPv4 and IPv6 networks. With regard to Windows Server operating systems, from packet sizes of 1152 bytes to 1536 bytes, Windows Server 2003 and Windows Server 2008 have the same UDP performance, their difference being less than 0.1% in both IPv4 and IPv6 networks; but from packet sizes of 64 bytes to 256 bytes Windows Server 2008 has on average 14.1% higher UDP performance than Windows Server 2003. There is quite a difference between Windows Server 2003 and Windows Server 2008 when the packet size is changed from 384 bytes to 1024 bytes, the UDP performance of Windows Server 2003 drops dramatically in both IPv4 and IPv6 networks, leaving Windows Server 2008 on average 36.54% higher performance than Windows Server 2003. The maximum difference even saw Windows Server 2008 reach 67% higher performance in packet size of 384 bytes.

The above results show that Windows Vista is still not able to give better UDP network performance than its predecessor Windows XP whether under IPv4 or IPv6 networks, while Windows 7 Beta has similar UDP network performance to Windows XP. On the other hand, Windows Server 2008 offers much better UDP performance than its predecessor Windows Server 2003, increasing on average 36.54% in both IPv4 and IPv6 networks.

6.1.3 DNS Performance

The results of section 5.3 indicate that a majority of IPv6 networks have better DNS performance than IPv4 networks. Under TCP traffic Windows XP has on average 2% higher DNS performance than Windows Vista in both IPv4 and IPv6 networks. Windows 7 gives 2.7% higher DNS performance than Windows Vista in IPv6 network.
but it performs 0.3% lower in IPv4. In Windows Server operating systems, Windows Server 2003 has better DNS performance than Windows Server 2008 under IPv4, while under IPv6 on the contrary, Windows Server 2008 has better DNS performance in both TCP and UDP traffic. Under UDP traffic, Windows XP still leads Windows Vista on average 1.1% in both IPv4 and IPv6 networks.

The above results show that Windows Vista is still not able to give better DNS performance than its predecessor Windows XP whether under IPv4 or IPv6 networks, but it is close to Windows XP, and Windows 7 Beta shows that it has better DNS performance than Windows Vista under IPv6 network. On the other hand, Windows Server 2008 has similar DNS performance as its predecessor Windows Server 2003 in both IPv4 and IPv6 networks.

### 6.1.4 Gaming Performance

The results of section 5.4 indicate that in gaming network performance modelled with Counter Strike, a majority of IPv6 networks have better performance than IPv4, except in the case of Windows Vista. Windows XP has 0.8% higher performance than Windows Vista in IPv4 network, and has 14.63% higher performance in IPv6. Although Windows 7 gives 2% lower performance than Windows Vista in IPv4 network, it gives 17.55% higher performance than Windows Vista in IPv6. In Windows Server operating systems, Windows Server 2003 has 2.56% higher performance than Windows Server 2008 in IPv4, and has 5.74% higher in IPv6 network.

In gaming network performance modelled with Quake 3, a majority of IPv4 networks give slightly better performance than IPv6 networks. All five Windows operating systems have similar performance in both IPv4 and IPv6 networks, their difference in gaming network performance for Quake 3 being less than 0.7%.

Therefore, the gaming network performance of Windows Vista is still not able to exceed its predecessor Windows XP; on the contrary, Windows XP has better gaming performance than Windows Vista. On the other hand, Windows Server 2003 gives slightly higher gaming performance than Windows Server 2008.
6.1.5 VoIP Performance
The results of section 5.5 indicate that different VoIP codecs have different network performance, but the five Microsoft Windows operating systems give very close VoIP performance in both IPv4 and IPv6 networks. Some significant facts that can be drawn from the findings include the following:

- VoIP in IPv4 networks is generally faster than in IPv6 network.
- Windows XP has better VoIP performance in IPv6 than in IPv4 network.
- Windows Vista has the best VoIP G.711 codec sample 2 performances compared with other Windows operating systems in both IPv4 and IPv6 networks.
- In VoIP G.729 codec sample 2, the performance of IPv6 in all Windows operating systems is better than that of IPv4 network.

The above results show the VoIP performance of Windows Vista is still not able to completely exceed its predecessor Windows XP; Windows 7 gives similar VoIP performance to Windows Vista; and Windows Server 2003 has similar VoIP performance to Windows Server 2008.

6.1.6 Performance Summary
From the discussion above, some facts are made apparent:

- IPv4 networks have higher performance than IPv6 networks for both TCP and UDP traffic. Zeadally & Raicu (2003) mention “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.” Visoottiviseth and Bureenok (2008) make the same observation.
- Windows Vista shows lower network performance than its predecessor Windows XP. However the latest Windows operating system Windows 7 Beta has overcome this shortcoming, it has better network performance than Windows Vista and comes close to that of Windows XP.
6.1.7 Findings in Literature

The above summary indicates that IPv4 networks have better performance than IPv6 networks. The main reason for this issue lies with the differences between the packet structures of these two protocols. IPv6 was designed based on IPv4 and to replace IPv4 in the near future: it adopts many new capabilities that do not exist in 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” (Zhang & Li, 2004). Although IPv6 reduces some fields in packet header, the addresses used by IPv6 are four times longer than the addresses used by IPv4; furthermore, IPv6 packet headers are twice the size of IPv4 packet headers. The following figure shows the detail of the IPv4 and IPv6 packet header structures.

![IPv4 and IPv6 Packet Header](Walton, 1999)

Unlike IPv4, IPv6 headers do not contain any ‘options’ field, “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” (Zhang & Li, 2004). 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 and particular capability. On the other hand, each IPv4 packet is made up of a packet header, options and data. The following figure shows the details of IPv4 and IPv6 packet structures.
IPv4 packets have a fixed structure which is the basic header plus variable length data. IPv6 packets have countless types of structures, which 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 kinds of possibility and the number of extension header is variable, therefore the number of IPv6 packet structure is enormous" (Zhang & Li, 2004). 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." Furthermore, Visoottiviseth and Bureenok (2008) also point out that "IPv4 yields the highest data throughput in both transmissions via TCP with no-delay option and UDP, followed by IPv6."

