

# Improving Network Performance: An Evaluation of TCP/UDP on Networks

A Doctor of Computing Thesis  
by

Shaneel Narayan

Supervisors

*Principal*

Professor Abdolhossein Sarrafzadeh

*Associate*

Dr Chandimal Jayawardena

Dr Iman Ardekani

Submitted in partial fulfilment of the  
requirements for the degree of  
Doctor of Computing

Department of Computing  
UNITEC Institute of Technology  
Auckland, New Zealand  
December 2014



*Dedicated to my mum and my little angel*



---

## Abstract

Computer networks are complex - they are a heterogeneous environment in which numerous services, such as electronic mail, web browsing, voice and multimedia data, traverse the globe daily. The needs and demands of end users continuously change, and to meet these, new technologies are being incorporated into this mega digital infrastructure at a phenomenal rate. In addition to ensuring that necessary functionalities are provided, it is vitally important to ensure that network performance is always at its optimum.

Fundamentally, networks are an environment where data, mostly in the form of TCP and UDP, are being propagated end-to-end between the sending and receiving nodes. There are numerous avenues of network performance that can be exploited in order to improve its performance. Research in this area is multi-faceted, and in this thesis the focus is on evaluating the behaviour of TCP and UDP end-to-end on networks in three scenarios, namely, networks with transition mechanisms, wireless based networks, and in the context of using virtual private network technologies as security protocols.

This thesis will give insights into the behaviour of common protocols on real networks. Therefore, performance metrics related to networks have been gathered from test-bed implementations. The collected data has been presented in graphs and heat maps, which have been evaluated to ascertain network related characteristics. In particular, key metrics have been identified, networking techniques within each context have been ranked, specific observations related to each network environment have been made, and finally, the impact of either version of the Internet Protocol or an operating system has been evaluated.



---

## Acknowledgments

A lifelong dream of completing a Doctorate has been fulfilled, and for that, first and foremost I would like to thank my Mum. She has inspired me and instilled in me the value of education. Her sacrifices, blessings, and encouragement have been my strength from inception to conclusion. Thank you Mum.

I would like to express my sincere gratitude to my principal supervisor Professor Abdolhossein Sarrafzadeh, and associate supervisors Dr Chandimal Jayawardena and Dr Iman Ardekani. Thank you for giving me an opportunity to work with you, your guidance has been priceless. Special acknowledgements go to my employer Unitec Institute of Technology and the Department of Computing for supporting me in this process for a number of years. I thank my Head of Department Professor Abdolhossein Sarrafzadeh for believing in me.

A big shout out to all those undergraduate and postgraduate students who have worked with me tirelessly on various network research projects over the last 14 years. Your enthusiasm and support has been brilliant.

I am forever grateful for my daughter Gitali's unconditional love. Moments with you are always priceless, and now with the completion of the Doctorate, I will spend more time with you.

Last but not least, I would like to thank God for being my strength. Without His blessings, this would have never come to fruition.





---

## List of Abbreviations

AES	Advance Encryption Standard
AH	Authentication Header
AMID	Additive-Increase Multiplicative-Decrease
APNIC	Asia Pacific Network Information Centre
ARP	Address Resolution Protocol
ARPA	Advance Research Project Agency
BYOD	Bring Your Own Device
CA	Certification Authority
CCK	Complementary Coding Key
CCMP	Chaining Message Authentication Code Protocol
CGA	Cryptographic Anomaly
CHAP	Challenge-Handshake Authentication Protocol
CIDR	Cluster-based Inter-Domain Routing
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance
D-ITG	Distributed-Internet Traffic Generator
DES	Data Encryption Standard

---

DHCP	Dynamic Host Control Protocol
DNS	Domain Name System
DoD	Department of Defense
DOS	Denial of Service
DSCP	Differentiated Service Code Point
DSSS	Direct-Sequence Spread Spectrum
DSTM	Dual Stack Transition Mechanism
DTTS	Dynamic Tunnelling Transition Solution
DUPACK	Duplicate Acknowledgement
EAP	Extensible Authentication Protocol
ECN	Explicit Congestion Notification
ELISIA	Evolvable Locator/ID Separation Internet Architecture
ERP	Extended Rate Physical
ESP	Encapsulated Security Payload
FTP	File Transfer Protocol
GPRS	General Packet Radio Services
GRE	Generic Routing Examinations
HMAC	Hash Message Authentication Code
HR-DSSS	High Rate - DSSS
HTTP	Hypertext Transfer Protocol
HTTPS	Secure HTTP
IANA	Internet Assignment Numbers Authority
ICAAAN	Internet Corporation for Assignment Name and Numbers
ICMP	Internet Control Message Protocol
ICV	Integrity Check Value
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force

---

IGMP	Internet Group Message Protocol
IKE	Internet Key Exchange
IMP	Interface Message Processor
IP	Internet Protocol
IPSec	Internet Protocol Security
ISAKMP	Internet Security Association and Key Management Protocol
ISATAP	Intra-Site Automatic Tunnel Addressing Protocol
ISM	Industrial Scientific Medical
ISO	International Organization for Standards
ISP	Internet Service Provider
IV	Initiation Vector
IVC	Integrity Value Check
L2TP	Layer 2 Tunnelling Protocol
LAN	Local Area Network
LDPC	Low-Density Parity-Check
MAC	Media Access Control
MANET	Mobile Ad hoc Network
MD	Message Digest
MIC	Message Integrity Check
MIMO	Multiple-Input and Multiple-Output
MPPE	Microsoft Point-to-Point Encryption
MTU	Maximum Transmission Unit
MU-MIMO	Multiple-User MIMO
NAPT-PT	Network Address Port Translation - Protocol Translation
NAT	Network Address Translation
NAT-PT	Network Address Translation - Protocol Translation
NBMA	Non-Broadcast Multiple Access

---

NDP	Neighbor Discover Protocol
NIC	Network Interface Card
NLB	Network Load Balancing
OFDM	Orthogonal Frequency-Division Multiplexing
OSI	Open Systems Interconnection
PAT	Port Address Translation
PEAP	Protected EAP
PET	Prefixing, Encapsulating, and Translation
PKI	Public Key Infrastructure
PPP	Point-to-Point Protocol
PPTP	Point-to-Point Tunnelling Protocol
PRL	Potential Router List
PSK	Pre-Shared Key
PSMP	Power Save Multi-Poll
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
RA	Router Advertisement
RARP	Reverse Address Resolution Protocol
RC	Rivest Cipher
RD	Reverse Direction
RFC	Request for Comments
RIPE NCC	The Rseaux IP Europens Network Coordination Centre
RIR	Regional Internet Registry
RS	Router Solicitation
SA	Security Association
SACK	Selective Acknowledgement
SDM	Spatial-Division Multiplexing

---

SEND	Secure Neighbor Discovery
SGI	Short Guard Interval
SHA	Secure Hash Algorithm
SKEME	Security Key Exchange Mechanism
SLAAC	Stateless Address Auto-configuration
SMTP	Simple Mail Transfer Protocol
SPD	Security Policy Database
SPI	Security Parameter Index
SSL	Secure Socket Layer
ssthresh	Slow Start Threshold
SSTP	Secure Socket Tunnelling Protocol
STBC	Space-Time Block Code
TCP	Transmission Control Protocol
TCPA	Teredo Client Protection Algorithm
TELNET	Terminal Network or Network Virtual Terminal Protocol
TEP	Tunnel End-Point
TKIP	Temporal Key Integrity Protocol
TTL	Time to Live
UDP	User Datagram Protocol
UHF	Ultra High Frequency
VHT	Very High Throughput
VoIP	Voice over Internet Protocol
VPN	Virtual Private Network
WEP	Wired Equivalent Privacy
WiGig	Wireless Gigabit
WiMax	Worldwide Interoperability for Microwave Access
WLAN	Wireless LAN
WPA	Wi-Fi Protected Access



---

# Contents

<b>Abstract</b>	<b>v</b>
<b>Acknowledgements</b>	<b>vii</b>
<b>List of Abbreviations</b>	<b>xiii</b>
<b>List of Publications</b>	<b>1</b>
<b>1 Introduction</b>	<b>5</b>
1.1 Thesis Objectives . . . . .	8
1.2 Thesis Contributions . . . . .	9
1.3 Thesis Structure . . . . .	10
<b>2 Background</b>	<b>11</b>
2.1 Introduction and Motivations . . . . .	11
2.2 A Layered Structure . . . . .	12
2.2.1 OSI Reference Model . . . . .	13
2.2.2 TCP Reference Model . . . . .	16
2.2.3 OSI and TCP Reference Model Comparison . . . . .	18
2.2.4 The Internet Model . . . . .	19
2.2.5 The UDP Protocol . . . . .	22
2.2.6 The TCP Protocol . . . . .	24
2.2.7 IP Header . . . . .	35
2.2.8 Comparison of IPv4 and IPv6 . . . . .	37

---

2.2.9	Exhaustion of IPv4 addresses . . . . .	40
2.2.10	IP Addressing . . . . .	41
2.2.11	TCP/UDP Related Research . . . . .	44
<b>3</b>	<b>Methodology</b>	<b>47</b>
3.1	Testing Networks in Realistic Conditions . . . . .	47
3.1.1	Simulation . . . . .	48
3.1.2	Emulation . . . . .	50
3.1.3	Test-bed Networks . . . . .	51
3.2	Experimental Test-bed Architecture . . . . .	52
3.2.1	Network Schematic . . . . .	53
3.2.2	Traffic Generation Software . . . . .	54
3.2.3	Traffic Generator . . . . .	55
3.3	Performance Metrics . . . . .	56
3.4	Test-bed Data Collection . . . . .	58
<b>4</b>	<b>TCP/UDP Behaviour across Transition Mechanisms</b>	<b>63</b>
4.1	Introduction and Motivation . . . . .	63
4.2	The IP Addressing Problem . . . . .	64
4.3	The Transition Mechanisms . . . . .	67
4.3.1	Dual Stack . . . . .	67
4.3.2	Address Translation . . . . .	69
4.3.3	Tunnelling . . . . .	73
4.4	Literature Analysis of Transition Mechanisms . . . . .	83
4.5	Test-bed Transition Mechanism Implementations . . . . .	86
4.6	Performance Metrics Measurement for various Transition Mechanisms . . . . .	92
4.7	Results Evaluation . . . . .	107
<b>5</b>	<b>TCP and UDP Behaviour in Wireless Environments</b>	<b>109</b>
5.1	Introduction and Motivation . . . . .	109
5.2	Wireless Preamble . . . . .	110
5.3	Wireless Standards . . . . .	112
5.3.1	IEEE802.11b . . . . .	114
5.3.2	IEEE802.11g . . . . .	115



## Contents

---

5.3.3	IEEE802.11n . . . . .	116
5.3.4	IEEE802.11ac . . . . .	117
5.4	Wireless Security Protocols . . . . .	120
5.4.1	Wired Equivalent Privacy . . . . .	120
5.4.2	Wi-Fi Protected Access . . . . .	122
5.4.3	Wi-Fi Protected Access 2 . . . . .	124
5.5	Some Key Research . . . . .	125
5.6	Wireless Test-bed Setup . . . . .	126
5.7	Performance Metrics Measurements from Wireless Test-beds . . . . .	127
5.8	Results Evaluation . . . . .	144
<b>6</b>	<b>TCP/UDP Behaviour in Virtual Private Networks</b>	<b>147</b>
6.1	Introduction and Motivation . . . . .	147
6.2	VPN Preamble . . . . .	147
6.3	VPN Protocols . . . . .	149
6.3.1	Layer 2 VPN Protocols . . . . .	150
6.3.2	Layer 3 VPN Protocols . . . . .	151
6.4	Key VPN Research . . . . .	167
6.5	VPN Test-bed Setup . . . . .	168
6.6	Performance Metrics Measurements from IPSec and SSTP VPN Protocols . . . . .	169
6.7	Results Evaluation . . . . .	183
<b>7</b>	<b>Discussion</b>	<b>185</b>
7.1	Preamble . . . . .	185
7.2	Networks with Transition Mechanisms . . . . .	186
7.3	Networks with Wireless Implementations . . . . .	188
7.4	Networks with VPNs . . . . .	190
7.5	Thesis Contributions . . . . .	192
<b>8</b>	<b>Conclusions</b>	<b>195</b>
	<b>References</b>	<b>197</b>



---

## List of Figures

1.1	Statistics - <i>Network Performance Index</i> . . . . .	7
1.2	Thesis - <i>Thesis Structure</i> . . . . .	10
2.1	Different Models - <i>OSI, TCP/IP, and Internet</i> . . . . .	18
2.2	Packet Structure - <i>UDP</i> . . . . .	23
2.3	Data Transfer Structure - <i>Reliable and Unreliable</i> . . . . .	24
2.4	Packet Structure - <i>TCP</i> . . . . .	26
2.5	Three Way Handshake - <i>TCP</i> . . . . .	29
2.6	TCP Congestion Control - <i>Slow Start Algorithm</i> . . . . .	31
2.7	TCP Congestion Control - <i>Saw Tooth</i> . . . . .	33
2.8	Internet Protocol - <i>IPv4 and IPv6 Header</i> . . . . .	36
2.9	Statistics - <i>IPv6 Address Usage</i> . . . . .	41
3.1	Experimentation Approaches - <i>Simulation</i> . . . . .	49
3.2	Experimentation Approaches - <i>Simulation, Emulation, Test-bed, and Real Networks</i> . . . . .	52
3.3	Performance Tools - <i>Test-bed Schematic</i> . . . . .	54
3.4	Performance Tools - <i>D-ITG Diagram</i> . . . . .	58
3.5	D-ITG - <i>Sender Command Line Interface</i> . . . . .	59
3.6	D-ITG - <i>Sender Sample Commands</i> . . . . .	60
3.7	D-ITG - <i>Decoder Sample Commands</i> . . . . .	60
3.8	D-ITG - <i>Decoded Data</i> . . . . .	61

## List of Figures

---

4.1	Statistics - <i>IPv4 Address Rundown</i> . . . . .	65
4.2	Transition Mechanisms - <i>Dual Stack Architecture</i> . . . . .	69
4.3	Transition Mechanisms - <i>NAT-PT Architecture</i> . . . . .	71
4.4	Transition Mechanisms - <i>NAT64 Architecture</i> . . . . .	73
4.5	Transition Mechanisms - <i>6to4 Architecture</i> . . . . .	77
4.6	Transition Mechanisms - <i>ISATAP Architecture</i> . . . . .	79
4.7	Transition Mechanisms - <i>Teredo Architecture</i> . . . . .	81
4.8	Test-bed Implementation - <i>Dual Stack Transition Mechanism</i> . .	86
4.9	Test-bed Implementation - <i>Configured Tunnel</i> . . . . .	87
4.10	Test-bed Implementation - <i>6to4</i> . . . . .	88
4.11	Test-bed Implementation - <i>ISATAP</i> . . . . .	89
4.12	Test-bed Implementation - <i>Teredo</i> . . . . .	90
4.13	Test-bed Implementation - <i>NAT64</i> . . . . .	91
4.14	Transition Mechanism Graph - <i>TCP Throughput</i> . . . . .	93
4.15	Transition Mechanism Heat Map: <i>TCP Throughput</i> . . . . .	94
4.16	Transition Mechanism Graph - <i>UDP Throughput</i> . . . . .	94
4.17	Transition Mechanism Heat Map - <i>UDP Throughput</i> . . . . .	95
4.18	Transition Mechanism Graph - <i>TCP Delay</i> . . . . .	96
4.19	Transition Mechanism Heat Map - <i>TCP Delay</i> . . . . .	97
4.20	Transition Mechanism Graph - <i>UDP Delay</i> . . . . .	97
4.21	Transition Mechanism Heat Map - <i>UDP Delay</i> . . . . .	99
4.22	Transition Mechanism Graph - <i>TCP Jitter</i> . . . . .	100
4.23	Transition Mechanism Heat Map - <i>TCP Jitter</i> . . . . .	100
4.24	Transition Mechanism Graph - <i>UDP Jitter</i> . . . . .	101
4.25	Transition Mechanism Heat Map - <i>UDP Jitter</i> . . . . .	102
4.26	Transition Mechanism Graph: <i>TCP CPU1 Usage Percentage</i> . .	103
4.27	Transition Mechanism Heat Map: <i>TCP CPU1 Usage Percentage</i>	103
4.28	Transition Mechanism Graph: <i>UDP CPU1 Usage Percentage</i> . .	104
4.29	Transition Mechanism Heat Map: <i>UDP CPU1 Usage Percentage</i>	104
4.30	Transition Mechanism Graph: <i>TCP CPU2 Usage Percentage</i> . .	105
4.31	Transition Mechanism Heat Map: <i>TCP CPU2 Usage Percentage</i>	105
4.32	Transition Mechanism Graph: <i>UDP CPU2 Usage Percentage</i> . .	106
4.33	Transition Mechanism Heat Map: <i>UDP CPU2 Usage Percentage</i>	106
5.1	IEEE 802.11 Protocol - <i>802.11 Frame Format</i> . . . . .	113

## List of Figures

---

5.2	IEEE 802.11 Protocol - <i>802.11ac Frame Format</i> . . . . .	119
5.3	Wireless Security Protocol - <i>WEP</i> . . . . .	122
5.4	Wireless Security Protocol - <i>WPA</i> . . . . .	123
5.5	Test-bed Implementation - <i>Wireless</i> . . . . .	126
5.6	Wireless Graph: <i>TCP Throughput</i> . . . . .	128
5.7	Extrapolated Graph: <i>TCP Throughput</i> . . . . .	128
5.8	Heat Map: <i>TCP Throughput</i> . . . . .	129
5.9	Wireless Graph: <i>UDP Throughput</i> . . . . .	131
5.10	Extrapolated Graph: <i>UDP Throughput</i> . . . . .	131
5.11	Heat Map: <i>UDP Throughput</i> . . . . .	132
5.12	Wireless Graph: <i>TCP Jitter</i> . . . . .	134
5.13	Extrapolated Graph: <i>TCP Jitter</i> . . . . .	134
5.14	Heat Map: <i>TCP Jitter</i> . . . . .	135
5.15	Wireless Graph: <i>UDP Jitter</i> . . . . .	136
5.16	Extrapolated Graph: <i>UDP Jitter</i> . . . . .	136
5.17	Heat Map: <i>UDP Jitter</i> . . . . .	137
5.18	Wireless Graph: <i>TCP Delay</i> . . . . .	138
5.19	Heat Map: <i>TCP Delay</i> . . . . .	139
5.20	Wireless Graph: <i>UDP Delay</i> . . . . .	140
5.21	Heat Map: <i>UDP Delay</i> . . . . .	141
5.22	Wireless Graph: <i>UDP Drop Rate</i> . . . . .	142
5.23	Heat Map: <i>UDP Drop Rate</i> . . . . .	143
6.1	VPN Protocol: <i>PPTP Implementation</i> . . . . .	153
6.2	VPN Protocol: <i>IPsec Implementation</i> . . . . .	155
6.3	IPSec Protocol: <i>AH and ESP Packet</i> . . . . .	157
6.4	IPsec Protocol: <i>AH IPv4 Header</i> . . . . .	158
6.5	IPSec Protocol: <i>ESP IPv4 Header</i> . . . . .	160
6.6	VPN Protocol: <i>SSTP Implementation</i> . . . . .	162
6.7	VPN Protocol: <i>OpenVPN Implementation</i> . . . . .	164
6.8	VPN Protocol: <i>SSL Implementation</i> . . . . .	165
6.9	Test-bed Implementation - <i>VPN</i> . . . . .	168
6.10	VPN Graph: <i>TCP Throughput</i> . . . . .	170
6.11	VPN Heat Map: <i>TCP Throughput</i> . . . . .	171
6.12	VPN Graph: <i>UDP Throughput</i> . . . . .	172

## List of Figures

---

6.13	VPN Heat Map: <i>UDP Throughput</i> . . . . .	173
6.14	VPN Graph: <i>TCP Delay</i> . . . . .	174
6.15	VPN Heat Map: <i>TCP Delay</i> . . . . .	175
6.16	VPN Graph: <i>UDP Delay</i> . . . . .	176
6.17	VPN Heat Map: <i>UDP Delay</i> . . . . .	177
6.18	VPN Graph: <i>TCP Jitter</i> . . . . .	178
6.19	VPN Heat Map: <i>TCP Jitter</i> . . . . .	179
6.20	VPN Heat Map: <i>UDP Jitter</i> . . . . .	180
6.21	VPN Graph: <i>UDP Jitter</i> . . . . .	181
6.22	Collective Graphs: <i>Throughput</i> . . . . .	182
6.23	Collective Graphs: <i>IPv4 Delay</i> . . . . .	182
6.24	Collective Graphs: <i>IPv6 Delay</i> . . . . .	183

---

## List of Tables

2.1	Application Layer - <i>TCP and UDP Ports</i> . . . . .	21
2.2	TCP Congestion Control - <i>TCP Algorithms</i> . . . . .	34
2.3	Statistics - <i>IPv4 Address Exhaustion</i> . . . . .	40
2.4	Internet Protocol - <i>IPv4 Classes</i> . . . . .	41
2.5	Internet Protocol - <i>IPv6 Unicast and Anycast</i> . . . . .	42
2.6	Internet Protocol - <i>IPv6 Global and Multicast Addresses</i> . . . . .	43
2.7	Internet Protocol - <i>IPv6 Abbreviation Technique</i> . . . . .	44
3.1	Performance Tools - <i>Comparison</i> . . . . .	55
4.1	Transition Techniques - <i>Translation</i> . . . . .	70
4.2	Transition Techniques - <i>Tunnelling</i> . . . . .	75
4.3	Related Research - <i>Transition Mechanism</i> . . . . .	85
5.1	IEEE 802.11 Protocol - <i>Wireless Network</i> . . . . .	112
5.2	IEEE 802.11 Protocol - <i>Wireless Standards</i> . . . . .	114
5.3	IEEE 802.11 Protocol - <i>Wireless Security</i> . . . . .	120
5.4	Related Research - <i>Wireless</i> . . . . .	125
6.1	VPN Protocol - <i>Layer 2 and Layer 3 VPN</i> . . . . .	150
6.2	Types of VPN - <i>Layer 2 VPN</i> . . . . .	150
6.3	Types of VPN - <i>Layer 3 VPN</i> . . . . .	151
6.4	VPN Algorithms - <i>Encryption Algorithms</i> . . . . .	151
6.5	Related Research - <i>VPN</i> . . . . .	167

List of Tables

---

7.1	Ranking - <i>Performance Metrics</i> . . . . .	186
7.2	Ranking - <i>Transition Mechanism TCP and UDP Delay</i> . . . . .	187
7.3	Ranking - <i>Transition Mechanism TCP and UDP Throughput</i> . . . . .	187
7.4	Ranking - <i>Wireless</i> . . . . .	188
7.5	Ranking - <i>Wireless Delay</i> . . . . .	189
7.6	Ranking - <i>Wireless Throughput</i> . . . . .	189
7.7	Ranking - <i>Wireless UDP Drop Rate</i> . . . . .	190
7.8	Ranking - <i>VPN TCP and UDP Delay</i> . . . . .	191
7.9	Ranking - <i>VPN TCP and UDP Throughput</i> . . . . .	191



---

## List of Publications

### Journal Articles

**Narayan, S.,** & Fitzgerald, M. (2012). Empirical network performance evaluation of security protocols on operating systems. *International Journal of Wireless and Microwave Technologies (IJWMT)*, 2(5), 19.

