TCP/UDP Network Performance Evaluation of Various IPSec Algorithms

An Empirical Test-bed Analysis of a Virtual Private Network Protocol

Rashikala Weerawarna

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Computing

UNITEC Institute of Technology, New Zealand
2013
Internet has become ubiquitous and the default communication channel for businesses, and it continues to grow worldwide. With the prolific development of the Internet, Virtual Private Network (VPN) is the most widely used tunnel, and it uses different protocols that guarantee secure data communication between multiple sites connected via public telecommunication channels such as Internet. VPN protocols provide secure communication links with data encryption and integrity, and they are implemented with various encryption algorithms. Encryption algorithms are Data Encryption Standard (DES), Triple Data Encryption Standard (3DES), Advanced Encryption Standard 128bit (AES128), Advanced Encryption Standard 256bit (AES256) and Blowfish (BF). Authentication and data integrity algorithms are Message-Digest 5 (MD5) and Secure Hash Algorithm (SHA1).

This research empirically evaluates the impact of various Internet Protocol Security (IPSec) algorithms (DES, 3DES, AES) on the performance of Virtual Private Network for Windows operating systems (Windows Server 2003, Windows Server 2008, Windows XP and Windows Vista) and for both Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). The parameters considered are throughput, jitter and latency. Results obtained indicate that IPSec algorithms do influence VPN performance and that different operating systems provide various results. Throughput, jitter and latency in a VPN tunnel can vary depending on the choice of operating system, protocol, and algorithm and packet size.
Acknowledgement

Firstly I would like to acknowledge my principle supervisor Shaneel Narayan, who guided and encourage me throughout this thesis study. I appreciate the knowledge and support given to achieve this qualification. Also I would like to thank Aaron Chen, who was my co-supervisor prior to his departure from Unitec. Both of them helped me to stay focused on the study and allowed me to complete my thesis successfully within my busy schedule.

Next I would like to give my appreciation to all the lectures in the Department of Computing at Unitec who provided the background knowledge for my postgraduate studies. Their experience and guidance empowered me as an academic.

Finally I would like to thank my loving parents who have always appreciated my desire to go for further studies and encouraged me to come to this position. I would like to add my thanks to my husband Chamaka, who has always encouraged me and helped me in my studies. Without his and my parent’s blessings I would not be able to achieve the results I desired.

Table of Contents

Abstract ....................................................................................................................................................... ii

Acknowledgement ......................................................................................................................................... iii

Table of Contents .......................................................................................................................................... iv

List of Figures ................................................................................................................................................. vii

List of Tables .................................................................................................................................................. ix

Chapter 1: Introduction ............................................................................................................................... 1

1.1 Overview ................................................................................................................................................. 1

1.2 Motivation for this study .......................................................................................................................... 3

1.3 Research problem .................................................................................................................................... 4

1.4 Thesis outline .......................................................................................................................................... 4

1.5 Chapter Summary .................................................................................................................................... 5

Chapter 2: Literature Review ....................................................................................................................... 6

2.1 Computer Networking and the Importance of a Network .................................................................... 6

2.2 Performance of a network ....................................................................................................................... 8

2.3 Relevant Researched Areas ..................................................................................................................... 10

2.3.1 Internet Protocol - IPv4 and IPv6 ................................................................................................. 10

2.3.2 Virtual Private Network ............................................................................................................... 16

2.3.3 VPN Protocols ............................................................................................................................... 22

2.3.4 The Transmission Protocols - TCP and UDP .............................................................................. 27

2.3.5 Cryptographic Algorithms ........................................................................................................... 29

2.3.6 Researched Metrics ....................................................................................................................... 32
2.4 Literature Analysis .................................................................................................................. 33
2.5 Chapter Summary .................................................................................................................... 38

Chapter 3: Methodology ................................................................................................................ 39
3.1 Research Methods ..................................................................................................................... 40
3.2 Research Methodology for this Study ...................................................................................... 41
3.3 Data Collection and Recording Methods ................................................................................ 42
3.4 Network Performance Monitoring Tools .................................................................................. 44
3.4 Chapter Summary ...................................................................................................................... 49

Chapter 4: Experimental Network Design .................................................................................. 50
4.1 Hardware Specifications ......................................................................................................... 50
4.2 Software Specifications ........................................................................................................... 51
4.3 Network Configurations .......................................................................................................... 51
4.3 Packet Payload sizes ................................................................................................................ 52
4.5 Chapter Summary ...................................................................................................................... 52

Chapter 5: Data Analysis .............................................................................................................. 53
5.1 Windows Server 2003 platform .............................................................................................. 53
5.1.1 Results for Throughput ....................................................................................................... 53
5.1.2 Results for Latency ............................................................................................................. 57
5.1.3 Results for Jitter .................................................................................................................. 60
5.1.4 Results on 3DES ................................................................................................................ 62
5.1.5 Results on DES ................................................................................................................... 66
5.1.6 Results for algorithms ...................................................................................................... 70
5.2 Windows Server 2008 platform ............................................................................................. 72
5.2.1 Results for Throughput ...................................................................................................... 72
5.2.2 Results for Latency ........................................................... 76
5.2.3 Results for Jitter ............................................................... 78
5.2.4 Results on 3DES ............................................................... 81
5.2.5 Results on DES ............................................................... 84
5.2.6 Results on AES ............................................................... 88
5.2.7 Results for algorithms ...................................................... 91
5.3 Windows Server 2003 VS 2008 platforms ........................................ 92
  5.3.1 Throughput ................................................................. 92
  5.3.2 Latency ...................................................................... 93
  5.3.3 Jitter ....................................................................... 94
  5.3.4 Results for algorithms ................................................. 95
5.4 Chapter Summary .................................................................. 97

Chapter 6: Discussion and Findings ................................................. 98
  6.1 Performance of TCP/UDP for IPSec algorithms on VPN using Microsoft Windows Server
      2003 ............................................................................... 98
  6.2 Performance of TCP/UDP for IPSec algorithms on VPN using Microsoft Windows Server
      2008 ............................................................................... 100
  6.3 Cross comparison between the performances of TCP/UDP for IPSec algorithms on two
  6.5 Chapter Summary .................................................................. 102

Chapter 7: Conclusion ................................................................. 103
  7.1 Summary of findings .......................................................... 104
  7.2 Future Work ...................................................................... 106

References .................................................................................. 107
List of Figures

Figure 1: Virtual Private Network ................................................................. 1
Figure 2: TCP/IP protocol suite ................................................................. 11
Figure 3: IPv4 header format ..................................................................... 14
Figure 4: IPv6 header format ..................................................................... 14
Figure 5: A VPN Connection ..................................................................... 17
Figure 6: Site-to-Site VPN ......................................................................... 18
Figure 7: A Remote Access VPN ................................................................. 18
Figure 8: Extranet VPN ............................................................................. 19
Figure 9: UDP Datagram .......................................................................... 29
Figure 10: TCP Segment .......................................................................... 29
Figure 11: VPN test bed ........................................................................... 43
Figure 12: Sample Line Graph ................................................................... 44
Figure 13: Network Diagram of the test-bed ............................................. 51
Figure 14: TCP IPv4 Throughput (IPSec Algorithms) – Microsoft Windows Server 2003 ................................................................. 54
Figure 15: UDP IPv4 Throughput (IPSec Algorithms) – Microsoft Windows Server 2003 ................................................................. 55
Figure 16: TCP IPv4 Latency (IPSec Algorithms) – Microsoft Windows Server 2003 ................................................................. 57
Figure 17: UDP IPv4 Latency (IPSec Algorithms) – Microsoft Windows Server 2003 ................................................................. 58
Figure 18: TCP IPv4 Jitter (IPSec Algorithms) – Microsoft Windows Server 2003 ................................................................. 60
Figure 19: UDP IPv4 Jitter (IPSec Algorithms) – Microsoft Windows Server 2003 ................................................................. 61
Figure 20: TCP and UDP IPv4 Throughput (3DES-MD5 IPSec Algorithm) – Microsoft Windows Server 2003 ................................................................. 63
Figure 21: TCP and UDP IPv4 Latency (3DES-MD5 IPSec Algorithm) – Microsoft Windows Server 2003 ................................................................. 64
Figure 22: TCP and UDP IPv4 Jitter (3DES-MD5 IPSec Algorithm) – Microsoft Windows Server 2003 ................................................................. 65
Figure 23: TCP and UDP IPv4 Throughput (DES-MD5 IPSec Algorithm) – Microsoft Windows 2003 ................................................................. 67
Figure 24: TCP and UDP IPv4 Latency (DES-MD5 IPSec Algorithm) – Microsoft Windows 2003 ................................................................. 68
Figure 25: TCP and UDP IPv4 Jitter (DES-MD5 IPSec Algorithm) – Microsoft Windows Server 2003 ................................................................. 69
Figure 26: Algorithm performance – TCP – IPSec - IPv4 – Microsoft Windows Server 2003 ................................................................. 69
Figure 27: TCP IPv4 Throughput (IPSec Algorithms) – Microsoft Windows Server 2008 ................................................................. 72
Figure 28: UDP IPv4 Throughput (IPSec Algorithms) – Microsoft Windows Server 2008 ................................................................. 74
Figure 29: TCP IPv4 Latency (IPSec Algorithms) – Microsoft Windows Server 2008 ................................................................. 76
Figure 30: UDP IPv4 Latency (IPSec Algorithms) – Microsoft Windows Server 2008 ................................................................. 77
Figure 31: TCP IPv4 Jitter (IPSec Algorithms) – Microsoft Windows Server 2008 .......................... 79
Figure 32: UDP Jitter IPv4 (IPSec Algorithms) – Microsoft Windows Server 2008 .......................... 80
Figure 33: TCP and UDP IPv4 Throughput (3DES-SHA1 IPSec Algorithm) – Microsoft Windows 2008 ...... 81
Figure 34: TCP and UDP IPv4 Latency (3DES-MD5 IPSec Algorithm) – Microsoft Windows Server 2008 .. 82
Figure 35: TCP and UDP IPv4 Jitter (3DES-SHA1 IPSec Algorithm) – Microsoft Windows Server 2008...... 83
Figure 36: TCP and UDP IPv4 Throughput (DES-MD5 IPSec Algorithm) – Microsoft Windows Server 2008
                                                                                            ............................................................................ 85
Figure 37: TCP and UDP IPv4 Latency (DES-SHA1 IPSec Algorithm) – Microsoft Windows Server 2008 ... 86
Figure 38: TCP and UDP IPv4 Jitter (DES-SHA1 IPSec Algorithm) – Microsoft Windows Server 2008....... 87
Figure 39: TCP and UDP IPv4 Throughput (AES256-SHA1 IPSec Algorithm) – Microsoft Windows Server
                             2008.................................................................................................................. 88
Figure 40: TCP and UDP IPv4 Latency (AES256-SHA1 IPSec Algorithm) – Microsoft Windows Server 2008
                                                                                            ............................................................................ 89
Figure 41: TCP and UDP IPv4 Jitter (AES256-SHA1 IPSec Algorithm) – Microsoft Windows Server 2008.. 90
Figure 42: Algorithm performance – TCP – IPSec- IPv4 – Microsoft Windows Server 2008...................... 91
Figure 43: TCP IPv4 Throughput (3DES-SHA1 IPSec Algorithm) – Microsoft Windows Server 2003 vs 2008
                                                                                            ....................................................................................... 92
Figure 44: TCP IPv4 Latency (3DES-SHA1 IPSec Algorithm) – Microsoft Windows Server 2003 vs 2008 ... 93
Figure 45: TCP IPv4 Jitter (3DES-SHA1 IPSec Algorithm) – Microsoft Windows Server 2003 vs 2008 ....... 94
Figure 46: Algorithm performance – TCP – IPSec - Microsoft Windows Server 2003 vs 2008............... 95
Figure 47: Algorithm performance Microsoft Windows Server 2003 vs 2008 ...................................... 96
List of Tables

Table 1: Related research on IPv4/IPv6 performance evaluation on different operating systems........15
Table 2: The position of security protocols on OSI model .........................................................23
Table 3: VPN protocol summary ...............................................................................................24
Table 4: OSI Reference Model ..................................................................................................28
Table 5: TCP/UDP Comparison ...............................................................................................28
Table 6: Related Research 1 ....................................................................................................35
Table 7: Related Research 2 ....................................................................................................36
Table 8: Comparison network performance monitoring tools ..................................................49
Table 9: Hardware Specification ...............................................................................................50
Table 10: Software Summary ..................................................................................................51
Chapter 1: Introduction

There is a great interest in organizations to expand their businesses to larger geographical areas. As a result of this enterprise, networks need to enable mobility across wired and wireless Local Area Networks (LANs). Since the needs of Internet have increased, the ultimate challenge for network designers is to allow the targeted networks to access sensitive corporate information in a secure manner via Internet. VPN technology is a technology that has been in existence for a while and it is a proven technology that allows businesses to transmit corporate data securely and economically over a large distance. The following figure (Figure 1) illustrates the basic architecture behind the VPN. This research study focuses on TCP and UDP performance evaluation on a VPN environment with different IPSec algorithms as parameters on four different Microsoft Windows operating systems.

![Virtual Private Network Diagram](image)

Figure 1: Virtual Private Network

A brief overview will be given in the next section to introduce the research thesis.

1.1 Overview

During the past few decades the growth of the VPN has gone beyond the imagination of its original makers and has become generally used in the networking industry. It is constructed as a private network within a public network infrastructure such as the global Internet by following the properties of a point-to-point private link, which provides a secure communication channel. As VPNs continue to evolve with a growing number of options, they
are much more attractive to small business owners for providing secured communication over a public infrastructure like Internet (Breslau, Chase, Duffield, Fenner, Mao, & Sen, 2006; Narayan, Brooking & De Vere, 2009). This unique VPN environment can be provided in many different architectural ways. Bresla et al. (2006) state that in a provider-based VPN, rather than connecting customer sites together using dedicated private lines, each customer site connects to one or more edge routers in the provider network. Such provider-based VPNs provide a scalable and secure way for a service provider to support many customers across its backbone.

One of the prime concerns about the VPN environment has been security over the private network territory. Its ability to provide secure communication by using the public network has created great interest among industries. As Breslau et al. (2006) note, VPN adds security procedures and protocols over insecure communication channels, and enterprise networks are increasingly turning to VPNs to connect geographically separate locations. Nowadays most of the organizations exist at different geographic locations and may span many geographical areas. This geographical distance challenges Information Technology (IT) infrastructure developers to think about their service security and efficiency. Narayan, Brooking & De Vere (2009) state that VPN has become a secure connection between geographically distributed network sites. In 2010 Narayan, Fitzgerald & Ram mentioned that there are numerous protocols, mechanisms and algorithms that can be used to implement VPN to encrypt and authenticate data as it travels between different locations.

Operating systems, protocols, and network media are at the core of the VPN systems that crucially facilitate better performance and security. According to earlier research, VPN provides different performance on different operating systems such as Microsoft Windows Server 2003, 2008, Microsoft XP, Vista, 7 and Linux. The VPN protocols like Point-To-Point Protocol (PPTP), Layer 2 Tunnelling Protocol (L2TP), IPSec have also been given prime observing targets with wired and wireless communication media.

All networking concepts are covered by Internet Protocol (IP). IPv4 (IP version 4) was introduced by the Internet Engineering Task Force (IETF) in the mid 70s. However, due to the
massive growth of Internet usage, IPv4 has run out of usable IP addresses, which creates a great need for another IP version. The evolution of new IP address versions has created massive research opportunities to researchers all over the world. It has become indeed a prime networking research study area.

A historical fact helps us to understand the background history and connect the present situation and how it arose in a way that everyone accepts. Since the Advanced Research Projects Agency Network (ARPANET) has been developed, researchers have worked on different networking research areas based on different networking architectures. VPN technology has been studied over a wide range of its occurrence, from performance observations of VPN protocols, payloads, and operating systems, to IP versions. Historical facts on VPNs research that I have come across have given me the motivation to conduct research on VPN that can benefit this fast moving networking industry. My motivation for this study is shown in the following section.

1.2 Motivation for this study

Realising the importance of the usage of VPN technology in industry motivated me to conduct research on this powerful and extremely important networking area. Rapid Internet growth is creating a requirement for a new IP version, which leads to a significant research opportunity for researchers. However, the complexity of converting the networking architecture to a fully IPv6 oriented platform and the inability of IPv6 to communicate with IPv4 directly have stopped further network upgrades in some business environments. Even though some networking giants like Microsoft, IBM, and Google have fully upgraded to IPv6, the IPv4 platforms are still in use. This thesis has been conducted on IPv4 and will provide information on the different performance of algorithms utilised in VPN protocol IPSec. It provides details of transmission protocols TCP/UDP and says which protocol gives the best network performance on VPN LAN and on which operating system. Research problems addresses in this research and its relevance are noted in the following section, 1.4.
1.3 Research problem

As it is explained above, there are some research questions that have been raised that address the prime objective of this thesis study.

The main research question is:

“Which combination of IPSec algorithm and operating system give the best network performance when measured for TCP/UDP traffic?”

The sub questions that need to be answered are:

- Does TCP/UDP network performance vary when implemented on a different client operating system on a Local Area Network?
- Does TCP/UDP network performance vary when implemented on a different server operating system on a Local Area Network?
- Which client operating system gives the best performance for IPSec cryptographic algorithm when measured for TCP/UDP traffic types?
- Which server operating system gives the best performance for IPSec cryptographic algorithm when measured for TCP/UDP traffic types?

The structure of this document thesis outline will be explained in the following section.