The previous section also indicates that Windows Vista has lower network performance than its predecessor, Windows XP; however, Windows Server 2008 has better network performance than its predecessor, Windows Server 2003. The main reason for these performance differences is that Windows Vista and Windows Server 2008 adopted the implementation of the TCP/IP protocol suite called Next Generation TCP/IP stack. The Next Generation TCP/IP stack is a complete redesign of TCP/IP functionality for both IPv4 and IPv6 networks. The new features of the Next Generation TCP/IP stack include:

- Dual IP layer architecture for IPv6
- Easier kernel mode network programming
- Support for a strong host model
- New security and packet filtering APIs
- New mechanisms for protocol stack offload
- New support for scaling on multi-processor computers
- New extensibility
- Reconfigure without having to restart the computer
- Automatic configuration of stack settings based on different network environments
- Supportability enhancements
  (Davies, 2005a)

The following figure shows the architecture of the Next Generation TCP/IP stack.

![Architecture of Next Generation TCP/IP Stack](image)

Figure 6-3: The Architecture of Next Generation TCP/IP Stack (Davies, 2005a)

The Next Generation TCP/IP stack in Windows Vista and Windows Server 2008 contain some performance enhancements to increase throughput in high-bandwidth, high-latency, and high-loss networking environments as follows:

- Receive window auto-tuning
- Compound TCP
- ECN support
- Enhancements for wireless traffic
- Improved routing path detection and recovery
  (Davies, 2005b)
Among these performance enhancements, ‘receive window auto-tuning’ and ‘compound TCP’ should be taken into account. These two enhancements might be the key elements that impact the performance of throughput as well as overall network performance.

“The TCP receive window size is the amount of data that a TCP receiver allows a TCP sender to send before having to wait for an acknowledgement. After the connection is established, the receive window size is advertised in each TCP segment. Advertising the maximum amount of data that the sender can send is a receiver-side flow control mechanism that prevents the sender from sending data that the receiver cannot store. A sending host can only send at a maximum the amount of data advertised by the receiver before waiting for an acknowledgment and a receive window size update” (Davies, 2005b).

Davies (2005b) points out that in Windows XP and Windows Server 2003, the correct size of the receive window is often hard to determine. Although Windows XP and Windows Server 2003 support scalable window, the maximum receive window size is still limiting to throughput because of the fixed maximum size that is utilised for all TCP connections (unless specified by the application). This setting enables increased throughput for some connections, but reduces throughput for others. In addition, the fixed maximum receive window size for a TCP connection does not change under different network conditions. In order to resolve this issue and correctly determine the value of the maximum receive window size for a connection based on the network, the Next Generation TCP/IP stack adopted ‘Receive Window Auto-Tuning’. This provides better throughput between TCP peers and the utilisation of network bandwidth increases during data transfer.

Davies (2005b) also points out that to better utilise the bandwidth of TCP connections for a large receive window size and large bandwidth-delay product, the Next Generation TCP/IP stack has adopted ‘Compound TCP’. ‘Compound TCP’ attempts to maximise throughput on these types of TCP connections by utilising monitored delay variations and losses.

For these reasons, the Next Generation TCP/IP stack should bring better throughput in Windows Vista and Windows Server 2008. However, the results of the experiment
suggest that the Next Generation TCP/IP stack does not bring better throughput in Windows Vista. Symantec researchers Newsham and Hoagland (2006) point out that “despite the claims of Microsoft developers, the Windows Vista network stack as it exists today is less stable than the earlier Windows XP stack. And a networking stack is a complex piece of software that takes many years to mature.” Therefore, from the results of experiment, the Windows XP operating system shows better network performance than Windows Vista as Windows Vista adopts less stable network stack. The results of experiment present that newer operating system Windows 7 Beta offers better network performance than Windows Vista as it has more matured network stack. In online gaming, Francia (2007) points out that “an extensive test of Battlefield 2, FEAR, and World of Warcraft on both Windows XP and Windows Vista has revealed that gaming performance is far better in Windows XP than in Vista, according to Bigfoot Networks. The results also showed that Vista’s networking stack not only didn’t improve online game play, it often introduced more lag than its predecessor, Windows XP.” That is the same results of the experiment.

The following figure shows comparison of network performance between Windows XP and Windows Vista with ‘Compact TCP’ enabled.

![Figure 6-4: Network Performance Comparison](image)

In figure 6-4, Smith (2007) states that “unfortunately for Vista, neither test is particularly favourable. Compared to XP when Compound TCP is disabled, Vista is anywhere between 25% and 50% slower than XP in terms of the total time required for these tests. The one bright spot however is that when enabled, Compound TCP is clearly having some effect even on our low-latency network. The 5% or so boost in
Vista’s low scores won’t bring it back above XP, but it clearly proves that Compound TCP does have a real-world effect on performance”. Although Compact TCP enabled in Windows Vista, its network performance is still lower than Windows XP.

The above literature indicates that Windows XP offers better network performance than Windows Vista. However, the results of the experiment show that the Next Generation TCP/IP stack brings better throughput in Windows Server 2008. Ward Ralston, senior technical product manager in the Microsoft Windows Server division, points out that “Windows Server 2008 and Windows Vista are introducing some of the biggest changes to our networking stack in recent memory. From the server side, I think this is one of the biggest overlooked features of Windows Server 2008. To that point, earlier this year we released the Tolly Group Whitepaper that showed how the improvements to our TCP/IP stack and SMB 2.0 Protocol gave us a 3.5x time-to-completion improvement improvement over Windows Server 2003” (Ralston, 2007).

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. The Next Generation TCP/IP stack does not bring better network performance to Windows Vista compared with Windows XP, yet it brings much better network performance to Windows Server 2008 compared with Windows Server 2003.

6.2 Further Research

This study focuses on the evaluation of network performance in Microsoft Windows operating systems; it includes different traffic types and different network layers. This study can help network administrators and network designers to target which Microsoft Windows operating system is more satisfactory for specific network tasks, enabling them to select an operating system based not only on its release date, but on the performance of the operating system relative to the particular tasks that are most important to the individual network in question. Further studies could be extended by:

- Increasing the traffic flow in order to make the network traffic of the experiment closely mimic a real network environment.
- Increasing types of traffic of application layer.
- Increasing the packet sizes and using a gigabit switch.
• Using different network measuring tools to compare the results.
7.0 Conclusion

This research is focused on the evaluation of network performance in Microsoft Windows operating systems. In order to achieve the goals of the research, an experiment has been utilized to gather initial data. In this experiment, five Microsoft Windows operating systems (Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008 and Windows 7) were tested with different types of traffic, such as TCP, UDP, DNS, VoIP and gaming in both 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. Some obvious strongly supported conclusions could be made based on the results of the experiment: IPv4 networks show better performance than IPv6 networks for both TCP and UDP traffic, Windows Vista shows lower network performance than its predecessor, Windows XP; and that Windows Server 2008 shows much better network performance than its predecessor, Windows Server 2003.