**Narayan, S.,** & Lutui, P. R. (2012). TCP/IP jumbo frames network performance evaluation on a test-bed infrastructure. *International Journal of Wireless and Microwave Technologies (IJWMT)*, 2(6), 29.

**Narayan, S.,** Graham, D., & Barbour, R. H. (2009). Generic factors influencing optimal LAN size for commonly used operating systems maximized for network performance. *International Journal of Computer Science and Network Security*, 9(6), 6372.

### Conference Papers

**Narayan, S.,** Jayawardena, C., Wang, J., Ma, W., & Geetu, G. (2015). Performance test of IEEE 802.11ac Wireless Devices. *Paper to be presented at the IEEE International Conference on Computer Communication and Informatics (ICCCI) (January 8-10, 2015)*.

**Narayan, S.,** Williams, C.J., Hart, D.K., & Qualtrough, M.W. (2015). Network performance comparison of VPN protocols on wired and wireless networks. *Paper to be presented at the IEEE International Conference on Computer Communication and Informatics (ICCCI) (January 8-10, 2015)*.

**Narayan, S.,** & Lutui, P. (2013, Dec). Network performance evaluation of jumbo frames on a network. In *Proceedings of the 6th IEEE International Conference on Emerging Trends in Engineering and Technology (ICETET)*. (p. 69-72). doi: 10.1109/ICETET.2013.16

Kolahi, S.S., **Narayan, S.,** Nguyen, D.D.T., & Sunarto, Y. (2011, March). Performance monitoring of various network traffic generators. In *Proceedings of the 13th IEEE International Conference on Computer Modelling and Simulation (UKSim)*. (p. 501-506). doi: 10.1109/UKSIM.2011.102

- Narayan, S., FitzGerald, M., & Ram, S.** (2010, Dec). Empirical network performance evaluation of IPSec algorithms on windows operating systems implemented on a test-bed. In *Proceedings of the IEEE International Conference on Computational Intelligence and Computing Research (ICCCIC)*. (p. 1-4).
- Narayan, S., Lutui, P.R., Vijayakumar, K., & Sodhi, S.** (2010, Dec). Performance analysis of networks with IPv4 and IPv6. In *Proceedings of the IEEE International Conference on Computational Intelligence and Computing Research (ICCCIC)*. (p. 1-4). doi: 10.1109/ICCCIC.2010.5705805
- Narayan, S., & Lutui, P.R.** (2010, Dec). Impact on network performance of jumbo-frames on IPv4/IPv6 network infrastructure: An empirical test-bed analysis. In *Proceedings of the 4th IEEE International Conference on Internet Multimedia Services Architecture and Application (IMSAA)*. (p. 1-4). doi: 10.1109/IMSAA.2010.5729409
- Narayan, S., & Tauch, S.** (2010, July). IPv4-v6 configured tunnel and 6to4 transition mechanisms network performance evaluation on linux operating systems. In *Proceedings of the 2nd IEEE International Conference on Signal Processing Systems (ICSPS)*. (Vol. 2, p. 113-117). doi: 10.1109/IC-SPS.2010.5555209
- Narayan, S., & Tauch, S.** (2010, July). IPv4-v6 transition mechanisms network performance evaluation on operating systems. In *Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT)*. (p. 664-668). doi: 10.1109/ICCSIT.2010.5564141
- Narayan, S., & Shi, Y.** (2010, July). Application layer network performance evaluation of VoIP traffic on a test-bed with IPv4 and IPv6 LAN infrastructure. In *Proceedings of the 8th Region IEEE International Conference on Computational Technologies in Electrical and Electronics Engineering (SIBIRCON)*. (p. 215-219). doi: 10.1109/SIBIRCON.2010.5555344
- Narayan, S., & Shi, Y.** (2010, July). TCP/UDP network performance analysis of windows operating systems with IPv4 and IPv6. In *Proceedings of the 2nd IEEE International Conference on Signal Processing Systems (ICSPS)* (Vol. 2, p. 219-222). doi: 10.1109/ICSPS.2010.5555285
- Narayan, S., Gordon, M., Branks, C., & Fan, L.** (2010, July). VoIP network performance evaluation of operating systems with IPv4 and IPv6 network implementations. In *Proceedings of the 3rd IEEE International Conference on Computer Science and Information technology (ICCSIT)*. (Vol. 5, p. 669-673). doi: 10.1109/ICCSIT.2010.5564004
- Narayan, S., Gordon, M., Branks, C., & Fan, L.** (2010, June). Network performance evaluation of VoIP on windows desktop operating systems with IPv4 and IPv6 network implementations. In *Proceedings of the IEEE International Conference on Computer Design and Applications (ICCD)*. (Vol. 5, p. 393-397). doi: 10.1109/ICCD.2010.5540931
- Narayan, S., & Tauch, S.** (2010, June). Network performance evaluation of IPv4-v6 configured tunnel and 6to4 transition mechanisms on windows server operating systems. In *Proceedings of the IEEE International Conference on Computer Design and Applications (ICCD)*. (Vol. 5, p. 435-440). doi: 10.1109/ICCD.2010.5540939
- Narayan, S., Sodhi, S., Lutui, P., & Vijayakumar, K.** (2010, June). Network performance evaluation of routers in IPv4/IPv6 environment. In *Proceedings of the IEEE International Conference on Wireless Communications, Networking and Information Security (WCNIS)*. (p. 707-710). doi: 10.1109/WCINS.2010.5541871

---

**Narayan, S.,** Feng, T., Xu, X. & Ardham, S. (2009, Dec). Impact of wireless IEEE802.11n encryption methods on network performance of operating systems. In *Proceedings of the 2nd IEEE International Conference on Emerging Trends in Engineering and Technology (ICETET)*. (p. 1178-1183). doi: 10.1109/ICETET.2009.121

**Narayan, S.,** & Shi, Y. (2009, Dec). Application layer network performance analysis of IPv4 and IPv6 on windows operating systems. In *Proceedings of the 4th IEEE International Conference on Computers and Devices for Communication (CODEC)*. (p. 1-4).

**Narayan, S.,** Feng, T., Xu, X., & Ardham, S. (2009, April). Network performance evaluation of wireless IEEE802.11n encryption methods on windows vista and windows server 2008 operating systems. In *Proceedings of the IEEE/IFIP International Conference on Wireless and Optical Communications Networks (WOCN)*. (p. 1-5).

**Narayan, S.,** Brooking, K., & de Vere, S. (2009, April). Network performance analysis of VPN protocols: An empirical comparison on different operating systems. In *Proceedings of the IEEE International Conference on Networks Security, Wireless Communications and Trusted Computing (NSWCTC)*. (Vol. 1, p. 645-648). doi: 10.1109/NSWCTC.2009.367

**Narayan, S.,** Shang, P., & Fan, N. (2009, April). Network performance evaluation of internet protocols IPv4 and IPv6 on operating systems. In *Proceedings of the IEEE/IFIP International Conference on Wireless and Optical Communications Networks (WOCN)*. (p. 1-5). doi: 10.1109/WOCN.2009.5010548

**Narayan, S.,** Shang, P., & Fan, N. (2009, April). Performance evaluation of IPv4 and IPv6 on windows vista and linux ubuntu. In *Proceedings of the IEEE International Conference on Networks Security, Wireless Communications and Trusted Computing (NSWCTC)*. (Vol. 1, p. 653-656). doi: 10.1109/NSWCTC.2009.368

**Narayan, S.,** Kolahi, S., Sunarto, Y., Nguyen, D., & Mani, P. (2008, Dec). Performance comparison of IPv4 and IPv6 on various windows operating systems. In *Proceedings of the 11th IEEE International Conference on Computer and Information Technology (ICCI)*. (p. 663-668). doi: 10.1109/IC-CITECHN.2008.4803056

**Narayan, S.,** Kolahi, S.S., Brooking, K. & de Vere, S. (2008, Dec). Performance evaluation of virtual private network protocols in windows 2003 environment. In *Proceedings of the IEEE International Conference on Advanced Computer Theory and Engineering (ICACTE)*. (p. 69-73). doi: 10.1109/ICACTE.2008.187

Kolahi, S., **Narayan, S.,** Nguyen, D., Sunarto, Y., & Mani, P. (2008, July). The impact of wireless LAN security on performance of different windows operating systems. In *Proceedings of the IEEE Symposium on Computers and Communications (ISCC)*. (p. 260-264). doi: 10.1109/ISCC.2008.4625636

**Narayan, S.,** Kolahi, S., Sunarto, Y., Nguyen, D., & Mani, P. (2008, May). The influence of wireless 802.11g LAN encryption methods on throughput and round trip time for various windows operating systems. In *Proceedings of the 6th Annual IEEE communication networks and services research conference (CNSR)*. (p. 171-175). doi: 10.1109/CNSR.2008.60



## Chapter 1

---

### Introduction

Information technology networks are at the core of communication in societies today. With the advent of the World Wide Web, social networking, multimedia streaming and similar, the Internet has become a global phenomenon allowing seamless, global communication and information sharing. On a micro scale, there has also been explosive growth in the number of small-scale networks, such as intranets and extranets, mainly resulting from increased business activities and continuous changes to the nature of conducting businesses. The way consumers purchase goods is also changing remarkably, as global interconnectivity has given rise to borderless economies. The unprecedented increase in its scale of establishment and its complexities, irrespective of the size of the actual network, has led to several challenging research problems in improving and enhancing network performance.

The fundamental manner in which networks are created have also evolved dramatically. It all started as a small collaborative network with a limited number of nodes created by researchers at the US Defence Advanced Research Projects Agency. The networks of today are vastly different, but they still are based on that original distributed packet-switching technology which facilitates data transmission end-to-end on communication links. That same basic building block, originating in the early 1980's, continues to dominate the networking arena in the 21<sup>st</sup> century, however it incorporates a number of new, necessary technologies. Some key developments are mentioned herewith.

- The version of the Internet protocol that is used widely has limitations, and this has necessitated a new version. Although this solves a number of the fundamental problems of its predecessor, uptake of the new

version has been extremely slow. Also the two versions happen to be incompatible. This has given rise to a number of interim solutions, known as transition mechanisms, which facilitate co-existence of the two versions on one network. It is important to understand the implications for implementing these.

- Wireless based technology plays a dominant role in all networks today. It boosts productivity and facilitates increased mobility, leading to increased effectiveness and improved responsiveness in business environments. Wireless, at the implementation level, is excellent as it minimises cabling, allowing networks to be easily expanded/reconfigured cost effectively. However, most networks today are an amalgamation of both wired and wireless counterparts. The back-end of the majority of network infrastructures are generally purely wired, while the *last mile data delivery* to a client node is mostly wireless. Currently this combination works well since it allows incorporation of fast cabled data transfer technologies, like fibre optics, and new standard twisted pair cables with wireless technologies. This gives rise to wired-to-wireless type networks.
- There has been substantial development in maximizing network security in recent times due to a proliferation of cyber-crimes, as fundamental networking technologies are insecure by design. A network needs to be secure at all different levels of operation, and this can be achieved by securing all entities (software and hardware) at granular level. Implementing security enhancements on a network downgrades performance, therefore the correct choice of technique and optimal configuration are of the uttermost importance.
- With the emergence of technologies, there is now a plethora of different network traffic types traversing communication pathways. While they may be using the same fundamental communication protocols as traditional data types, networks nowadays are experiencing increased traffic. Voice, multimedia streaming, games traffic and similar all have to be appropriately prioritised so that they can coexist on cables, together with traditional data forms.

In order to move with the times many other aspects of networking have undergone both incremental and significant change since the early 1980s. The evolutionary nature of networks poses a number of challenges and questions, and of interest to the work in this thesis are the questions related to network performance, particularly in the context of communication protocols on infrastructures implemented with new technologies.

Network performance has always been a hot topic for both academics and practitioners. Its significance on a global scale is highlighted in Figure 1.1 (data date: December 2014), where it can be clearly seen that network performance in various parts of the world are different. This is indicated by dissimilar performance index values, where a higher value is indicative of a comparatively better performing network. To enhance network performance so that end-users get the best experience on information technology infrastructures, network practitioners and academia continually research, redesign and upgrade networks.

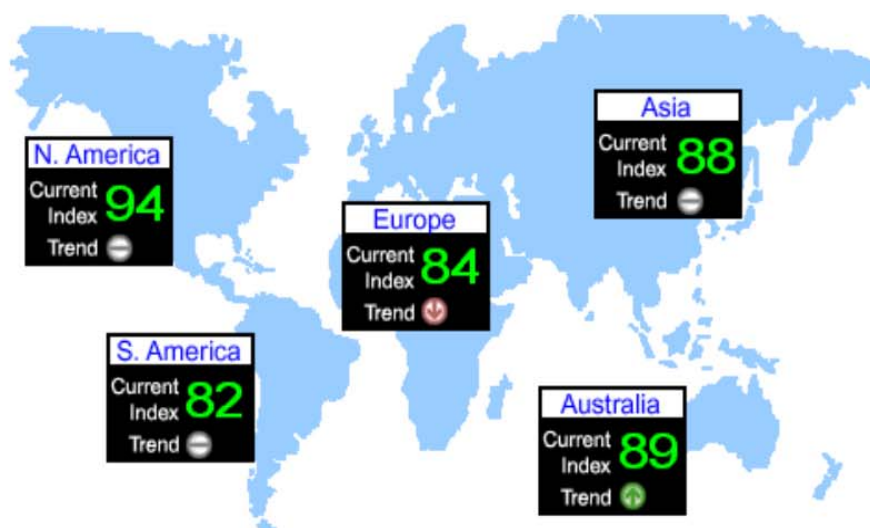


Figure 1.1: Global network performance index

Source: <http://www.internettrafficreport.com>

In this thesis, two commonly used transport layer protocols, the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP), have been evaluated for network performance in three different network scenarios, namely, networks with transition mechanisms, wireless based networks, and in the context of using virtual private network technologies as security protocols. In each of the three contexts, various networks have been implemented on test-beds and then performance related metrics were measured and analysed with a view to evaluating the network behaviour of the two protocols. It was envisaged that this undertaking would lead to a better understanding of the two protocols' network performance behaviours.

## 1.1 Thesis Objectives

The primary focus of this thesis is to evaluate network performance characteristics of transport layer protocols in three different scenarios. In each of the three contexts, various networks have been implemented to evaluate protocol behaviour. Thus the core objectives of this thesis are:

1. To exploit TCP and UDP traffic behaviour on networks that have various IPv4/IPv6 transition mechanisms implemented. Dual stack, translation and tunnelling mechanisms have been analysed with different transport layer data types to ascertain if there are any performance difference between some commonly used mechanisms.
2. To analyse TCP and UDP traffic behaviour on wireless networks implemented with various IEEE802.11 standards. Here the objective is to evaluate wireless environments configured with different wireless standards (IEEE802.11g,n,ac) and security protocols so that each environments network behaviours can be differentiated for the two traffic types.
3. To evaluate TCP and UDP network behaviour on networks with various virtual private network security protocols. There are a number of technologies that can be used to secure data in motion on a network. The degree by which they impact on network performance when transport layer traffic crosses through these implementations will be analysed.



4. To ascertain if the choice of operating systems influences the behaviour of TCP and UDP traffic types. The goal here is to empirically determine if the network operating system used has any effect of the network performance on traffic types.

5. To determine, based on different performance metrics, which specific technology, in each of the three different contexts, gives the best network performance. Subsequently this will lead to an ordinal ranking of the specific technologies that are commonly used within each context of focus in the thesis.

Accomplishing the above thesis objectives will enhance knowledge related to the two protocols' behaviour on networks. This will be very useful for professional practitioners who may be making decisions related to the choice of technology they are to implement on network infrastructures.

## 1.2 Thesis Contributions

This thesis is a culmination of work undertaken in the arena of network performance over a number of years. Subsequently, three journal articles and more than 20 IEEE international conference papers have been published (see *List of Publications*). The focus of each of these research publications has been on a different aspect of network performance evaluation, and details related to performance metrics in that particular context have been presented in those peer, blind-reviewed research outputs. The common theme in all publications is performance of TCP and UDP traffic types analysed on test-bed implementations.

Following on from that, the purpose of this doctoral undertaking is to bring together all those prior works into one cohesive thesis on network performance evaluation. The thesis will enhance professional practice knowledge in the area of network performance. As stated earlier, networks continue to evolve, and the contents of this thesis will enable network practitioners to make informed decisions when changing or incorporating technologies, such as transition mechanisms, wireless, or virtual private networks into the

infrastructure. The presented contents will be beneficial for those wanting to get the best performance out of a network infrastructure.

### 1.3 Thesis Structure

The thesis is organised as follows: Chapter 2 provides background information on key concepts relating to the topic and finishes off by highlighting other research work with a similar focus. In Chapter 3, the thesis methodology is presented with an emphasis on detailing the test-bed schematic employed. Data collection and analysis process have been outlined. Following that, Chapters 4, 5 and 6 contain intricacies related to the three network contexts namely, transition mechanisms, wireless and virtual private networks respectively. In each of these three chapters, the technical details related to the context are provided, followed by test-bed details specific to the technology. At the end of each of these chapters, empirical results from the test-bed analysis are presented. The results presented in Chapters 4, 5 and 6 are analysed and discussed in Chapter 7 collaboratively, and finally, conclusions related to network performance behaviour of TCP and UDP are drawn in Chapter 8. Thesis contributions have also been stated in the final chapters.