1.4 Thesis outline

This thesis has seven Chapters along with sub topics that lead the research study in a step-by-step progression throughout the research. Chapter one provides an introduction and overview of the purpose of this research study, followed by the structure of this thesis document. The novelty of the research is presented in Chapter two by reviewing multiple threads of research on this topic and also related study areas that are discussed clearly in this Chapter. This introduces the importance of network and network performances followed by all the components related to this study such as VPN, IPv4, IPv6, VPN protocols, TCP and UDP, cryptographic algorithms, metrics and prior research and studies that have explored different operating systems. Research gaps are identified and research analysis done to reveal the
research question. The research methodology is presented in Chapter three along with the data recording methods used in the data collection phase of this study. Chapter four explains the specifications of all hardware and software employed in this research as well as the test-bed network design for this experiment. All data gathered during the experiments are shown in Chapter five with the test phases in line with the charts. The results of this research together with in depth discussions of the findings are presented in Chapter six. The research conclusions are made in Chapter seven. It summarises the research findings that provide a better solution for a VPN environment. Prospects for future work arising from the research are also discussed in Chapter Seven.

1.5 Chapter Summary

This Chapter has provided an overview of the research study by explaining the historical facts of the research area, the motivation behind the study, and the research questions that are going to be answered and its relevance. Research contributions from this study are also overviewed in this Chapter, and finally the outline of the thesis document is clearly explained.

Because of the research problem, the topic area and the purpose, previous research literature is reviewed in order to define a theoretical framework to use for the study and to understand what others have done in the same areas. Therefore, to achieve good research results a literature review of the research area is carried out in the next section, Chapter two.
Chapter 2: Literature Review

This section gives an overview of the research domain and the prior research studies that have been explored in regards to this research stream. It considers the demands, such as what analysis has been done before, what questions have been addressed and answered before, and what results have been produced. The existing data availability with the same research theme was analysed to confirm this research distinctiveness.

Firstly, general networking and the importance of networking performance will be overviewed in terms of understanding the overall networking philosophy as a foundation for this research study. This is followed by the main components of this research, such as VPN, IPv4, IPv6, VPN protocol IPSec, TCP/UDP and algorithms that are overviewed in terms of understanding the technological philosophy. Furthermore, the research that had been done on wired VPN with different operating systems will be discussed to identify the research gaps in order to develop this research study. Literature was gathered from credible resources like IEEE Computer Society, ACM, Conference papers and books published in the networking stream. Reviewing the literature begins in section 2.1 with exploring the computer networking research stream and the importance of networking.

2.1 Computer Networking and the Importance of a Network

Since this research area is mainly wrapped around the term “network“, it is a fair beginning for the research to define and examine its industry acceptance. Ferguson & Huston (1998) state that a network consists of any number of devices which can communicate through some arbitrary method, and devices of this nature include computers, routers, printers, and may reside in geographically diverse locations. A computer network can be defined as a group of systems that are interconnected to allow sharing of resources such as files, printers, services, and that the main two aspects of a setting up a network are the hardware used to connect the system together and the software installed on the computer to allow them to communicate with each component in network (Burg & Kenny, 2000). They further mentioned that early
networking goes back one and half decades before the invention of LAN technology in the early 1970s with the US Air Force SAGE network implementation.

Information and communication (ICT) are two of the most important strategic items for the success of every enterprise system. Even though nearly every organization uses a substantial number of computers, other technological tools (printers, scanners, projectors, cameras) and communication tools (telephones, fax, personal handheld devices), still may be isolated. However, although managers today are able to use the newest applications, many departments still do not communicate and much needed information cannot be readily accessed. To overcome these obstacles in effective usage of information technology, computer networks are necessary in the ICT environment in every enterprise system. Only with the help of computer networks can a borderless communication and information environment be built that allows users to access the network in many different ways.

To cater for the demand of the networking stream there are different types of computer networks available in the world that vary in size, topology and medium. However, variations do not matter, as using the computer network concept is important regardless of the variations. According to Burg and Kenny (2000), in the mid-1990s, LAN was created with personal computers and other devices linked together, and had become the prominent computer architecture in institutions. Smith (2002) said that the “importance of computer networking is nowhere more evident than in the phenomenal growth of the World Wide Web”. The ability to access remote programs and remote databases either of the same organization or from other enterprises or public sources through a computer network makes communication faster than other facilities. Beside the above major reasons, an organization should have a computer network because of the factor of cost reduction through sharing hardware and software resources, the high reliability of multiple sources of supply, and the greater flexibility from connecting various devises from various vendors. Considering all the above, it is obvious that there is a great need for computer networking all over the enterprise systems. When networking people realised the importance of network performance it became a necessity. As a result of this understanding, network performance has been researched enormously by various
researchers in many different domains. Previous research done in the network performance area will be reviewed in the next section, 2.2.

2.2 Performance of a network

Networking is mostly about connecting devices together to communicate in some fashion, to transmit and received data, and to share information, services, and resources. When data is sent from one place to another, a significant characteristic of any network is its speed, which determines the overall performance of a network. While network performance is a crucial task in network administration, network performance evaluation has become one of the major threads in the ICT world. Park, Kirn and Crovella (1997) stated that there are number of papers that have studied the implications of long-range dependence on traffic modelling and network performance evaluation. Also, that different aspects of network performance may vary and there are generic factors which influence the performance of a LAN.

Research studies of network performance can be done based on the different hardware, software, protocols, services, technologies, traffic information, and use within the network. Park, Kirn and Crovella (1997) classified network performance research avenues into two categories; one deals primarily with traffic characterization and modelling issues, with the other concentrating on the performance evaluation side. Hong and Li (2009) studied the impact of information on network performance and propose a general information theoretical framework which can be applied to any network. Narayan, Graham & Barbour (2009) have done a literature review on generic factors that influence the performance of LAN, mainly focusing on the performance and metrics of commonly used operating systems that have been implemented to create IT infrastructure. Their literature findings show that performance analysis, internet protocols and wireless are the major themes in literature.

Protocols and network media play a major role in a network to establish network communication channels and maintain performance. Narayan, Graham & Barbour (2009) state that IP is the basic building block used to enable information technology communication
channels and to improve the performance of the overall operating system and the network, which proves that the performance of the IP stack needs to be improved. Qiang & Nagurney (2008) published information on unified network performance measures and stated that in order to be able to evaluate the vulnerability and the reliability of a network, a measure that can quantifiably capture the efficiency or performance of a network must be developed.

Network performance has been tested by different network parameters. These parameters are called performance metrics that can be measured to evaluate the performance of a network. Many researchers have observed performance metrics such as delay, jitter, and throughput as well as Central Processing Unit (CPU) utilization when investigating performance evaluation in the real network environment. Narayan, Graham and Barbour (2009) stated that when analysing network performance for both effectiveness and efficiency, the performance metrics need to be selected carefully. They also say that in performance analysis literature, the most common evidence is throughput, round trip time (latency) and CPU utilization. Hong and Li (2009) studied the relationship between network information and network performance and stated that network information, channel state, traffic information and network topology are essential factors in a network performance. They further mentioned that the more you have network information, the more protocol will be efficient which leads more network performance. On the other hand collecting and disseminating of network information consumes more bandwidth.

Throughput is one of the valuable metrics in network performance and does the end-to-end measurement. Narayan, Graham & Barbour (2009) mentioned that “the most common metric evident in literature is throughput”, by explaining that the importance in understanding total network performance. They further said that the percentage of CPU utilization is also a valid metric in literature. Since the CPU resources are utilised by other processors running on the nodes, Narayan, Graham and Barbour (2009) mentioned that the percentage of CPU utilization of nodes also a valid metric. All the above literature statements prove that the common network performance metrics are throughput, jitter, CPU usage, TTL, and latency, which react in many different ways according to the protocols and architecture of the network.
The following section, 2.3, reviews the previous research areas that are specifically relevant to this research in order to define a theoretical framework for this research study. It reviews IP versions, VPN technology, VPN protocol IPSec, TCP/UDP, cryptographic algorithms and finally the metrics researched.

2.3 Relevant Researched Areas

This research includes several network areas such as IPv4 and IPv6, VPN, VPN protocols, TCP, UDP, cryptographic algorithms and research metrics, which will be discussed in the next few sections.

2.3.1 Internet Protocol - IPv4 and IPv6

Components in a network cannot communicate each other unless we install software. We call such a software program a protocol, which enables the rituals of communication in a network. IP is one of the most ubiquitous features of networking, so that any of the hardware, software or any networking product needs to be generally accepted, regardless of whether it is from a computer vendor like IBM, Microsoft, Apple, Sun, Novell, Compact, Netscape. IP is capable of linking networks, which requires mainly three things. Firstly, every component in the network needs to be uniquely identified; secondly, it must be able to send and receive data to and from other components in a format that everyone can understand, and finally, data transmission need to be reliable (Loshin, 1999; Salus, 2000; Raicu & Zeadally, 2003). IP did not start becoming common until the late 1980s and early 1990s. According to Das (2008), Loshin (1999) and Salus (2000) even then it was marketed as a specialty until the 1995 Transmission Control Protocol and Internet Protocol (TCP/IP) was incorporated into personal computer products. Since then TCP/IP has become the protocol suite used by most small, medium and large networking environments and has operated across the globe as the protocol of the Internet.

TCP/IP is a full suite of protocols as shown in Figure 2, and it contains five layers called Physical, Link, Network, Transport and Application. The representation from the sender’s end to the receiver’s end uses layers which are conceptually derived from the Open System...
Interconnection (OSI) model with seven layers called Physical, Data Link, Network, Transport, Session, Presentation and Application. Although OSI is the only the internationally accepted set of standards for communication, it is barely used by the manufacturers. TCP/IP was already in use and was popular among the technical community (Loshin, 1999). The entire networking philosophy is wraparound with OSI model with different architectural ways.

![TCP/IP protocol suite](image)

The numbers of products that incorporate IP from computer hardware and software to mobile computer devices and home entertainment products with wireless Internet connectivity, demonstrates how important IP is to the world of communication infrastructure. IP is one of the protocols in the suite which provides globally unique addresses in dotted quad notation, transmits data in packets and performs routing between IP based networks. IP is basically responsible for a unique address, connectionless communication, which means that no effort is made to set up a dedicated end-to-end virtual connection, routing, which means the process of moving data from one network to another by forwarding packets via gateways, and unicasting, broadcasting and multicasting. There are two versions of IP, Internet Protocol version 4 (IPv4) and Internet Protocol version 6 (IPv6). IPv4 was developed in the mid 70s, and IPv6 was developed in the late 90s, (Das, 2008; Loshin, 1999; Salus, 2000). IPv4 turned out to be the most widely deployed network layer protocol and the only standard internetwork-layer protocol used on the Internet. More information about the two IP versions will be explained in the next two sections.
2.3.1.1 IPv4

Internet Protocol version 4 or IPv4 is the most widely used network layer protocol on the Internet. IPv4 is a data oriented protocol used over packet switched networks that neither guarantees the delivery nor the correctness of the data delivered. IPv4 offers 32-bit (4Bytes) address space of $2^{32}$ that is equivalent to about 4,294,967,296 (4.3 billion: $2^{32}$) unique addresses on the network. The IPv4 protocol is described in IETF RFC 791 (September 1981). (Loshin, 1999; Salus, 2000; Das, 2008; Narayana, Lutui, Vijayakumar, & Sodhi, 2010; Narayan & Shi, 2010; Narayan & Tauch, 2010) In 2008 total unallocated address space remained at 16% and it was commonly accepted that IPv4 would run out of addresses by the end of 2010 or 2011 (Narayan, Kolahi, Sunarto, Nguyen & Mani, 2008). Before the Internet exceeded the expectation of the original developers limit, IPv4 was the most suitable IP version for Internet without any doubt.

IPv4 limitations

Due to the tremendous growth of the Internet, public IP addresses of IPv4 exceeded the limits of the IP architecture, which led the TCP/IP engineers and designers to rethink the need for an upgrade as early as the late 1980s. The IP upgrade was mainly motivated by the IP address space crisis and also some improvements were being targeted at the same time. Although IP performed remarkably well everyone believed that there was a needed for improvement in performance, security and the auto configuration feature. However as a result of this space crunch identification, in the late 1999 IETF started an IPng (IP next generation) development. After many discussions under RFC 1550, around 1995 IETF chose the IPv6 version as the final IPng proposal, and base specifications were specified in RFC 2460. With an increasing number of networked devices, there was a real threat of address space exhaustion that was successfully prevented by the much larger address space offered by the IPv6 (Loshin, 1999; Salus, 2000; Raicu & Zeadally, 2003). As a result of this space crunch, a new era of IP, version IPv6, has arisen. This new version’s structure and capabilities will be explained in the following section.
2.3.1.2 IPv6

IPv6 offered an address space of $2^{128}$ or approximately 5x1028 ($2^{128}$) unique addresses to be used over the Internet. The IPv6 protocol is described in the IETF RFC 2460 (1998). IPv6 addresses are four times as long as IPv4 addresses: the basic representation of an IPv4 address is in the form X:X:X:X:X:X:X:X, where X refers to four digit hexadecimal integers (16 bits). IPv6 addresses are broken into two portions, the subnet prefix and the interface ID. For example 1020:0:0:C9B4:FF12:48AA:1A2B/60 indicates that the first 60 bits refers to a subnet prefix, for routing purposes. There are basically three types of IPv6 addresses, the unicast, multicast and anycast. Broadcast addresses are no longer available with IPv6. As per the RFC 2373, the unicast address is an identifier for a single interface, so the packet sent to a unicast address is delivered to the interface identified by that address. The multicast address is an identifier for a set of interfaces. When the packet is sent to a multicast address, it is delivered to all interfaces identified by that address. The anycast address is an identifier for a set of interfaces. (Loshin, 1999; Salus, 2000; Raicu & Zeadally, 2003; Das, 2008; Narayan & Shi, 2009). The format of the architecture of IPv6 has proved to be a valuable answer for the IPv4 space crunch because of its additional features.

2.3.1.3 IPv4 and IPv6 comparison

When the RFC 2460 the changed from IPv4 to IPv6, it expanded the addressing capabilities by improving the address size from 32 to 128, it simplified the header information by dropping or making optional of some IPv4 header information, it improved support for extension and options by changing the way IP header options are encoded for efficient forwarding, it added a new capability to enable the labelling of the packets, and it also added extensions to support authentication, data integrity and data confidentiality (Salus, 2000). Dias (2008) and Narayan & Tauch (2010) stated that the resulting IPv6 protocol offers various improvements, such as a significantly larger address space (from $2^{32}$ to $2^{128}$), enhanced the user benefits, a simplified header structure, and enhanced security. It is fully interoperable and offers a smooth transition, contains provision for flexible protocol, auto configuration, multicasting, better provisions for ad-hoc networking that is required by the different wireless devices and integrated Quality of
Service (QoS). Govil (2007) stated that IPv6 offers many enhancements like increased addressing capacity and capabilities, built in IPSec security, mobility QoS control, and improved routing efficiency. On the next page Figure 3 illustrates the IPv4 and Figure 4 illustrates the IPv6 header format to show the difference between the header information of the two IP versions.

(Loshin, 1999; Salus, 2000)

Even though IPv6 resolved the address space crunch, it has not succeeded in the transformation from IPv4 to IPv6 as we expected and as IETF planned. Narayan, Lutui, Vijayakumar and Sodhi (2010) stated that there are many reasons that IPv6 uptake is slow, such as that IPv4 to IPv6 conversion is a massive undertaking due to the configuration changes of many computers on the global network and the supporting networking infrastructure hardware.
with the use of Network Address Translation (NAT). They divided this transaction mechanism into three categories;

1. Dual stack (where devices are loaded with both IPv4 and IPv6 stack to allow communication with both types of devices)

2. IPv4/IPv6 translation (accepts from one version, converts to another and then sends to destination in the desired version)

3. IPv4/IPv6 tunnelling (where IPv6 packets are encapsulated with IPv4 and then sent on convolutional IPv4 network)

(Narayan, Lutui, Vijayakumar and Sodhi, 2010; Narayan, Kolahi, Sunarto, Nguyen & Mani, 2008)

Most researchers have concentrated on this building block of Internet architecture. On account of the concentration on this research thread, extensive studies have been conducted related to performance evaluation of IPv4 and IPv6 on different operating systems. Narayan, Shang and Fan (2009) have done research on performance evaluation of IPv4 and IPv6 on Windows Vista and Linux Ubuntu and concluded that IPv4 gives a slightly better throughput than IPv6 with packets sizes that are larger than 256Bytes; however, the performance is almost identical for small packet sizes. The IPv4 and IPv6 performance evaluation done in different operating systems is tabulated in Table 1.

<table>
<thead>
<tr>
<th>Researcher(s)</th>
<th>Operating System(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anand (2000)</td>
<td>Linux Kernal 2.4.17 (IPv6)</td>
</tr>
<tr>
<td>Ettikan (2000)</td>
<td>FreeBSD (KAME IPv6)</td>
</tr>
<tr>
<td>Ariga, et.al (2000)</td>
<td>FreeBSD (KAME IPv6)</td>
</tr>
</tbody>
</table>

Table 1 : Related research on IPv4/IPv6 performance evaluation on different operating systems

(Narayan, Lutui, Vijayakumar and Sodhi, 2010)
Since IP version is a major component in every kind of networking, it gave an impetus to this VPN study. It is vital to understand the IP versions when conducting research into networking framework. However since this thesis is mainly concern with VPN technology, the next section, 2.3.2, reviews previous research done on VPN architecture.

2.3.2 Virtual Private Network

VPN utilization has become common in the business world with the exponential development of Internet. There has always been a clear division between public and private networks. A public network is a large collection of unrelated peers that exchange information freely with each other, such as Internet and public telephone systems. A private network, on the other hand is composed of computers owned by a single organization, which share information specifically with each other. While the public networks can be accessed by any given person, regardless of commonality of Internet access, private networks assure communication only between people in a specific group that has the common requirement of accessing similar information with each other. The line between public and private networks has always been drawn by the gateway router. In the early 1990s, emergence of the VPN concept blurred the line between public and private networks and moved the network technology to a new stage. Nobody stepped forward to patent the idea at the beginning, because there was neither a private nor secure connection between a server and its multiple interfaces (Scott, Wolfe, & Erwin, 1999; Bestvpnserver, 2012). However, VPN was vastly improved with the help of the biggest company, Microsoft, in the computer software industry. Therefore, the conclusion can be made is that it is worth the effort to conduct a research on this ubiquitous computing (ubicomp) technology. In the next section, a thorough VPN study begins by revealing the literature that understands the architecture and functions of VPN.