The research questions of this report are presented below, with summary answers to conclude the study.

Main research question:

*Does a newer Microsoft operating system give better network performance?*

The results of the experiment show that newer Microsoft Windows client operating system (Windows Vista) does not give better network performance. Windows Vista gives lower network performance than Windows XP, and even the latest Microsoft Windows client operating system (Windows 7) still offers slightly lower network performance than Windows XP. However, in Microsoft Windows Server operating systems, Windows Server 2008 gives much better network performance than Windows Server 2003.

Research sub-questions:

*What are the differences between the network performance of different Windows operating systems?*

The IPv4 network is faster than the IPv6 network. Windows XP gives on average 3% higher network performance than Windows Vista on TCP traffic and 5% higher network performance on UDP traffic. Windows Server 2008 gives on average 5.5%
higher network performance than Windows Server 2003 on TCP traffic, and 36% higher network performance on UDP traffic.

*Which Windows operating system has the best network performance?*
In Windows client operating systems, Windows XP has better network performance, and in Windows Server operating systems, Windows Server 2008 has better network performance.

*What suggestions may improve the network performance of Windows operating systems?*
Although IPv6 solves several of IPv4’s shortcomings, it affects both TCP and UDP performance (Zeadally & Raicu, 2003). Therefore, unless IPv6 is necessary for the network, utilising the IPv4 network will improve network performance. In Windows Vista, enabling ‘compound TCP’ will bring about 5% boosts in network performance. On the other hand, the network performance of Windows operating systems may be improved if all unnecessary network protocols are removed.
Appendices

TCP Results

The following table shows the results of TCP throughput in five Microsoft Windows operating systems. The results are represented in megabit per second.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>29.975081</td>
<td>29.972257</td>
<td>29.957318</td>
<td>29.985595</td>
<td>29.960623</td>
<td>29.979613</td>
<td>29.988728</td>
<td>29.969049</td>
<td></td>
<td></td>
</tr>
<tr>
<td>256</td>
<td>59.999535</td>
<td>59.978030</td>
<td>60.000595</td>
<td>59.967560</td>
<td>59.999456</td>
<td>59.999462</td>
<td>59.999511</td>
<td>59.999464</td>
<td>59.999461</td>
<td>59.999463</td>
</tr>
<tr>
<td>384</td>
<td>85.816965</td>
<td>83.426901</td>
<td>83.207266</td>
<td>82.683977</td>
<td>83.599236</td>
<td>81.377648</td>
<td>83.675909</td>
<td>81.624845</td>
<td>83.278998</td>
<td>81.436117</td>
</tr>
<tr>
<td>512</td>
<td>85.588620</td>
<td>83.196560</td>
<td>82.548661</td>
<td>81.412414</td>
<td>83.862185</td>
<td>81.909162</td>
<td>83.852901</td>
<td>81.739187</td>
<td>85.054174</td>
<td>82.531851</td>
</tr>
<tr>
<td>640</td>
<td>86.348357</td>
<td>85.178168</td>
<td>83.087230</td>
<td>82.151179</td>
<td>85.353887</td>
<td>83.681079</td>
<td>85.361518</td>
<td>83.675129</td>
<td>84.931064</td>
<td>83.030483</td>
</tr>
<tr>
<td>768</td>
<td>84.046431</td>
<td>82.340937</td>
<td>82.549302</td>
<td>80.304782</td>
<td>83.919631</td>
<td>81.205577</td>
<td>84.072255</td>
<td>81.407308</td>
<td>85.003455</td>
<td>83.201772</td>
</tr>
<tr>
<td>896</td>
<td>87.745835</td>
<td>87.315941</td>
<td>86.635253</td>
<td>81.996144</td>
<td>85.867482</td>
<td>84.179421</td>
<td>86.064655</td>
<td>83.913774</td>
<td>86.814936</td>
<td>84.547538</td>
</tr>
<tr>
<td>1024</td>
<td>90.563934</td>
<td>88.967599</td>
<td>79.354569</td>
<td>78.135024</td>
<td>86.924992</td>
<td>85.403420</td>
<td>86.989897</td>
<td>84.962440</td>
<td>87.653014</td>
<td>88.307755</td>
</tr>
<tr>
<td>1152</td>
<td>91.386153</td>
<td>89.833491</td>
<td>80.022009</td>
<td>78.294450</td>
<td>88.843347</td>
<td>87.075019</td>
<td>88.212248</td>
<td>85.885912</td>
<td>90.086400</td>
<td>89.204800</td>
</tr>
<tr>
<td>1280</td>
<td>91.955814</td>
<td>90.613392</td>
<td>79.545148</td>
<td>77.211449</td>
<td>89.309424</td>
<td>87.732858</td>
<td>89.354303</td>
<td>87.835358</td>
<td>90.487592</td>
<td>89.709305</td>
</tr>
<tr>
<td>1408</td>
<td>92.262983</td>
<td>90.990259</td>
<td>77.959521</td>
<td>75.523398</td>
<td>88.519525</td>
<td>87.254289</td>
<td>88.321313</td>
<td>87.063736</td>
<td>91.240590</td>
<td>89.650172</td>
</tr>
<tr>
<td>1536</td>
<td>84.402095</td>
<td>81.967653</td>
<td>81.463569</td>
<td>78.445264</td>
<td>83.962438</td>
<td>81.710597</td>
<td>83.899355</td>
<td>81.289349</td>
<td>83.808507</td>
<td>81.276625</td>
</tr>
</tbody>
</table>