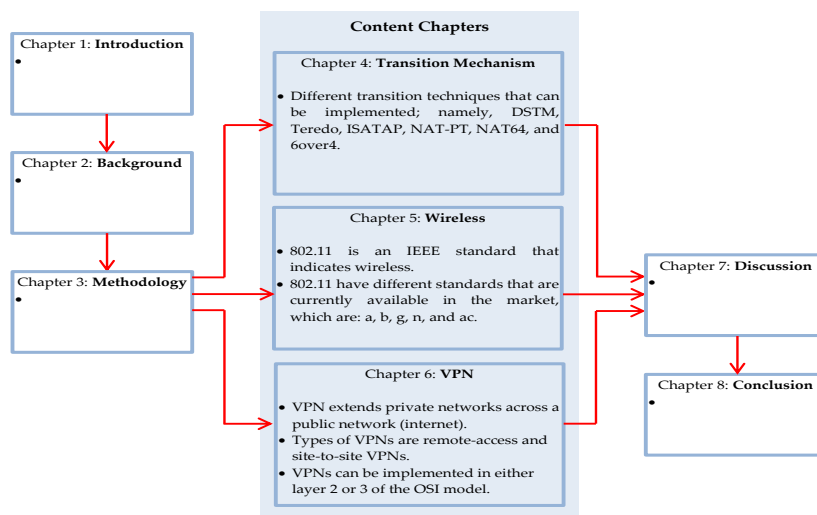


Figure 1.2: Thesis layout overview

### 2.1 Introduction and Motivations

The objective of this thesis is to investigate the network performance behaviour of TCP and UDP traffic types in various network scenarios, especially that of networks which are wireless, or have either transition mechanisms, or security enhancements related to data transmission. End-to-end data delivery was revolutionised by Cerf and Kahn (1974) in their landmark paper which detailed the TCP and outlined concepts related to encapsulation, datagrams, and functions of a gateway. By transferring the responsibility of error correction in communication from the Interface Message Processor (IMP) to the host machine, the Advance Research Project Agency Network (ARPANET) network became the focus for all communication development (Neigus, 1972). Then in 1977, a network consisting of ARPANET, packet radio and packet satellite was successfully demonstrated, thus giving rise to a new era in modern communication.

Thereafter, TCP/IP was split into two distinct protocols: TCP and Internetworking Protocol (IP) (Postel, 1981b). With this, split datagram routing was handled by IP, while TCP performed tasks like segmentation, reassembly, and error detection. The new internetworking protocol became TCP/IP, which was later incorporated into computer network systems like UNIX (K. Kim, Sung, & Lee, 1997). By 1981, TCP/IP was formalised and its inclusion as networking software in computer operating systems gave rise to data communication on desktop computers (Socolofsky & Kale, 1991). Eventually the original ARPANET protocol was abolished and TCP/IP became the standard for various networks, and the Internet. The protocol has been in use, and in continuous development, for more than three decades; what

started off as experimental communication technology has now become the powerhouse of the most used and complex network in the world.

In this chapter, background concepts related to common network protocols and network performance are discussed. In this context, the different models commonly used to compartmentalise and understand networks are first presented. This is followed by a discussion on the two communication protocols at the core of this thesis, their different characteristics are detailed, and finally the Internet protocol versions are discussed. Intertwined in these discussions, some key research undertaken in the area of network performance are mentioned. The actual context in which the protocols have been performance evaluated will be presented in later chapters. In summary, this chapter mainly presents the intricacies of TCP/IP important to know, in order to understand network performance.

## **2.2 A Layered Structure**

Networks and the Internet are extremely complicated systems, since there are numerous components working together to provide a platform for end-to-end communication. Applications, software, protocols, operating systems, switches, routers, link level media, and more, are all supposed to work together to seamlessly facilitate connectivity. Given this immense complexity, network architecture has been organised in layers. This aligns with the basic principles and philosophies of the Internet architecture design, which in summary states that complexity must be controlled if one hopes to efficiently scale a complex problem (Bush & Meyer, 2002). By doing so, functions of communication systems have been split into abstraction layers, giving rise to standardisation in data transmission. Functionality at each layer relies on the services offered by the layer directly below.

### 2.2.1 OSI Reference Model

There are two prevailing models that exist as the foundation for the layered structure in communication. The International Standards Organisation (ISO), established in 1947 and responsible for agreements on international standards, introduced the Open System Interconnection (OSI) model in the late 1970s (Forouzan, 2001). OSI enables one to understand and design a network architecture that is flexible, reliable, and interoperable, allowing end systems to communicate regardless of their underlying architecture (Staalhagen, 1996) (Day & Zimmermann, 1983). The second model, known as the TCP reference model (originally known as the Department of Defense (DoD) model) was developed prior to the OSI model. The TCP model has interactive modules, each providing a specific functionality in communication, although each module is not necessarily interdependent. However, in the OSI model, functions belonging to specific layers rely on the working of the other layers, while in the TCP model, since it uses relatively independent protocols as various layers, modules can be mixed and matched to suit the needs of the system. Nonetheless, both OSI and TCP models provide mechanisms by which complex heterogeneous environments can be sensibly segmented. The two models are detailed next.

#### The Physical Layer

The physical layer (layer 1) is mainly responsible for coordinating transmission of bit streams over the physical medium between the end nodes. It defines the rules that transmission media follow in relation to moving data bits (zeros and ones) on a circuit (Staalhagen, 1996). The protocols in that layer are link dependant, that is, it relies on the actual details of the transmission medium (i.e. fibre-optics, twisted-pair copper wire etc.) (Heller, Heindinger, Schneelee, Fischer, & Klose, 2010) (Xia, Bataineh, Hassoun, & Kryzak, 1999). Following are the concerns at this layer: characteristics of the interfaces between the devices and the transmission medium are defined; data bits are encoded into electrical or optical signals; the transmission rate is defined; sender and receiver clocks are synchronised; either a point-to-point or a multi-point link between the devices are established; and finally the direc-

tion of transmission between the devices (simplex, half-duplex, full-duplex) is defined.

### **The Data Link Layer**

The primary task of layer 2 is to ensure that the raw data received at the physical layer is transformed into a circuit that appears error free to the upper layers of the OSI model. This is done by masking the real errors, and is accomplished by breaking the sender data into data frames, and then transmitting the frames sequentially. This layer also adds a header to the frame to define the physical addresses of the sender and receiver nodes (Postel & Reynolds, 1988) (Staalhagen, 1996). The main responsibilities of the data link layer are: to divide the data stream into frames of manageable size; physical addressing of the frames to include sender and receiver addresses; imposing a flow control mechanism that will prevent overwhelming the receiver; adding reliability to data transmission by implementing a mechanism to detect and retransmit lost or damaged data; and controlling access on links where two or more devices are connected to the same link.

### **The Network Layer**

This layer is responsible for source to destination moving of network layer packets, known as datagrams, possibly across different networks. The data link layer is responsible for transportation within a network, while the network layer ensures that each packet reaches its final destination, even if it is across multiple networks (Hornig, 1984) (Staalhagen, 1996). So at this layer a header containing logical addresses of the sender/receiver is added to the packets coming from upper layers. On completing that, the datagram starts the process of being routed to the destination by determining the next network node the message should be sent to, so that the best possible route (depending on the implemented routing protocol) is found (R. Hong, 2011). The network layer contains the IP protocol and numerous routing protocols, however this layer is simply referred to as the IP layer.

### **The Transport Layer**

The fourth layer of the OSI mainly deals with end-to-end connectivity issues, such as determining the procedures that have to be followed for data entering and departing various networks. The previous network layer is responsible for end-to-end delivery of individual packets without realising the relationship between the packets - the transport layer, on the other hand, ensures that the entire message arrives at the destination intact, and in the correct sequence (R. Hong, 2011). That is, it is responsible for both error control and flow control for the entire transportation process (Staalhagen, 1996). This layer: adds a header with a service-point address (port address) allowing a network node to get the entire message to the correct process on a destination computer; segments a message into transmittable segments and appends a sequence number (later uses this information to reassemble the message); maintains a connection control mechanism; maintains flow control end-to-end; and finally, ensures error free (damage, loss or duplication) delivery by using retransmissions when necessary.

### **The Session Layer**

This layer is the network dialogue controller since it is responsible for initiating, maintaining, and terminating interactions between the communicating systems. The managing and structuring of sessions, during which session participants undertake activities, like logging onto equipment, file transfers and performing security checks, are all done at this layer. At the end of the communication process, the session is also terminated by mechanisms at this layer, however, if there is a premature session termination, in-built redundancies are able to re-establish the session (Staalhagen, 1996). In summary, the session layer allows the two systems to enter into a dialogue, and adds checkpoints (for synchronisation purposes) into the data stream.

### **The Presentation Layer**

The lower layers of the OSI, mentioned above, are mainly responsible for moving bits around a communication infrastructure - the presentation layer is concerned with the syntax and semantics of the information exchanged

between the communicating devices. It formats the data for presentation to the user so that different interfaces on various communications devices, and applications on computers, need not worry about data formats, that is, it is concerned with displaying, formatting, and editing user inputs and outputs (Staalhagen, 1996). Specific responsibilities of the presentation layer include: translating data so that the sender-dependant data format is changed into a common format, which at the receiving end is changed into a receiver-dependant format; data encryption, so that sensitive information can be carried across systems; and finally, data compression, where the number of bits contained in the information is reduced.

### **The Application Layer**

This is the upper most layer (layer 7) of the OSI reference model. At this layer, a variety of protocols that are commonly utilised by end users are presented, facilitating network access via services such as electronic mail, file transfers, database access, and similar (C. Nguyen, Vialatte, & Rieu, 1989). Other network-centric applications, such as network monitoring and network management facilities, are also available at this layer.

### **2.2.2 TCP Reference Model**

The OSI model has been used as the underpinning mechanism to explain network concepts for a long time; however, underpinning the OSI model is the reference model used in the ARPANET project. The TCP reference model, initially described by Carf and Kahn (1974), and later refined by Braden (1989) has only four layers (Renbo & Xiong, 2009), brief details of which are presented next.

### **The Link Layer**

The link layer is mainly concerned with the relationship between hosts and the transmission links. Early material relating to this model had little information related to this layer, however requirements at this layer led to the choice of packet-switched network technology based on concepts of a connectionless layer running across multiple networks. So, this layer describes



what links, like serial lines or Ethernet, must do to meet the requirements of the next layer up, the Internet Layer.

### **The Internet Layer**

In the TCP/IP reference model, the Internet layer corresponds closely to the network layer in the OSI model. This layer is crucial to the architecture of a network since it enables network hosts to initiate communication by injecting packets into any network, facilitating their travel to the destination, potentially via different network routes. Arriving at the destination, the higher layers in this architecture will rearrange and assemble the packets. That is, the Internet layer is responsible for delivering IP packets to any location that it is destined for.

### **The Transport Layer**

Like in the OSI model, the transport layer is designed to allow communicating source and destination nodes to carry on a conversation. To do this, there are two protocols defined at this layer: TCP and UDP. TCP is the connection-oriented protocol that enables error free data transmission, while UDP is the connectionless protocol that can be used by applications not requiring checks and balances commonly found in TCP type connections.

### **The Application Layer**

The application layer in the TCP/IP model is the highest layer, and in addition to containing the higher-level protocols required by a user of a network, it also includes session and presentation functions that exist on discreet levels in the OSI model. Incorporating the functions of all three layers into one is worthy, since session and presentation layers generally are of little use to most applications. In the application layer, there are numerous protocols that are typically used on networks, including the Terminal Network (TELNET), File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP), Hypertext Transfer Protocol (HTTP), Domain Name System (DNS) and the like.

### 2.2.3 OSI and TCP Reference Model Comparison

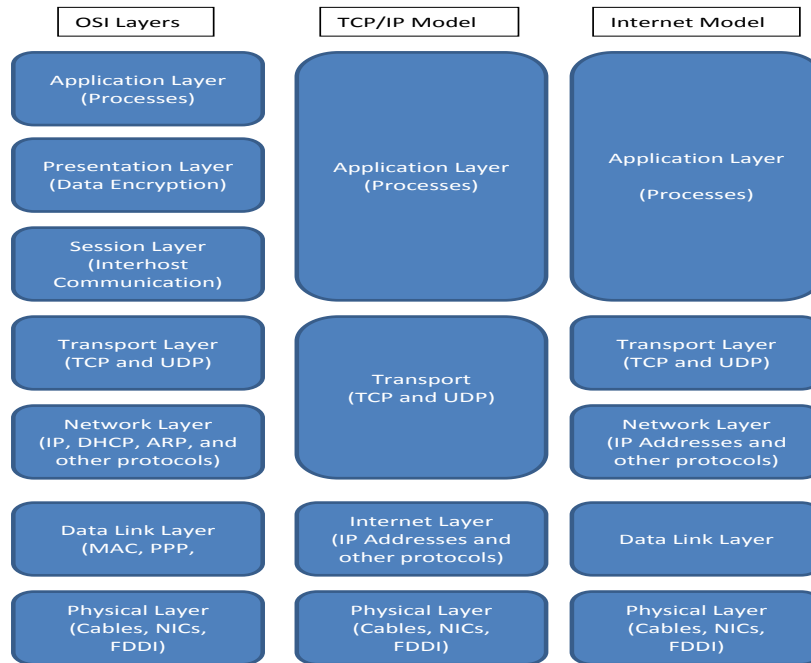


Figure 2.1: Different network models and their corresponding layers

The functionalities at the layers in both the OSI and the TCP reference models, as shown in Figure 2.1, are roughly similar. Layers up to, and including transport, provide end-to-end network independent transport services for communication, while layers above transport are application-oriented for the users. However, there are some key differences between the two models. First, the OSI model makes a very clear distinction between services, interfaces, and protocols, unlike the TCP model (Muskinja, Tovornik, & Terbuc, 2003). Here, services define the semantics of a layer, interfaces are related to informing the processes above the layer of what parameters are, in relation to interaction between the layers, and protocols relate to how a layer executes instructions within the layer to achieve desired goals. By separating the

three, OSI is able to fit in well with the modern concepts of object-oriented programming. However, in the TCP model, these boundaries between the services, interfaces, and protocols are blurred. Second, the OSI model is generic and not biased towards any particular routed protocol like TCP/IP. This is mainly due to the fact that the model itself was delineated well before the designing of any particular protocol. This was not the case for the TCP model: the model came first, followed by the protocols. From a design point of view, protocols like TCP/IP are a perfect fit into the TCP reference model. Thirdly, in relation to connectionless versus connection-orientated communication, the OSI model supports both at the network layer, but at the transport layer it only supports connection-oriented communication. However, the TCP model supports only connectionless at the network layer, but both at the transport layer - this matters since transport layer services are visible to the end user, so this gives TCP model users a choice. Finally, the lack of distinction between the physical and data link layer in the TCP model is of concern. The physical layer communication characteristics of copper wires, fibre optics, or wireless and data link layer issues related to start/end frames and desired degrees of reliability are distinct sets of characteristics that are only differentiated in the OSI model.

#### **2.2.4 The Internet Model**

Historically, OSI is the most common model used to describe network activities. However, a more simplified, five-layer model dominates current hardware and software designs. This model runs from a physical layer, up through the link, network and transport layers to the application layer. The details of each layer are discussed herewith.

##### **The Physical Layer**

Activities at this layer are link and media dependant. Here, individual bits within each frame, handed down from the link layer, are moved from one node to the next. The Internet model does not define any specific protocols at this layer, however, it supports all the standard and commonly used proprietary protocols.

### **The Link Layer**

At this layer, routing activities take place whereby datagrams can be moved through a series of routers between source and destination nodes. The services provided at this layer are dependant on specific link-layer protocols that are employed over the link. So if the specific protocols have details about reliable delivery of data, the link layer will ensure that such is achieved. Since data traverses several links in the path from source to destination, a datagram may be subject to multiple link-layer protocols at different links along the path.

### **Network Layer**

Here, datagrams are moved between hosts. The multiple links that exist in networks and between networks are combined at this layer so that data can be successfully sent between distant computers. Paths between the nodes are found and protocols, like the Address Resolution Protocol (ARP), Reverse Address Resolution Protocol (RARP), Internet Control Message Protocol (ICMP)(Postel, 1981a) and Internet Group Message Protocol (IGMP) (Fenner, 1997), also exist at this layer (Renbo & Xiong, 2009). The Internet's celebrated IP protocol is included here.

### **The Transport Layer**

The transport layer is responsible for transporting application-layer messages between the endpoints of applications used by the nodes. This is done using the two transport layer protocols, TCP and UDP, either of which is capable of transporting application-layer messages. This layer also strengthens the delivery guarantees of the Network Layer, and provides delivery abstractions, such as reliable byte streams to match the needs of different applications.

### **The Application Layer**

In OSI model application, the presentation and session layers activities are combined into one layer called the application layer in the Internet Model. Applications that make use of the network reside at this layer, some of which

are HTTP, SMTP and FTP. In addition to these, network specific services, such as DNS, are also present. Protocols used at the application layer may be distributed across multiple systems in the path of communication between the sender and receiver.

Application	Application Protocol	Transport Protocol	Port Number
File Transfer	FTP	TCP	20 and 21
Secured Shell	SSH	UDP/ TCP	22
Remote Terminal Access	Telnet	TCP	23
Electronic Mail	SMTP	TCP	25
Name Translation	DNS	Typically UDP	53
File Transfer	TFTP	UDP	69
Web	HTTP	TCP	80
Mail Transfer	POP	TCP	109 and 110
Web	HTTPS	TCP	443
Simple File Transfer	SFTP	TCP	115
Address Allocation	DHCP	UDP	67 and 68
	DHCPv6	UDP/ TCP	546 and 547
Streaming Multimedia	RTSP	UDP /TCP	554
Internet Telephony	Typically proprietary	UDP/TCP	
Network Management	SNMP	Typically UDP	161
Routing Protocol	RIP	Typically UDP	520

Table 2.1: Different applications for both TCP and UDP, and their corresponding port number(s)

The three reference models discussed earlier have multiple services and protocols at different layers. The five-layer Internet model will be used as a basis in this thesis, and of interest are some of the protocols at network and transport layer. In particular, the TCP and UDP protocols' network performance analysis is a common thread in this research. TCP and UDP protocols are commonly used by Internet applications [see Table 2.1] and are discussed next.

### 2.2.5 The UDP Protocol

The Internet model has both TCP and UDP at its transport layer, and of the two UDP is the simpler protocol. It is a connectionless (RFC 768)(Postel, 1980), unreliable protocol that provides a mechanism for applications to send encapsulated IP datagrams, without the need to establish a connection between the communicating nodes (P. Liu, Meng, Ye, & Gu, 2002) (Brownlee & Claffy, 2002). UDP does not have a flow control mechanism and lacks functions related to acknowledgement of received packets (Gopinath, Kumar, & Sharma, 2013) (R. Hong, 2011). Error checking is not a prevailing feature: if it detects an error, the packet is dropped silently. UDP can be described as a no-frills, bare-bones transport protocol.

A transport layer protocol has to provide a mechanism for connecting processes between the communicating nodes. Connecting processes entails establishing a connection between the sender and receiver, segmenting the packet stream into transportable units, numbering them, and then sending one by one, from sender to receiver. At the receiving end, the transport layer waits until all the different units common to a process arrives, checks them and then passes the error free entities to the process as a stream. However, UDP does not perform any of the activities mentioned here - it just receives a data stream for a process and delivers it unreliably. So UDP is the simplest of the transport protocols, providing process-to-process communication instead of a host-to-host complete process.

In spite of the drawbacks of UDP and its inherent disadvantages, it is a widely used communication protocol due to some of its integral advantages.

It is a simple protocol that has only 8 Bytes overhead per segment. If a sender needs to send information, and reliability is not a concern, UDP is the protocol of choice (Cai, Zhang, & Song, 2010). UDP will immediately package data inside a UDP segment and swiftly pass it to the network layer. Also the interaction between the sender and the receiver is at a bare minimum, that is, there are no preliminaries at the beginning of the process. UDP does not establish and maintain a connection state, which includes entities like receive and send buffers, congestion control mechanisms and acknowledgement and sequence number parameters.

The UDP segment structure was originally defined in RFC 768. It has a header of 8 bytes, fixed in size, which is followed by the payload (Renbo & Xiong, 2009). As shown in Figure 2.2, the source and destination nodes are identified by the ports, and when a packet arrives, the payload is transferred to the process attached to the destination port (Partridge & Pink, 1993). The port then delivers the embedded segment to the correct application. The source port field is copied to the incoming segment so that a reply can be sent via it to the communication origin is required (Akhtar & Siddiqui, 2011). The UDP length field can be a minimum of 8 bytes (size of the header) or up to 65515 bytes. Header checksum field is an optional field that can be used for adding extra reliability to the header information, or to the data itself.