2.3.2.1 What does a VPN do?

This section gives an overview of VPN architecture by explaining its functions and features. A VPN is a way of simulating a private network over the public network that can be created using hardware, software, or combination of both. It creates a secure connection between
peers over a public network using encryption, authentication, packet tunnelling and firewalls. Since it depends on the use of a virtual connection, it is called “virtual”. The term “VPN,” or Virtual Private Network, has become almost ubiquitous in the networking industry as a secure communication technology. Scott, Wolfe, & Erwin (1999) pointed out that the main idea behind VPN is to create a private, secure point-to-point communication channel via tunnelling and/or encryption across a private or a public network like Internet. They further mentioned that secure virtual connection can be created between a machine and a network, two machines or two networks. As an example, the VPN can provide a secure connection between a salesperson’s laptop and the Intranet web server running the database. Microsoft Technet (2001) stated that VPN creates links across shared or public networks and it is an extension of a private network. It enables you to transfer data between two nodes across a shared or public network like Internet by following the properties of a point-to-point private link. Figure 5 illustrates how Microsoft Technet sees the architecture of VPN.

![Figure 5: A VPN Connection](image)

Narayan, Kolahi, Brooking, & de Vere (2008) came up with a somewhat formal characterization of the VPN term: “VPN is a technology that provides secure communication for data as it transits through insecure regions of information technology infrastructure.” In 2009 Narayan, Kolahi, & de Vere stated that; “VPN is commonly used in business situations to provide secure communication channels over public infrastructure such as Internet. And VPN is a proven technology that does provide security strong enough for business use.” In 2010 Narayan, Fitzgerald & Ram defined VPN as “an inexpensive methodology to secure connections between network sites that exist at different geographic locations.” According to Diab, Tohme and Bassil (2007), VPN is considered as a stronger security solution for the communication between nodes in an intranet over the unsecured IP networks. McGregor & Lee (2000) also provided a more
approximate, simpler and formal description. The methods of constructing the VPN solutions differ depending on the various VPN application scenarios.

VPN architectures can be divided into three common scenarios: site-to-site Intranet VPN, remote access VPN and extranet VPN. Multiple network sites are located at different geographical locations but connected with each other within the same organization using VPN named as site-to-site Intranet VPN (Figure 6).

![Figure 6: Site-to-Site VPN](image)

So each site can have multiple subnets that create a cooperate intranet, and using VPN, each site can be formed into a large intranet. If a VPN is used to connect a single remote network device, such as a portable computer, to connect to a cooperating network via any form of connectivity such as cable modem or Digital Subscriber Line (DSL), this is called a remote access VPN (Figure 7).

![Figure 7: A Remote Access VPN](image)

If the network resources within one corporation are opened for access to other corporations for various purposes, such as business transactions, this network architecture is considered to be an extranet VPN or point-to-point network (Figure 8). Apart from that, there are software
based VPN systems that use existing Internet connections. Because the client software connection looks like it is dialling up, it is called Dial-up VPN. By using an existing Internet connection, a secure "tunnel" is created between two points allowing a remote user to connect to a remote network. This can be set up with various types of software or hardware, but it requires third party software to be loaded on remote workstations (Rodriguez, 2008; Diab et al., 2007; MicrosoftTechnet, 2003; Yuan et al., 2001; Bestvpnserver, 2012). Pena & Evans (2000) described the VPN software solution as an alternative VPN solution implemented in software that provides economical and accessible advantages and has a significant impact on performance, producing high CPU usage and limiting network throughput.

The essence of creating this cohesive VPN architecture is that it assembles many computer technological components to provides practical solutions for organizational communication. There are mainly four categories of components: tunnelling, authentication, access control and data security, which make both the “virtual” and “private” aspects of VPN (Scott, Wolfe, & Erwin, 1999; Yuan et al., 2001). In the next few sub topics, information about the above mentioned components will be reviewed.

**Tunnelling**

Tunnelling is an architectural concept in which virtual topology is created on top of the physical topology with one or more repeated protocol layers. It is simply a method of using a network infrastructure to transfer data for one network over another network. A packet travelling from
host A to host H passes through each node along the path, in this case B through G and there is no way to skip any of these nodes. However, if node C takes the original packet and places it completely within a new packet addressed to node F then as the new packet passes through nodes D and E, these nodes would only know the packet as if it were addressed to F. They would not know the original destination H. Therefore, in this case it can be said that the original packet is tunnelled from D to F, represented by the heavy line in Figure 8. Tunnelling can be used in any network layer but the most common layers are layer 2 - Data link layer and layer 3, Network layer. In Layer 2 tunnelling, a link layer frame is placed into the payload of a protocol data unit from some other layer. Examples for layer 2 protocols are Point-To-Point Protocol (PPTP), Layer Two Forwarding Protocol (L2F) and L2TP. A layer 3 tunnelling and a layer 3 packet are placed into the payload of some other layer. IPSec protocols Authentication Header (AH) and Encapsulating Security Payload (ESP) tunnel modes are good examples of layer 3 protocols (Diab et al., 2007; Yuan et al., 2001). Tunnelling is one of the main features in the VPN process.

**Authentication**

Authentication is an essential technique in VPN since it ensures communication between correct users or hosts. This technique typically performs at the beginning of the session by allowing the user to log in to a system with a username and password and also perform randomly during the session. Most VPN authentication systems are based on a shared key system, in which the keys are run through hashing algorithms and generate a hash value. The other end holds its own hash value and compares it to the one it received from the other end. The actual hash value sent across the Internet is meaningless to an observer, which prevents sniffing. The Challenge Handshake Authentication Protocol (CHAP) is an example of this process (Yuan et al., 2001). According to MicrosoftTchnet (2003), there are three types of authentication for VPN connections. In the User authentication the VPN connection is established after the VPN server authenticates the VPN client attempting the connection and verifies that the VPN client has the appropriate permissions. The user attempting the L2TP/IPSec or PPTP connection is authenticated using Point-to-Point (PPP) based user authentication protocols like Microsoft Challenge-Handshake Authentication Protocol (MS-CHAP), Microsoft Challenge-Handshake Authentication Protocol version 2 (MS-CHAP v2),
Extensible Authentication Protocol-Transport Layer Security (EAP-TLS), Password Authentication Protocol (PAP) and Shiva Password Authentication Protocol (SPAP). The Computer authentication with L2TP/IPSec performs computer-level authentication with IPSec, L2TP/IPSec connections and also verifies that the remote access client computer is trusted. Data authentication and integrity verifies that the data being sent on an L2TP/IPSec VPN connection originated at the other end of the connection and was not modified in transit. L2TP/IPSec packets include a cryptographic checksum based on an encryption key known only to the sender and the receiver.

**Access control**

When the authentication process is completed the communication sources can decide whether to continue the session or reject the session. This secure communication allows authorised access to resources. Access controls contain two features: one is the information (including the identity of the entity that is requesting access and the resources to be accessed), which is on the access controller’s decision. And the second feature is how the access controller made the decision based on the information available, which is basically a decision-making process. (Yuan et al., 2001) The access control mechanism is capable of assuring the security of VPN architecture.

**Data security**

The early stage of internet design was an open environment that was used for communication among the users without any central control and thus mutual mistrust was not of primary concern. This environment was a place where attacks could be quite easy and hard to prevent, detect and trace. Also it was difficult to ensure the main security goals regarding confidentiality, integrity and availability. Because of all these issues, a secure VPN over the public Internet has become vital for the IT world (Adeyinka, 2008). VPN belongs to a network that uses IP tunnels that use cryptographic techniques to provide robust privacy and security and it forms a virtual network over the Internet (Khanvilkar & Khokhar, 2004). Further, remote users get all the benefits of a private network and operations get the benefit of high security with low operational costs when using VPN.
With the productive growth of the Internet VPN has been popular in the business environment due to its cost effectiveness, security improvements and geographically distribution. Due to the exponential growth of Internet usage, however, security has become a prominent concern (McGregor & Lee, 2000). Khanvikar and Khokhar (2004) mentioned that VPN has gained immense popularity among commercial and defence organizations due to its ability to provide secure connections at lower cost. Narayan, Fitzgerald and Ram (2008) said that the VPN solution has become an economical methodology that provides a more secure connection between geographically distributed sites. Narayan, Brooking & de Vere (2009) stated that organizations span large geographical areas and it is the ultimate goal for network designers to provide secure and efficient communication channels thorough It infrastructure. Due to VPN’s cost effective technology for secure data transmission over long distances, organizations can escape from the traditional physical cable installation over large distances (Narayan, Brooking & de Vere, 2009). According to Scott, Wolfe, & Erwin, 1999; Yuan et al. (2001), an internet service can be viewed in two ways: topologically and architecturally. Topologically, it is a collection of networks connected by gateways or routers. Architecturally, it is a collection of protocol layers that convert user data into manageable units and control the transmission. It is must to have a look at of VPN protocols need to be examined as they play a major role in networking. Section 2.3.3 reviews information from the history of VPN protocols.

2.3.3 VPN Protocols

This section defines the rules that govern the VPN as protocols. To harmonise all the different products and services in a network, several protocols have been developed over the years. In the VPN context, one of the major concerns is to provide secure connection. Narayan, Kolahi, Brooking, & de Vere (2008) stated that business environments have implemented a VPN that uses different protocols to provide secure and authentic data transfer between multiple sites connected over a public communication infrastructure. This VPN architecture with security infrastructure is provided by various VPN protocols.

VPN tunnels serve three major purposes in VPNs, such as encapsulating one protocol within another, which helps different protocols to be transported over an IP infrastructure, route
privately addressed packets through a publicly addressed infrastructure and provide private data integrity and confidentiality. Using encryption techniques, VPN is capable of preventing datagram capture and analysis while they are in the public network. Commonly, both layer 2 and layer 3 are useful for VPNs. (Wu, 2009; Diab et al., 2007) There are basically four categories of VPN tunnelling protocols. Layer 2 protocols such as PPTP, Layer Two Forwarding protocol (L2F), and L2TP are specifically design to tunnel PPP and protocols for session traversal across firewalls securely (SOCKS v5). Wu (2009) illustrates the tunnelling protocols position at OSI model as per Table 2.

<table>
<thead>
<tr>
<th>OSI Layer</th>
<th>Security Technology</th>
<th>Security Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Layer</td>
<td>Application Agent</td>
<td></td>
</tr>
<tr>
<td>Presentation Layer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Session Layer</td>
<td>Session Layer Agent</td>
<td>SOCKS v5 / SSL</td>
</tr>
<tr>
<td>Transport Layer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network Layer</td>
<td>Packet Layer</td>
<td>IPSec</td>
</tr>
<tr>
<td>Data Link Layer</td>
<td></td>
<td>PPTP/L2F/L2Tp</td>
</tr>
<tr>
<td>Physical Layer</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: The position of security protocols on OSI model

Since this arrangement is particularly useful for mobile network connections PPP, which is called multiprotocol due to the capability of carrying datagrams from multiple protocols, it has become popular for connecting two devices over the phone lines. Basically what is happening in this connection is firstly, users connect to the network on demand through public switch telephone network (PSTN) or ISDN. Then the remote user sets up a PPP connection from the remote computer to the remote access server (RAS) that is at home or work. When the PPP connection is established the remote computer starts sending IP or any datagram inside the PPP frame. Then the RAS removes the datagrams from the PPP frames and inserts them into the private network. When one intranet is connecting to another, Layer 3 tunnelling plays a most useful role for VPN’s security tunnelling within an IP network. Internet Protocol Security (IPSec), Secure Socket Layer (SSL) protocol, PPTP and Secure Socket Tunnelling Protocol (SSTP), L2TP are commonly used protocol in VPN to ensure security. Narayan, Kolahi, Brooking, & de Vere (2008) and Narayan, Brooking, & de Vere (2009) named IPSec, PPTP and SSL as widely
used VPN protocols in the industry, both in open source implementation and commercial products.

Security is the main feature of VPN implementation as it provides integrity and encryption of data transaction by VPN protocols by implementing different algorithms. The following table (Table 3) illustrates the summary of VPN protocols. Some of the above protocols will be discussed in the next few sections.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Developed by</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPTP</td>
<td>Microsoft, Ascend Communications, 3Com</td>
<td>No longer used</td>
</tr>
<tr>
<td>L2TP</td>
<td>Cisco company</td>
<td>Mainly used in Cisco routers</td>
</tr>
<tr>
<td>IPSec</td>
<td>Internet Engineering Task Force (IETF)</td>
<td>Security Scheme of Internet protocol</td>
</tr>
<tr>
<td>SOCKS v5</td>
<td>David Koblas, a systems engineer for MIPS computer</td>
<td>Security protocol of TCP layer</td>
</tr>
</tbody>
</table>

Table 3: VPN protocol summary

2.3.3.1 Internet Protocol Security (IPSec)

Internet Protocol Security (IPSec) is a protocol suite defined by the IETF which was originally developed to achieve the goal of enabling the protection of all types of Internet protocol (IP) communications by protecting multiple peers at the layer 3 network layer, in both the IPv4 and IPv6 environments. It is widely implemented in VPNs. IPSec VPNs have been deployed by many organisations to provide enterprise-level secure remote access by protecting the IP packet exchanged between remote networks or hosts and an IPSec gateway located at the edge of a private network. VPN security running on the TCP/IP protocol suite is ensured by the IPSec (Wu, 2009; Adeyinka, 2008; Narayan, Kolahi, Brooking, & de Vere, 2008; Diab et al., 2007; Yuan et al., 2001; Ferguson & Schneier, 2000; Ariga et al., 2000). According to the above research, IPSec supports a network in many ways, such as by supporting level peer authentication, data origin authentication, data integrity, data confidentiality, and replay protection. It supports a series of cryptographic security services such as DES, 3DES and AES.

IPSec is designed to allow encryption and authentication of network traffic between the host machines over an existing TCP/IP, hence it is associated with the authentication mechanism
which manages the authentication and encryption phases between the client and the gateway computers. Using an IPSec is a way of providing additional security as IPSec can implement two data encryption modes known as Transport and Tunnel. Transport mode encrypts the data portion (payload) of each packet, leaves the header untouched and is most commonly used to secure data communication within a network. On the other hand, tunnel mode encrypts both the header and the payload. It is more secure and used for securing data communication that traverses unknown third party networks. It is also used for network-to-network communication (Adedayinka, 2008; Narayan, Kolahi, Brooking, & de Vere, 2008). All the above security mechanisms are determined by different types of protocols.

The IPSec framework provides security services at the IP layer by enabling a system to select the required security protocols that determine the algorithm(s) to use for the service(s). IPSec packet level security is provided mainly by two protocols: AH and ESP. As per Narayan, Kolahi, Brooking, & de Vere (2008), AH provides guaranteed connectionless integrity and data origin authentication of the IP datagrams, and it also protects against replay packets. They also mentioned that ESP provides origin authenticity, confidentiality protection, integrity of a packet, authentication-only implementation and encryption-only implementation.

IPSec based VPNs are used in most organizations as remote access technology to establish a secure connection in entire private networks by protecting the IP packet exchanged between remote networks or hosts. Many organizations find that IPSec meets the requirements of users already using the technology. However, in this implementation IT administrators must determine who should have remote access to the network since IPSec VPNs require a client to be installed on each user machine. Therefore, this solution becomes resource intensive and cost prohibitive for large enterprises and can create more responsibilities on deployment, configuration and maintenance (Adedayinka, 2008). Recently people have sought other alternatives for IPSec VPN due to the lack of understanding of IPSec standards, and the complexities of setting up and maintaining IPSec VPNs.
2.3.3.2 Secure Socket Tunnelling Protocol (SSTP)

SSTP is released and owned by Microsoft Co-operation. (Microsoft Tech-net, 2007) According to MicrosoftTechnet (2007), SSTP is one of the protocols that it uses for VPN connections. It allows traffic to pass through firewalls that block the PPTP and Internet Protocol security/ Layer 2 Tunnelling Protocol (L2TP/IPSec) traffic. All the above protocols are used in VPN connections but there are differences between them in how they are executed with different operating systems.

The mechanism provided by the SSTP encapsulates the PPP traffic over the SSL channel of the Hypertext Transfer Protocol Secure (HTTPS) protocol. In this process the PPP supports strong authentication methods like EAP-TLS. With the use of HTTPS, traffic will flow through TCP port 443 (the port commonly used for web access) and transport-level security with enhanced key negotiation, integrity, and encryption will be provided by the SSL (Microsoft Tech-net, 2007). Their data flow for an SSTP-based VPN connection execution is as follows:

1. The SSTP client establishes a TCP connection with the SSTP server between a dynamically allocated TCP port on the SSTP client and TCP port 443 on the SSTP server.
2. The SSTP client sends an SSL Client-Hello message, indicating that the SSTP client wants to create an SSL session with the SSTP server.
3. The SSTP server sends its computer certificate to the SSTP client.
4. The SSTP client validates the computer certificate, determines the encryption method for the SSL session generates an SSL session key and encrypts it with the public key of the SSTP server’s certificate, and then sends the encrypted form of the SSL session key to the SSTP server.
5. The SSTP server decrypts the encrypted SSL session key with the private key of its computer certificate. All future communication between the SSTP client and the SSTP server is encrypted with the negotiated encryption method and SSL session key.
6. The SSTP client sends an HTTP over SSL request message to the SSTP server.
7. The SSTP client negotiates an SSTP tunnel with the SSTP server.
8. The SSTP client negotiates a PPP connection with the SSTP server. This negotiation includes authenticating the user’s credentials with a PPP authentication method and configuring settings for Internet Protocol version 4 (IPv4) or Internet Protocol version 6 (IPv6) traffic.
9. The SSTP client begins sending IPv4 or IPv6 traffic over the PPP link.