Table A-1: IPv4 & IPv6 TCP Throughput of Microsoft Windows
The following table shows the results of TCP round trip time in five Microsoft Windows operating systems. The results are represented in millisecond.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>1.0092</td>
<td>1.0773</td>
<td>1.1172</td>
<td>1.3032</td>
<td>1.0250</td>
<td>1.1143</td>
<td>0.9620</td>
<td>1.0491</td>
<td>0.9922</td>
<td>1.0702</td>
</tr>
<tr>
<td>128</td>
<td>1.1591</td>
<td>1.1063</td>
<td>1.1531</td>
<td>1.1827</td>
<td>1.0792</td>
<td>1.0766</td>
<td>1.0716</td>
<td>1.0403</td>
<td>1.0885</td>
<td>1.1090</td>
</tr>
<tr>
<td>256</td>
<td>0.7778</td>
<td>0.7617</td>
<td>0.8477</td>
<td>0.9694</td>
<td>0.8639</td>
<td>1.0705</td>
<td>0.8595</td>
<td>0.8754</td>
<td>0.8103</td>
<td>0.8206</td>
</tr>
<tr>
<td>384</td>
<td>2.7598</td>
<td>2.3685</td>
<td>6.6848</td>
<td>3.2139</td>
<td>5.9148</td>
<td>7.0345</td>
<td>5.9054</td>
<td>6.7511</td>
<td>7.4132</td>
<td>7.7917</td>
</tr>
<tr>
<td>512</td>
<td>5.4951</td>
<td>2.4561</td>
<td>5.5544</td>
<td>3.2690</td>
<td>5.0846</td>
<td>5.7191</td>
<td>6.3060</td>
<td>5.6191</td>
<td>4.9243</td>
<td>7.2853</td>
</tr>
<tr>
<td>896</td>
<td>1.8935</td>
<td>1.4259</td>
<td>2.2945</td>
<td>3.3366</td>
<td>5.8340</td>
<td>1.5279</td>
<td>4.9275</td>
<td>4.7595</td>
<td>5.4144</td>
<td>7.7157</td>
</tr>
<tr>
<td>1152</td>
<td>1.3463</td>
<td>1.3738</td>
<td>4.4878</td>
<td>3.3029</td>
<td>1.4955</td>
<td>1.4968</td>
<td>1.9181</td>
<td>3.9073</td>
<td>2.4764</td>
<td>1.7739</td>
</tr>
<tr>
<td>1280</td>
<td>1.3579</td>
<td>1.3505</td>
<td>4.1565</td>
<td>3.2727</td>
<td>1.6161</td>
<td>1.5989</td>
<td>1.5662</td>
<td>1.5239</td>
<td>2.1953</td>
<td>1.7040</td>
</tr>
<tr>
<td>1408</td>
<td>1.2735</td>
<td>1.3148</td>
<td>4.1643</td>
<td>3.3922</td>
<td>1.5303</td>
<td>1.5168</td>
<td>1.7042</td>
<td>1.4713</td>
<td>1.4746</td>
<td>1.9277</td>
</tr>
<tr>
<td>1536</td>
<td>1.5624</td>
<td>1.6196</td>
<td>4.5015</td>
<td>3.0264</td>
<td>1.8438</td>
<td>1.8259</td>
<td>1.8551</td>
<td>1.8249</td>
<td>1.9355</td>
<td>1.8552</td>
</tr>
</tbody>
</table>

Table A-2: IPv4 & IPv6 Round Trip Time of Microsoft Windows
The following table shows the results of TCP jitter in five Microsoft Windows operating systems. The results are represented in millisecond.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>0.0400</td>
<td>0.0415</td>
<td>0.0457</td>
<td>0.0450</td>
<td>0.0352</td>
<td>0.0363</td>
<td>0.0366</td>
</tr>
<tr>
<td>128</td>
<td>0.0430</td>
<td>0.0428</td>
<td>0.0460</td>
<td>0.0459</td>
<td>0.0360</td>
<td>0.0365</td>
<td>0.0375</td>
</tr>
<tr>
<td>256</td>
<td>0.0340</td>
<td>0.0331</td>
<td>0.0400</td>
<td>0.0410</td>
<td>0.0296</td>
<td>0.0302</td>
<td>0.0322</td>
</tr>
<tr>
<td>384</td>
<td>0.0331</td>
<td>0.0350</td>
<td>0.0418</td>
<td>0.0396</td>
<td>0.0326</td>
<td>0.0354</td>
<td>0.0328</td>
</tr>
<tr>
<td>512</td>
<td>0.0495</td>
<td>0.0502</td>
<td>0.0574</td>
<td>0.0549</td>
<td>0.0451</td>
<td>0.0493</td>
<td>0.0487</td>
</tr>
<tr>
<td>640</td>
<td>0.0623</td>
<td>0.0633</td>
<td>0.0730</td>
<td>0.0754</td>
<td>0.0639</td>
<td>0.0587</td>
<td>0.0627</td>
</tr>
<tr>
<td>768</td>
<td>0.0614</td>
<td>0.0751</td>
<td>0.0855</td>
<td>0.0887</td>
<td>0.0566</td>
<td>0.0736</td>
<td>0.0706</td>
</tr>
<tr>
<td>896</td>
<td>0.0676</td>
<td>0.0535</td>
<td>0.0597</td>
<td>0.1093</td>
<td>0.0728</td>
<td>0.0689</td>
<td>0.0789</td>
</tr>
<tr>
<td>1024</td>
<td>0.0314</td>
<td>0.0364</td>
<td>0.1249</td>
<td>0.1411</td>
<td>0.0826</td>
<td>0.0774</td>
<td>0.0875</td>
</tr>
<tr>
<td>1152</td>
<td>0.0361</td>
<td>0.0470</td>
<td>0.1501</td>
<td>0.1659</td>
<td>0.0871</td>
<td>0.0903</td>
<td>0.0948</td>
</tr>
<tr>
<td>1280</td>
<td>0.0370</td>
<td>0.0407</td>
<td>0.1602</td>
<td>0.1939</td>
<td>0.1023</td>
<td>0.0940</td>
<td>0.1070</td>
</tr>
<tr>
<td>1408</td>
<td>0.0715</td>
<td>0.0806</td>
<td>0.1565</td>
<td>0.2299</td>
<td>0.1133</td>
<td>0.1231</td>
<td>0.1176</td>
</tr>
<tr>
<td>1536</td>
<td>0.0242</td>
<td>0.0101</td>
<td>0.1723</td>
<td>0.1078</td>
<td>0.0174</td>
<td>0.0156</td>
<td>0.0191</td>
</tr>
</tbody>
</table>