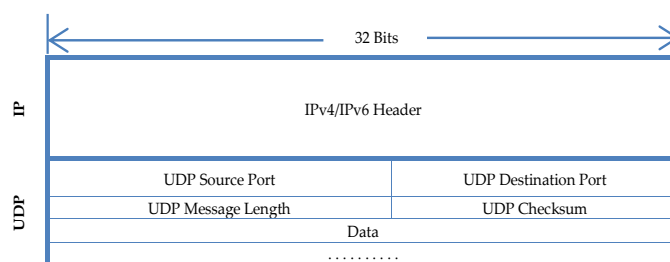


Figure 2.2: UDP packet structure

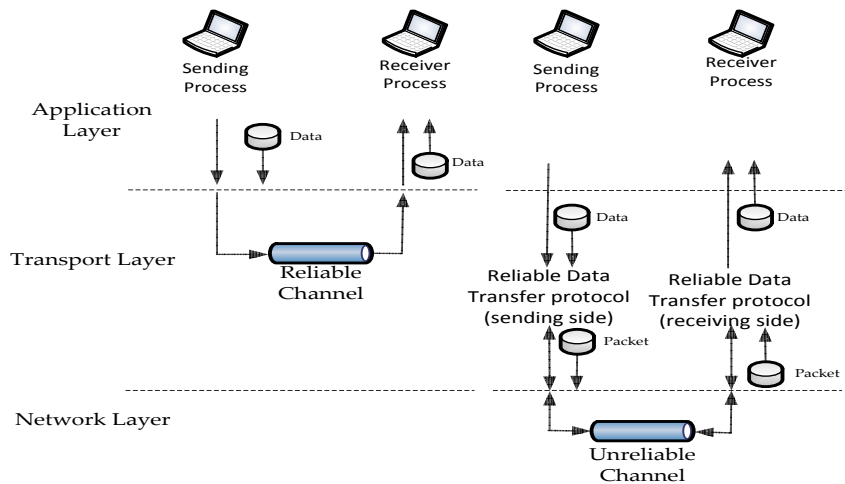


Figure 2.3: Reliable and unreliable data transfer structure

The UDP protocol is used in communication situations where error correction facilities are not needed, or where there is a need for a single short message exchange between applications. However, in most communication channels, reliable data transmission is warranted, where loss of even one single bit cannot be tolerated (i.e. transactions related to a customer's bank details). In such situations, TCP is the favoured protocol, as discussed next.

### 2.2.6 The TCP Protocol

The TCP protocol, unlike UDP, is a connection-oriented protocol that transfers data between the source and the destination reliably, traversing unreliable internetworks. In 1981, the US DoD standardised the protocol, prior to which there were nine earlier additions of ARPA TCP (R. Hong, 2011) (Brownlee & Claffy, 2002). The importance of the TCP protocol as the basis for today's communication infrastructure is highlighted by the number of subsequent RFCs written after RFC793 (Postel, 1981c), each suggesting improvements and fixes for errors and inconsistencies. The full collection of all RFCs related to the protocol has been road-mapped in a separate RFC document, RFC 4614 (Duke, Braden, Eddy, & Blanton, 2006). The TCP protocol



is dynamic, and capable of adapting to various properties of the communications infrastructure, whilst providing a robust environment for data transmission.

There are a number of distinctive TCP features. One of its key features is in its design, where it has been optimised for accurate and reliable delivery, rather than a fast and timely delivery. Consequently, TCP data transmission may be subjected to relatively long transmission delays due to corrections related to transmission errors (Caceres & Iftode, 1995) (T.-H. Nguyen, Park, Youn, & Jung, 2013) . Thus, TCP is not suitable for real time data transmission applications, such as Voice-over IP (VoIP). In addition, all TCP connections are full duplex, that is, data is transmitted in both directions during the communications process. TCP is also a point-to-point protocol, implying that it is a must that there are always only two end points in the communications process (for this reason the protocol is not suitable for applications that utilise multicasting and broadcasting mechanisms). The use of sockets plays an important role when communicating using the TCP protocol. The sockets, being the end points on the sending and the receiving nodes, are given socket numbers that can be used by multiple applications for connection purposes. When establishing sockets, socket numbers for each end node point are created, which consist of the IP address and a 16-bit local port number. Flow and congestion controls are also prevalent in TCP data transmission. Here, the transfer rates are managed by the receiver to ensure that data is reliably received by the sender (this is done by the receiver continuously hinting to the sender how much data it can receive in a given period).

When transmitting using the TCP protocol, each byte in the communication process has its own sequence number. These 32-bit numbers are carried, together with the packets, in both directions of the transmission and are used to ensure that all data in motion arrives at the destination intact (otherwise retransmission procedures are initiated) (Safa, Karam, Assi, & Mcheick, 2011). When it comes to transferring data between the sender and the receiver, TCP segments are used. These are made up of a fixed 20-byte header, which is normally followed by some optional parameters. Generally the size of the

entire segment will be governed by the amount of data it can carry, which is restricted by two parameters: each segment has to fit into a 65535 byte payload (header inclusive), and must fit in with the size of the Maximum Transmission Unit (MTU). The actual structure of the TCP packet is discussed next.

The TCP packet, as shown in Figure 2.4, has an IP header and the TCP header. In the TCP header, the first two fields are the port addresses of the source and destination nodes, similar to that in UDP packet structure (Ahmad, 2001). These two addresses are the end points of the communication channel to be established. Following this is the sequence and the acknowledgement numbers, where the former is related to data transfer direction and the latter is used in the reverse direction. Both these numbers relate to the position of the actual octet in the complete message stream identifying either the position of the first octet relative to the start of the data stream, or in the reverse data stream. Next, the options field is a variable length field that can accommodate headers if they are of variable lengths.

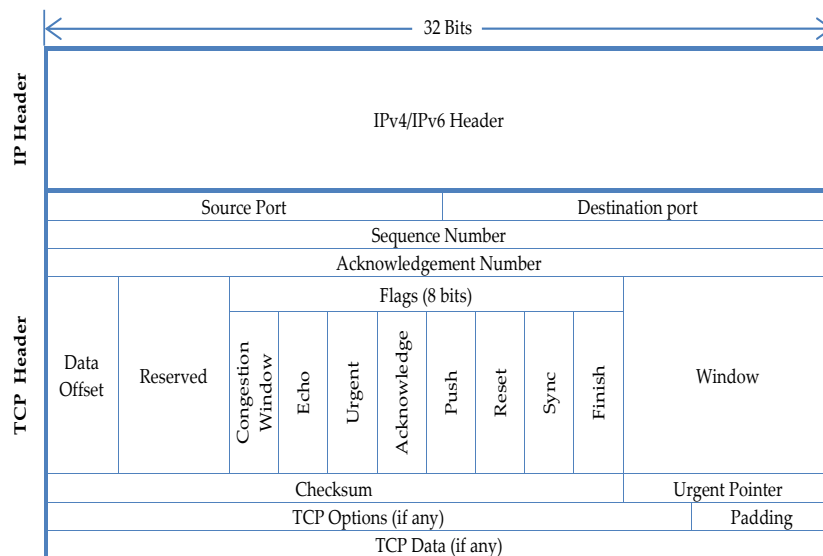


Figure 2.4: TCP packet structure

The eight 1-bit flags follow next. The congestion window is a flag set by the sender to indicate that it received a TCP segment and has responded with a congestion control mechanism (Ramakrishnan, Floyd, & Black, 2001), while the echo flag plays a dual role, depending on the value of the SYN flag (Fukushima & Goto, 1999). Both these flags use Explicit Congestion Notification (ECN) (Edwan, Guan, Oikonomou, & Phillips, 2010) (Ramakrishnan et al., 2001) in the process of controlling congestion on a network. The urgent field value dictates if the urgent pointer is being used, while the acknowledgement field indicates if this field is significant. Data pushed through to the application, activates the push field, indicating that upon arrival, the new data does not need to buffer, but can go directly to the application. Following this, the reset flag can be activated if the connection between the sender and the receiver needs to be reactivated, assisting the SYN bit with re-establishment of the connection. The final flag (finish) is used to indicate that no more data needs to be transmitted.

In TCP connections, the amount of data that can be sent before being acknowledged is controlled using a variable-sized sliding window. The window size field manages this using the window scale option, to allow senders and receivers to negotiate a window size. Following on, the checksum field adds extra reliability to both the header and the data, while the optional field can be used to provide extra facilities if required by the regular header. In every packet sent is a timestamp value, which is echoed by the receiver. A Selective Acknowledgement (SACK) option allows a receiver to tell the sender about the range of sequence numbers it has received (Leerujikul & Ahmed, 2001). As described in RFC 2018 and RFC 2883 (Mathis, Mahdavi, Floyd, & Romanow, 1996), (Floyd, Mahdavi, Mathis, & Podolsky, 2000), SACK enables the sender to determine what data the receiver already has, and what needs to be retransmitted (W. Xu, Xu, Wu, & Ou, 2011). The actual transfer of data occurs only after session parameters have been negotiated, as discussed next.

### TCP Connection Establishment and Release

At the beginning of the TCP data transmission process, a three-way message exchange procedure known as a three-way handshake is performed. As shown in Figure 2.5, the purpose of this procedure is to synchronise the connection ends by negotiating the parameters that will be used during the TCP session. The handshake mechanism also ensures that both sides are ready for the transmission process, and that all transmissions will occur only after a successful session establishment, and that transmission will stop at session termination.

The actual session is setup when the sending node sends a segment with a SYN flag, together with a proposed first sequence in the appropriate TCP header field ( $\text{syn}=X$ ) (Park, Lee, & Kim, 2003). On receiving this information, the receiver takes a note of what has been sent and then returns a segment with both the SYN and the ACK flags set with its own assigned values for the reverse direction, for both the sequence number ( $\text{syn}=Y$ ) and the acknowledgement field (Shakkottai, Srikant, Brownlee, Broido, & Others, 2004). This is sent together with  $\text{syn}=X+1$  and  $\text{Ack}=X+1$  to acknowledge that the initial values from the incoming direction have been received. On the sender receiving the reciprocated information, it takes note of the  $Y$  value and sends back an acknowledgement with the value  $Y+1$  (H. Zeng, Peng, Li, Xu, & Jin, 2009) (Cardwell, Savage, & Anderson, 2000). After successfully undertaking the three-way handshake, both sides in the communication channel are configured and can start independently sending data to each other.

At the end of the communication process, the session between the sender and receiver has to be terminated. This can be initiated by either of the communicating nodes by sending a TCP segment with a FIN bit, indicating that there is no more data left to be transmitted (Shukla & Brecht, 2006). On receiving the FIN bit, the other end shuts down the communication channel one way for any new data. However data may continue to flow in the opposite direction. This reverse direction can also be shut down in a similar way when a FIN packet is sent from the other node to its counterpart. So in total, four TCP segments have to be sent to terminate a session.

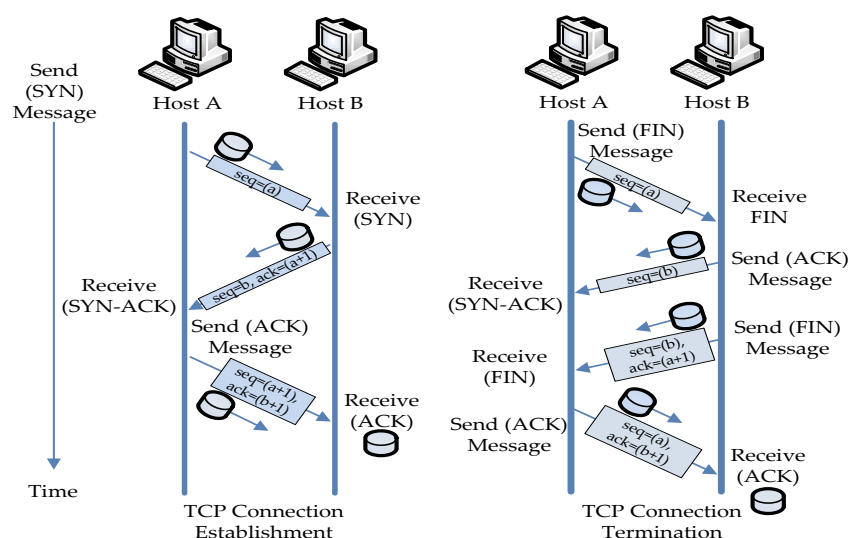


Figure 2.5: Three way handshake for connection establishment and termination

### Congestion Control

One of the prevalent features of a TCP protocol is its ability to control congestion. During data transmission, when there is more data than what the network can handle, congestion builds up, creating network queues and resulting in performance degradation and packet loss (Nagle, 1984). This is essentially due to a source node sending packets into the network unreservedly, resulting in the receiver reacting to the event (Forouzan, 2003). At this stage the network layer informs the transport layer of the situation, which reacts by reducing the transmission rate (Shah, Bilal, Khan, & Rehman, 2007). The role that TCP plays during data transmission is critical since it is the key protocol for controlling congestion on crowded networks.

To control congestion on a network, a number of algorithms and procedures have been designed, some inbuilt into the TCP and other related protocols. RFC 5681 (Allman, Paxson, & Blanton, 2009) outlines TCP's four intertwined algorithms: slow start, congestion avoidance, fast retransmit,

and fast recovery (X. Sun, 2012) (W. Xu et al., 2011). When these four algorithms are used together, they are known collectively as the Additive-Increase Multiplicative-Decrease (AMID) algorithm. The details of the four key algorithms used for controlling network congestions are presented here-with.

The slow start algorithm is used to control the amount of overloaded data being sent into a network. As discussed by Abed, Ismail, and Jumari (2011), it was designed primarily to control TCP's aggressive start-up behaviour, where the sender quickly flooded the communication channel to a maximum, which lead to data losses and buffer overflows, resulting in poor network performance.

Here, in the first round-trip time, only one packet is injected into the network by the sender, and on the destination acknowledging its receipt, two packets are sent in the next round, and then four, each after being acknowledged by the receiver. A slow start works well, irrespective of the communication channel's bandwidth and associated round trip times (Arpaci & Copeland, 2000). This is possible since its mechanism uses an ACK clock to match the sender's transmission rate to that of the communication link. Slow start algorithms can quite aggressively transfer data, resulting in a negative impact on network performance. This behaviour is controlled using a threshold parameter *ssthresh*, which works together with the size of another variable, known as *cwnd* (Khalifa & Trajkovic, 2004) (Abed et al., 2011). During transmission, when the *cwnd* value exceeds *ssthresh*, the sending node instigates a congestion avoidance algorithm.

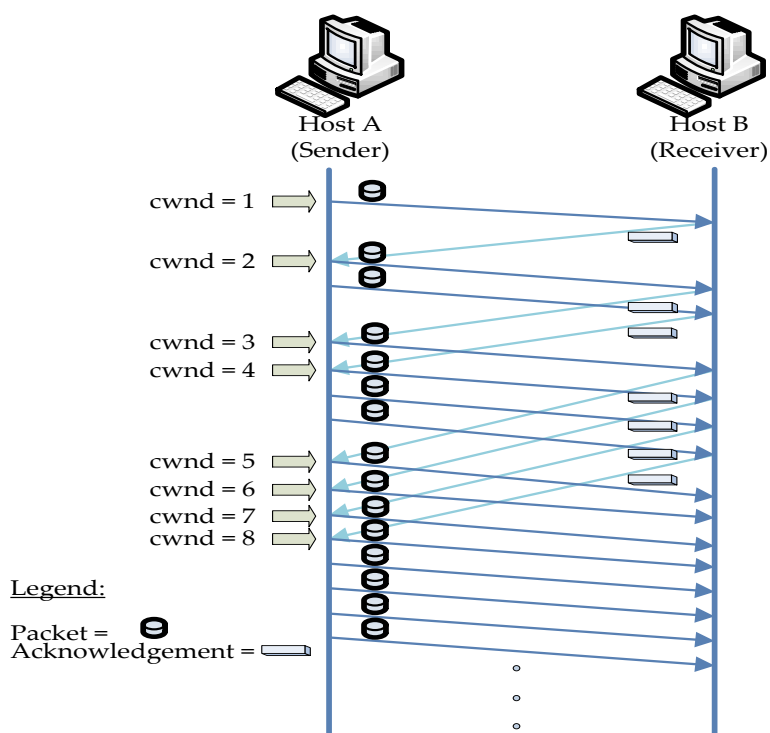


Figure 2.6: Representation of the slow start algorithm

The congestion avoidance algorithm is the primary mechanism for controlling congestion on the network. It is used by the TCP sender at a stage when the network tolerance has been exceeded. The algorithm maintains a steady state of data transmission by injecting new data streams into the network at a rate equivalent to its ACK's receiving rate (Arpaci & Copeland, 2000). Based on the premise that any communication channel can take a little bit more data, the congestion avoidance algorithm utilises any additional bandwidth along the path to maintain a stable data stream. When the algorithm is in action, the sender is able to dynamically adapt to any incidental changes in the network path condition. During congestion avoidance, the value of *cwnd* is increased by one segment per round trip time, on receiving every new ACK for data sent (Shah et al., 2007) (Alcock & Nelson, 2011). This incremental

method allows the sender to gently search for any available bandwidths that can be utilised whilst remaining fair to any other TCP sessions being used by the same network link.

When a data segment is lost during a TCP sender's transmission session, that segment has to be retransmitted. Fast retransmit is the specific mechanism that manages this, and attempts to reduce time that a sender has to wait before it can retransmit a lost segment (Qureshi, Othman, & Hamid, 2009). The basis on which fast retransmit decides how much to delay transmission prior to sending lost segments again is Duplicate Acknowledgement (DUPACK) (Karafillis, Fouli, ParandehGheibi, & Medard, 2013). On receiving three DUPACKs from the receiver (indicating lost segments), the sender immediately retransmits what appears to be the missing segments, without waiting for its timeout period (Waghmare, Parab, Nikose, & Bhosale, 2011). Each time the fast retransmit algorithm is invoked, one single data segment is sent, and immediately after retransmission is performed, the responsibility for further transmission is handed to the fast recovery algorithm.

In a situation where there is moderate congestion on a network, the fast recovery algorithm enhances data transmission rates, especially for large windows. The algorithm drops the congestion window to 1 each time network congestion is detected, and alleviates the problem by removing the slow-start phase (Saini & Dhaka, 2009). This occurs when the TCP sender is in receipt of DUPACKs, which informs that further packets have been lost (Stevens, 1997) and indicates there is still data flowing between the communicating nodes and that there is no need to reduce the flow abruptly using the slow start algorithm. The fast recovery algorithm is a major improvement to TCP that has been implemented since the release known as TCP Reno. The algorithm is normally used in conjunction with the Fast Retransmit algorithm.



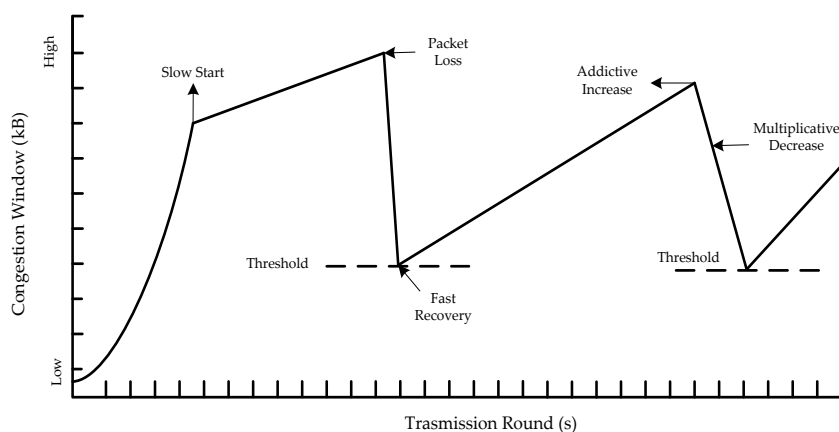


Figure 2.7: "Saw Tooth" pattern produced by TCP congestion algorithm

The congestion control mechanism improves network performance of the TCP protocol. A release known as TCP Tahoe uses the slow-start, congestion avoidance and fast retransmit algorithms together to maximise data transfer (Sikdar, Kalyanaraman, & Vastola, 2003). Tahoe works on the principle of packet conservation, whereby it does not need to inject any further packets into the data stream if it is running at the maximum available bandwidth. TCP Reno works on the same basic principles of Tahoe, however the way lost packets are detected is more sophisticated, ensuring that the data pipeline is not emptied each time packet loss occurs due to transmission (Waghmare et al., 2011)(Sharma & Tyagi, 2013). Later it was found that a TCP sender often had to wait for the timer to expire in order to recover from multiple packet loss - this leads to unnecessary delays, and instigated the release of SACK, which corrects this behaviour (Cong & Miki, 2000). However, at that time, many TCP hosts were not supporting its use, which led to the release of a modified Reno. TCP New-Reno (the successor of TCP Reno) has even better packet loss capability since it has the ability to detect multiple packet losses (Mathis et al., 1996). Thus in relation to TCP modifications, both TCP Reno with SACK and TCP-New Reno are plausible solutions for tackling the same congestion control problem.