(Microsoft Tech-net, 2007)
Massive research has been undertaken on VPN performance since its introduction relating to many areas such as protocols used, operating systems, algorithms used, and security implementation. In relation to this research, the transmission control protocols TCP and UDP will be the next focused area (section 2.3.4) to be reviewed using various literature articles.

2.3.4 The Transmission Protocols - TCP and UDP

This section has reviewed the literature of transmission control protocols that relate to this research study. Internet Protocol suite (TCP/UDP) probably is the oldest and most popular networking standard used in the networking backbone since the ARPANET started in 1973. These file transmission protocols provide the functions of the transport layer in the OSI model, which was developed by International Standardisation Organisation (ISO) in the late 1970s.

This reference model has seven layers (Table 4) that define vendor independent protocols and standards for the interconnection of computer and network equipment. At the sending computer, the data is passed down the layers. Each layer adds information to the data for use by its peer layer in the receiving computer. At the receiving computer, the data is passed up the layers. Each layer processes the information attached by its peer layer and then passes the data to the higher layer. In effect, during communication, processes running in each layer on each computer communicate with each other as peers. Each layer of the OSI model has different protocols associated with it to handle the data that defines rules and procedures in order to communicate with processes on other systems (Bansal et al., 2004; Shelly, Cashman, & Serwatka, 2004; Xylenomenos & Polyzos, 1999).
As a result of the rate of internet growth, Defence Advanced Research Project Agency (DARPA) has developed a simplified model, the TCP/IP model. Unlike the OSI model the TCP/IP model, it has only four layers. The TCP/IP model is illustrated in Table 6 with the protocols that each layer uses. The application-to-application delivery is achieved through a set of functions performed at the transport layer. TCP transfers information in a different way than the UDP. The following Table 5 gives an overview of TCP and UDP protocols and Figures 9 and 10 illustrate a comparison between data packets.

<table>
<thead>
<tr>
<th>OSI Reference Model</th>
<th>Layer</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>7</td>
<td>Application</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>Presentation</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>Session</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>Transport</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Network</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Data Link</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Physical Layer</td>
</tr>
</tbody>
</table>

Table 4: OSI Reference Model

TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) are two transport layer protocols in the TCP/IP model. TCP establishes a (virtual) connection between sender and receiver. (connection-oriented) There is no virtual connection (connectionless).

Virtual: the application layer “thinks” that a single path has been created; in reality packets can travel different physical paths: reliable connection. The protocol is used when the application needs to send one packet quickly without the overhead of connection creation and termination: unreliable connection.

TCP packets are called segments UDP packets are called datagrams

The data delivery is slower and more complicated, and most application protocols like SMTP, HTTP, FTP and TELNET use TCP. UDP is designed for speed and is suitable for applications like video-conferencing and ping.

Table 5: TCP/UDP Comparison
VPN uses UDP to get through NAT devices. IPSec traffic is being encapsulated in UDP and wraps an IPSec packet. The native IPSec packet would have an IP protocol header with value 50. NAT gateways drop the packet rather than pass it and IPSec does not have a port number. So, with large number of access, only one can get through the VPN as the VPN concentrator only does IP level interpretation. When a NAT device will be detected on the path, UDP ports will be used to encapsulate of IPSec ESP traffic. By encapsulating inside of a UDP packet, the address of the IP header gets translated when it goes through the NAT device and allows many VPN points behind a NAT device. Additional header strips off when the packet reaches its destination, but leaves the original IPSec packet, that pass all other validations. If there no NAT devices between routers, ESP would not need to be encapsulated in UDP. (IBM, 2014)

Many researches have study on TCP and UDP performance over ethernet LAN by measuring different metrics like throughput, jitter, delay, CPU usage, datagram loss. Next section 2.3.5 will review cryptographic algorithms.

### 2.3.5 Cryptographic Algorithms

This section presents information about cryptographic algorithms used in VPN with IPSec which have found from history of literature.
IPSec provides security depends on the level of security required by the users and application with employed different cryptographic algorithms. The main concern of using an encryption algorithm is to provide better security to a network which prevents unauthorized attacks. However the challenge for the researches is to keep balance between the security and performance in a network simultaneously. There is a wide range of cryptographic algorithms are used in VPN with IPSec such as DES, AES, MD5 and Secure Hash Algorithm 1 (SHA-1) (Xenakis, et al., 2006). Narayan, Kolahi, Brooking, & de Vere (2008) mentioned that VPN uses various encryption and data integrity protocols. While 3DES and BF are the commonly used protocols for Encryption and MD5 and Secure Harsh Algorithm (SHA) are the commonly used protocols for integrity. Following section will present most prominent algorithms used with IPSec.

2.3.5.1 DES

The DES was developed by researchers at IBM in 1972 and most of technological standard agencies adopted DES as a central standard for encryption of commercial data. It is widely used cryptographic algorithms is a symmetric algorithm where the same key is used for both encryption and decryption between IPSec peers. This shared key is a secret key and key size varies from 8 to 64 bits for odd parity. Basically DES begins encryption process by using 64bit key as an input key which contains 56-bits for actual key, the least significant bit, and parity bit. This was design for encryption and decryption (Pasham & Trimberger, 2001; Xenakis, et al., 2006). As DES Feistel cipher, it needs a same amount of processing for both encryption and decryption. Ferguson & Schneier (2000) noted that DES cannot be considered to be very secure as it has very limited key length. Therefore DES was replaced by a stronger alternative, that was the most obvious candidate 3DES.

2.3.5.2 3DES

In 1998, DES led to the introduction of 3DES, which has triple iterations of the basic DES encryption. This encryption method follows the encryption steps below.

1. Data block is DES encrypted using an internal key (56bits key)
2. Encrypted block is decrypted using a different 56bits (second) key
3. The new block is re-encrypted using the internal key (56bits key) again and it is equivalent to using a 168-bit encryption key.

3DES uses different keying methods and either all three keys can be independent of each other or the first and last keys can be the same as the second unique key. The drawback of 3DES algorithm is that it runs three times slower than DES as it has triple repetition of encryption on the same platform (Ferguson & Schneier, 2000; Pasham & Trimberger, 2001; Xenakis, et al., 2006; Agrawal & Sharma, 2010). As a result of this, the 3DES issue of AES has been introduced by the developers.

2.3.5.3 AES

The newest addition to IPSec is AES, introduced in 2002 by two Belgian cryptographers as replacements of DES algorithm, which has a stronger encryption level. AES uses a very resilient algorithm called Rijndael block cipher. It is also a symmetric block that supports different keys, and block sizes vary from 128, 192 or 256 bits. The initial block of this is passed through repeatedly 9, 11 and 13 times to round information transformation function with different block sizes. Each processing round goes through four steps, as below:

1. Substitute Bytes – Uses an S-box to perform a byte by byte substitution of the block
2. Shift rows – A simple permutation
3. Mix column – A substitution method where data in each column from the shift row step is multiplied by the algorithm’s metric
4. Add round key – The key for the processing round is XORed with the data

(Agrawal & Sharma, 2010)

This method is durable and versatile as it serves as a Message Authentication Code (MAC) algorithm, as a hash function and as a pseudo random number generator. Ferguson & Schneier (2000) said that AES standards will become the default encryption method for most systems.

There is rather limited literature about the overheads of IPSec in conjunction with security algorithms. However, Xenakis, et al. (2006) intended to fill the gap in literature about security
algorithms in conjunction with IPSec. The MD5 and Secure Hash Algorithm 1 (SHA-1) are one-way hash functions used with the above mentioned cryptographic algorithms. As these two do not include secret keys they cannot be used as MAC algorithms. Instead they use Hashing for Message Authentication (HMAC), which is a secret key authentication algorithm that provides a framework for incorporating various hashing functions (Xenakis, et al., 2006). The combined mechanisms HMAC-SHA1 and HMAC-MD5 are offered data authentication and integrity services to IPSec. Both IPSec packetization and ciphering increase the final size of the transmitted packets, that creates space overhead. The IPSec packetization overhead depend on the security protocol, ESP or AH. The employed algorithms in IPSec are selected from the analysed ciphers such as DES, 3DES, AES, HMAC-MD5, HMAC-SHA1. Two different IPSec protocols, (ESP and AH) with different cryptographic algorithms (DES, 3DES, AES, MD5 and SHA1) provide different level of security. DES, 3DES and AES provide no security but pure confidentiality. MD5 and SHA1 provide pure authentication. Other combinations such as DES+MD5, 3DES+SHA1 provide combined authentication and confidentiality. The security of IP Authentication Header is dependent on the cryptographic algorithms used, which provides a mechanism for a data integrity and authentication for IP packets using different encryptions. In the authentication data section of the AH has Integrity Check Value (ICV) for a packet. This ICV is computed using the authentication algorithm such as MD5, DES, 3DES or AES. IPsec key management procedures will be used to manage key exchange between the two parties. (Xenakis, et al., 2006; Kessler, 2014). Khanvilkar & Khokhar (2004), Lin, Chang & Chung (2003), McGregor & Lee (2000) have studied MD5, SHA1, DES, 3DES on different platforms (Xenakis, et al., 2006). Metrics researched in this study will be reviewed in the next section.

2.3.6 Researched Metrics

This section explains the performance metrics used in the previous research studies. Most of the networking performance studies have focused on delay, jitter, throughput, and CPU utilization as network performance metrics.
According to the literature, it is evident that the most common metric research in the networking research is throughput. Narayan, Kolahi, Sunarto, Nguyen, & Mani mentioned in 2008 that throughput value gives vital insight as a metric of a network performance that measures the rate at which data gets transmitted from one node to another over a network on a sufficiently long time period. They did their research on performance evaluation on IPv4 and IPv6 on different operating systems and measured the throughput values for TCP and UDP traffic generated and sent between nodes. Since this measurement does end-to-end it is valuable to understand this metrics as it provides network total performance (Zeadally & Raicu, 2003; Munasinghe & Shahrestani, 2005). However this measurement can be vary dependent on network conditions such as hardware used, and processor capabilities. Kolahi et al. (2008) also stated that throughput depends on several conditions over the network like the processor limitations and hardware designs.

Another important network performance metric is latency, which tests for both transmission protocol, TCP and UDP (Narayan, et al., 2008). As per Zeadally & Raicu (2003), latency is the time taken by a packet to travel from one node to another and back. This can vary depending on several factors such as network medium used, and distance between the start and end node.

Jitter is also one of the main metrics used in the network performance research area. After reviewing what others have done in different areas related to this research study, the literature will be analysed in the next section, 2.4.

2.4 Literature Analysis

After reviewing the massive range of articles, reviews and books, it could be said that a vast number of experiments have been done on the network performance stream by different researchers since the ARPANET was developed. This literature review gives the evidence that networking performance is the most valuable feature in network administration, regardless of the architecture, components, platforms, topology, and surrounding features. Different types of
research areas have evolved due to the different network architectures such as LAN, Metropolitan Area Network (MAN), Wide Area Network (WAN), VPN, Intranet, Internet, and Cloud.

With all these massive networking architectural areas, VPN architecture has become the clear winner in the industrial world. The ability of simulating a private network over a public network like Internet is the reason that VPN has become more popular in the business world. Some of the security concerns made the industry have doubts about the VPN area. However, most of the prime IT vendors like Microsoft realised the significance of VPN, and helped the concept to be shine again in the networking industry. This literature review indicated that massive amounts of research have been done on the VPN area. Tales 6 and 7 illustrate the research that has been done on network performance analysis on VPN.

The four main areas that most of the researchers were interested in are network performance, protocols, and security and communication media. As we all know, a common factor of network research is network protocols, which play a major role in network administration. Every hardware and software vendors agree that IP is the most fascinating feature in networking. The literature review clearly shows IP addressing issues that arose from 2008. The evolution of IPv6 has accelerated the research on the IPv4 and IPv6 network performance monitoring stream. The transitional time from IPv4 to IPv6 is not an easy task, which explains the high weighting in networking research areas. The researchers are still continuously working on the area of IPv6 to evaluate the best networking performance. With the rapid growth of Internet usage and the demand for computer networking has led the demand for hardware and software manufacturing. Many researchers are concerned about the protocols and algorithms used on different environments. Table 6 illustrates research done by different researchers on VPN protocols and algorithms.
<table>
<thead>
<tr>
<th>Year</th>
<th>Researcher</th>
<th>Title</th>
<th>Platform(s) - VPN protocol(s)/algorithm(s)</th>
<th>Wired/Wireless</th>
</tr>
</thead>
<tbody>
<tr>
<td>2010</td>
<td>Kosta, Dalal, Jha</td>
<td>Security comparison of wired and wireless network with firewall and VPN</td>
<td>IPSec</td>
<td>Wired, Wireless</td>
</tr>
<tr>
<td>2009</td>
<td>Narayan, Kolahi, Brooking, de Vere</td>
<td>Network performance analysis of VPN protocols: An empirical comparison on different operating systems</td>
<td>Windows Server 2003, Windows Vista, Linux Fedora Core 6, IPSec, PPTP, SSL</td>
<td>Wired</td>
</tr>
<tr>
<td>2009</td>
<td>Wu</td>
<td>Implementation of Virtual Private Network based on IPSec protocol</td>
<td>Windows Server 2003, IPSec</td>
<td>Wired</td>
</tr>
<tr>
<td>2008</td>
<td>Adeyinka</td>
<td>Analysis of problems associated with IPSec VPN technology</td>
<td>Literature Analysis</td>
<td>Wired</td>
</tr>
</tbody>
</table>

Table 6: Related Research 1

(Narayan, Fitzgerald & Ram, 2010)

Table 7 in the next page gives a brief idea of different research areas covered in the VPN domain.
Since a major concern about VPN is security, researchers have given more thought to VPN tunnelling protocols such as PPTP, L2TP, IPSec, SOCKS v5 and SSTP. The above literature findings also give evidence for the amount of research done on the areas of VPN, VPN protocols, algorithms, IPv4 and IPv6 on various operating system platforms. Research has covered different protocols and different algorithms but there are still research gaps that can be identified in the VPN domain. Quite a lot of research has been done on transmission protocols with different VPN protocols based on different networking metrics on a very few operating systems. However, the literature review process has revealed little evidence of research on TCP/UDP based on IPSec VPN protocol that compares cryptographic algorithms on
different operating systems. Most of the researchers have left this cryptographic algorithm comparison for future research work. Therefore, it is worth doing research with a specific VPN protocol on transmission protocols TCP/UDP with various encryptions over different operating systems. The researchers also noted that the VPN experiments that have been done can be extended to include a greater range of operating systems, communication mediums, protocols and metrics. Even though table 5 and 6 show that authors have evaluated different VPN protocols on different operating system platforms, there are still research gaps in VPN with regard to performance evaluations. Hence a gaps in these studies has created the foundation of this research, and they lead to the following main and sub questions that are going to be addressed and answered in this study.

The main research question is:

“Which combination of IPSec algorithm and operating system gives the best network performance when measured for TCP/UDP traffic?”

The sub questions that are going to be answered are:

- Does TCP/UDP network performance vary when implemented on a different client operating system on a Local Area Network?
- Does TCP/UDP network performance vary when implemented on a different server operating system on a Local Area Network?
- Which client operating system gives the best performance for IPSec cryptographic algorithm when measured for TCP/UDP traffic types?
- Which server operating system gives the best performance for IPSec cryptographic algorithm when measured for TCP/UDP traffic types?

After a massive review of literature regarding VPN it has been concluded that the above queries will be answer in this research. The literature concerns so far in this study will be summarised in the next section, 2.5.
2.5 Chapter Summary

This Chapter gave an overview of networking and networking performance to understand the Chapter. Then it reviewed the research that has been done on different research areas related to this study such as VPN performances, IPv4 and IPv6, VPN, VPN tunnelling protocols, TCP/UDP, cryptographic algorithms and the metrics that have been used. Research analysis has been done to identify the research gaps on IPSec, algorithms and operating systems when measured for TCP/UDP traffic performance on VPN. The literature analysis explained the issues in this study area relating to previous experiments. The research done on different operating system platforms were examined to finalise the research questions. The next Chapter presents the methods and techniques used by this research.
Chapter 3: Methodology

This Chapter gives an idea of how the research has been done and explains the data gathering process and data analysis procedure used when searching for systematic method of investigation to establish the facts of this study. One can define research as a systematic and scientific search for pertinent information on a specific topic to find answers to a problem. In fact, research is the art of scientific investigation. According to Gareth Morgan, in Beyond Method, SAGE Publications, 1983, “research is a process through which researchers reveal or discover or create knowledge”. When the initial question arises, there are different ways for searching for the answer in the research field. It is the way of thinking: examine critically the various aspects of the professional work that move towards the methodology (Grix, 2001). This refers to the choice of research strategy taken by a scholar.

The knowledge and facts have been established, in a step by step process and in a logical manner to trigger the research method. Research methodology defines and incorporates the principles, practices, and procedures required to carry out such research and meet the research objectives. The basic types of research are Descriptive vs. Analytical: Descriptive includes surveys and fact-finding enquiries of different kinds as well as analytical research. On the other hand, the researcher has to use the facts or information that is already available, and analyse these to make a critical evaluation of the material. Applied vs. Fundamental: Research can either be applied (or action) research or fundamental (basic or pure) research. Applied research aims at finding a solution to an immediate problem facing a society or an industrial/business organisation, whereas fundamental research is mainly concerned with generalisations and with the formulation of a theory. Quantitative vs. Qualitative: Quantitative research is based on the measurement of the quantity or amount that is applicable to phenomena that can be expressed in terms of quantity. On other hand, qualitative research is concerned with a qualitative phenomenon. Conceptual vs. Empirical: Conceptual research is that related to some abstract idea(s) or theory that is generally used by philosophers and thinkers to develop new concepts or to reinterpret existing ones. Empirical research, on the other hand, relies on experience or observation alone, often without due regard for system and theory. There are also other types
of research available which are variations of one or more of the above stated approaches, based on either the purpose of research, or the time required to accomplish research, on the environment in which research is done, or on the basis of some other similar factor (Burns, 2000; Zikmund, 2003) It is vital to understand which research methods are available to decide which method is suitable for this research. In the next section some research methods will be overviewed and finally, the research method utilised in this research will be justified.