Table A-3: IPv4 & IPv6 TCP Jitter of Microsoft Windows
UDP Results

The following table shows the results of UDP throughput in five Microsoft Windows operating systems. The results are represented in megabit per second.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>14.544786</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>14.994285</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>14.993675</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>14.932903</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>14.932903</td>
</tr>
<tr>
<td>128</td>
<td>30.000043</td>
<td>29.648865</td>
<td>26.466882</td>
<td>26.092139</td>
<td>27.918998</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>28.684901</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>29.990202</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>29.984429</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>29.825543</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>29.758534</td>
</tr>
<tr>
<td>256</td>
<td>60.000084</td>
<td>57.655035</td>
<td>52.263276</td>
<td>50.854048</td>
<td>55.508991</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>55.434549</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>59.986996</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>59.973373</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>59.495382</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>59.362785</td>
</tr>
<tr>
<td>384</td>
<td>81.880496</td>
<td>78.526913</td>
<td>49.181068</td>
<td>47.962807</td>
<td>81.279669</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>77.900232</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>81.897005</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>78.395487</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>81.424272</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>78.067451</td>
</tr>
<tr>
<td>512</td>
<td>83.734464</td>
<td>81.118151</td>
<td>56.293207</td>
<td>54.703039</td>
<td>83.945632</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>81.525577</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>83.615958</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>80.764758</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>83.446193</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>80.662253</td>
</tr>
<tr>
<td>640</td>
<td>85.043502</td>
<td>82.826766</td>
<td>61.539994</td>
<td>59.898893</td>
<td>84.431491</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>82.650470</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>84.468695</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>82.501314</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>84.633969</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>82.368054</td>
</tr>
<tr>
<td>768</td>
<td>85.529107</td>
<td>83.551222</td>
<td>65.844817</td>
<td>64.207601</td>
<td>85.090049</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>83.535258</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>85.411584</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>82.991330</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>85.496350</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>83.320417</td>
</tr>
<tr>
<td>896</td>
<td>87.158615</td>
<td>84.108371</td>
<td>69.747630</td>
<td>68.642801</td>
<td>84.712192</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>83.757665</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>85.150648</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>83.080535</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>85.925806</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>84.139474</td>
</tr>
<tr>
<td>1024</td>
<td>87.117245</td>
<td>83.351355</td>
<td>72.472095</td>
<td>70.487215</td>
<td>84.423785</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>83.436352</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>84.751385</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>83.368435</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>85.018009</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>84.636038</td>
</tr>
<tr>
<td>1152</td>
<td>92.366360</td>
<td>90.872624</td>
<td>90.748994</td>
<td>90.456246</td>
<td>92.159949</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>90.456246</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>92.219595</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>90.678688</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>92.348718</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>90.814967</td>
</tr>
<tr>
<td>1280</td>
<td>92.869420</td>
<td>91.507606</td>
<td>92.748586</td>
<td>91.347625</td>
<td>92.850481</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>91.478017</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>92.671539</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>91.327156</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>92.848387</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>91.450996</td>
</tr>
<tr>
<td>1408</td>
<td>93.284561</td>
<td>92.033864</td>
<td>93.203998</td>
<td>91.917428</td>
<td>93.271163</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>92.005711</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>93.116050</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>91.880495</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>93.264628</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>91.971265</td>
</tr>
<tr>
<td>1536</td>
<td>90.357746</td>
<td>87.411074</td>
<td>90.234542</td>
<td>87.268387</td>
<td>90.326210</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>87.347366</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>90.225397</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>87.268460</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>90.301427</td>
</tr>
</tbody>
</table>

Table A-4: IPv4 & IPv6 UDP Throughput of Microsoft Windows
The following table shows the results of UDP round trip time in five Microsoft Windows operating systems. The results are represented in millisecond.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>0.1410</td>
<td>1.8316</td>
<td>1.1109</td>
<td>1.3762</td>
<td>2.2326</td>
<td>3.9472</td>
<td>0.1727</td>
<td>0.2815</td>
<td>0.3749</td>
<td>0.7481</td>
</tr>
<tr>
<td>128</td>
<td>0.1634</td>
<td>1.4414</td>
<td>0.7301</td>
<td>0.7816</td>
<td>1.4954</td>
<td>2.2106</td>
<td>0.1820</td>
<td>0.2599</td>
<td>0.4800</td>
<td>0.4491</td>
</tr>
<tr>
<td>256</td>
<td>0.2080</td>
<td>1.0642</td>
<td>0.5295</td>
<td>0.5512</td>
<td>0.8642</td>
<td>1.3111</td>
<td>0.2231</td>
<td>0.3715</td>
<td>0.5890</td>
<td>0.4061</td>
</tr>
<tr>
<td>384</td>
<td>1.6479</td>
<td>1.6094</td>
<td>2.5428</td>
<td>2.5829</td>
<td>1.9829</td>
<td>1.3587</td>
<td>1.6427</td>
<td>1.5700</td>
<td>1.8642</td>
<td>1.6402</td>
</tr>
<tr>
<td>512</td>
<td>1.7244</td>
<td>1.7598</td>
<td>2.2931</td>
<td>2.3245</td>
<td>1.9514</td>
<td>1.6403</td>
<td>1.7976</td>
<td>1.7328</td>
<td>1.8917</td>
<td>1.7662</td>
</tr>
<tr>
<td>640</td>
<td>1.7948</td>
<td>1.8641</td>
<td>2.3300</td>
<td>2.3127</td>
<td>1.9760</td>
<td>1.7241</td>
<td>1.8121</td>
<td>1.7700</td>
<td>1.9873</td>
<td>1.8038</td>
</tr>
<tr>
<td>768</td>
<td>1.8221</td>
<td>1.9134</td>
<td>2.3683</td>
<td>2.3318</td>
<td>2.0946</td>
<td>1.7718</td>
<td>1.8179</td>
<td>1.7816</td>
<td>2.0678</td>
<td>1.8750</td>
</tr>
<tr>
<td>896</td>
<td>1.8827</td>
<td>1.9717</td>
<td>2.4105</td>
<td>2.3450</td>
<td>2.1722</td>
<td>1.8644</td>
<td>1.7543</td>
<td>1.7916</td>
<td>2.0615</td>
<td>1.9887</td>
</tr>
<tr>
<td>1024</td>
<td>1.8545</td>
<td>1.9049</td>
<td>2.3057</td>
<td>2.2662</td>
<td>2.1381</td>
<td>1.8640</td>
<td>1.7040</td>
<td>1.7325</td>
<td>2.0277</td>
<td>1.8423</td>
</tr>
<tr>
<td>1152</td>
<td>1.3341</td>
<td>1.3489</td>
<td>1.6018</td>
<td>1.5646</td>
<td>1.1384</td>
<td>0.9589</td>
<td>1.1298</td>
<td>1.0910</td>
<td>1.4516</td>
<td>1.2224</td>
</tr>
<tr>
<td>1280</td>
<td>1.3260</td>
<td>1.3153</td>
<td>1.5736</td>
<td>1.5576</td>
<td>1.6206</td>
<td>1.1858</td>
<td>1.1081</td>
<td>1.1152</td>
<td>1.3579</td>
<td>1.2227</td>
</tr>
<tr>
<td>1408</td>
<td>1.4119</td>
<td>1.4276</td>
<td>1.6155</td>
<td>1.5787</td>
<td>1.6785</td>
<td>1.2713</td>
<td>1.1892</td>
<td>1.1984</td>
<td>1.4508</td>
<td>1.2633</td>
</tr>
<tr>
<td>1536</td>
<td>1.4634</td>
<td>1.3996</td>
<td>1.6868</td>
<td>1.6845</td>
<td>1.6019</td>
<td>1.2455</td>
<td>1.2280</td>
<td>1.2031</td>
<td>1.4185</td>
<td>1.3334</td>
</tr>
</tbody>
</table>