The TCP header has several pieces of information that are vital to establishment and use of a communications channel. In relation to TCP performance issues, there are a few key fields, discussed here. At the initialisation of the connection, the *maximum-receive-segment-size* option is used to inform the destination of the maximum segment size (Postel, 1983) (Borman, 2012). This option sets the sizes of both the maximum receive segment size, and the size of the TCP window. So, this parameter will enable segments to pass across without the need for fragmentation, thus enhancing network performance. The *window-scale* option in the TCP header also plays a critical part in enhancing network performance, as it addresses the issue of maximum window size, adjusting it as per transmission requirements (Jacobson, Braden, & Borman, 1992). It allows the TCP sender to effectively change the window size so that more data can be held in flight depending on the network bandwidth and network delay parameters. These two parameters need to be negotiated at the start of the actual TCP session so that the entire session has the largest possible packet size (avoiding fragmentation) and so that the window size suits the bandwidth/delay attributes of the communication pathway.

TCP Algorithm	Year	Key feature(s)
TCP Tahoe	1988	Slow-start, collision avoidance, fast retransmit algorithms
TCP Reno	1993	Enhanced collision avoidance, fast recovery algorithm
TCP SACK	1996	Selective retransmission
TCP New Reno	2004	For TCP connections not able to use SACK.

Table 2.2: Different TCP algorithms

### 2.2.7 IP Header

In data transmission, to make a segment into a packet, the UDP, or the TCP header, is preceded by the IP header. The IP header prefixes information related to the IP version, source/destination IP addresses and other similar information that assist with finding the path from the source to the destination node. There are two versions of IP (and subsequently IP headers) currently being used: version 4 and version 6. Figure 2.8 provides an overview of their structure.

The IPv4 packet header is 20 bytes long and has 14 fields, of which one is optional. The first field (4 bits in length) has a value of 4, indicating the version of the header (Raicu & Zeadally, 2003) (Ahmad, 2001). The next field is the Internet header field (4 bits), which indicates the length of the IP header itself in 4-byte *words*, minimum value being 5, indicating a length of 20 bytes. The Type of Service (ToS) field is 8 bits long and is generally used in simple implementations of Quality of Service (QoS) (Almquist, 1992), providing the ability to effectively manage bandwidth and similar attributes. The total length field states the size of the total packet (in bytes), inclusive of the packet header. Since only 16 bits are available for this, the maximum value for this field is 65535, while the smallest is 20. The identification field information assists with reassembling a fragmented packet. Since each original packet has a unique identification value, which is the same for each fragment of the original, reassembly based on this information is possible. To assist with fragmented packet reassembly, the next three bits are various flags used in the process. Time To Live (TTL) is a value that governs when a packet should be dropped from the transmission pathway by counting the number of hops that a particular packet has taken, decrementing each step by one. Next is the 8 bit protocol field identifying the header that follows the IP header. The header checksum field value is used in calculations related to errors and identification of changes to the packet. The next two fields contain the source and the destination 32 bit addresses. These addresses may be changed if either of the nodes is operating in an environment where the Network Address Translation (NAT) is in use. The final field in the IP version 4 header is the *options* field - this is seldom used and can contain options that

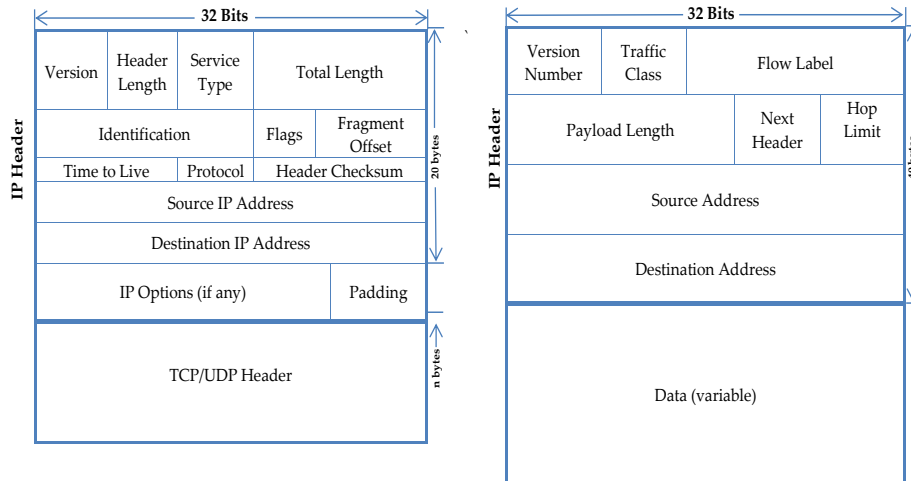


Figure 2.8: Structure of an IPv4 and IPv6 header

is related to security, routes, and time stamps.

The IPv6 header is a less complex version of its predecessor. There are a number of fields that have been eliminated, while some have been moved to the extension header (like the fragmentation and checksum fields), however the entire IPv6 header is of 40 bytes fixed length (Raicu & Zeadally, 2003) (Deering & Hinden, 1998). The increased header size is mainly due to the move away from 32 bit to 128 bit addressing (Halsall, 1996). This four fold increase in address size and accounting for every bit of IPv6 header undoubtedly has increased its size. Overall, the IPv6 header is comparatively simple, and is, in theory, much more efficient than its counterpart.

In the main, the fields in the IPv6 header replicate some of the functions of the IPv4 header (with some modifications). The version field, again, is 4 bits long and still specifies a value indicating IPv6 is being used (actual value:0110). The traffic class (8 bits) field is similar to the IPv4 service field and is capable of indicating traffic based on Differentiated Service Code Point (DSCP) (Nichols, Blake, Baker, & Black, 1998). Information in this field is also

used for congestion control purposes. Next is the flow label, 20 bit long field that allows tracking of specific traffic at the network layer of the OSI. Payload length indicates the length of the data portion, while next header indicates the next field that will follow, similar to IPv4 operation. Hop limit is similar to the TTL field in IPv4, where the maximum allowable number of hops for a particular packet are indicated. Finally, source and destination addresses (in 128 bit form) are stated.

### 2.2.8 Comparison of IPv4 and IPv6

IPv6 is the evolutionary successor of version 4 and has been designed to address some of the major shortcomings of IPv4 network environments. Some of these key features and how they alleviate IPv4 issues are discussed here-with.

#### Extended Address Space

With its hexadecimal address structure, IPv6 has approximately  $7.9 \times 10^{28}$  times more unique addresses than IPv4. The exponential growth of the Internet has brought a new generation of products that are based on an embedded IP reference model. This has rapidly depleted IP addresses, thus increasing IPv6 address space is necessary for providing Internet gateway to products like mobile platforms, hand-held devices, and similar.

In the IPv4 address space, use of Internet non-routable private IP addresses has been prevalent. These IP addresses, together with NAT and Port Address Translation (PAT), enable networks to connect to the Internet with the aid of minimal public IP addresses (Zhou, van Renesse, & Marsh, 2002). However, this type of access to the Internet has limitations. With IPv6, total end-to-end connectivity from source to destination is possible, thus eliminating the use of NAT and PAT type network translation technologies as a solution to extend the address space.

### **Efficient Routing**

Increased address space in IPv6 allows organisations and Internet Service Providers (ISPs) to advertise aggregated IP address prefixes. This is possible, as with increased addresses, organisations need to utilise only one prefix and as such they need to advertise just that. This leads to the creation of a hierarchical network and route aggregation. Subsequently, entries in routing tables on the Internets backbone routers dramatically shrink in size, resulting in increased routing efficiency.

### **Autoconfiguration**

IPv6 has inbuilt autoconfiguration capabilities. On large network sites, this simplifies network node configuration by automating the process. This is possible since stateless host automatic configuration is embedded in the new version and this simplifies the Dynamic Host Configuration Protocol (DHCP) configuration and administration (Narten, 1999). DHCPv6 is both a stateful and a stateless protocol and in the latter form, does not need a DHCP server on the network, since there is no state to maintain. As a network node boots up, a local-link address is automatically created for each IPv6 enabled interface (Mayer, Chan, Grillo, & Thomas, 2007). This occurs because the nodes are using Stateless Address Autoconfiguration (SLAAC) (Thomson, Narten, & Jinmei, 2007), which is a component of the Network Discovery Protocol (NDP) (Narten, Nordmark, Simpson, & Soliman, 2007) (Rafiee & Meinel, 2013a). Autoconfiguration simplifies network administration since devices that are plug and play capable can seamlessly integrate into any network (Rafiee & Meinel, 2013b). Further, should a need to change IP addresses network wide arise, it can be achieved effortlessly.

### **Neighbor Discovery**

Network nodes use NDP to determine the link-layer addresses for neighbours that reside on attached links, as specified in Narten et al. (2007). Using ICMP and solicited-node multicast addresses (Hinden & Deering, 2006) a node can determine its neighbours on the local network (the local link) (Narten, 1999). This IPv6 embedded technology, which has replaced IPv4s

ARP, also enables nodes to find neighbouring routers that can be utilised to forward traffic to some other networks (Xiaorong, Jun, & Shizhun, 2013) (Narten, 1999). The node keeps track of other active nodes (including routers) on the network, and searches for alternative routes should a need arise.

### **Improved Security**

There are a number of security advancements/benefits in IPv6 that are fundamentally non-existent in the older version. One of the first actions of most network attackers is to scan ports as a reconnaissance technique to gather as much information as possible about a victim network. With IPv4, this was easily achieved due to the availability of only a limited number of IP addresses (some studies estimate that the entire IPv4 based Internet can be scanned in approximately 10 hours). However, with IPv6, this time frame is dramatically increased, thus negating security risks using port scanning as the initial step. The use of Cryptographically Generated Address (CGA) also contributes to better security in IPv6. Here, a public signature key is bound to an IPv6 address allowing a user to provide a proof of ownership when using a specific IPv6 address (Alsadeh, Rafiee, & Meinel, 2012) (Rafiee & Meinel, 2013a) (Aura, 2005). This provides protection from IP spoofing and facilitates messages to be signed using the sender's private key (Shen, Lee, Sun, & Jiang, 2011). IP Security (IPSec) is a mechanism to secure data that is in motion between parts of a network, and its usage on IPv6 based networks is encouraged (Kent & Seo, 2005). Initially, IPSec usage with IPv6 was mandatory (Kent & Atkinson, 1998) (Goode, 1998), however as from December 2011, the recommendation of the Internet Engineering Task Force (IETF) has changed this from mandatory to recommended (Jankiewicz, Loughney, & Narten, 2011). Nonetheless, IPSec improves security by providing authenticity, integrity, confidentiality and access control to each IP packet traversing a network. Finally, using SEcure Neighbor Discovery (SEND) (Arkko, Kempf, Zill, & Nikander, 2005) (Rafiee & Meinel, 2013a), as opposed to just NDP, boosts security.

### 2.2.9 Exhaustion of IPv4 addresses

IPv4 addresses have already been depleted. In February 2011, the last batch of remaining IPv4 addresses was allocated by the Internet Corporation of Assigned Names and Numbers (ICANN) to the five Regional Internet Registries (RIRs) (Goth, 2012). This did not mean that there were no more IPv4 addresses after that date, or an absolute end to version 4, however it did mark a definitive point in time signalling the scarcity of IPv4 addresses. It was an indication that there was an urgent need to plan and implement strategies (that is, change over to IPv6) to continue business as usual, without expecting to receive additional IPv4 addresses (Taggart & Rudolph, 2009).

This report generated at 02-Oct-2014 08:09 UTC.

---

IANA Unallocated Address Pool		
Exhaustion: <b>03-Feb-2011</b>		
Projected RIR Address Pool		
Exhaustion Dates:		
RIR	Projected Exhaustion Date	Remaining Addresses in RIR Pool (/8s)
APNIC:	<b>19-Apr-2011</b> (actual)	0.8226
RIPE	<b>14-Sep-2012</b> (actual)	0.9758
NCC:	<b>10-Jun-2014</b> (actual)	0.2184
ARIN:	<b>15-Mar-2015</b>	0.6809
AFRINC:	<b>06-Jun-2019</b>	3.0116

---

Table 2.3: IPv4 address exhaustion statistics

Source: <http://www.potaroo.net/tools/ipv4/index.html>

In spite of the near exhaustion IPv4 addresses, the uptake of IPv6 has been slow (see Figure 2.9). An estimation of IPv6 usage globally has been conducted by a number of organisations and researchers (Asia Pacific Network Information Centre (APNIC), Cisco, Google etc.) (Courtney, 2012). In the main, these estimates are based on monitoring aspects of traffic to websites that are hosted on IPv6 networks, or have some IPv6 feature enabled. The search engine, Google, estimates approximately 4.5% of traffic to its site (as of 4<sup>th</sup> quarter, 2014) is from IPv6 network - this is based on DNS AAAA records. While most countries, in all regions of the world, are still early adopters of IPv6 (usage between 0 and 1%), Belgium and Germany are the leaders, with adoption rates of approximately 30% and 12% respectively.



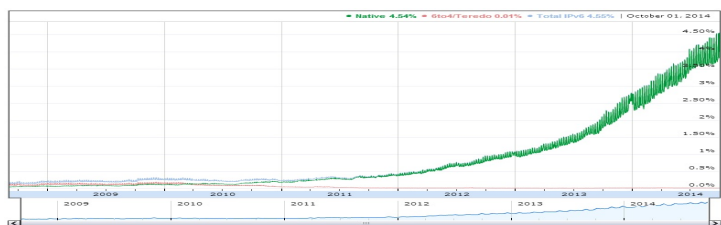


Figure 2.9: IPv6 address usage percentage per year

Source: <https://www.google.com/intl/en/ipv6/statistics.html>

### 2.2.10 IP Addressing

IPv4 addresses are 32 bits in size, grouped in four octets, each 8 bit long, and separated by dots, and normally written in dotted decimal notation. Each bit in the octet has a binary weighting, with the minimum being 0 and maximum being 255. In this representation, there are five classes (A,B,C,D,E), of which the first three are for commercial use (Rooney, 2011) (Postel, 1981b). As indicated in Table 2.4, classes D and E are for multicasting and experimental purposes respectively. The actual class of the address is determined by the value of the first octet, and associated with each class is a default subnet mask. In each class there are public and private IP addresses. Network nodes that are configured with public IP addresses are Internet capable, while private IP addresses cannot be routed via Internet routers.

Class	1 <sup>st</sup> Octet	Leading Bit	Default Subnets	Private Addresses	Number of Networks	Number of Hosts per Network
Class A	1 - 127	0	255.0.0.0	10.0.0.0 - 10.255.255.255	126	$256^3 - 2$
Class B	128 - 191	10	255.255.0.0	172.16.0.0 - 172.31.255.255	16,384	$256^2 - 2$
Class C	192 - 223	110	255.255.255.0	192.168.0.0 - 192.168.255.255	2,097,152	254
Class D	224 - 239	1110	Multicast		N/A	N/A
Class E	240 - 255	1111	Experimental and Research		N/A	N/A

Table 2.4: IPv4 classes and its range of addresses

There are a limited number of networks and hosts in each class of network. By default, class A has the least number of networks, while class C has the most, but class A has the most number of hosts while class C has the least. Using subnetting (classful and classless), a network can be divided into smaller networks (or subnets). This is possible since subnetting allows bits normally used by the host field to be designated for the network field. In situations where there is a need for several networks to be combined into a lesser number of networks, that is, to reduce the number of network IDs, supernetting can be implemented.

IPv6 uses 128 bit addressing, requiring eight 16 bit hexadecimal colon-delimited blocks. These addresses are not only longer, but they also incorporate letters in addition to numbers. IPv4 addresses can be divided into unicast, broadcast, or multicast, however, IPv6 classes are unicast, multicast, and anycast. Traditional IPv4 broadcast has been replaced with the use of multicast addressing. In each of the three mentioned groupings, the 128 bit addresses are logically divided, and rules have been established for associating the bit groups with special addressing features.

IPv6 Notation	Binary Prefix	Fraction of Address Space	Allocation
::1/128	00... 1 (128 bits)	1/8	Special Unicast (Loopback)
2000::/3	001	1/8	Global Unicast and Anycast
FC00::/7	1111 110	1/128	Unique Unicast and Anycast
FE80::/10	1111 1110 10	1/1024	Link Local Unicast

Table 2.5: IPv6 unicast and anycast addresses

The unicast address in IPv6 is similar to that in IPv4, where it is used for identifying a single interface. In this category there are four types of addresses: global, link local, unique local and special(see Table 2.5). Global unicast addresses are publicly routable and similar to IPv4 public addresses. The link local addresses are private, non-routable, and confined to a single network segment, so these can be utilised in setting up a small Local Area Network

(LAN) seamlessly. Unique local addresses are also meant for private addressing, but being unique allows multiple subnetted network segments to be joined without causing address conflicts. Special addresses, the fourth unicast type, are basically loopback addresses.

The equivalent of IPv4 broadcast addresses in IPv6 is multicast. With multicasting, any data sent to the multicast address is delivered to every interface in that particular group. The primary difference between this and traditional broadcast is that with the latter, data is sent to every single host on a particular segment, but with multicast only hosts that are part of a particular multicast group will be the recipients. Also IPv6 multicast is routable, however, routers that have the capability to forward multicast on a specific network have to be members of the multicast group. With anycast (Hartman, 2006) a single address is assigned to a number of network hosts, but the ultimate destination is the first available node in that particular set. In the IPv6 environment, this is particularly useful where there are Network Load Balancing (NLB), or an automatic fail-over implementation.

IPv6 Notation	Binary Prefix	Fraction of Address Space	Allocation
::FFFF/96	00... 1111 1111 1111 1111	1/8	Prefix for embedding IPv4 Addresses
2001:0000::/32	0010 0000 0000 0001 0000 0000 0000 0000	1/8	Teredo
2001:0DB8::/32	0010 0000 0000 0001 0000 1101 1011 1000	1/8	Nonroutable Addresses
2002::/16	0010 0000 0000 0010	1/8	6to4
3FFE::/16	0011 1111 1111 1110	1/8	6Bone
FF00::/8	1111 1111	1/256	Multicast

Table 2.6: IPv6 global and multicast addresses

### IPv6 Address Structure

The structure of the IPv6 address is very different to IPv4. IPv6 addresses are made up of three fields (Hinden & Deering, 2003). First is the global routing prefix; information in this field is used by network routers to send packets to local routers that service the networks specified in the global prefix of the version 6 addresses. The subnet ID field indicates subnets within the organisation and the interface ID designates the address of the source, or the intended recipients interface address. There is also the concept of IPv4-mapped IPv6 addresses. While these addresses are not routable on the Internet, they are useful when implementing IPv4-IPv6 translation mechanisms (discussed in Chapter 4). Use of an IPv4-mapped IPv6 address is discouraged within an IPv6 packet on a communications link. The actual format of the address is that the 32 bit IPv4 address is prefixed with 80 zeros and 16 one bits.

IPv6 Address form	Abbreviation rule
2001:0DB8:5F62:AB41:0000:0000:0000:0801	Original Format
2001:DB8:5F62:AB41:0:0:0:801	Leading zeros eliminated within nibbles.
2001:DB8:5F62:AB41::801	Double colon used to represent one or more consecutive sets of zero nibbles.

Table 2.7: IPv6 standards for abbreviating zeros

#### 2.2.11 TCP/UDP Related Research

A number of fundamental topics important to this thesis have been presented in this chapter and significant works of various authors have been highlighted. Since the focus of this thesis is on the network performance of the two protocols in selected network scenarios, research that emphasises performance issues related to the topic are mentioned herewith. However, network performance research pertaining to the three areas in which the two protocols have been performance evaluated will be mentioned in appropriate chapters later in this thesis.