3.1 Research Methods

One of the important aspects of a research is to make a decision on research method to be used. The above description of the types of research brings to light the fact that there are two major competing methods: the scientific empirical tradition and the naturalistic phenomenological mode. According to Burns (2000), in the scientific approaches, quantitative approaches are employed and the naturalist approach emphasises the importance of the subjective experience of individuals, the qualitative approach.

This form of rigorous quantitative analysis approach can be further sub-classified into inferential, experimental and simulation approaches to research. The purpose of the inferential approach to research is to form a database from which to infer the characteristics or relationships of a population. This means a survey research where a sample of a population is studied (questioned or observed) to determine its characteristics, and it is then inferred that the population has the same characteristics. The experimental approach has much greater control over the research environment and some variables are manipulated to observe their effect on other variables. The simulation approach involves the construction of an artificial environment within which relevant information and data can be generated, which permits observation of the dynamic behaviour of a system (or its sub-system) under controlled conditions (Burns, 2000). So, with the given values of the initial conditions, parameters and exogenous variables, a simulation is run to represent the behaviour of the process over time, which is useful in building models for understanding future conditions.
This research project approaches the research theme using the experimental approach of quantitative methodology as the research idea can be quantified, measured, expressed numerically. It is comprised predominantly of experiments in a controlled laboratory environment, and the information about the phenomenon are expressed in numeric way and analysed using statistical methods. All the experiments are implemented on the test bed environment at the Unitec, Institute of Technology lab premises. The next section, 3.2, explains the research methodology used in this research study.

3.2 Research Methodology for this Study

The primary objective of this research is to analyse the IPSec, TCP, UDP, and cryptographic algorithms performances of VPN. The results of this comparative analysis will show which combination of IPSec algorithm and operating system gives the best network performance when measured for TCP/UDP traffic. The study has employed the following main and sub questions which are addressed and answered in this study.

The main question:

“Which combination of IPSec algorithm and operating system gives the best network performance when measured for TCP/UDP traffic?”

The sub questions to be answered are:

- Does TCP/UDP network performance vary when implemented on a different client operating system on a Local Area Network?
- Does TCP/UDP network performance vary when implemented on a different server operating system on a Local Area Network?
- Which client operating system gives the best performance for IPSec cryptographic algorithm when measured for TCP/UDP traffic types?
- Which server operating system gives the best performance for IPSec cryptographic algorithm when measured for TCP/UDP traffic types?

The research was conducted in an experimental, controlled test bed setup in a computer laboratory environment and results were extracted from the data collected during the data
collection project phase of study. This pure quantitative data will be the evidence for the statement mentioned in the above categorization of the study as an experimental approach of quantitative methodology. The following section will help to broaden the knowledge and understanding of the methods of data collection and analysis that have been done in the study.

3.3 Data Collection and Recording Methods

Data collection and recording methods are significant activities in a research project. As the quantitative approach is very systematic, the quantitative data collection method relies on random sampling and structured data collection instruments that fit diverse experiences into predetermined response categories which produce results that are easy to summarize, compare, and generalize. Depending on the research question, data will be collected in order to catch all possible combinations of variables with different controls. This produces situational characteristics for their influence on the dependent, or outcome, variable. Typical quantitative data gathering strategies include: Experiments/clinical trials, observing and recording well-defined events, obtaining relevant data from management information systems, and administering surveys with closed-ended questions.

Information was collected in two different approaches in this study. The first phase was conducted by doing the literature review and then the by collecting the actual experimental data. All the information gathered from resources such as, books, articles and conference proceedings were taken from sources like IEEE, and web information was retrieved from credible web sources that are reviewed to build a better understanding of the research area. Also, the knowledge gained from the literature review revealed whether experiments had been done in this research area. The gaps identified assisted as a guide to conduct this research.

The second information collection approach was the actual experimental data collection process, which is the core resource for the results of the research. As this study was based on the quantitative approach, the data collection of variables that affect the IPv4/Ipv6 network performance of VPN protocols on wired and wireless LANs was done by setting up a test bed in
a computer laboratory and introducing dependent variables to measure the performance of the network. Dependent variable measurement consistency in data gathering was of the greatest importance throughout the experimental set up. The test bed environment was implemented as follows.

![VPN test bed diagram](Diagram)

**Figure 11: VPN test bed**

The experiment was conducted on four different operating systems, which included client and server operating systems on the Microsoft Windows Platform. Microsoft Windows 7 Professional and Microsoft Windows Vista Enterprise editions were given as the platform to the clients and Microsoft Windows Server 2003 R2 Enterprise and Microsoft Windows Server 2008 R2 Enterprise editions were given as the platform to the servers. The study was concluded by doing all the possible combinations of the main dependent variables of the research.

The results of the experiment were gathered by the use of network performance monitoring software called D-ITG. Both ITGSend, which generates the required traffic and ITGRecv, which is located on the receiving end to collect and store the data were run at the same time. ITGDec was used at the receivers’ end to decode the data to a text format. When this process had been done all the data was fed into Microsoft Excel to plot the graphs and draw the final conclusions. The final conclusions were made by analysis of the most highlighted resources of the data analysis phase, which was plotted on Microsoft Excel graphs as below.
The next section presents the networking performance monitoring tool that is used in this research and some more tools that are frequently used in the networking research area.

### 3.4 Network Performance Monitoring Tools

This section provides an overview of network performance monitoring tools and at the end, the tool used in this research will be explained and justified.

**What is a traffic generator used for?**

Network performance monitoring tools are used to analyse the performance of a network by generating network traffic. As per Avallone et al. (2004), traffic a generator assists in analysing the network and evaluating the performance, in terms of different metrics such as throughput, delay, packet loss and jitter analysis, of networks like Wired LAN, WLAN, GPRS, and Bluetooth. Also it can be used for testing device capabilities (devices like PC desktop, Laptop/Notebook, Pocket PC, Advanced Mobile, and Phone), QoS architecture, routing algorithms, and carrying out Scalability and Protocol behaviour analysis (Kolahi et al., 2011; Vasileios, 2013). There is not much literature evidence on comparing network performance tools; however, the following sections provide an overview of the network performance monitoring tools used by researchers. Finally, the tool used in this research will be described thoroughly.
3.4.1 Iperf

Iperf is commonly used to analyse network metrics such as bandwidth, delay, packet loss and window size. It can be used to evaluate both TCP and UDP traffic and is able to run on both the Windows and Linux platform. The latest version of Iperf is designed to work with IP versions 4 and 6. It can be found as a command line tool and also as a Graphical User Interface (called Jpref) (Kolahi et al., 2011; Vasileios, 2013). Network researchers have been using this tool to study IPv6 related network efficiency and its security impact on wireless LAN.

3.4.2 Netperf

Netperf, a benchmark tool, was developed by Hewlett-Packard and can be used to measure network metrics throughput and end-to-end latency on many different types of networks. It can be used for both TCP and UDP evaluations with IP versions 4 and 6. It can also be used on different platforms like Windows, UNIX and Linux. It has two separate files, which work on the server side and client side (Kolahi et al., 2011). This tool has been used by researchers to study TCP performance over Ethernet and Wireless LANs.

3.4.3 IP Traffic

IP Traffic is a tool developed by ZTI-Telecom as a commercial tool. It is commonly used as a data generation, monitoring and testing tool for IP supporting networks. Therefore, it can be used for TCP, UDP and Internet Control Message Protocol (ICMP) protocols and can be used on a Windows platform. IP traffic requires two parts: Traffic-Generator and Traffic Answering, like other performance tools (Kolahi et al., 2011; Vasileios, 2013). This tool is used by researchers to study security the impact on wireless LAN by measuring TCP and UDP throughput.

3.4.4 Distributed Internet Traffic Generator (D-ITG)

D-ITG was implemented in 2003 by Alessio Botta, Alberto Dainotti and Walter de Donato, who has a research interest in analysing network traffic stream. Their analysis was performed by adopting several techniques that come from information theory, signal processing, statistical analysis, pattern recognition and modelling. D-ITG has become popular by showing many different interesting properties when compared to other traffic generators. Avallone et al.
(2004) stated that D-ITG is a packet-level traffic generator. It allows for simultaneously generating multiple flows by managing a single flow in the multi-threaded applications environment. It can produce realistic traffic patterns from different Internet Protocols like TCP, UDP, ICMP, and Voice over Internet Protocol (VoIP) (Avallone et al., 2004; Kolahi et al., 2011). As per the Avallone et al. (2004), DIT-G has already reached the highest performance on all the platforms by supporting all the protocols, and traffic patterns on heterogeneous network scenarios. Avallone et al. (2004) mentioned that D-ITG has improved performance features, namely, generated bit rate, received bit rate, scalability, usability. There are also new features added like supported stochastic processes, introducing a Log server and introducing the daemon mode. It is a multi-platform traffic generating tools that supports Windows, Linux, and Linux Familiar.

According to Botta, Dainotti and de Donato (2007);

- “D-ITG is a platform capable to produce traffic at packet level accurately replicating appropriate stochastic processes for both Inter Departure Time (IDT) and Packet Size (PS) random variables (exponential, uniform, cauchy, normal, pareto,)”.

- “D-ITG supports both IPv4 and IPv6 traffic generation and it is capable to generate traffic at network, transport, and application layer.”
D-ITG Architecture

(Avallone et al., 2004)

Avallone et al. (2004) said that in D-ITG a new protocol has been introduced named Traffic Specification Protocol (TSP). This protocol assists the sender and receiver to decide the experiment parameters and control the traffic generation by using TSP. It also creates a connection between the sender and the receiver, authenticates a receiver, exchanges information on a generation process, closes a sender-receiver connection and detects generation events. D-ITG implements the TPS protocol over a TCP signalling channel between ITGSend and ITGRecv.

ITGSend

ITGSend has three modes named Single flow mode, multiple flows mode and daemon mode. In Single mode ITGSend generates a single flow and the single thread is responsible for the generation of the flow and the management of the signalling channel through the TSP protocol.
In the multiple flows mode it generates a set of flows and it operates as a multithreaded application. One of the threads implements the TSP protocol and drives the generation process, while the others generate the traffic flows.

**ITGRecv**

ITGRecv always acts as a concurrent daemon. It listens for new TSP connections and when a request arrives, a signal flow is received by a separate thread.

**ITGManager**

ITGManager can control all the traffic crossing the network.

**ITGLog**

It collects statistics on the generation process between the ITGSend and ITGRecv, such as flow number, sequence number, source address, destination address, transmission time, receiving time, and packet size. This information can be stored either in a local log file or in a remote log using the log server ITGLog. These log files are processed at a later stage by DITGDec in order to network parameters.

**ITGDec**

ITGDec enables determining the average values of throughput, delay, jitter and packet loss not only on the whole duration of the experiment, but also on windows of the desired duration. (Avallone et al., 2004)

Many researchers use D-ITG for a variety of research and engineering purposes. Specifically, D-ITG if one of the most well known traffic generators within the networking research community. Narayan and Shi (2010) used D-ITG as the primary tool to evaluate the performance of Internet Protocols on VoIP traffic due its capability of working with both IPv4 and IPv6 protocols. It generates traffic at the network, transport and application layer and sends it from sender to generator node and can measure performance related metrics. The
following table illustrates a comparison of the above mentioned network performance monitoring tools.

<table>
<thead>
<tr>
<th></th>
<th>Iperf</th>
<th>Netperf</th>
<th>IP Traffic</th>
<th>D-ITG</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Interface</strong></td>
<td>Command Line</td>
<td>Command Line</td>
<td>Command Line</td>
<td>Command Line</td>
</tr>
<tr>
<td><strong>Multi-platform</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>User guide</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Protocols</strong></td>
<td>TCP &amp; UDP</td>
<td>TCP,UDP,SCTP,DLPI</td>
<td>TCP,UDP,IGMP</td>
<td>TCP, UDP, ICMP, DNS, Telnet, VoIP</td>
</tr>
<tr>
<td><strong>Log file</strong></td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Packet Departure</strong></td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Metrics</strong></td>
<td>Throughput</td>
<td>Throughput</td>
<td>Throughput</td>
<td>Throughput</td>
</tr>
<tr>
<td></td>
<td>Jitter</td>
<td>Packet Loss</td>
<td>Jitter</td>
<td>Jitter</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CPU usage</td>
<td>Packet Loss</td>
<td>Packet Loss</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Response Time</td>
<td>Round-trip-time</td>
<td>Round-trip-time</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>One-way-delay</td>
<td></td>
</tr>
</tbody>
</table>

Table 8: Comparison network performance monitoring tools

After examining all the above network monitoring tools, it was realised that D-ITG is the appropriate tool for this research study. In 2004 Avallone et al. also preferred D-ITG over Iperf as it does not produce a log file.

3.4 Chapter Summary

This Chapter summarised the research method approaches, the methodology and how the data collection had been done to complete the study. This Chapter further explained the reason for using a quantitative approach and test bed experimental study for this research. The research is mainly based on a main research question and four sub questions, which need to be answered by conducting experiments and collecting data and finally analysing the collected data by plotting the graphs on Microsoft Excel. Since the study employed numerical values, the research had been done using a quantitative approach, which leads to a systematic way of thinking. Chapter four presents the experimental study for this research.
Chapter 4: Experimental Network Design

As mentioned in the above Chapter, the experiment was done in a computer laboratory. This Chapter gives the details of the resources used in the experimental setup that was used to study TCP/UDP network performances with various IPsec algorithms on a wired LAN. Data was collected a number of different experimental scenarios by changing the different parameters. Four operating systems employed in the experimental designed and three network metrics were analysed with different payload sizes. Each experimental scenario was done 20 times and repeated three times. Two transmission protocols, TCP and UDP, were tested in IPsec mode with different algorithms.

4.1 Hardware Specifications

This section gives an overview of the hardware specification utilised in the test bed environment. (Table 9: Hardware specification) The entire hardware configuration was kept identical on the two server machines and two client machines in order to main the consistency and accuracy of the results. The computers were connected using CAT5e cross over cables for the wired LAN.

<table>
<thead>
<tr>
<th>Hardware component</th>
<th>Server Specification</th>
<th>Client Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motherboard</td>
<td>Lenovo</td>
<td>Lenovo</td>
</tr>
<tr>
<td>Processor</td>
<td>Intel® Core (TM) i5 CPU 760 @ 2.80GHz</td>
<td>Intel® Core 2 Duo CPU E6300 @ 1.86GHz</td>
</tr>
<tr>
<td>BIOS</td>
<td>Lenovo 2JKT40AUS</td>
<td>Lenovo 2JKT40AUS</td>
</tr>
<tr>
<td>Memory</td>
<td>8GB Samsung DDR3</td>
<td>8GB Samsung DDR3</td>
</tr>
<tr>
<td>Storage</td>
<td>Segate 150GB SATA</td>
<td>Segate 150GB SATA</td>
</tr>
<tr>
<td>2 x Ethernet NIC</td>
<td>Intel® PRO/1000 GT Desktop Adapter</td>
<td>Intel® PRO/1000 GT Desktop Adapter</td>
</tr>
</tbody>
</table>

Table 9: Hardware Specification

The software specification utilised in this research is presented in the next section, 4.2.
4.2 Software Specifications

Two client side and two server side Microsoft Windows operating systems were used in this study. D-ITG networking monitoring software tool was used to check the performance of the network. ITGSend, ITGRecv and ITGDec components were used to send, received and decode the data. Table 10 below outlines the specifications of the software used.

<table>
<thead>
<tr>
<th>Software</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Windows 7</td>
<td>Professional 64-bit SP1</td>
</tr>
<tr>
<td>Microsoft Windows Vista</td>
<td>Enterprise 64-bit SP1</td>
</tr>
<tr>
<td>Microsoft Windows 2003</td>
<td>Enterprise 64-bit</td>
</tr>
<tr>
<td>Microsoft Windows 2008 R2</td>
<td>Enterprise 64-bit</td>
</tr>
<tr>
<td>Distributed Internet Traffic Generator (D-ITG)</td>
<td>2.8.0-rc1</td>
</tr>
</tbody>
</table>

Table 10: Software Summary

The network design for the test bed for this study will be explained in the next section, 4.3.

4.3 Network Configurations

A network test-bed was designed for this study using four computers. Two computers were used as sender and receiver and, in the middle, two server machines were used. Three private networks were utilised on the test- bed. As in Figure 13 shown below, four computers were connected using Cat5e crossover cables. Since some of the hardware supports Gigabit Ethernet, 100Mbps link has been used. The D-ITG send and receive were configured on client machines and used Microsoft Windows Vista and 7. The server machines were configured with Microsoft Windows Server 2003 and 2008.

Figure 13: Network Diagram of the test-bed
4.3 Packet Payload sizes

The actual amount of data in a packet (without the header information of a packet) being sent, which is packet payload size, varied from 64Bytes to 1536Bytes in this research. Variations are as below:

<table>
<thead>
<tr>
<th>Payload Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>64Bytes</td>
</tr>
<tr>
<td>128Bytes</td>
</tr>
<tr>
<td>256Bytes</td>
</tr>
<tr>
<td>512Bytes</td>
</tr>
<tr>
<td>768Bytes</td>
</tr>
<tr>
<td>1024Bytes</td>
</tr>
<tr>
<td>1152Bytes</td>
</tr>
<tr>
<td>1280Bytes</td>
</tr>
<tr>
<td>1344Bytes</td>
</tr>
<tr>
<td>1408Bytes</td>
</tr>
<tr>
<td>1536Bytes</td>
</tr>
</tbody>
</table>

The reasons for this choice for the packet size range are, most packet sizes observed on networks and Internet are within this range and supported to most common applications such as VoIP.