Table A-5: IPv4 & IPv6 UDP Round Trip Time of Microsoft Windows
The following table shows the results of UDP Jitter in five Microsoft Windows operating systems. The results are represented in millisecond.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>0.0110</td>
<td>0.0091</td>
<td>0.0182</td>
<td>0.0177</td>
<td>0.0138</td>
<td>0.0102</td>
<td>0.0109</td>
<td>0.0110</td>
<td>0.0100</td>
<td>0.0101</td>
</tr>
<tr>
<td>128</td>
<td>0.0094</td>
<td>0.0100</td>
<td>0.0188</td>
<td>0.0168</td>
<td>0.0141</td>
<td>0.0109</td>
<td>0.0101</td>
<td>0.0106</td>
<td>0.0092</td>
<td>0.0100</td>
</tr>
<tr>
<td>256</td>
<td>0.0100</td>
<td>0.0093</td>
<td>0.0138</td>
<td>0.0153</td>
<td>0.0133</td>
<td>0.0132</td>
<td>0.0101</td>
<td>0.0092</td>
<td>0.0090</td>
<td>0.0098</td>
</tr>
<tr>
<td>384</td>
<td>0.0360</td>
<td>0.0330</td>
<td>0.0143</td>
<td>0.0180</td>
<td>0.0271</td>
<td>0.0190</td>
<td>0.0308</td>
<td>0.0335</td>
<td>0.0366</td>
<td>0.0392</td>
</tr>
<tr>
<td>512</td>
<td>0.0575</td>
<td>0.0542</td>
<td>0.0303</td>
<td>0.0182</td>
<td>0.0500</td>
<td>0.0436</td>
<td>0.0575</td>
<td>0.0562</td>
<td>0.0617</td>
<td>0.0631</td>
</tr>
<tr>
<td>640</td>
<td>0.0820</td>
<td>0.0804</td>
<td>0.1126</td>
<td>0.1108</td>
<td>0.0769</td>
<td>0.0695</td>
<td>0.0802</td>
<td>0.0811</td>
<td>0.0838</td>
<td>0.0862</td>
</tr>
<tr>
<td>768</td>
<td>0.1038</td>
<td>0.1014</td>
<td>0.1368</td>
<td>0.1388</td>
<td>0.0973</td>
<td>0.0955</td>
<td>0.0989</td>
<td>0.1066</td>
<td>0.0893</td>
<td>0.1050</td>
</tr>
<tr>
<td>896</td>
<td>0.1244</td>
<td>0.1263</td>
<td>0.1569</td>
<td>0.1589</td>
<td>0.0814</td>
<td>0.1225</td>
<td>0.1266</td>
<td>0.1294</td>
<td>0.1169</td>
<td>0.1221</td>
</tr>
<tr>
<td>1024</td>
<td>0.1172</td>
<td>0.1198</td>
<td>0.1789</td>
<td>0.1827</td>
<td>0.1391</td>
<td>0.1383</td>
<td>0.1415</td>
<td>0.1487</td>
<td>0.1384</td>
<td>0.1395</td>
</tr>
<tr>
<td>1152</td>
<td>0.0057</td>
<td>0.0070</td>
<td>0.0092</td>
<td>0.0090</td>
<td>0.0082</td>
<td>0.0093</td>
<td>0.0084</td>
<td>0.0088</td>
<td>0.0076</td>
<td>0.0080</td>
</tr>
<tr>
<td>1280</td>
<td>0.0067</td>
<td>0.0064</td>
<td>0.0083</td>
<td>0.0079</td>
<td>0.0073</td>
<td>0.0077</td>
<td>0.0055</td>
<td>0.0049</td>
<td>0.0055</td>
<td>0.0068</td>
</tr>
<tr>
<td>1408</td>
<td>0.0079</td>
<td>0.0098</td>
<td>0.0097</td>
<td>0.0120</td>
<td>0.0098</td>
<td>0.0106</td>
<td>0.0097</td>
<td>0.0120</td>
<td>0.0050</td>
<td>0.0054</td>
</tr>
<tr>
<td>1536</td>
<td>0.0062</td>
<td>0.0071</td>
<td>0.0080</td>
<td>0.0090</td>
<td>0.0089</td>
<td>0.0108</td>
<td>0.0096</td>
<td>0.0080</td>
<td>0.0077</td>
<td>0.0078</td>
</tr>
</tbody>
</table>

Table A-6: IPv4 & IPv6 UDP Jitter of Microsoft Windows
DNS Results

The following table shows the results of TCP throughput of DNS in five Microsoft Windows operating systems. The results are represented in kilobit per second.

<table>
<thead>
<tr>
<th>Windows</th>
<th>XP</th>
<th>Server 2003</th>
<th>Vista</th>
<th>Server 2008</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv4 DNS</td>
<td>0.960615</td>
<td>0.969264</td>
<td>0.942993</td>
<td>0.957537</td>
<td>0.939562</td>
</tr>
<tr>
<td>IPv6 DNS</td>
<td>0.966117</td>
<td>0.964184</td>
<td>0.945917</td>
<td>0.971533</td>
<td>0.970980</td>
</tr>
</tbody>
</table>

Table A-7: IPv4 & IPv6 TCP throughput of DNS of Microsoft Windows

The following table shows the results of TCP round trip time of DNS in five Microsoft Windows operating systems. The results are represented in millisecond.