There are numerous researches related to the TCP and UDP protocols. Their performance on various links is of interest: Benko, Malicsko, and Veres (2004), Meyer (1999), Othman, Zakaria, and Ab Hamid (2007), and Wennstrom, Brunstrom, and Rendon (2004) have evaluated it over a General Packet Radio Service (GPRS) network, while Tsiknas and Stamatelos (2012), Omprakash and Sabitha (2011), Martin, Li, Pressly, and Westall (2010), and D. Kim, Cai, Na, and Choi (2008) have researched them in the context of the Institute of Electrical and Electronics Engineer's (IEEE) 802.16 Worldwide Interoperability for Microwave Access (WiMax) technology. These researches have mainly focused on either measuring performance of the protocols on the specific link, or on suggesting some improvement to the actual protocol that enhances transmission attributes. Performance evaluation of various TCP variants has also been undertaken by many. Yue, Zhang, Ren, Li, and Zhong (2012), and Alrshah and Othman (2013) evaluated transport layer protocols on long distance links and high bandwidth connections, and established that the protocol is suited on such connections, and that the performance is acceptable.

Improving the performance of the protocols, especially TCP over multipath, has also been of interest. Multipath is an extension of the actual TCP protocol, where multiple flows can be derived from a single TCP session. Research in this area has mainly been in the application of this technique to different scenarios especially live streaming of data ((Bui & Zhu, 2007), (Jurca & Frossard, 2007), (Javed, Suchara, He, & Rexford, 2009), (D. Lee, Carpenter, & Brownlee, 2010)) and in wireless implementations ((X. Chen, Zhai, Wang, & Fang, 2004), (A. Singh, Xiang, Konsgen, & Goerg, 2013), (Lam, Chapin, & Chan, 2011)). To improve multipath performance in such scenarios, reducing high bit error or drop rate centric improvements to existing algorithms, or proposing new ones, is evident.

TCP protocol variances and their associated differences is of interest to a lot of researchers. Early research has mainly focused on TCP Vegas, with emphasis on either comparing attributes of it with other next generation variants (mainly TCP Reno), or analysing its network performance on a partic-

ular physical link type. Hengartner, Bolliger, and Gross (2000) specifically compared Vegas with Reno, while S. Xu, Saadawi, and Lee (2000), S. Xu and Saadawi (2001), Tandjaoui and Badache (2004) and Charalambous, Frost, and Evans (1999) compared multiple TCP variances with TCP Vegas. Most authors primarily found that a newer, improved TCP gives better performance than the predecessor, some stating improvements by almost 30%. In recent times, TCP variations in different scenarios, for example, with the Mobile Ad hoc Network (MANET) Sharma and Tyagi (2013), with QoS Varshini and Chaurasia (2010) and with mobile adhoc Bhanumathi and Dhanasekaran (2010), Ahmed, Zaidi, and Ahmed (2004) have been evaluated. Testing new releases of TCP performance on satellite links has also been of interest in research: Obata, Tamehiro, and Ishida (2011), Luglio, Sanadidi, Gerla, and Stepanek (2004) and F. Peng, Cardona, Shafiee, and Leung (2012) have analysed TCP Reno's performance on satellite communication links while Jacob, Srijiith, Duo, and Ananda (2002) did the same with SACK.

Research in the area of congestion control techniques is also evident in literature. While TCP congestion research is present ((Kotsiolis, Antonopoulos, & Koubias, 2010), (Nossier, 2004), (X. Sun, 2012)), recently, research in implementing UDP congestion control techniques has become a hotspot ((Chowdhury, Lahiry, & Hasan, 2009), (Sharma & Tyagi, 2013), (Shekhar & Ramanatha, 2010) and (Ullah & Khan, 2008)). In relation to UDP, end-to-end congestion control semantics of TCP techniques are being applied to the protocol, with the aim of improving overall reliability and throughput.

## Chapter 3

---

# Methodology

Measuring network performance has been an area of extensive research since the days of early networking. The types of networks, and their associated intricacies, have drastically changed over the years. Initially mainly research in the context of wired networks was predominant, then came wireless, and nowadays there is a focus on not just traditional networking, but also on contemporary technologies like enterprise networks, virtualisation and cloud computing. As shown in the previous chapter, researchers are also keen to experiment with performance issues at a more detailed level, evident by extensive research on the TCP protocol itself.

In relation to measuring network performance, there are a few approaches that can be taken. Like the networks themselves, the techniques used to evaluate network performance have also evolved over the years. The focus of this chapter is to highlight the techniques used by researchers in this research area. It will then outline the methodology employed in this thesis, leading onto detailing the test scenarios and the data collection process.

### 3.1 Testing Networks in Realistic Conditions

There are a number of methodologies commonly employed to test networks under varied conditions for entities like robustness, reliability, and performance. Each of the techniques has its own strengths and weaknesses, and they all aim to somehow represent the complexities of modern networks so that such systems can be analysed. An alternative to using these techniques is to perform tests on real networks, but this is scarcely done due to high costs and availability issues (Krop, Bredel, Hollick, & Steinmetz, 2007). Repeating exact conditions on real networks is a tall order, thus in situations where ex-

act experiments have to be repeatedly run, real networks may not be ideal (Judd & Steenkiste, 2005). Simulation, emulation and test-bed techniques are predominantly used for testing networks.

### 3.1.1 Simulation

Simulation techniques have always been the platform of choice for performing tests on various aspects of networks. They strive to accurately model and predict behaviour of real network environments in different scenarios. Simulation suites tend to offer an excellent set of features whilst providing a realistic representation of network components. This method of studying networks facilitates researchers to evaluate aspects of the infrastructure in a controlled manner, allowing them to seamlessly create different network topologies, incorporate operating system behaviours, send customised traffic patterns, change network scenarios and to collect data for analysis. Study using simulation solves problems related to repeatability, configurability, manageability, and modifiability (Judd & Steenkiste, 2005). Thus, simulators provide a rich environment yielding a number of benefits to researchers, like validation of network components, a platform to test new developments, and an opportunity to study a large network infrastructure.

A number of simulators are currently prevalent in the network performance research arena. NS-2/3 (Altman & Jimnez, 2012), OPNET (X. Chang, 1999), MATLAB (Ali, Abdulmaowjod, & Mohammed, 2010), J-SIM Insane (Sobeih et al., 2005), OMNeT++ (Wang, Liu, & Hu, 2005), GloMoSim (Ahvar & Fathy, 2007) and JiST/SWANS (Tippanagoudar, Mahgoub, & Badi, 2007) are some of the open source suites that can simulate discrete events on networks. NS-2 is commonly used for wired/wireless simulations, but its new release also supports networks like MANET. It has a C++ core engine and is resource hungry and platform dependent. NS-3, the successor of NS-2, addresses some of the drawbacks of NS-2, however the new version is still platform dependant and yet to realise its full potential. OpenSim is the other very commonly used simulator that allows a wide range of network protocols to be tested on networks. Commercial platforms, like OPNET, offer excellent GUI and help facilities, however, it comes at a huge cost (Imran, Said,



& Hasbullah, 2010). GLoMoSim is an example of a MANET simulator capable of multiplexing multiple simulated nodes. MATLAB suits network protocol development and analysis and has good visualisation capabilities. JiST/SWANS is a Java Virtual Machine based simulator which allows easy development of simulation models based on network entity concepts. Barr, Haas, and Van Renesse (2004) have shown that JiST/SWANS can outperform its counterparts, however of them all, this is the least popular simulator.

Network simulators have limitations. Some see simulators as the artificial offering of a synthetic environment (Göktürk, 2005) Simulations are generalised representations which lack details present in real systems (Heidemann et al., 2001), so results obtained from research using this may be skewed due to a high level of abstraction. Therefore, simulation based studies are generally used for qualitative purposes only. There are issues related to their accuracy and computational loading, which have led researchers to explore other methodologies that can be used to represent real network situations.

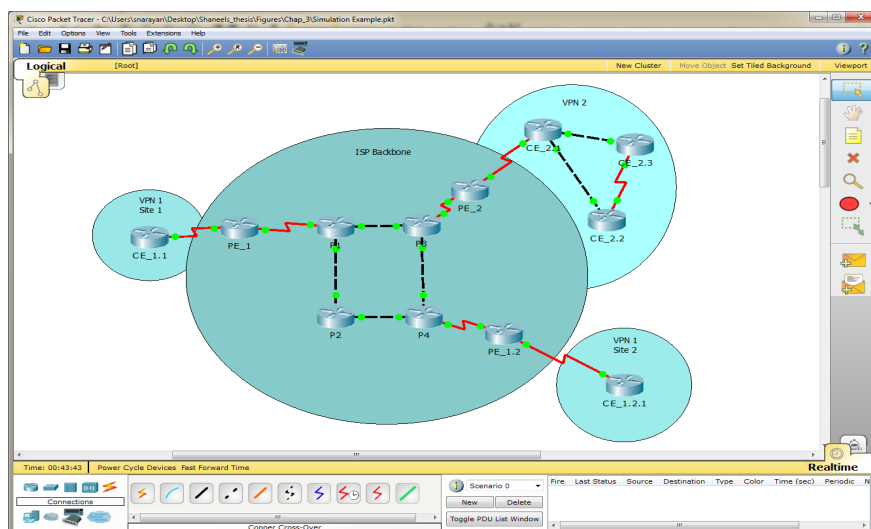


Figure 3.1: Sample diagram of simulation environment using Cisco packet tracer

### 3.1.2 Emulation

Emulation is a hybrid technique that combines hardware devices with some software whereby part of the network is implemented on hardware and the rest are simulated. Emulation based research has a network scenario represented by a combination of one or more surrogate systems (Göktürk, 2007) allowing one to conduct experiments using some real components and combining them with simulations. Emulation retains advantages offered by simulation (repeatability, manageability, etc) in addition to having the advantage of mitigating difficulties associated with realism. The emulation environment offers researchers the advantage of observing a real-world network scenario working in a controlled environment with a high degree of reproducibility.

Emulation techniques have numerous uses. They highly suit training environments such as medical training for surgeons, pilot training in avionics, or nuclear power plant personnel training (Göktürk, 2007). In network performance research, they provide a unique set of characteristics with respect to the degree of abstraction, accuracy, and scalability. NS-2/3 can be used for emulating network situations since they have an emulation interface which allows network traffic to pass between real-world networks and a simulated environment (Breslau et al., 2000). This provides a powerful platform to analyse network behaviour in end-to-end systems. MobiNet is a sophisticated environment that emulates a MANET, however, the setup is complicated and there are scalability issues (Krop et al., 2007). NEMAN is an emulator that can manage multiple network devices, where each device represents a virtual node in the emulation. NEMAN offers the added advantage of incorporating real software within the emulation. When emulators are used in performance analysis studies, they are highly prone to what is known as monitoring overhead problem (Göktürk, 2007) - this is where the hardware and software probes being utilised have an effect on the actual network being emulated.

### 3.1.3 Test-bed Networks

Simulation and emulation attempts to represent a real world network scenario, but the result is the creation of a system that reduces realism. Assumptions made regarding parameters of the actual network when setting up a simulation, or an emulation, over simplifies the network traffic experiences on a real physical network. The alternative is to research network performance on real-world networks, however, there are some serious issues in doing this. In addition to accessibility issues, there are significant repeatability and control issues. This is mainly due to the behaviour at the physical layer being tightly coupled with the physical environment and conditions under which the actual research is being conducted (Judd & Steenkiste, 2005). Also, with simulations and emulations, assumptions and simplifications may lead to inaccurate final results (Imran et al., 2010). To solve these problems, physical test-beds strive to bridge the gap between simulation and real deployment. Test-beds are hardware based, so depending on their size and sophistication, can be a pricey alternative to software environment based testing. A high degree of configuration may also be required during the setup phase and there will be a need to continuously monitor and maintain the environment during its usage phase. Depending on its size, provisions for remote monitoring may also be necessary.

In relation to network performance, there are a number of large scale test-beds developed specifically for this purpose. They provide environments that can be accessed remotely to run and analyse network scenarios; some common ones are MoteLab (Werner-Allen, Swieskowski, & Welsh, 2005), SensorScope (Barrenetxea et al., 2008), Emulab (Johnson et al., 2006), Signet-Lab (Crepaldi et al., 2007), GNOMES (Welsh, Fish, & Frantz, 2003) and ORBIT (Raychaudhuri et al., 2005). Use of some of these comes at a cost, while some (like Emulab) are publicly available for free.

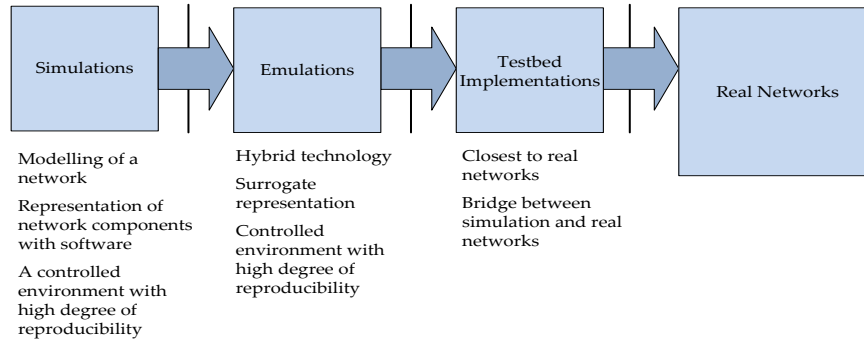


Figure 3.2: Diagram of different experimentation approaches

### 3.2 Experimental Test-bed Architecture

The objective of this thesis is to evaluate the network performance of TCP and UDP traffic types in different network scenarios. These protocols were originally designed for end-to-end data transportation using simple protocols on fixed cables, on networks purely dedicated for this purpose. Nowadays the situation is different, where networks are more complex due to entities like different IP versions and traffic types, diversification in physical communications media with the dominance of wireless technology, and introduction of additional protocols related to data security. As a result of these added complexities (and more), it is necessary to gain accurate insights into the performance of the two commonly used protocols in an heterogeneous environment, as data traverses end-to-end during a communication process. To undertake this study, the approach taken in this thesis was to utilise test-beds.

The use of a test-bed in network performance evaluations is widely evident in research. Actual test-bed centric network performance research will be detailed in subsequent chapters, contextualised to network scenarios in which TCP and UDP have been performance analysed. Here, however, is a sample of researches that have employed a methodology similar to that in this thesis. Taank (2008) evaluated end-to-end behaviour of the TCP protocol on a wired-to-wireless network. In this, a test-bed was employed, and evalua-

tions of TCP senders over IEEE802.11 Wireless LANs (WLANs) were carried out. Network performance of various operating systems is also a common theme in test-bed based research. Göktürk (2007) has evaluated the network performance of various operating systems, with the aim to compare the behaviour of one operating system with another. Test-beds with routers configured with other networking devices representing real-world scenarios have also been completed by Michalski (2012).

In the above mentioned researches, the idea was to evaluate network performance end-to-end without delving into the details of the performance of individual network components in the path of data transmission. Whilst there are numerous researches that do evaluate network behaviour at a fine-grained, component level, (most concentrate on a selected few attributes in the communications pathway).

The aim of this thesis is to evaluate the performance of the selected protocol holistically, in the context of the environment that it is implemented in. In summary then, this thesis explores end-to-end performance issues when data is transmitted using TCP or UDP via three different scenarios on a network implemented on an in-house test-bed. This way, the data gathered from the analysis is closer to performing a real-world test, rather than what would be attained from other traditional methodologies.

### 3.2.1 Network Schematic

The schematic of the test-beds used in this thesis is shown in Figure 3.3. Two network nodes, shown at the far ends of the diagram, act as senders and receivers of the TCP or UDP traffic that traverses the actual network housed in the middle block. The aim is to retain this configuration of sender connected to a middle block, with the receiver at the other end, whilst only making changes in the middle block. Changes made in the middle block will be to the network components and configurations, so that the different network scenarios in which the two protocols are being tested can be implemented on the test-bed. The intricate details of the network in the block will be revealed in the subsequent three chapters of this thesis, however details of the

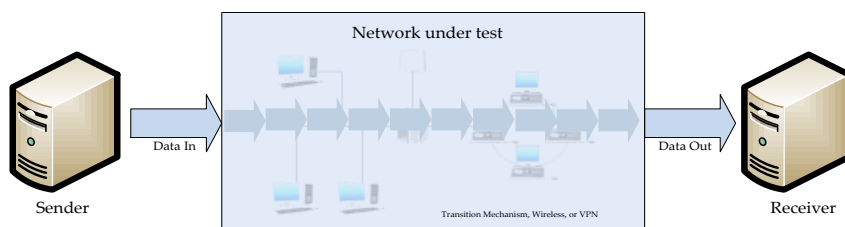


Figure 3.3: Test-bed schematic

sending and receiving nodes will be discussed here. The purpose of the two end nodes in the test-bed is to act as the sender and receiver of different types of data traversing the test network.

They will also be utilised in measuring performance related attributes of the network under test. In such a test, the aim is to flood the transmission path with data and to measure network performance related metrics. This can be conveniently done by employing one of a number of off-the-shelve traffic generation software.

### 3.2.2 Traffic Generation Software

Traffic generation software are widely used on test-beds for network performance evaluations. Traffic generation tools have been evaluated for functionality, purpose and features in a number of similar studies (Narayan, n.d; Srivastava et al., 2014a,2014b). The tools were implemented on a test-bed and each tested under various conditions to ascertain their key differentiators, some of the results obtained are presented in Table 3.1. Here it is seen that although most of the tools basically perform the same task, facilitating network performance testing by flooding various data types on networks and then measuring metrics, their features and capabilities are highly varied. At one end of the spectrum some tools are fully graphical and have an interactive GUI, while others use CLI with a different set of features.

Features	D-ITG	Netperf	Iperf	PRTG	Tamosoft	Net Stress	Calasoft Capsa	IP Traffic Test and Measure
Evaluated Version	2.8.1	2.6.0	2.0.5-2	13.4.6	1.0	2.0.9686	7.7.2	2.6.9
Latest Released Date	07/2013	06/2012	07/2010	10/2013	07/2013	01/2011	11/2013	04/2011
Supported Platform	Linux, Windows	Windows, Linux	Windows, Linux	Windows	Windows, Linux	Windows	Windows	Windows
Network Protocol	IPv4, IPv6	IPv4, IPv6	IPv4, IPv6	IPv4, IPv6	IPv4, IPv6	IPv4, IPv6	IPv4, IPv6	IPv4, IPv6
Transport Protocol	TCP, UDP	TCP, UDP	TCP, UDP	TCP, UDP	TCP, UDP	TCP, UDP	TCP, UDP	TCP, UDP, ICMP
User Interface	CLI, GUI	CLI	CLI	GUI	GUI	GUI	GUI	GUI
Open Source	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
Cost	Free	Free	Free	US \$2700/- year	Free	Free	US \$995/- year	Trial Version
Reported Metrics	Throughput, Jitter and Delay	Throughput	Throughput	Throughput	Throughput, Loss or RTT	Throughput	--	Throughput, Delay
Measurement Metric	Per hop Capacity	Achievable TCP Throughput	Achievable TCP Throughput	End-to-End Capacity	End-to-End Capacity	Per-hop Capacity	End-to-End Capacity	Per hop Capacity
Sync Clock	Yes	No	No	No	No	No	No	Yes (optional)

Table 3.1: Comparison of various network traffic generation tools

### 3.2.3 Traffic Generator

The primary traffic generation tool used in this research is D-ITG. In addition to generating popular TCP/UDP traffic, D-ITG emulates Telnet, VoIP (three different CODECS), DNS and various games traffic (Botta, Dainott, & Pescapè, 2012). Traffic generated with this tool supports probability distributions such as Pareto, Exponential, Poisson and Gamma distributions (Kolahi, Narayan, Nguyen, & Sunarto, 2011). D-ITG, on generating network traffic, is able to measure common performance metrics and output the results to a coded file. This can ultimately be decoded into a text file, results from which can be extracted into a spreadsheet for data analysis purposes.