4.5 Chapter Summary

This Chapter presented the experimental test-bed for this research study. Hardware and software configuration were presented in the first section in this Chapter, which then moved to explain the network diagram of this research and then explained the packet payload sizes used. After collecting data from the research experiment, the next step will be the data analysis which is the vital phase in a research study. Chapter five presents the data analysis of this research.
Chapter 5: Data Analysis

This Chapter analyses the results obtained from the research experiment. Gathered data will be analysed and presented in 3 different sections: Windows Server 2003 platform, Windows Server 2008 platform and Server Platform Comparisons. Data analysis is mainly concerned with TPC and UDP results based on the metrics chosen and discussed in 2.3.6: throughput, jitter and delay. To ensure high data accuracy, all tests were executed 20 times and results have been plotted into line graphs and column graphs.

5.1 Windows Server 2003 platform

This section is concerned with the analysis of the transmission protocols TCP and UDP results from the experiments conducted on Windows Server 2003 platform. Throughput, latency and jitter results are analysed in the next section followed by 3DES and DES algorithms. At the end of the section algorithm comparison is plotted on a column graph.

5.1.1 Results for Throughput

Throughput is one of the main metrics analysed in the research on TCP and UDP with different algorithms with different Windows operating systems. This section provides a clear analysis of the experiments conducted for bandwidth with TCP and UDP.

Firstly in figure 14 it presents a line graph for TCP throughput values for 3DES and DES algorithms when implemented with Windows Vista and Windows 7 in a Windows Server 2003 platform.
From the graph it is evident that all the values are below 90Mbps. The throughput increases in values from a lower payload size up to payload size 1344 then there is a decrement after that. It can be seen that DES and 3DES show a lower performance with small and large packet size and higher performance with medium packet size. Packet fragmentation happens when the packet based network protocol use to address smaller maximum transmission unit sizes in the network. As standard maximum transmission unit for an Ethernet network is 1500Bytes, it could see a reflection in performance when packets exceed this size. Sometimes fragmentation can be expected at the smaller packet sizes as it leaves less room for data with IPSec. Throughput values for packets in the range 64 to 1344 with DES are close increment and average approximately 45Mbps, and it is approximately 75Mbps for 3DES. Also, it is worth noting that 3DES with Windows Vista and Windows 7 shows lower values while DES shows much higher values with both Windows Vista and Windows 7. The greatest difference between 3DES and DES is for TCP, approximately 35%. Windows Vista 3DES-MD5 and Windows Vista DES-MD5 clearly show much higher throughput values with smaller packet size. This graph illustrates a higher throughput value for DES-MD5 with Windows Vista and a lower throughput
value for 3DES- SHA1 with Windows Vista. A higher increment can be seen compared to TCP, from lower packet size to medium packet size. There is no distinction that can be made between Windows Vista and Windows 7.

As per the analysis it can be concluded that:

- TCP bandwidth for DES with Windows Vista and Windows 7 on Windows Server 2003 records the highest bandwidth in all the packet sizes.
- For both the client operating systems, 3DES have lower bandwidth values than DES algorithm on the Windows Server 2003 platform.
- TCP bandwidth values for all the algorithms show lower performance with smaller and larger packet sizes and higher performance with the medium size packets.

Figure 15 below presents UDP throughput values for 3DES and DES algorithms with Windows Vista and Windows 7 on Windows Server 2003.

![Figure 15: UDP IPv4 Throughput (IPsec Algorithms) – Microsoft Windows Server 2003](image-url)
The above graph shows that all values are below 90Mbps as TCP (Figure 13). Like TCP, 3DES algorithms show lower throughput values with both Windows Vista and Windows 7 and higher throughput values with DES algorithms. The average throughput value for 3DES is approximately 43Mbps and it is 72Mbps for DES. Throughput values for all the algorithms are increased packets from 64 to 1344 but it decreases after the 1344 packet size. Similar to TCP, UDP throughput also gives lower throughput values with small and large packet size and higher throughput values with medium packet size. However, it can be seen that UDP throughput for DES with both Windows Vista and Windows 7 for small packet size to medium packet size is significantly higher than the TCP throughput. The throughput difference between 3DES and DES for UDP is approximately 23%. The packet size over 1344 Windows 7 values are the lowest, averaging around 62Mbps.

Unlike TCP, UDP gives higher throughput with DES-SHA1 when it is implemented with Windows 7. Similar to TCP throughput, it is difficult to make a distinction between Windows Vista and Windows 7 operating systems.

The following conclusions can be made as per the above results:

- The UDP bandwidth for DES with Windows Vista and Windows 7 on Windows Server 2003 records the highest bandwidth in all the packet sizes. This is the same as the TCP results.
- For both the client operating systems 3DES has lower bandwidth values than DES algorithm on the Windows Server 2003 platform, which is the same as the TCP results.
- The UDP bandwidth values for all the algorithms show lower performance with smaller and larger packet sizes and higher performance with medium size packets, the same as the TCP results.
- The UDP bandwidth for DES with both Windows Vista and Windows 7 for small packet size to medium packet size is significantly higher than the TCP bandwidth.

Jitter and latency are the other metrics of concern in this analysis, and they are discussed in the following sections.
5.1.2 Results for Latency

This section presents the time delay experienced in the test environment on the client operating systems Microsoft Windows Vista and Windows 7 on Microsoft Windows Server 2003 with TCP and UDP. The results are compared using the line graphs shown below.

![Figure 16: TCP IPv4 Latency (IPSec Algorithms) – Microsoft Windows Server 2003](image)

Figure 16 shows the TCP latency for 3DES and DES algorithm on the Windows Vista and Windows 7 with Windows Server 2003 environment. All latency shown in the above graph is within the margins of 0ms and 20ms. However, most values are concentrated between 2ms and 12ms. The latency trend towards the payload size shows nearly similar fluctuations across all the algorithms with all the operating systems. TCP latency values in the graph shows that for all the packet sizes, Windows 7 latency values are significantly lower than the Windows Vista based scenarios. Higher latency performance can be seen with small and large packet sizes. However, medium packet size gives slightly lower latency performances. A clear distinction is seen between latency for packet sizes 256 and 1408 and the rest. It is shown that latency
become more reliable when it reaches the packet fragmentation point. These considerable increases could cause once the packet size reaches the point of fragmentation. Latency values steeply increase to almost 12ms with Windows Vista 3DES-SHA1. Windows Vista performs almost 75% better than Windows 7.

The previous results conclude the following:

- TCP latency gives higher performance with Windows Vista than Windows 7 on Windows Server 2003 for all the packet sizes.
- For all the algorithms, TCP latency gives a lower performance on Windows Server 2003 with medium size packets than the small and large packet sizes.

The analysis of UDP latency on client operating systems Windows Vista and Windows 7 on Windows Server 2003 will be compared in the following section.

![Figure 17: UDP IPv4 Latency (IPSec Algorithms) – Microsoft Windows Server 2003](image-url)
In the Figure 17 latency performance is within a margin 0ms to 35ms. Latency increases considerably with 3DES and DES at the range of 256 and 1152 packet size. More stable UDP latency can be seen when the packet size increases after 1152Bytes. Only a small increase in delay is observed, once packet have fragmented. Unlike TCP latency, UDP latency performs well with Windows 7 than with Windows Vista. UPD latency steeply increases to almost 30ms with Windows 7 at packet size 1024. It is seen that Windows 7 performs 78% better than the Windows Vista. UDP latency works better with medium packet sizes rather than small and large packet sizes.

The above results conclude the following:

- UDP latency gives higher performance with Windows 7 than Windows Vista with small and medium size packets on Windows Server 2003.
- For all the algorithms, UDP latency gives lower performance with small size packets, higher performance with medium size packets size and stable performance with large packet sizes on Windows Server 2003.

The last network characteristic results that are discussed in the analysis is Jitter, which is a variation in the packet delay as described in the next section.
5.1.3 Results for Jitter

Figure 18 below shows TCP jitter for 3DES and DES algorithms with Windows Vista and Windows 7 as clients on Windows Server 2003.

From the graph it is evident that the general trend of jitter exhibits mostly upward gradient. All the jitter performances are within a range of 0.1ms to 0.7ms. The top performer within the group is Microsoft Windows Vista with 3DES-SHA1. It outperforms at 1536 payload size. Also, it is worth noting that Windows 7 with 3DES-MD5 shows a higher jitter value at the packet size 256Bytes. Windows Vista performs approximately 66% better than Windows 7 on Windows Server 2003.

As per the results above it is found that:

- When the packet sizes increase, the TCP jitter values increase well on the Windows Server 2003 for DES and 3DES algorithms.
• For TCP jitter, on Windows Server 2003, Windows Vista performs approximately 66% better than Windows 7.

• For TCP jitter, 3DES-SHA1 with Windows Vista records the highest value on Windows Server 2003.

In the following section you will be able to understand the Jitter analysis with the connectionless protocol UDP on the server 2003.

Figure 19 above shows UDP jitter for 3DES and DES with Windows Vista and Windows 7 as clients on Windows Server 2003. The graph shows that in all the scenarios, jitter values are almost similar and range from 0.1ms to 1.0ms. It is mostly consistent for all packet sizes up to 1024Bytes. When lower jitter performance shows at lower packet sizes, higher packet sizes manage to show higher jitter performance. This could have caused with the fragmentation. Larger packet size (1152Bytes and above) jitter values average approximately 0.7ms and the rest are round 0.25ms. Similar to TCP, UDP also manages to show incremental jitter.
performance with increase of packet size. This can be compared with the delays values of Windows Vista with 3DES-SHA1, which has values well above the other scenarios. This behaviour is similar to TCP delays.

Based on the results it is found that:

- When the packet sizes increase, the UDP jitter values increases as well on Windows Server 2003 for DES and 3DES algorithms, which is the same as TCP.
- UDP delay values are lower and consistent with the smaller packet size on Windows Server 2003, and medium and larger packet sizes give higher jitter values.
- For UCP jitter, 3DES-SHA1 with Windows Vista records the highest value on the Windows Server 2003, which is the same as TCP.

After understanding the behaviour of throughput, latency and jitter metrics in the research, it is worth looking at how those metrics behaved with both connection-oriented and connectionless protocols with data encryption standards applied in the research environment. Accordingly, the following sections summarises how the researched VPN environment configured with encryption modules 3DES and DES.

5.1.4 Results on 3DES

This section is comprised of the TCP and UDP results with 3DES-MD5 with all three metrics: throughput, latency and jitter.

5.1.4.1 Throughput

In the section below Figure 20 shows TCP and UDP throughput values for 3DES-MD5 on Windows Vista and Windows 7 with Windows Server 2003.
It can be seen that for small and medium packet size 3DES-MD5, throughput for TCP and UDP on all the operating systems gradually increases as the packet size increases. However, for the large packet size (1344Bytes and above) TCP throughput is slightly lower (20%) than the UDP on both operating systems. Also from packet size 1408, TCP gives the same throughput for both operating systems. UDP throughput with Windows Vista is noticeable compared to other scenarios.

The above results lead to the following conclusions;

- TCP with 3DES-MD5 show slightly lower bandwidth than UDP with both Windows Vista and 7 on the Windows Server 2003.
How TCP and UDP with 3DES-MD5 affect packet delay on the Microsoft Windows server 2003 environment will be analysed in the following section.

5.1.4.2 Latency

Figure 21 shows TCP and UDP latency with 3DES on Windows Server 2003.

![Figure 21: TCP and UDP IPv4 Latency (3DES-MD5 IPSec Algorithm) – Microsoft Windows Server 2003](image)

UDP Latency started at a higher rate at small packet size than the TCP latency and ended with a lower rate at large packet size. For all the packets, TCP latency values are shown to be between 3ms to 11ms. UDP latency value with Windows 7 is significantly higher at packet size 768Bytes.
then the rest of the scenarios, and performs 78% better than the other scenarios. It could cause before fragmentation.

As per the results, the following conclusions can be made:

- UDP with 3DES-MD5 shows long delays with smaller packets and with the large packets it experiences small delay times with both Windows Vista and Windows 7 on Windows Server 2003.
- TCP with 3DES-MD5 shows low latency with smaller packets and high latency with large packets with both Windows Vista and Windows 7 on Windows Server 2003.

The delay variations experienced in the research with TCP and UDP configured on 3DES-MD5 is analysed below.

### 5.1.4.3 Jitter

Figure 22 shows that TCP and UDP jitter values for 3DES-MD5 on Windows Vista and Windows 7 with Windows Server 2003.

![Figure 22: TCP and UDP IPv4 Jitter (3DES-MD5 IPSec Algorithm) – Microsoft Windows Server 2003](image-url)
It is observed that as packet size increases, the TCP and UDP jitter values increase as well for all the operating systems. It shows an approximately linear trend with the gradient value 0.0778. While UDP with Windows Vista gives higher jitter values (34%) at the larger packet size of 1344 Bytes, TCP with Windows 7 gives high jitter (54%) values at the packet size of 256 Bytes.

The following conclusions are made from the above results:

- Both UDP and TCP jitter values with 3DES-MD5 show an approximately linear trend on both Windows Vista and 7 with Windows Server 2003.
- UDP jitter values are higher than TCP with large packet sizes with both Windows Vista and 7 on Windows Server 2003.

After analysing TCP and UDP with improved data encryption standards it is necessary to discover how TCP and UDP perform with normal DES. This scenario is analysed in the following section consecutively with throughput, latency and jitter.

### 5.1.5 Results on DES

Firstly, throughput results which configured with DES-MD5 are analysed in the following section.

#### 5.1.5.1 Throughput

Figure 23 shows TCP and UDP throughput values for DES-MD5 on Windows Vista and Windows 7 with Windows Server 2003.
TCP throughput range from 32.39Mbps to 87.12Mbps, and UDP values are from 13.34Mbps to 88.75Mbps. For smaller packet sizes (up to 512Bytes) there is a significant incline in throughput as the packet size increases. UDP shows almost the same behaviour for both the operating systems from packet size 512 to 1536. All the results are similar when the packet size is at the optimum point before the fragmentation. This shows approximately the logarithmic trend.

The following conclusion can be made from the above results:

- UDP bandwidth is higher than TCP bandwidth with DES-MD5 for both Windows Vista and 7 on Windows Server 2003 with medium and large packet sizes.

Regardless of the processor it takes a finite amount of time to transfer the data. How the delay can vary with TCP and UDP with DES-MD5 will be analysed in the following section.
5.1.5.2 Latency

Figure 24 shows TCP and UDP latency values for DES-MD5 on Windows Vista and Windows 7 with Windows Server 2003.

![Microsoft Windows 2003 Latency-TCP/UDP - DES-MD5](image)

**Figure 24: TCP and UDP IPv4 Latency (DES-MD5 IPsec Algorithm) – Microsoft Windows 2003**

UDP shows much higher latency with the smaller packet size than the TCP, but it shows the opposite behaviour at the larger packet size (1344Bytes and above). TCP exhibits a clear difference with Windows Vista at the packet size of 768Bytes. It is a 79% better performance than the other scenarios. It is difficult to determined Vista performance with DES at the packet size 768Bytes.

As per the results above, the following conclusion can be made:

- UDP with DES-MD5 suffers from longer delays than TCP with the smaller packet size with both Windows Vista and 7 on Windows Server 2003. This is the same as 3DES-MD5.
Considering latency would not give a complete analysis of network performance. Therefore, it is important to analyse network jitter to conduct a complete analysis of expected network behaviour. How jitter can vary with TCP and UDP with DES-MD5 will be analysed in the following section.

5.1.5.3 Jitter

Figure 25 shows TCP and UDP jitter values for DES-MD5 on Windows Vista and Windows 7 with Windows Server 2003.

![Figure 25: TCP and UDP IPv4 Jitter (DES-MD5 IPSec Algorithm) – Microsoft Windows Server 2003](image)

It is shown in this figure that as packet size increases, throughput values increase as well. Therefore, it shows an approximately linear trend with the gradient value 0.0463. However, the maximum values for TCP and UDP on different operating systems vary. UDP with Windows Vista starts as a lower value and exhibits the highest value at the largest packet size.
The conclusions that can be made from the above results are as follows:

- TCP and UDP jitter values increase gradually when the packet size increases, with DES-MD5 on both Windows Vista and Windows 7 with Windows Server 2003.

The above two sections (5.1.4 and 5) gave an overview of analysed information on throughput, latency and jitter on a VPN environment configured with encryption modules 3DES and DES. How Microsoft Windows Vista and 7 perform with different encryption algorithms on Microsoft Windows 2003 environment configured with TCP will be argued in the following section.

5.1.6 Results for algorithms

Figure 26 below presents and compares the TCP throughput performance for 2 algorithms with Windows Vista and Windows in a Windows Server 2003 platform and for packet size 1536Bytes.

In both the graphs it is seen that 3DES has the lowest throughput for TCP and the difference between the two operating systems is insignificant. DES-MD5 has the highest TCP throughput with Windows Vista and the lowest is 3DES-SHA1 with Windows Vista. 3DES algorithm gives a lower TCP throughput than the DES algorithm. It is also seen that Windows Vista operating
system gives slightly higher throughput values than Windows 7 except for 3DES-SHA1. The greatest difference between Windows Vista and Windows 7 for DES-MD5 is approximately 7%, which is not significant.

The above results conclude the following:

- Windows Vista has higher TCP bandwidth than Windows 7 with DES and 3DES algorithms on Windows Server 2003.
- DES shows higher TCP bandwidth with Windows Vista than Windows 7 on Windows Server 2003

To conclude the comparisons between Windows 2003 and Windows 2008 server environment research results, it is important to analyse the same scenarios with the Microsoft Windows 2008 environment. This task will be considered in the next section.
5.2 Windows Server 2008 platform

This section presents research the metrics for TCP and UDP on the Microsoft Windows Server 2008 environment with Windows Vista and 7. Firstly it shows throughput results in the section below.

5.2.1 Results for Throughput

Figure 27 shows the TCP throughput for 3DES, DES and AES on the Windows Vista and Windows 7 with the Windows Server 2008 operating system.