<table>
<thead>
<tr>
<th>Windows</th>
<th>XP</th>
<th>Server 2003</th>
<th>Vista</th>
<th>Server 2008</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv4 DNS</td>
<td>0.2550</td>
<td>0.2755</td>
<td>0.2997</td>
<td>0.2891</td>
<td>0.3075</td>
</tr>
<tr>
<td>IPv6 DNS</td>
<td>0.2692</td>
<td>0.2853</td>
<td>0.3020</td>
<td>0.2981</td>
<td>0.3132</td>
</tr>
</tbody>
</table>

Table A-8: IPv4 & IPv6 TCP Round Trip Time of DNS of Microsoft Windows

The following table shows the results of TCP jitter of DNS in five Microsoft Windows operating systems. The results are represented in millisecond.

<table>
<thead>
<tr>
<th>Windows</th>
<th>XP</th>
<th>Server 2003</th>
<th>Vista</th>
<th>Server 2008</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv4 DNS</td>
<td>0.0274</td>
<td>0.0274</td>
<td>0.0282</td>
<td>0.0278</td>
<td>0.0266</td>
</tr>
<tr>
<td>IPv6 DNS</td>
<td>0.0251</td>
<td>0.0341</td>
<td>0.0286</td>
<td>0.0276</td>
<td>0.0290</td>
</tr>
</tbody>
</table>

Table A-9: IPv4 & IPv6 DNS TCP Jitter of Microsoft Windows

The following table shows the results of UDP throughput of DNS in five Microsoft Windows operating systems. The results are represented in kilobit per second.

<table>
<thead>
<tr>
<th>Windows</th>
<th>XP</th>
<th>Server 2003</th>
<th>Vista</th>
<th>Server 2008</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv4 DNS</td>
<td>0.961017</td>
<td>0.980006</td>
<td>0.943790</td>
<td>0.977221</td>
<td>0.957591</td>
</tr>
<tr>
<td>IPv6 DNS</td>
<td>0.964764</td>
<td>0.957932</td>
<td>0.960341</td>
<td>0.969225</td>
<td>0.951050</td>
</tr>
</tbody>
</table>

Table A-10: IPv4 & IPv6 UDP Throughput of DNS of Microsoft Windows
The following table shows the results of UDP Round Trip Time of DNS in five Microsoft Windows operating systems. The results are represented in millisecond.

<table>
<thead>
<tr>
<th>Windows</th>
<th>XP</th>
<th>Server 2003</th>
<th>Vista</th>
<th>Server 2008</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv4 DNS</td>
<td>0.2508</td>
<td>0.2684</td>
<td>0.2921</td>
<td>0.2800</td>
<td>0.3054</td>
</tr>
<tr>
<td>IPv6 DNS</td>
<td>0.2697</td>
<td>0.2834</td>
<td>0.2981</td>
<td>0.2903</td>
<td>0.3082</td>
</tr>
</tbody>
</table>

Table A-11: IPv4 & IPv6 UDP Round Trip Time of DNS of Microsoft Windows

The following table shows the results of UDP jitter of DNS in five Microsoft Windows operating systems. The results are represented in millisecond.

<table>
<thead>
<tr>
<th>Windows</th>
<th>XP</th>
<th>Server 2003</th>
<th>Vista</th>
<th>Server 2008</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv4 DNS</td>
<td>0.0315</td>
<td>0.0280</td>
<td>0.0262</td>
<td>0.0269</td>
<td>0.0293</td>
</tr>
<tr>
<td>IPv6 DNS</td>
<td>0.0258</td>
<td>0.0260</td>
<td>0.0290</td>
<td>0.0270</td>
<td>0.0271</td>
</tr>
</tbody>
</table>

Table A-12: IPv4 & IPv6 UDP Jitter of DNS of Microsoft Windows
Gaming Results

The following table shows the results of two games’ throughput in five Microsoft Windows operating systems. The results are represented in kilobit per second.

<table>
<thead>
<tr>
<th>Windows</th>
<th>Throughput (Kbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>XP</td>
</tr>
<tr>
<td>IPv4 CS</td>
<td>8.499851</td>
</tr>
<tr>
<td>IPv6 CS</td>
<td>8.730722</td>
</tr>
<tr>
<td>IPv4 Quake3</td>
<td>74.015124</td>
</tr>
<tr>
<td>IPv6 Quake3</td>
<td>74.251560</td>
</tr>
</tbody>
</table>

Table A-13: IPv4 & IPv6 Gaming Throughput of Microsoft Windows

The following table shows the results of two games’ round trip time in five Microsoft Windows operating systems. The results are represented in millisecond.

<table>
<thead>
<tr>
<th>Windows</th>
<th>Round Trip Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>XP</td>
</tr>
<tr>
<td>IPv4 CS</td>
<td>0.1415</td>
</tr>
<tr>
<td>IPv6 CS</td>
<td>0.1519</td>
</tr>
<tr>
<td>IPv4 Quake3</td>
<td>0.1505</td>
</tr>
<tr>
<td>IPv6 Quake3</td>
<td>0.1707</td>
</tr>
</tbody>
</table>

Table A-14: IPv4 & IPv6 Gaming Round Trip Time of Microsoft Windows

The following table shows the results of two games’ Jitter in five Microsoft Windows operating systems. The results are represented in millisecond.

<table>
<thead>
<tr>
<th>Windows</th>
<th>Jitter (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>XP</td>
</tr>
<tr>
<td>IPv4 CS</td>
<td>0.0146</td>
</tr>
<tr>
<td>IPv6 CS</td>
<td>0.0102</td>
</tr>
<tr>
<td>IPv4 Quake3</td>
<td>0.0132</td>
</tr>
<tr>
<td>IPv6 Quake3</td>
<td>0.0148</td>
</tr>
</tbody>
</table>

Table A-15: IPv4 & IPv6 Gaming Jitter of Microsoft Windows
VoIP Results

The following table shows the results of VoIP throughput in five Microsoft Windows operating systems. The results are represented in kilobit per second.