D-ITG has the facility to incorporate a log server during network tests. This server is optional, however, its use really simplifies file management. The D-ITG Log server, as shown in 3.4, runs on a different host than the sender and

receiver, and receives and stores log information from multiple senders and receivers (Avallone, Guadagno, Emma, Pescape, & Ventre, 2004). The sender component of D-ITG can operate in three different modes (single flow, multiple flow or in daemon), and in all tests in this thesis, single flow mode was preferred. The receiver works concurrently with the sender, which ultimately listens and receives data flow that traverses the network under test. A manager node can also be implemented that can serve as a control station for all tests - this was not used in any of the test-bed analyses in this thesis as it was not deemed necessary to do so. Instead, the sender and receiver nodes were manually handled. D-ITG is a widely used network traffic generation tool used in research and engineering applications. It is a simple-to-use CLI application that seamlessly tests a network for various traffic types and output results as log files. Its developers (Botta et al., 2012) have listed approximately 150 research and engineering undertakings that it has been utilised in.

### 3.3 Performance Metrics

There are a number of parameters that can be measured on a network to evaluate its network performance. Irrespective of the type of setup they are measured on (simulation, emulation, or a test-bed), these measurements, known as performance metrics, give quantifiable values that allows one to gauge a particular network (Bradner & McQuaid, 1999). Some common performance metrics are discussed herewith.

**Bandwidth and Throughput:** Bandwidth refers to a channel capacity in a network communication pathway. Measured in Megabits/second, it is also known as the net bit rate. This is the maximum amount of data that can theoretically travel on a communication pathway in a given time, however, in practice, the maximum transfer rate will be lower. The actual achieved rate of successful data transfer is referred to as throughput (FitzGerald, 2011). Throughput on a network is controlled by the available bandwidth, signal-to-noise ratio, and the available hardware and software. In summary, bandwidth is the theoretical limit, while throughput is the practical limit of data



transfer on a network channel.

**Latency, Delay and Round Trip Time:** Latency and delay are synonyms describing the amount of time it takes for data bits to travel between two network nodes (Bradner, 1991). This is generally indicative of the time it takes for data to travel one-way between the sender and the receiver, commonly measured in milliseconds. Delay is generally caused by the distance between the nodes, other network equipment in the pathway, error and error rectification, congestion, queuing, and the data processing capabilities of the nodes involved. Round trip time is another such measure, however, this is more of a two-way measurement since its measurement involves only one node (Constantine, Forget, Geib, & Schrage, 2011). However, round trip time measurements generally exclude the processing time at the destination node.

**Jitter:** Jitter is closely related to the previous metrics and describes the variation in delay in receiving packets (Demichelis & Chimento, 2002). It indicates the time difference in packet inter-arrival time and generally is not a major issue on data networks (TCP/IP protocol can counteract its effects). However, in relation to VoIP and multimedia applications, jitter can play a major role in communication quality. Jitter is normally caused by network congestion, data queuing issues, and configuration issues.

To ascertain the network performance characteristics of TCP and UDP, throughput, delay, and jitter were measured on different network scenarios. How this was done on the test-bed, and the use of D-ITG is detailed next.

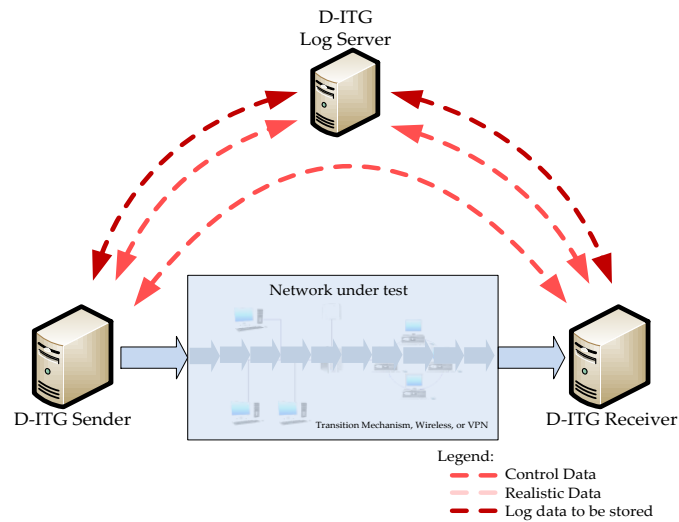
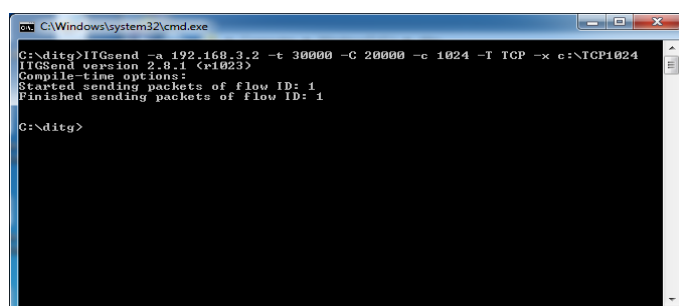


Figure 3.4: D-ITG mechanism with log server

### 3.4 Test-bed Data Collection

Figure 3.4 shows the generic form of the network that was setup to test performance of the TCP and UDP traffic types. Here it is seen that the D-ITG traffic generator has been incorporated into the test-bed, with the rectangle depicting the actual test-bed. The details of this rectangle are the subject of the next three subsequent chapters, however, the manner in which D-ITG has been incorporated into the test-bed, in principle, is the same in all test cases. So on configuring the complete setup of a network in the rectangle, the D-ITG sender, receiver, and log server nodes were connected to the test-bed in a manner ensuring that data really flowed from the sender to the receiver, traversing the network under test. Once the connectivity of the entire setup was positively tested, D-ITG traffic generation was initiated and performance metrics measured. The command to send emulated traffic types from the sender to receiver is shown in Figure 3.5. Here it is seen that TCP emulated data was sent for 30 seconds ( $t=30000$ ), at a rate of 20000 packets per second ( $C=20000$ ), with packet size set at 1024 Bytes ( $c=1024$ ). In such tests it is necessary to ensure that the sample size is large enough

(Tanenbaum & Wetherall, 2011) to ensure that the links are tested to capacity. The parameters chosen in the D-ITG command (t and C values especially) ensure that the emulated traffic really floods the sender/receiver link. This is necessary in ascertaining performance metric like throughput. For each net-



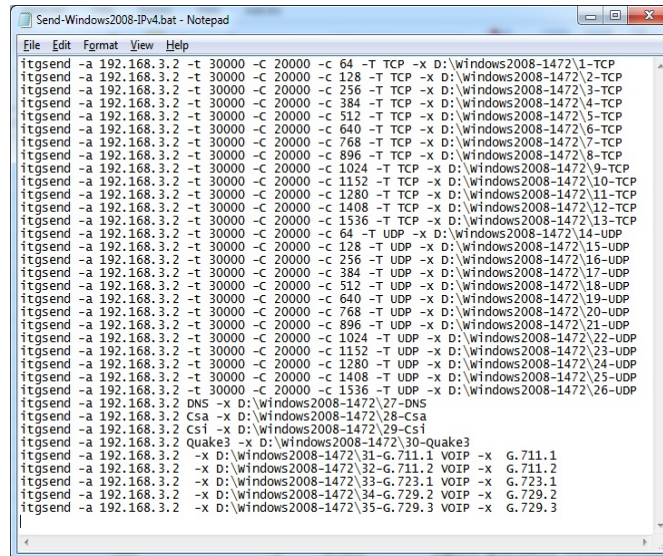
```
C:\Windows\system32\cmd.exe
C:\d\itg>ITGsend -s 192.168.3.2 -t 30000 -C 20000 -c 1024 -I TCP -x c:\TCP1024
ITGSend version 2.8.1 (<1023>)
Compile-time options:
Started sending packets of flow ID: 1
Finished sending packets of flow ID: 1

C:\d\itg>
```

Figure 3.5: D-ITG sender command line interface

work scenario, both TCP and UDP traffic types were emulated and tested. This was done for a number of packet sizes ranging from 64 to 1536 Bytes. To ensure that all these tests were done efficiently, a batch file was created as shown in Figure 3.6. This batch file was a handy automation tool in most scenarios, however, in tests that experienced a high drop rate (especially on wireless networks), each line in the test file had to be run manually.

The output file resulting from each run of the test is a D-ITG coded file saved on the log server. In order to make sense of the contents of this, a decoding process is necessary, the syntax of which is shown in Figure 3.7. Consequently, a text file is produced (Figure 3.8) that contains performance related metrics in readable form. For the purposes of data analysis and manipulation, contents of each decoded text file was transferred into a spreadsheet.

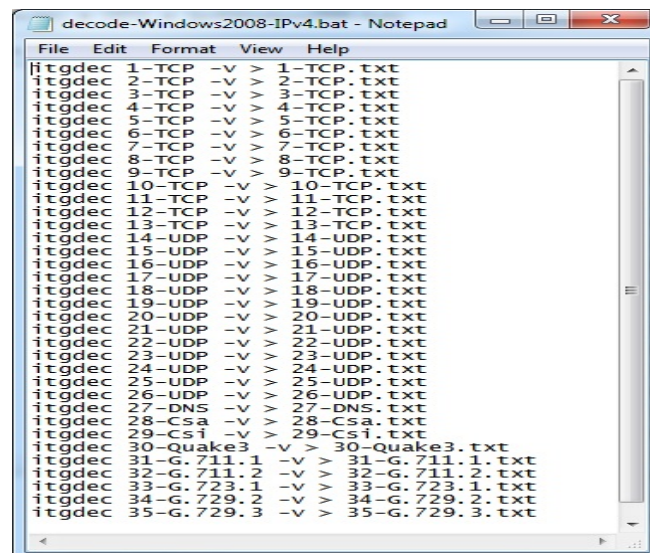


```

Send-Windows2008-IPv4.bat - Notepad
File Edit Format View Help
itgsend -a 192.168.3.2 -t 30000 -c 20000 -c 64 -T TCP -x D:\windows2008-1472\1-TCP
itgsend -a 192.168.3.2 -t 30000 -c 20000 -c 128 -T TCP -x D:\windows2008-1472\2-TCP
itgsend -a 192.168.3.2 -t 30000 -c 20000 -c 256 -T TCP -x D:\windows2008-1472\3-TCP
itgsend -a 192.168.3.2 -t 30000 -c 20000 -c 384 -T TCP -x D:\windows2008-1472\4-TCP
itgsend -a 192.168.3.2 -t 30000 -c 20000 -c 512 -T TCP -x D:\windows2008-1472\5-TCP
itgsend -a 192.168.3.2 -t 30000 -c 20000 -c 640 -T TCP -x D:\windows2008-1472\6-TCP
itgsend -a 192.168.3.2 -t 30000 -c 20000 -c 768 -T TCP -x D:\windows2008-1472\7-TCP
itgsend -a 192.168.3.2 -t 30000 -c 20000 -c 896 -T TCP -x D:\windows2008-1472\8-TCP
itgsend -a 192.168.3.2 -t 30000 -c 20000 -c 1024 -T TCP -x D:\windows2008-1472\9-TCP
itgsend -a 192.168.3.2 -t 30000 -c 20000 -c 1152 -T TCP -x D:\windows2008-1472\10-TCP
itgsend -a 192.168.3.2 -t 30000 -c 20000 -c 1280 -T TCP -x D:\windows2008-1472\11-TCP
itgsend -a 192.168.3.2 -t 30000 -c 20000 -c 1408 -T TCP -x D:\windows2008-1472\12-TCP
itgsend -a 192.168.3.2 -t 30000 -c 20000 -c 1536 -T TCP -x D:\windows2008-1472\13-TCP
itgsend -a 192.168.3.2 -t 30000 -c 20000 -c 64 -T UDP -x D:\windows2008-1472\14-UDP
itgsend -a 192.168.3.2 -t 30000 -c 20000 -c 128 -T UDP -x D:\windows2008-1472\15-UDP
itgsend -a 192.168.3.2 -t 30000 -c 20000 -c 256 -T UDP -x D:\windows2008-1472\16-UDP
itgsend -a 192.168.3.2 -t 30000 -c 20000 -c 384 -T UDP -x D:\windows2008-1472\17-UDP
itgsend -a 192.168.3.2 -t 30000 -c 20000 -c 512 -T UDP -x D:\windows2008-1472\18-UDP
itgsend -a 192.168.3.2 -t 30000 -c 20000 -c 640 -T UDP -x D:\windows2008-1472\19-UDP
itgsend -a 192.168.3.2 -t 30000 -c 20000 -c 768 -T UDP -x D:\windows2008-1472\20-UDP
itgsend -a 192.168.3.2 -t 30000 -c 20000 -c 896 -T UDP -x D:\windows2008-1472\21-UDP
itgsend -a 192.168.3.2 -t 30000 -c 20000 -c 1024 -T UDP -x D:\windows2008-1472\22-UDP
itgsend -a 192.168.3.2 -t 30000 -c 20000 -c 1152 -T UDP -x D:\windows2008-1472\23-UDP
itgsend -a 192.168.3.2 -t 30000 -c 20000 -c 1280 -T UDP -x D:\windows2008-1472\24-UDP
itgsend -a 192.168.3.2 -t 30000 -c 20000 -c 1408 -T UDP -x D:\windows2008-1472\25-UDP
itgsend -a 192.168.3.2 -t 30000 -c 20000 -c 1536 -T UDP -x D:\windows2008-1472\26-UDP
itgsend -a 192.168.3.2 -t DNS -x D:\windows2008-1472\27-DNS
itgsend -a 192.168.3.2 -t Csa -x D:\windows2008-1472\28-Csa
itgsend -a 192.168.3.2 -t Csi -x D:\windows2008-1472\29-Csi
itgsend -a 192.168.3.2 -t Quake3 -x D:\windows2008-1472\30-Quake3
itgsend -a 192.168.3.2 -x D:\windows2008-1472\31-G.711.1 VOIP -x G.711.1
itgsend -a 192.168.3.2 -x D:\windows2008-1472\32-G.711.2 VOIP -x G.711.2
itgsend -a 192.168.3.2 -x D:\windows2008-1472\33-G.723.1 VOIP -x G.723.1
itgsend -a 192.168.3.2 -x D:\windows2008-1472\34-G.729.2 VOIP -x G.729.2
itgsend -a 192.168.3.2 -x D:\windows2008-1472\35-G.729.3 VOIP -x G.729.3

```

Figure 3.6: D-ITG sample sender commands



```

decode-Windows2008-IPv4.bat - Notepad
File Edit Format View Help
itgdec 1-TCP -v > 1-TCP.txt
itgdec 2-TCP -v > 2-TCP.txt
itgdec 3-TCP -v > 3-TCP.txt
itgdec 4-TCP -v > 4-TCP.txt
itgdec 5-TCP -v > 5-TCP.txt
itgdec 6-TCP -v > 6-TCP.txt
itgdec 7-TCP -v > 7-TCP.txt
itgdec 8-TCP -v > 8-TCP.txt
itgdec 9-TCP -v > 9-TCP.txt
itgdec 10-TCP -v > 10-TCP.txt
itgdec 11-TCP -v > 11-TCP.txt
itgdec 12-TCP -v > 12-TCP.txt
itgdec 13-TCP -v > 13-TCP.txt
itgdec 14-UDP -v > 14-UDP.txt
itgdec 15-UDP -v > 15-UDP.txt
itgdec 16-UDP -v > 16-UDP.txt
itgdec 17-UDP -v > 17-UDP.txt
itgdec 18-UDP -v > 18-UDP.txt
itgdec 19-UDP -v > 19-UDP.txt
itgdec 20-UDP -v > 20-UDP.txt
itgdec 21-UDP -v > 21-UDP.txt
itgdec 22-UDP -v > 22-UDP.txt
itgdec 23-UDP -v > 23-UDP.txt
itgdec 24-UDP -v > 24-UDP.txt
itgdec 25-UDP -v > 25-UDP.txt
itgdec 26-UDP -v > 26-UDP.txt
itgdec 27-DNS -v > 27-DNS.txt
itgdec 28-Csa -v > 28-Csa.txt
itgdec 29-Csi -v > 29-Csi.txt
itgdec 30-Quake3 -v > 30-Quake3.txt
itgdec 31-G.711.1 -v > 31-G.711.1.txt
itgdec 32-G.711.2 -v > 32-G.711.2.txt
itgdec 33-G.723.1 -v > 33-G.723.1.txt
itgdec 34-G.729.2 -v > 34-G.729.2.txt
itgdec 35-G.729.3 -v > 35-G.729.3.txt

```

Figure 3.7: D-ITG decoder sample command lines

```

ITGDec version 2.8.0-rc1 (r459)
Compile-time options: ipv6
-----
Flow number: 1
From 192.168.1.1:49911
To 192.168.3.2:8999
-----
Total time = 29.989000 s
Total packets = 599787
Minimum delay = 0.022000 s
Maximum delay = 0.081000 s
Average delay = 0.024776 s
Average jitter = 0.000076 s
Delay standard deviation = 0.002255 s
Bytes received = 153545472
Average bitrate = 40960.478042 kbit/s
Average packet rate = 20000.233419 pkt/s
Packets dropped = 0 (0.00 %)
Average loss-burst size = 0.000000 pkt
-----
***** TOTAL RESULTS *****
-----
Number of flows = 1
Total time = 29.989000 s
Total packets = 599787
Minimum delay = 0.022000 s
Maximum delay = 0.081000 s
Average delay = 0.024776 s
Average jitter = 0.000076 s
Delay standard deviation = 0.002255 s
Bytes received = 153545472
Average bitrate = 40960.478042 kbit/s
Average packet rate = 20000.233419 pkt/s
Packets dropped = 0 (0.00 %)
Average loss-burst size = 0 pkt
Error lines = 0
-----

```

Figure 3.8: D-ITG sample decoded data in text format

There were a number of test-beds that were implemented in the process of this research, and consequently a great deal of primary data was generated. Due to the chosen method of data collection (test-bed), it was necessary to ensure that all results reported in the thesis were accurate and free from anomalies. That is, a high degree of accuracy was necessary and this was achieved by ensuring that multiple runs (number depending on the actual context in which TCP/UDP was tested) of D-ITG with the same parameters was performed for each value reported in this work. Due to the nature of the technology, wireless based network analysis required a greater number of repeat runs in spite of taking all necessary precautions to minimise the effects of external factors like signal interference. Overall, three to five repeat runs were necessary (to attain 95% confidence interval) for each value reported in this thesis. The reported value was the average of the readings taken after filtering any outliers.



## Chapter 4

---

# TCP/UDP Behaviour across Transition Mechanisms

### 4.1 Introduction and Motivation

The number of nodes connected to the networks is on an exponential incline, with each node requiring a unique address. In the original scheme of things there are only a limited number of these allocatable addresses, so it is necessary to changeover to a new version that offers an increased number of addresses. However, this changeover is complex and poses a number of problems, including the need to make configuration changes to all devices connected to the Internet. The sheer size of the problem, and incompatibility between the two versions, is delaying a complete switch over to the new version, which in addition to providing an increased address space has a number of other critical benefits.

The subject of this chapter is transition mechanisms. These facilitate coexistence of the two versions of the Internet Protocol. It is an interim solution that will be in use for a long time, therefore network performance analysis in such environments is important. In this chapter, commonly used transition mechanisms are described mainly by dividing them into three distinct categories dual stack, translation techniques and tunnelling technique (Arkko & Baker, 2011). Then literature in relation to transition mechanisms is evaluated, which is followed by identifying prior experimental based research related to performance evaluation . Next, a series of test-beds are described that were implemented to evaluate behaviour of TCP and UDP traffic type across various transition mechanisms. Finally, performance metrics from the test-beds are presented.

## 4.2 The IP Addressing Problem

The original version of the IP was developed in the early 1980s. Today one of the major concerns for network practitioners globally is how to change over from IPv4 to IPv6. As early as 1994, the IETF was warned that the exponential growth of the Internet, would quickly lead to a be a global shortage of IPv4 addresses (Bradner & Mankin, 1995) (Stallings, 1996) (Zeadally, Wasseem, & Raicu, 2004). Theoretically, IPv4 has approximately 4.3 billion addresses (Cerf, 2004); however, in practice it cannot support more than 250 million, uniquely addressed nodes (Zander, Andrew, Armitage, & Huston, 2013) (Postel, 1981b). This is mainly due to the hierarchical nature of the Internet and the resulting inefficiencies in address assignment (RFC 1751) (McDonald, 1994). To circumvent this problem a number of temporary solutions were created including: implementing DHCP (Droms, 1999) (Alexander & Droms, 1997) (Droms, 1997) to avoid permanent allocation of IP addresses to nodes; implementing Classless Inter-Domain Routing (CIDR) (Fuller & Li, 2006) removing fixed boundaries in IP addressing plans; implementing NAT (Audet & Jennings, 2007); allowing nodes with Private IP addresses access to Internet service; tightly controlling allocation of IP addresses by RIR; and reclaiming unused IPv4 address space. However, even with the implementation of these temporary mitigations, IPv4 addressing cannot keep up with the ever increasing demand for devices needing connectivity to the Internet (Waddington & Chang, 2002). Since IPv6 offers approximately 340 undecillion addresses, ultimately it is necessary to change the entire Internet infrastructure to be purely IPv6 based.