![Figure 27: TCP IPv4 Throughput (IPSec Algorithms) – Microsoft Windows Server 2008](image-url)
Most of the throughput values are concentrated between 42Mbps to 58Mbps and 67Mbps to 87Mbps. As could be seen in Figure 25, as the packet size increases from 64 to 1344Bytes, throughput escalates. This value dramatically decreases from 1344 to 1408Bytes. There is also a slight increase in throughput values for packet sizes bigger than 1408Bytes. From there the throughput decreases and again it escalates at 1536Bytes. Similar to Windows 2003, it can be seen that 3DES gives a lower performance than DES and AES. The highest throughput value can be seen with AES128-SHA1 on both Windows Vista and 7. Also, the highest TCP throughput in Windows Server 2008 is noted at the packet size of 1344Bytes where 7DES-SHA1 provides 87.7Mbps. The TCP throughput on Windows Server 2008 with Windows 7 performs 31% better than Windows Vista. TCP throughput performs higher with small and large packet sizes compared to 3DES and DES. 3DES-SHA1 is prominent with medium packet sizes.

As per the results the following conclusions can be made:

- TCP bandwidth for AES with Windows Vista and Windows 7 on Windows Server 2008 records the highest bandwidth in all the packet sizes.
- For both the client operating systems, 3DES has lower bandwidth values than DES and AES algorithm on Windows Server 2008 platform. This is similar to the Windows Server 2003 results.
- TCP bandwidth values for all the algorithms show lower performance with smaller and larger packet sizes and higher performance with the medium size packets. This is also similar to the Windows Server 2003 results.
- The TCP throughput on Windows Server 2008 with Windows 7 performs 31% better than with Windows Vista.

The UDP throughput values of the two client operating systems Windows Vista and Windows 7 in Windows Server 2008 platform with three algorithms are shown in the next figure, 28.
It can be seen that UDP throughput increases from packet size 64 to 1344Bytes. For all these packets, DES and AES have the higher UDP throughput. A clear distinction can be made between Windows Vista 3DES and Windows 7 3DES. A difference can be seen when comparing 3DES algorithm to UDP throughput performance on Windows Server 2003. UDP throughput with 3DES in Windows Vista performs 26% better with Windows 7. The lowest UDP point of difference can be seen at the packet size of 64Bytes for AES-128-SHA1 with Windows 7 and 1408Bytes for 3DES- MD5 with Windows 7. Also, AES performs with both the client operating
systems 36% better than 3DES. With the fragmentation point performance of all the protocols decrease significantly.

The conclusions made from the above results are as follows:

- UDP bandwidth for AES and DES with Windows Vista and Windows 7 on Windows Server 2008 records the highest bandwidth in all the packet sizes.
- UDP bandwidth values for all the algorithms show lower performance with smaller and larger packet sizes and higher performance with medium size packets. This is also similar to the Windows Server 2003 results.
- For both the client operating systems, 3DES has lower bandwidth values than DES and AES algorithm on the Windows Server 2008 platform. This is similar to the Windows Server 2003 results.
- The UDP throughput for 3DES on Windows Server 2008 with Windows Vista performs better than with Windows 7.

Time delay observed as data transmitted from one point to another within the Windows Server 2008 platform will be illustrated in the section below.
5.2.2 Results for Latency

In this section, Figure 29 shows TCP latency for 3DES, DES and AES algorithms on the Windows Vista and Windows 7 as clients with Windows Server 2008.

![Figure 29: TCP IPv4 Latency (IPSec Algorithms) – Microsoft Windows Server 2008](image)

Most latency values are concentrated between 2.5 and 5.5ms, but a clear distinction can be seen between the smaller packets and the rest. For the smaller packet size 128Bytes, the latency values steeply increase to almost 17ms. In Figure 29 it is noted that Windows Vista performs 85% better than Windows 7 with TCP latency. Windows Vista on DES-SHA1 is
prominent up to the packet size 1024Bytes, which indicates that Windows Vista 3DES-SHA1 shows the higher performance. The lowest TCP latency can be seen with Windows 7 on AES256-SHA1. All protocols perform gradually decreased latency from fragmentation point.

The above results lead to the following conclusions:

- TCP latency gives higher performance with Windows Vista than Windows 7 on Windows Server 2008 for all the packet sizes, which is the same as for Windows Server 2003.
- For all the algorithms, TCP latency gives a higher performance with smaller size packets than the rest on Windows Server 2003.

The amount of time it takes a packet to travel from source to destination with UDP on the Windows Server 2008 environment will be presented in the following section.

![Microsoft Windows Server 2008 Latency - UDP](image-url)

*Figure 30: UDP IPv4 Latency (IPSec Algorithms) – Microsoft Windows Server 2008*
Figure 30 shows that UDP latency for 3DES, DES and AES algorithms on Windows Vista and Windows 7 as clients with Windows Server 2008. UDP latency shows a clear difference when compared with TCP latency. It can be seen that mostly the latency trends are downwards except for a few combinations. It starts at approximately 9.5ms and decreased to approximately 2.5ms. Lower UDP latency can be seen mostly with large packet sizes (Packet size 1152Bytes and above). 3DES-SHA1 with Windows 7 outperforms at the packet size of 1536Bytes. It is difficult to make a clear distinction between the operating systems or algorithms on UDP latency with Windows Server 2008. It caused more delays, once the packet fragmentation point has been reached.

The above results conclude the following:

- UDP latency trends are more downwards with Windows Vista than with Windows 7 on Windows Server 2008.
- For all the algorithms, UDP latency gives a lower performance at large packet sizes on Windows Server 2008.

How the latency varies on the Windows Server 2008 environment will be analysed in section 5.2.3 below.

### 5.2.3 Results for Jitter

In this section, Figure 31 presents the TCP jitter values for 3DES, DES and AES algorithms with Windows Vista and Windows 7 as clients on the Windows Server 2008 environment.
No clear distinction between the operating systems or algorithms can be made for jitter on Windows Server 2008. The jitter value 0.9ms is recorded for larger packet sizes (higher than 1408Bytes). From the graph it is evident that the general trend of the jitter exhibits mostly an upward gradient, which is similar to TCP jitter on Windows Server 2003 (Figure 18).

As per the results above it is found that:

- When the packet sizes increase, the TCP jitter values increases as well on Windows Server 2008 for DES, 3DES and AES algorithms. This is the same as for Windows 2003.

In the section below, Figure 32 shows UDP jitter values for 3DES, DES and AES algorithms on Windows Vista and Windows 7 as clients with Windows Server 2008.
Figure 32: UDP Jitter IPv4 (IPSec Algorithms) – Microsoft Windows Server 2008

Most of the scenarios exhibit slow and consistent delays with small packet size up to 1024Bytes and this is almost similar for the range from 0.1 to 0.4ms. The smaller packet size jitter values average approximately 0.25ms. After packet size 1152 there is an increment up to packet size 1536Bytes. The larger packet size jitter values average approximately 0.7ms. It is worth noting that there is a clear distinction between 3DES-SHA1 with Windows 7 and the other scenarios. It is also seen that 3DES-MD5 performs 48% better than DES-SHA1 on Windows Vista with Windows Server 2008.

Based on the results above, it is found that:
• When the packet sizes increase, the UDP jitter values increase as well on Windows Server 2008 for DES, 3DES and AES algorithms, which is the same as UDP jitter on Windows Server 2003.

• UDP delay values are lower and consistent with the smaller packet size on Windows Server 2003, and medium and larger packet sizes give higher jitter values as UDP jitter on Windows Server 2003.

• For UCP jitter, 3DES-SHA1 with Windows 7 records the highest value on Windows Server 2008.

The following section exhibits how the researched VPN environment behaved with encryption modules 3DES, DES and AES on the Windows Server 2008 environment.

5.2.4 Results on 3DES

This section presents the impact of 3DES behaviour on network throughput, latency and jitter.

5.2.4.1 Throughput

Figure 33 below shows TCP and UDP throughput with 3DES-SHA1 on Windows Server 2008 environment.

![Figure 33: TCP and UDP IPv4 Throughput (3DES-SHA1 IPSec Algorithm) – Microsoft Windows 2008](image-url)
TCP values start at higher throughput values with small packet size than the UDP throughput, but TCP gives lower throughput with larger packet size than UDP throughput. TCP values drop by approximately 22% from UDP with both Windows Vista and Windows 7. It is seen that UDP has a better performance with Windows Vista than with the other scenarios.

- It is concluded that bandwidth is higher for UDP with 3DES-SHA1 on Windows Vista with Windows Server 2008.

In the next section 5.2.4.2 shows how network time delay varies on the Windows Server environment with 3DES-MD5 algorithm.

5.2.4.2 Latency

Figure 34 shows TCP and UDP latency for 3DES-MD5 on Windows Vista and Windows 7 with Windows Server 2008.

![Figure 34: TCP and UDP IPv4 Latency (3DES-MD5 IPSec Algorithm) – Microsoft Windows Server 2008](image-url)
All latency shown is within a margin 0ms and 85ms. When compared to UDP, TCP illustrates significantly different performance in both the operating systems. UDP exhibits a relatively stable latency performance level with every payload size in both the operating systems. After payload size 1152, UDP performances are lower than the previous payload size. UDP latency values show an interesting trend: UDP gives 92% better performance with Windows 7 than the other scenarios.

The above results conclude that:

- UDP latency with 3DES-MD5 gives a stable performance with every payload size on Windows Server 2008.

The last metrics, the jitter results, are shown in the section below to analyse the TCP and UDP behaviours with 3DES-SHA algorithm.

### 5.2.4.3 Jitter

Figure 35 shows the TCP and UCP jitter values for 3DES-SHA1 algorithm on Windows Vista and Windows with Windows Server 2008.

**Figure 35: TCP and UDP IPv4 Jitter (3DES-SHA1 IPSec Algorithm) – Microsoft Windows Server 2008**
The jitter values changes range from 0.01ms to 1.30ms. It is shown that as packet size increases, the jitter values increase as well. It shows an approximately linear trend with the gradient value 0.0907, which is higher than with Windows Server 2003. As in the previous graph, UDP exhibits better performance than TCP. UDP on Windows 7 with 3DES-SHA1 has most jitter values higher than Windows Vista.

The following conclusions can be made from the above results:

- Both UDP and TCP jitter values with 3DES-MD5 show an approximately linear trend on both Windows Vista and 7 with Windows Server 2008, which is the same as for Windows 2003.
- UDP jitter values are higher than the TCP with large packet sizes with both Windows Vista and 7 on Windows Server 2008. Same as Windows Server 2003.

How the TCP and UDP results perform with DES algorithm will be present in the following section.

5.2.5 Results on DES

Firstly, how throughout values change with DES-MD5 will be presented in the section below.

5.2.5.1 Throughput

Figure 36 shows TCP and UDP throughput values for DES-MD5 on Windows Vista and Windows 7 with Windows Server 2008.
Figure 36: TCP and UDP IPv4 Throughput (DES-MD5 IPSec Algorithm) – Microsoft Windows Server 2008

TCP throughput values give a better performance than the UDP at smaller packets, averaging around 40MBps and vice versa at the larger packets, which average 75Mbps. It can be seen that TCP and UDP do not exhibit much difference with medium size data packets. All operating systems performed sudden increase after the optimum point due to fragmentation.

- As per the results above, it is concluded that TCP bandwidth value is higher than the UDP one at smaller payloads.

The following section shows how the network time delay varies with DES-SHA1 on the Windows Server environment.

5.2.5.2 Latency

Figure 37 shows TCP and UDP latency for DES-SHA1 on Windows Vista and Windows 7 with Windows Server 2008.
The start is very interesting, compared to the other metrics. DES-SHA starts at a higher latency with smaller packet sizes and gradually decreases with larger packet sizes. The starting value is average at around 17ms and it shows a latency average of around 2.5ms with larger packets. It shows an approximately negative linear trend with the gradient value -3.487. TCP with DES-SHA on Windows 7 shows a lower performance than the other scenarios.

The following conclusions can be made from the above results:

- TCP latency is consistent with DES-SHA1 on Windows 7 with Windows Server 2008.
- UDP latency is higher than TCP with smaller packets on Windows Server 2008.

The next section exhibits the variation in the time between the packets arriving on the research network environment.
5.2.5.3 Jitter

Figure 38 shows TCP and UDP jitter values for DES-MD5 on Windows Vista and Windows 7 with Windows Server 2008.

![Figure 38: TCP and UDP IPv4 Jitter (DES-SHA1 IPSec Algorithm) – Microsoft Windows Server 2008](image)

DES-MD5 gives upward gradient jitter values with all the operating systems and it ranges from 0.10ms to 0.80ms. A clear distinction can be seen in the four different scenarios from the data packet 1024Bytes. UDP is prominent with DES-MD5 on Windows Vista from packet size 1024Bytes.

As per the results above it can be concluded:

- TCP jitter shows a logarithmic trend with DES-MD5 on Windows Vista and 7 with Windows Server 2008.
- UDP with DES-MD5 gives higher jitter values on Windows Vista at large payloads on Windows Server 2008.
AES algorithm results for TCP and UDP for the Windows Server 2008 environment will be discussed in the next section.

5.2.6 Results on AES

Firstly, in this section network throughput results will be analysed.

5.2.6.1 Throughput

Figure 39 shows TCP and UDP throughput values for AES256-SHA1 on Windows Vista and Windows 7 with Windows Server 2008.

![Graph showing throughput values for AES256-SHA1 on Windows Vista and Windows 7 with Windows Server 2008.](image)

Figure 39: TCP and UDP IPv4 Throughput (AES256-SHA1 IPSec Algorithm) – Microsoft Windows Server 2008

All throughput values increase as packet size increases. (Up to 1344Bytes) This trend is similar to all UDP throughput values for all the algorithms. It shows a logarithmic with 0.878 R² value. The performance drops for all operating systems, when the fragmentation point is reached.
As per the results above it can conclude that:

- With AES256-SHA1, UDP bandwidth is lower than TCP at smaller payloads with both Windows Vista and 7 on Windows Server 2008.

The section 5.2.6.2 below shows network latency variations on the Windows Server 2008 environment configured with AES256-SHA1.

5.2.6.2 Latency

Figure 40 shows TCP and UDP latency values for AES256-SHA1 on Windows Vista and Windows 7 with Windows Server 2008.

![Figure 40: TCP and UDP IPv4 Latency (AES256-SHA1 IPSec Algorithm) – Microsoft Windows Server 2008](image)

Almost all the TCP and UDP values for all the scenarios flatten off, indicating values closer to zero except UDP with Windows Vista with smaller packet size (up to 512Bytes).

From the above results it can be concluded that:
• TCP and UDP values for AES256-SHA1 with all the payloads flatten off, indicating values closer to zero on Windows Server 2008.

Section 5.2.6.3 presents network jitter variations on the Windows Server 2008 environment configured with AES256-SHA1.

5.2.6.3 Jitter

Figure 41 shows TCP and UDP jitter values for AES256-SHA1 on Windows Vista and Windows 7 with Windows Server 2008.

![Figure 41: TCP and UDP IPv4 Jitter (AES256-SHA1 IPSec Algorithm) – Microsoft Windows Server 2008](image)

It can be seen that as packet size increases, the jitter values increase as well for all the scenarios. There is an approximately positive linear trend with the gradient value 0.0676.
Therefore, the conclusion is:

- TCP and UDP jitter exhibits a positive linear trend with AES256-SHA1 on Windows Server 2008.

The DES, 3DES and AES algorithm results variation on Windows Server 2008 will be presented in the following section.

5.2.7 Results for algorithms

Figure 42 presents and compares the TCP throughput performance for 3 algorithms with Windows Vista and Windows in the Windows Server 2008 platform and for packet size 1536Bytes.

![Algorithm performance on MS Windows 2008 - TCP](image)

**Figure 42: Algorithm performance – TCP – IPsec- IPv4 – Microsoft Windows Server 2008**

The 3DES, DES and AES performances are almost identical across both client operating systems in the Microsoft Windows Server 2008 server environment. However DES shows significantly greater performance than 3DES and AES. In both, Windows Vista values are slightly higher than its counterpart for all the algorithms for AES128-SHA128 algorithm.

- It can be concluded that AES algorithm performs well with both Windows Vista and 7 on Windows Server 2008.

The next section presents a comparison between the scenarios on the Windows Server 2003 and 2008 Server platforms.
5.3 Windows Server 2003 VS 2008 platforms

This section shows how network throughput, latency and jitter vary on both server operating systems.

5.3.1 Throughput

This section presents the TCP throughput results with 3DES-SHA1 on Microsoft Windows Server 2003 and 2008.

![Throughput Graph](image)

**Figure 43: TCP IPv4 Throughput (3DES-SHA1 IPsec Algorithm) – Microsoft Windows Server 2003 vs 2008**

3DES performance is considerably increased in throughput when the payload size increases from 64Mbps to 1344Mbps. However, it drops from 1344Mbps onwards. 3DES-SHA1 shows a nearly close performance across Microsoft Windows Vista with both Microsoft Windows server operating systems. 3DES-SHA1 performance across Microsoft Windows 7 is significantly different as it shows higher performance in Microsoft Windows Server 2003 but lower performance in Microsoft Windows Server 2008. The top operating system combination from
this group is Microsoft Windows 7 with Microsoft Windows Server 2003, and the lowest is Microsoft Windows 7 with Microsoft Windows Server 2008. Based on the results above the following conclusions can be made:

- For TCP traffic for 3DES, Windows 7 with Windows Server 2003 records a higher bandwidth than Windows 7 with Windows Server 2008.

The next section will compared the latency results implemented on the two server environments.

5.3.2 Latency

This section shows how the TCP time delay varies with 3DES-SHA1 on Microsoft Windows Server 2003 and 2008. Figure 44 presents TCP latency with 3DES-SHA1 on Windows Vista and Windows 7 as clients on Windows Server 2003 and 2008 environments.

3DES performs higher with all the operating systems at smaller packet sizes than the rest. It can be clearly seen that there is significant latency performance with Windows Vista 3DES-SHA1 on
Windows Server 2008, which steeply increases to 18ms. Also, 3DES-SHA performs approximately 63% better with Windows Vista than with Windows 7 on Windows Server 2003. It can also be concluded that Windows Vista with Windows Server 2003 performs approximately 76% higher than Windows 7 with Windows Server 2008.