<table>
<thead>
<tr>
<th>Throughput (Kbps)</th>
<th>XP</th>
<th>Server 2003</th>
<th>Vista</th>
<th>Server 2008</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv4 VoIP G.711.1</td>
<td>73.606753</td>
<td>73.605462</td>
<td>73.604316</td>
<td>73.606446</td>
<td>73.605991</td>
</tr>
<tr>
<td>IPv6 VoIP G.711.1</td>
<td>73.607775</td>
<td>73.604490</td>
<td>73.604813</td>
<td>73.604443</td>
<td>73.604368</td>
</tr>
<tr>
<td>IPv4 VoIP G.711.2</td>
<td>68.813080</td>
<td>68.811084</td>
<td>68.818141</td>
<td>68.817185</td>
<td>68.814328</td>
</tr>
<tr>
<td>IPv6 VoIP G.711.2</td>
<td>68.808760</td>
<td>68.811219</td>
<td>68.819402</td>
<td>68.816679</td>
<td>68.815099</td>
</tr>
<tr>
<td>IPv4 VoIP G.723.1</td>
<td>8.741083</td>
<td>8.740757</td>
<td>8.740990</td>
<td>8.740738</td>
<td>8.740557</td>
</tr>
<tr>
<td>IPv6 VoIP G.723.1</td>
<td>8.741474</td>
<td>8.739987</td>
<td>8.740449</td>
<td>8.740445</td>
<td>8.740516</td>
</tr>
<tr>
<td>IPv4 VoIP G.729.3</td>
<td>11.092142</td>
<td>11.092691</td>
<td>11.09222</td>
<td>11.092051</td>
<td>11.092289</td>
</tr>
<tr>
<td>IPv6 VoIP G.729.3</td>
<td>11.092179</td>
<td>11.092772</td>
<td>11.091784</td>
<td>11.092631</td>
<td>11.091919</td>
</tr>
</tbody>
</table>

Table A-16: IPv4 & IPv6 VoIP Throughput of Microsoft Windows

The following table shows the results of VoIP round trip time in five Microsoft Windows operating systems. The results are represented in millisecond.

<table>
<thead>
<tr>
<th>Round Trip Time (ms)</th>
<th>XP</th>
<th>Server 2003</th>
<th>Vista</th>
<th>Server 2008</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv4 VoIP G.711.1</td>
<td>0.1530</td>
<td>0.1641</td>
<td>0.1813</td>
<td>0.1654</td>
<td>0.1644</td>
</tr>
<tr>
<td>IPv6 VoIP G.711.1</td>
<td>0.1650</td>
<td>0.1790</td>
<td>0.1944</td>
<td>0.1741</td>
<td>0.1760</td>
</tr>
<tr>
<td>IPv4 VoIP G.711.2</td>
<td>0.1808</td>
<td>0.1849</td>
<td>0.2048</td>
<td>0.1925</td>
<td>0.1990</td>
</tr>
<tr>
<td>IPv6 VoIP G.711.2</td>
<td>0.1928</td>
<td>0.1996</td>
<td>0.2127</td>
<td>0.2008</td>
<td>0.2019</td>
</tr>
<tr>
<td>IPv4 VoIP G.723.1</td>
<td>0.1362</td>
<td>0.1421</td>
<td>0.1572</td>
<td>0.1446</td>
<td>0.1532</td>
</tr>
<tr>
<td>IPv6 VoIP G.723.1</td>
<td>0.1470</td>
<td>0.1567</td>
<td>0.1713</td>
<td>0.1502</td>
<td>0.1621</td>
</tr>
<tr>
<td>IPv4 VoIP G.729.2</td>
<td>0.1294</td>
<td>0.1333</td>
<td>0.1483</td>
<td>0.1378</td>
<td>0.1416</td>
</tr>
<tr>
<td>IPv6 VoIP G.729.2</td>
<td>0.1424</td>
<td>0.1503</td>
<td>0.1640</td>
<td>0.1466</td>
<td>0.1539</td>
</tr>
<tr>
<td>IPv4 VoIP G.729.3</td>
<td>0.1350</td>
<td>0.1413</td>
<td>0.1535</td>
<td>0.1440</td>
<td>0.1504</td>
</tr>
<tr>
<td>IPv6 VoIP G.729.3</td>
<td>0.1466</td>
<td>0.1545</td>
<td>0.1718</td>
<td>0.1502</td>
<td>0.1598</td>
</tr>
</tbody>
</table>

Table A-17: IPv4 & IPv6 VoIP Round Trip Time of Microsoft Windows
The following table shows the results of VoIP jitter in five Microsoft Windows operating systems. The results are represented in millisecond.

<table>
<thead>
<tr>
<th>Windows</th>
<th>Jitter (ms)</th>
<th>XP</th>
<th>Server 2003</th>
<th>Vista</th>
<th>Server 2008</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv4 VoIP G.711.1</td>
<td>0.0090</td>
<td>0.0211</td>
<td>0.0139</td>
<td>0.0145</td>
<td>0.0117</td>
<td></td>
</tr>
<tr>
<td>IPv6 VoIP G.711.1</td>
<td>0.0100</td>
<td>0.0270</td>
<td>0.0163</td>
<td>0.0140</td>
<td>0.0111</td>
<td></td>
</tr>
<tr>
<td>IPv4 VoIP G.711.2</td>
<td>0.0078</td>
<td>0.0105</td>
<td>0.0113</td>
<td>0.0119</td>
<td>0.0159</td>
<td></td>
</tr>
<tr>
<td>IPv6 VoIP G.711.2</td>
<td>0.0102</td>
<td>0.0194</td>
<td>0.0098</td>
<td>0.0081</td>
<td>0.0092</td>
<td></td>
</tr>
<tr>
<td>IPv4 VoIP G.723.1</td>
<td>0.0094</td>
<td>0.0137</td>
<td>0.0157</td>
<td>0.0121</td>
<td>0.0139</td>
<td></td>
</tr>
<tr>
<td>IPv6 VoIP G.723.1</td>
<td>0.0089</td>
<td>0.0210</td>
<td>0.0135</td>
<td>0.0088</td>
<td>0.0120</td>
<td></td>
</tr>
<tr>
<td>IPv4 VoIP G.729.2</td>
<td>0.0076</td>
<td>0.0107</td>
<td>0.0112</td>
<td>0.0105</td>
<td>0.0103</td>
<td></td>
</tr>
<tr>
<td>IPv6 VoIP G.729.2</td>
<td>0.0068</td>
<td>0.0189</td>
<td>0.0123</td>
<td>0.0082</td>
<td>0.0093</td>
<td></td>
</tr>
<tr>
<td>IPv4 VoIP G.729.3</td>
<td>0.0085</td>
<td>0.0134</td>
<td>0.0122</td>
<td>0.0107</td>
<td>0.0126</td>
<td></td>
</tr>
<tr>
<td>IPv6 VoIP G.729.3</td>
<td>0.0086</td>
<td>0.0182</td>
<td>0.0141</td>
<td>0.0080</td>
<td>0.0103</td>
<td></td>
</tr>
</tbody>
</table>

Table A-18: IPv4 & IPv6 VoIP Jitter of Microsoft Windows
Reference