The conclusions made from the above results are as follows:

- For TCP latency for 3DES, Windows 7 with Windows Server 2003 records a higher bandwidth than Windows 7 with Windows Server 2008. It is approximately 76% higher.
- TCP latency with 3DES-SHA gives approximately 63% better values with Windows Vista than Windows 7 on Windows Server 2003.

In the next section the variation of network jitter results on Microsoft Windows Server 2003 and 2008 are shown.

### 5.3.3 Jitter

Figure 45 below presents 3DES-SHA1 TCP jitter values on Windows Vista and Windows 7 as clients with Windows Server 2003 and 2008 environments.

![Figure 45: TCP IPv4 Jitter (3DES-SHA1 IPSec Algorithm) – Microsoft Windows Server 2003 vs 2008](image)

As per the above graph, Jitter performance is clearly different with different combinations of operating systems. Jitter for all the operating systems increases gradually with the payload size.
3DES performance in Windows Vista is higher in Microsoft Windows Server 2003 than Microsoft Windows Server 2008. However, 3DES performs lower in Microsoft Windows 7 with Microsoft Windows Server 2003 in some payload sizes and higher with Microsoft Windows Server 2008. 3DES performs significantly higher in Microsoft Windows Vista with Microsoft Windows Server 2003 but gives lower performance in Microsoft Windows 7 with Microsoft Windows Server 2008. 3DES-SHA1 on Windows Vista provides 52% better performance than Windows 7 on the Windows Server 2003 environment. Also 3DES-SHA1 on Windows Vista provides 26% better performance than Windows 7 on the Windows Server 2003 environment.

The above results lead to the following conclusions:

- TCP jitter values for 3DES-SHA1 increase gradually with all the packet sizes.
- For TCP jitter for 3DES, Windows 7 with Windows Server 2003 records a higher value than Windows 7 with Windows Server 2008. It is approximately 61% higher.

Section 5.3.4 presents TCP throughput results from experiments conducted on Microsoft Windows Server 2003 and 2008 operating systems implemented on a network employing Windows Vista with different encryption algorithms on a data packet of 1536Bytes.

**5.3.4 Results for algorithms**

The following figure presents and compares the TCP throughput performance for 2 algorithms with Windows Server 2003 and 2008 platforms and for a packet size of 1536Bytes.

![Algorithm performance on MS Windows 2003 - 2008 - Windows Vista -TCP](image)

*Figure 46: Algorithm performance – TCP –IPSec - Microsoft Windows Server 2003 vs 2008*

The conclusions made from the above results are:


The section below presents TCP throughput results from experiments conducted on Microsoft Windows Server 2003 and 2008 operating systems implemented on a network employing Windows 7 with different encryption algorithms on a data packet of 1536Bytes.

![Algorithm performance on MS Windows 2003 - 2008 - Windows 7- TCP](image)

**Figure 47: Algorithm performance Microsoft Windows Server 2003 vs 2008**

The above graph illustrates that 3DES performs slightly better and DES-SHA1 performs nearly identically across Microsoft Windows 7 in Microsoft Windows Server 2003 and Microsoft Windows Server 2008 platforms. However DES-MD5 performs slightly higher with Microsoft Windows 2008. The conclusions made from the above results are:
- For TCP traffic, 3DES and DES-SHA1 with Windows 7 records a higher bandwidth with Windows Server 2003, and DES-MD5 with Windows 7 records a higher bandwidth with Windows Server 2008.

### 5.4 Chapter Summary

This Chapter has analysed the experimental results and provided an overview of the research study by presenting line and column graphs. It provides a cross comparison between the performances of TCP and UDP with different data encryption algorithms on IPv4 with Windows Vista and Windows 7 client operating systems and Windows Server 2003 and 2008 server operating systems. Performances are mainly concerned with three metrics: throughput, jitter, and delay. Conclusions are made from the analysed results. The discussions and major findings of this Chapter will be covered in Chapter six.
Chapter 6: Discussion and Findings

Experimental results were analysed in the Chapter five and summary results will be discussed in detail in this Chapter. The aim of this study was to analyse the performance of TCP and UDP with IPv4 on various operating systems with various IPsec algorithms. Four operating systems were used in the experiment: Microsoft Windows Vista and 7 as client operating systems and Microsoft Windows Server 2003 and 2008 as server operating systems. The data collection phase was done using a traffic generator tool (D-ITG) and three metrics were extracted: throughput, jitter and delay. Collected data was analysed in the Chapter five and findings are highlighted and discussed in the following sections.

6.1 Performance of TCP/UDP for IPsec algorithms on VPN using Microsoft Windows Server 2003

This section discusses the corresponding results extracted from data collected in the Microsoft Windows Server 2003 platform. Analysis of the throughput with Microsoft Windows Vista and 7 on the Windows Server 2003 platform running on the IPv4 using transmission protocols TCP and UDP shows the following behaviours.

Both TCP and UDP bandwidth for DES with Microsoft Windows Vista and Microsoft Windows 7 on Microsoft Windows Server 2003 record the highest bandwidth in all the packet sizes. 3DES, on the other hand, has lower bandwidth values than DES algorithm on the Windows Server 2003 platform. Performance can be degraded with 3DES than DES due to the complexity of the 3DES algorithm. McGregor & Lee (2000) also noted that encryption with 3DES can be reduce system performance than DES. According to Ferguson & Schneier (2000) the reason behind these results might be the limited key length of DES algorithm. Also both transmission protocols bandwidth values for all the algorithms show lower performance with smaller and larger packet sizes and higher performance with medium size packets. However UDP bandwidth for DES with both Microsoft Windows Vista and Microsoft Windows 7 for small packet size to medium packet size is significantly higher than the TCP bandwidth. Narayan & Shi noted in 2010,
comparing TCP throughput values with UDP, TCP values are slightly lower for most operating systems. This could be caused as TCP has to wait for acknowledgment packets. Both transmission protocols give higher bandwidth even with DES-MD5, but only with medium and large payloads. Microsoft Windows Vista has higher TCP bandwidth than Microsoft Windows 7 with DES and 3DES algorithms on Microsoft Windows Server 2003.

TCP latency exhibits higher performance with Windows Vista than with Microsoft Windows 7 on the Windows Server 2003 for all the packet sizes. However, UDP gives higher latency with Microsoft Windows 7 than with Windows Vista with small and medium size packets on Microsoft Windows Server 2003. TCP latency performances with all the algorithms are lower at medium size packets than the small and large packet sizes. UDP latency gives lower performance with small size packets, higher performance with medium size packets, and stable performance with large packet sizes on Windows Server 2003. Narayan & Shi (2010) said that TCP latency values rise as packet sizes increase on most Windows operating systems. UDP with 3DES-MD5 and DES-MD5 shows long delays with smaller packets and with the large packets it experiences small delay times with both Microsoft Windows Vista and Windows 7 on Microsoft Windows Server 2003. However, TCP with 3DES-MD5 shows low latency with smaller packets and high latency with large packets with the same environment.

As per the results identified in the Chapter five, when the packet sizes increase, both TCP and UDP jitter values increases as well on Windows Server 2003 for DES and 3DES algorithms. Basically it exhibits a linear trend. TCP jitter values are higher with Windows Vista than with Windows 7 on Windows Server 2003. 3DES-SHA1 with Microsoft Windows Vista records the highest value on Microsoft Windows Server 2003 for both the transmission control protocols. UDP jitter values are higher than TCP with large packet sizes with both Microsoft Windows Vista and 7 on Microsoft Windows Server 2003. The Microsoft Windows Server 2008 results are discussed in the next section.
6.2 Performance of TCP/UDP for IPSec algorithms on VPN using Microsoft Windows Server 2008

Analysis of the network metrics throughput, latency and jitter with different encryption algorithms in the Microsoft Windows Server 2008 environment will be discussed in this section. According to the analysis of IPv4 performance while employing the transmission protocols TCP and UDP, the following behaviours can be seen.

TCP bandwidth for AES with Microsoft Windows Vista and Windows 7 on Microsoft Windows Server 2008 records the highest bandwidth in all the packet sizes. However, with the same environment the highest bandwidth can be seen with UDP for AES and DES. Also for both the client operating systems, 3DES has lower bandwidth values than DES and AES algorithm on the Windows Server 2008 platform. Similar to Windows Server 2003 results this can be caused due to the complexity of 3DES. Ferrante et al. (2000), also mentioned that 3DES algorithm is much slower than AES. This is similar to the Windows Server 2003 results. With all the algorithms, both TCP and UDP exhibit lower performance with smaller and larger packet sizes and higher performance with medium size packets. This is also similar to the Windows Server 2003 results.

The TCP throughput on Microsoft Windows Server 2008 with Windows 7 performs 31% better than with Windows Vista. For both the client operating systems, 3DES has lower bandwidth values than the DES and AES algorithm on the Windows Server 2008 platform. As mentioned earlier Ferrante et al. (2000), also agreed on saying that 3DES algorithm is much slower than AES. This is similar to the Windows Server 2003 results. The UDP throughput for 3DES on Windows Server 2008 with Windows Vista performs better than with Windows 7. With AES256-SHA1, UDP bandwidth is lower than TCP at smaller payloads with both Windows Vista and 7 on Windows Server 2008. As per the analysis, it is concluded that AES algorithm performs well with both Microsoft Windows Vista and 7 on Windows Server 2008.

Similar to Windows Server 2003, TCP latency gives higher performance with Windows Vista than Windows 7 on Windows Server 2008 for all the packet sizes. UDP latency trends are more

Similar to Windows 2003, when the packet size increases, TCP and UDP jitter values increase as well on Windows Server 2008 for DES, 3DES and AES algorithms (it shows a linear trend). UDP delay values are lower and consistent with the smaller packet sizes on Windows Server 2003, and medium and larger packet sizes gives higher jitter values as UDP jitter on Windows Server 2003. For UCP jitter, 3DES-SHA1 with Windows 7 records the highest value on Windows Server 2003. Compared to TCP jitter, UDP jitter values are much higher with larger packet sizes with both Windows Vista and 7 on Windows Server 2008. This behaviour is similar to Windows Server 2003. TCP jitter shows a logarithmic trend with DES-MD5 on Windows Vista and 7 with Windows Server 2008. UDP with DES-MD5 gives higher jitter value on Windows Vista at large payloads on Windows Server 2008. Jitter of both the transmission protocols exhibits a positive linear trend with AES256-SHA1 on Windows Server 2008.

After analysing the research metrics on the Windows 2003 and 2008 Server environments, it is worth doing a cross comparison between the two server operating systems. Therefore, the next section will discuss the behaviours of the two Windows Servers with different research scenarios.

6.3 Cross comparison between the performances of TCP/UDP for IPSec algorithms on two Windows Server Operating Systems (2003/2008)

The research metrics throughput, latency and jitter behaviours can vary on two different Windows Server environments. This section will help to present a fair comparison between the server operating systems.
With regards to throughput TCP traffic for 3DES, Windows 7 with Windows Server 2003 records a higher bandwidth than Windows 7 with Windows Server 2008. Kolahi et al. also noted in 2008 that Windows Server 2003 gives the highest bandwidth, for both TCP and UDP in 3DES. For TCP latency for 3DES, Windows 7 with Windows Server 2003 records higher bandwidth than Windows 7 with Windows Server 2008. It is approximately 76% higher. TCP latency with 3DES-SHA gives approximately 63% better values with Windows Vista than with Windows 7 on Windows Server 2003. TCP jitter values for 3DES-SHA1 increase gradually with all the packet sizes.

With regards to TCP jitter for 3DES, Windows 7 with Windows Server 2003 records a higher value at approximately 61% than Windows 7 with Windows Server 2008.

Analysis of the algorithms concludes that for TCP traffic, 3DES with Windows Vista records a higher bandwidth with Windows Server 2003 than with Windows Server 2008, and DES records a higher bandwidth with Windows Server 2008 than with Windows Server 2003. TCP traffic with 3DES and DES on Windows 7 records a higher bandwidth with both Windows Server 2003 and 2008.

### 6.5 Chapter Summary

This Chapter discussed the research findings in detail under three main sections. The conclusions of this empirical test-bed analysis of a various IPSec algorithms with TCP and UDP on four different Microsoft Windows operating systems will be explained in next Chapter seven. And also future study areas were identified and listed for future research.
Chapter 7: Conclusion

Conclusions from the research area are drawn in this section and also further study areas will be mentioned at the end. This research study was conducted to analyse TCP/UDP performance on the VPN network employed with IPSec algorithms. Performances were checked with TCP and UDP on different Microsoft Windows platforms. Results were extracted from the data collected from the test-bed experimental setup. Three main performance metrics were analysed: throughput, latency and jitter. As mentioned earlier. There were four operating systems that were employed in this research: Microsoft Windows Vista and 7 as client operating systems and Microsoft Windows Server 2003 and 2008 as server operating systems. The importance of VPN usage in the industry gave the motivation for this research study. After reviewing the existing studies in the same area it was realised that of conducting this research was necessary. The quantitative approach was selected and research was conducted as a test bed experimental study after examining different research methodologies. Test data were generated and collected using the networking monitoring tool D-ITG. Experimental data were analysed by plotting line and column graphs on Microsoft Excel. From the empirical test-bed analysis of the research, performance related metrics throughput, jitter and latency were measured for TCP and UDP traffic between two nodes. Therefore, the following conclusions can be drawn as a summary of the study.

- Windows Vista gives higher bandwidth than Windows 7 with DES and 3DES algorithms on Windows Server 2003.
- Latency values differ for the operating systems as well where Windows Vista values are higher than Windows 7 on Server Windows 2003 for TCP and vice versa for UDP.
- Jitter values differ for the operating systems as well where Windows Vista values are higher than those of Windows 7 for TCP traffic.
- Larger packet size on both Windows Vista and 7 on Windows Server 2003 gives higher jitter for UDP than TCP.
- TCP jitter is higher with Windows Vista than with Windows 7 on Windows 2003. It is 66% better than Windows 7.
- UDP jitter is also higher with Windows Vista than with 7 on Windows Server 2008.
Both TCP and UDP for DES with Windows Vista and 7 on Windows Server 2003 give the highest bandwidth for all the packets.

TCP bandwidth is highest with AES on Windows Vista and 7 in Windows Server 2008.

TCP throughput is 31% better with Windows 7 on Windows Server 2008 than with Windows Vista.

TCP exhibits higher latency with Windows Vista than with Windows 7 on Windows 2003 and Windows Server 2008 for all packet sizes. It is 75% better than Windows 7.

UDP exhibits higher latency with Windows 7 on Windows 2003 and Windows 2008 for all packet sizes.

### 7.1 Summary of findings

After conducting the research, the data were analysed and discussed in the previous sections. As a conclusion, the research questions are answered below.

The main question:

> “Which combination of IPSec algorithm and operating system gives the best network performance when measured for TCP/UDP traffic?”

According to the information provided in the previous sections in this document there were four operating systems employed in the VPN environment with different IPSec algorithms. Basically two transmission protocols, TCP and UDP, were involved in this study. The findings of the study concluded that DES algorithm performs well with both the Microsoft Windows Vista and 7 client operating systems when measurers for TCP and UDP. DES algorithm performs well with the Microsoft Windows 2003 server operating system when measured for TCP and UDP. Also this study concluded that AES algorithm performs well with the Microsoft Windows 7 client operating system when measured for TCP with Microsoft Server 2008, at 31%. 3DES algorithm performs well with the Microsoft Vista client operating system with the Microsoft 2008 server operating system when measured for UDP. The research questions considered in this research study are addressed in the next section.
Does TCP/UDP network performance vary when implemented on a different client operating system on a Local Area Network?

The findings of this study concluded that Microsoft Vista throughput is higher with DES in Microsoft windows 2003 and also Microsoft Windows Vista throughput is higher with AES-128-SHA1 in Microsoft Windows 2008. UDP throughput is higher than TCP throughput on both Microsoft Windows Vista and 7.

Does TCP/UDP network performance vary when implemented on a different server operating system on a Local Area Network?

In this study the findings concluded that on both TCP and UDP, throughput performed higher with the Microsoft Windows 2008 server environment.

Which client operating system gives the best performance for IPSec cryptographic algorithm when measured for TCP/UDP traffic types?

In this study it is concluded that Windows Vista performed well with DES-MD5 on the Windows Server 2003 and 2008 server environments when measured for TCP. Windows 7 exhibits high performance on Windows 2008 when measured for UDP.

Which server operating system gives the best performance for IPSec cryptographic algorithm when measured for TCP/UDP traffic types?

The finding of this study concluded that Microsoft Windows 2003 performed better than Windows 2008 when measured for TCP/UDP traffic types.

This empirical evaluation of four operating systems has proven that IPSec algorithms on VPN give different network performance metric values for different combinations of operating systems, protocols and algorithms. Further study areas will be noted in the next section.
7.2 Future Work

This research study tried to answer one main question and four sub questions and results were answered and presented in these chapters. The main focus of the study was to find which combination of IPSec algorithm and operating system gives the best network performance when measured for TCP/UDP traffic. The first objective was to find whether TCP/UDP network performance varies when implemented on a different client operating system on a Local Area Network. The second objective was to find whether TCP/UDP network performance varies when implemented on a different server operating system on a Local Area Network. The third objective was to find which client operating system gives the best performance for IPSec cryptographic algorithm when measured for TCP/UDP traffic types. The final objective was to find which server operating system gives the best performance for IPSec cryptographic algorithm when measured for TCP/UDP traffic types. This research experiment was undertaken only under the Windows Platform by analysing various metrics. However, this research could be extended to various interesting studies as below.

- Conduct the same study on other operating system environments
- Conduct the same study on a wireless network
- Conduct the same study on hardware routers

As there are vast variations on the VPN research area, there is a need for further research on the evaluation of the performance of many parameters involved in the VPN environment. This work will be extended to include a greater range of operating systems, protocols and metrics.
References