Running out of IP addresses in a communication network can pose massive problems, and although there is a short supply of IPv4 addresses globally, IPv6 adoption has been slow (Leavitt, 2011) .



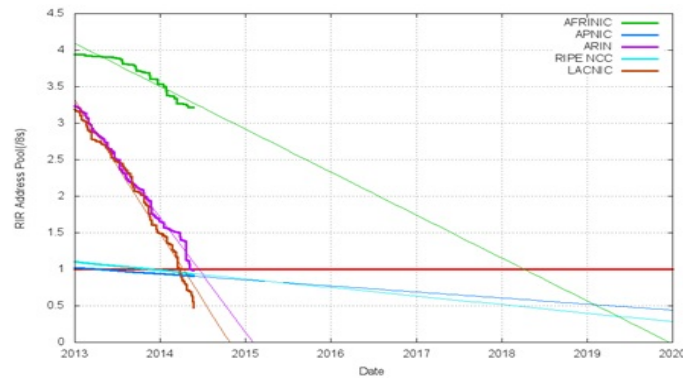


Figure 4.1: RIR IPv4 address run-down model per year

Source: <http://www.potaroo.net/tools/ipv4/index.html>

The Internet Assigned Number Authority (IANA) exhausted its pool of unallocated top-level addresses as early as 3 February 2011 (Boucadair, Grimault, Levis, Villefranque, & Morand, 2009) (Goth, 2012) (P. Wu, Cui, Wu, Liu, & Metz, 2013). Two of the five RIRs, APNIC and Rseaux IP Europens Network Coordination Centre (RIPE NCC), exhausted their addresses in April 2011 and September 2012, respectively (Housley, Curran, Huston, & Conrad, 2013). At the current rate of usage it is predicted that in the next few years, the three other RIRs will also exhaust their allocations. The uptake of IPv6 addresses has been surprisingly slow (Cerf, 2004). Google, which tracks the number of clients connecting to its services from different IP version based networks, predicted that as at mid-2013, a total of approximately 1.3% networks globally would be IPv6 based, and that only nine countries in the world would see the proportion of IPv6 users rise above one. This has been the case even though commonly used operating systems have been IPv6 ready for a while. Microsoft, for example, have been making its operating systems IPv6 ready since mid-2000 (Ladid, 2001) while Apples Mac OSX systems have had the capability since around 2006. The general view was that in the initial phase, the uptake of IPv6 would be slow, but would later gain momentum and accelerate along exponentially. This has not been the case.

The switchover from IPv4 to IPv6 is not that straightforward since the two versions of the protocol are not compatible, and there is no deadline by which the change must happen, unlike, for example, the Y2K problem. A seamless changeover from IPv4 to a completely IPv6 based Internet would be technically difficult to achieve. Until the entire Internet is purely IPv6 based, transition mechanisms will be the primary tools that will allow interoperability of both IP versions. That is, the changeover has to be done in stages, and transition mechanisms will be used during the migration stage so that IPv4 and IPv6 networks can coexist (Callon & Haskin, 1997), irrespective of how long the migration takes.

The Internet core will eventually change from IPv4 to IPv6. However, in the initial stages IPv6 islands will appear in the IPv4 cloud and connectivity for IPv6 networks will be required not just to the IPv4 Internet itself but also to other independent IPv4 or IPv6 islands. Towards the end of the migration process, the Internet cloud will predominantly be IPv6 based, and at that stage the reverse of what is described above will be required. That is, communication will have to be established between pockets of remaining IPv4 islands with either the IPv6 cloud or the independent IPv4 or IPv6 islands. For all communication channels that are to be established, transition mechanisms can be employed.

## 4.3 The Transition Mechanisms

Transition mechanisms allow interoperation between IPv4 and IPv6 networks during the migration phase from pure IPv4 to IPv6 environment. There are numerous such mechanisms, and this section will discuss the most common ones. Transition mechanisms can be divided into three categories (Dual Stack, Translation Techniques and Tunnels).

### 4.3.1 Dual Stack

One of the most common straightforward techniques adopted for migrating to IPv6 networks is the Dual Stack Transition Mechanism (DSTM). This technique enables communication between IPv4 and IPv6 networks (J. Chen, Jia., & Li, 2011), where hosts and routers on the network are implemented with both IPv4 and IPv6 protocol stacks (J.-M. Chang, Chao, Chen, & Lai, 2012). When a node needs to communicate with an IPv4 node, it uses the IPv4 protocol stack, and reverts back to establishing communication via the IPv6 stack when it needs to communicate with an IPv6 based node (Wei, Zhang, & Zhang, 2009). This setup enables a node to support both IPv4 and IPv6 based applications and services on a network.

Dual stack architecture is based on configuring a DSTM address server, dual stack capable client nodes and a Tunnel End-Point (TEP) router. The client nodes are IPv6 nodes in an IPv6 domain that want to communicate with IPv4 based nodes elsewhere, while the server manages the IPv4 address pool (Chakraborty, Dutta, & Biradar, 2009). The address server is responsible for allocating client nodes with IPv4 addresses within the IPv6 network, and also provides the TEP address. The server guarantees uniqueness of the IPv4 address allocated to the node for that period of time. When a host from the IPv6 domain wants to communicate with a node that has only an IPv4 address, it asks the DSTM server for a temporary IPv4 address (Xiaohong, 2013), which is issued to the client, together with information related to the TEP. The client, on receiving this information configures an IPv4 stack and from that point onwards, this client encapsulates all IPv4 packets and tunnels them, traversing the IPv6 domain to the TEP. On receiving the packet,

the TEP decapsulates it and transmits the packet to the IPv4 destination host. The TEP maintains a mapping of IPv4/IPv6 addresses and is also responsible for creating and suppressing the tunnel that temporarily exists between the dual stack node and TEP router.

Dual stack as a transition solution on a network, has various merits and weaknesses. First, the use of dual stack on a network infrastructure is transparent to the network. Nodes that are in the IPv6 domain encapsulate all IPv4 packets prior to transmission, thus, there is no need for the network to maintain IPv4 routing information. Secondly, the applications that are used on the clients need no configuration changes for them to continue working on dual-stack hosts. Also, allocation of IPv4 addresses in an IPv6 domain is effortless, since this can be easily done using DHCPv6, thus, reducing IPv4 related administrative overheads. However, this transition mechanism's prime weakness is that it does not support asymmetric paths, that is, return packets initiating from IPv4 hosts to dual-stacked clients in the IPv6 domain must enter the network through the same TEP that initially forwarded the traffic between the communicating nodes (T. Liu, Guan, Zheng, & Qu, 2009). Finally, there are significant network performance issues related to dual stack, mainly to do with the delay experienced at the initial stage of communication establishment between the sending and receiving nodes (S. Lee, Shin, Kim, Nordmark, & Durand, 2002).

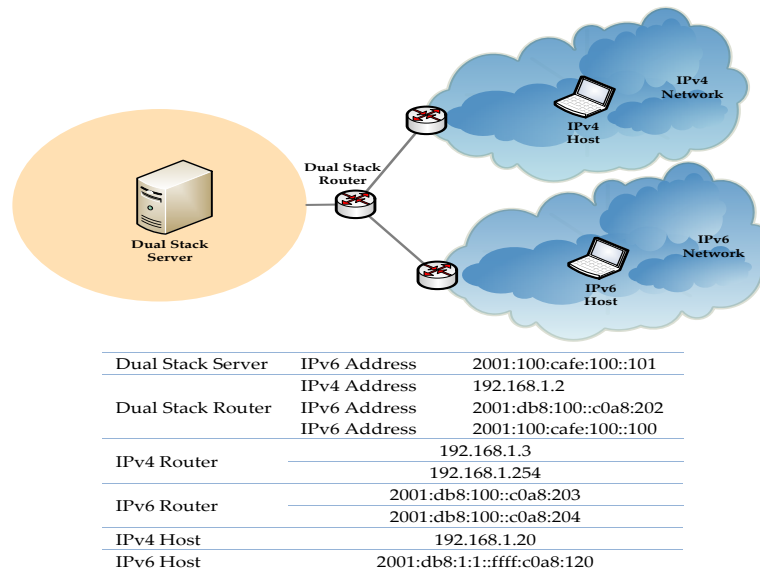


Figure 4.2: Architecture of dual stack implementation

### 4.3.2 Address Translation

There are a number of network address and protocol address translation techniques that can be used during the IPv4/IPv6 transition phase. NAT, in its classic form, is a technique that has been used to translate one IPv4 address into another address of the same version (Wing, 2010) (Srisuresh & Egevang, 2001). This has mainly been used on access networks in organisations using private IP address, where IP addresses from inside the network are mapped to public IP addresses (Srisuresh & Holdrege, 1999). By doing so, the Internet based services become accessible from within the organisation even when each host is not configured with a publically routable IP address (Kohler, Morris, & Poletto, 2002). Based on classic NAT are address translation techniques that can be implemented as solutions to enable co-existence of IPv4 and IPv6 based networks. These techniques allow establishment of communication between IPv4-only and IPv6-only devices.

Translation Mechanism	Description	Address Mapping	Weakness
NAT-PT	Translates IPv4/IPv6 addresses, keeps pool of IPv4 addresses in IPv6 domain.	Stateful, one-to-one mapping	Low feasibility on IPv4 to IPv6 routing and high processing for large-scale connections
NAPT-PT	Translates IPv4/IPv6 addresses, does not keep pool of IPv4 addresses, and IPv6 hosts can use a single IPv4 address.	Stateful, one-to-many mapping	Redirecting issues and unable to translate packet fragments.
NAT64	Translates IPv4/IPv6 addresses, translates IP header and IP address using various algorithms.	Stateful, vice-versa mapping	Pre-flow state maintenance and common translation issues.

Table 4.1: Different translation techniques in IPv4-IPv6 transition

### Network Address Translation Protocol Translation

Network Address Translation-Protocol Translation (NAT-PT) allows transparent routing of packets between end nodes in the IPv6 domain with end nodes in the IPv4 realm. A few variations of the NAT-PT exist. In Basic-NAT-PT (Tsirtsis & Srisuresh, 2000), a pool of IPv4 addresses are maintained in IPv6 domain and these addresses are allocated to IPv6-only nodes that intend to communicate with IPv4 nodes in some other domain (Chuanhuang & Haonan, 2012). Here, a one-to-one mapping of IPv4 to IPv6 addresses is made, thus, there is a need to maintain as many IPv4 addresses in the pool as IPv6 nodes in the network. A variation of this is Network Address Port Translation + Protocol Translation (NAPT-PT), where only a single IPv4 address is used to enable v6 nodes to communicate transparently with v4 nodes. This is achieved by translating TCP/UDP ports of the communicating IPv6 node into that of the specified IPv4 address. The advantage NAPT-PT has over NAT-PT is that there is no need to maintain a large pool of IP addresses (Shi et al., 2007); however a large number of IPv6 hosts (as many as the theoretical number of TCP/UDP ports) can effectively use a single IPv4 address to communicate with IPv4 nodes.

There are a number of limitations of NAT-PT and NAPT-PT address translation techniques. On establishment of a communication session between two nodes, all requests and responses pertaining to that session have to be sent

via the same NAT-PT router. To achieve this, the router can generally only be placed at one location on the network (at the perimeter) creating a single point of failure. Also, since the actual address translation is performed by NAT-PT, applications that have IP address information in the higher OSI layers will become dysfunctional. End-to-end security, as is possible in pure IPv6 networks, is completely lacking (Y.-G. Hong, Shin, & Kim, 2003). Another major concern is that end-to-end IPsec security cannot be configured since, IPsec requires end nodes to be configured with the same version of the IP. NAT-PT is also tightly coupled with DNS (W. Peng, Zhou, Wang, & Yang, 2009). Finally, implementing NAT-PT is complicated since it requires a significant amount of configuration.

Numerous technical and operational difficulties are encountered when NAT-PT is implemented on a network. For this reason IETF recommends that NAT-PT should not be used as a general purpose transition mechanism and has deprecated its use (Aoun & Davies, 2007).

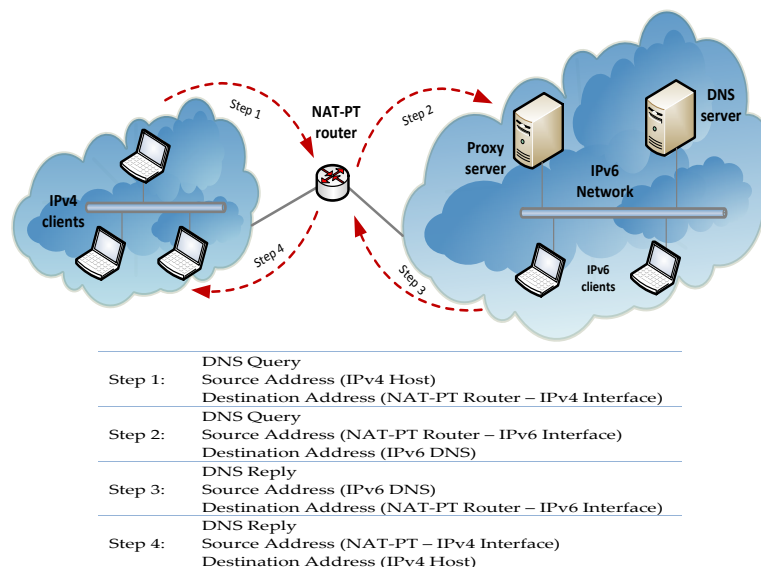


Figure 4.3: Architecture of NAT-PT implementation

### Network Address Translation IPv6 to IPv4

NAT IPv6 to IPv4 (NAT64) is the transition technique that is the successor to NAT-PT. Like its predecessor, NAT64 facilitates communication between IPv6-only, to IPv4-only hosts (Jankiewicz, Chan, & Green, 2006). However, total separation of DNS functionality and the NAT64 mechanism makes it a far superior technology. On an infrastructure, NAT64 can be implemented in two states, namely stateless or stateful. In stateless NAT64, the state of the session, or binding information, is not preserved, which means that every IPv6 node that is communicating with a v4 node needs a dedicated IPv4 address. This is achieved by algorithmically mapping IPv4 addresses to each communicating node. Stateless NAT64 does not solve the problem of depleting IPv4 addresses, however, it will facilitate nodes with different versions of the IP to communicate. Stateful NAT64 is the more desired method of implementing NAT64, and is capable of translating not only IPv6 addresses to IPv4, but also translating IPv4 to IPv6 (Bagnulo, Garcia-Martinez, & Van Beijnum, 2012) (Ding, Savolainen, Korhonen, & Kojo, 2012) (Bagnulo, Matthews, & van Beijnum, 2011). It also supports both IPv6 initiated and IPv4 initiated communications. Being a stateful method, it creates or modifies bindings and session state information while performing translation. Stateful NAT64 provides communication using TCP, UDP and ICMP between nodes by translating both the IP header and IP address using various algorithmic functions.

Comparing stateless and stateful NAT64, it is observed that the former is capable of a one-to-one mapping between IPv6-only and IPv6-only node, while the latter can be used in scenarios where there is a need to setup one-to-many translation. So with stateful implementations, there is no limitation on the number of end points, therefore, this is more suited as a transition solution that can be implemented on a large scale by carrier network providers. Stateful also conserves IPv4 addresses, however, lacks end-to-end address transparency. One of the mandatory requirements for stateless NAT64 is that there is a need to have an IPv4-translatable IPv6 address, but with the other, there is no restriction on the characteristics of the IPv6 address assigned (Hodzic & Mrdovic, 2012). Also, stateless NAT64 requires either manual or DHCPv6-



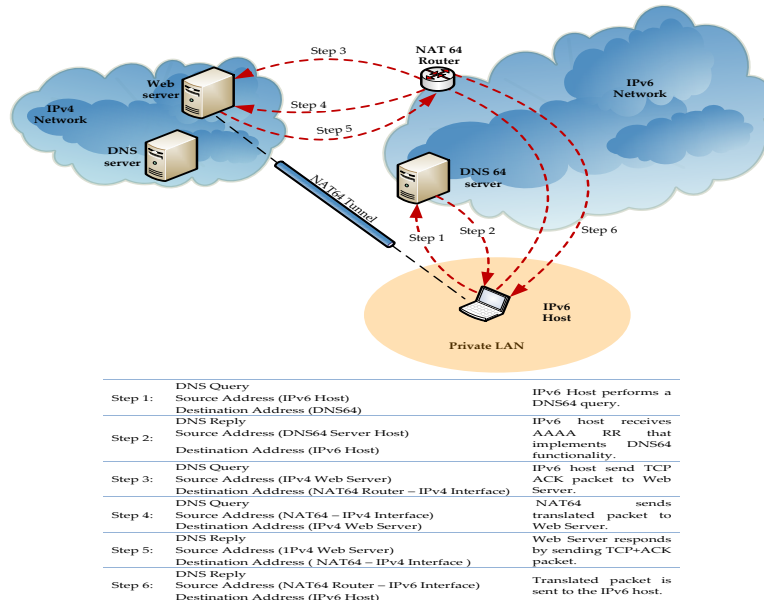


Figure 4.4: Architecture of NAT64 implementation

based address assignment for IPv6 hosts, but with stateful NAT64, the mode of IPv6 address assignment is inconsequential.

### 4.3.3 Tunnelling

Tunneling transition techniques provide mechanisms to utilise an already existing IPv4 infrastructure to carry IPv6 traffic. This is achieved by enabling hosts and routers to tunnel IPv6 datagrams over IPv4 topology by encapsulating the datagrams within IPv4 packets, primarily using IPv4 as the link layer for IPv6 communication (Xiaohong, 2013) (Hou, Zhao, & Ma, 2010). The key advantage of this technique is that the v6 protocol can work without hindrance to the old version, whilst also allowing connectivity between nodes using the newer version of the protocol. For ISPs that are in early stages of IPv6 deployment, tunnelling is the desired option, since it does not require a total upgrade of the network, yet can provide connectivity between IPv6 islands. ISPs and similar network providers need to enable IPv6 only

on the edge routers of the networks, facilitating rapid deployment of IPv6 services to customers. Therefore, tunnelling techniques enable incompatible networks to be bridged.

Tunnelling transition mechanisms can generally be configured in three different ways: router-to-router; host-to-router (or router-to-host); and host-to-host (Steffann, van Beijnum, & van Rein, 2013). Router-to-router tunnelling is normally implemented using IPv4/IPv6 dual stacked routers (at the edge of the network) to establish communication between two IPv4 or IPv6 networks over an existing IPv4 infrastructure (Raicu & Zeadally, 2003). A logical link is established between the edge routers which connects the source with the destination. All other routers within each IPv4 or IPv6 domain forward traffic to the IPv4/IPv6 edge routers when data needs to be sent from one IP domain to the other. With host-to-router tunnelling, an IPv4/IPv6 dual stacked host residing in an IPv4 domain creates an IPv6 over IPv4 tunnel to reach an edge IPv4/IPv6 router. The IPv4/IPv6 node configures an IPv4/IPv6 tunnel interface, representing the tunnel, through which the IPv6 packet is transmitted. The tunnel between the host and the router acts as a single hop. On receiving the packet at the IPv4/IPv6 router, another tunnel interface is created and using this, the router tunnels the IPv6 packet to the IPv6 node over the existing IPv4 or IPv6 infrastructure. Finally, the host-to-host tunnelling configuration is typically used between an IPv4/IPv6 dual stacked node, with another similar node, where both are residing in one IPv4 domain. When a communication channel needs to be established between the two nodes, an interface representing the tunnel is established on each, which is then used to send data between the source and the destination.

Tunnelling Mechanism	Description	Address Mapping	Weakness
6to4	IPv4 encapsulates IPv6 packets, tunnel is created automatically.	Stateless, IPv4 embedded in IPv6	Routing scalability issue: no optimization of paths
6over4	IPv4 encapsulates IPv6 packets, tunnel is created automatically.	None	Needs IPv4 multicast support
Dual Stack Translation Mechanism	IPv4 encapsulates IPv6 addresses, implementing gateway and server function.	Binding, per flow	Complicated implementation
ISATAP	IPv4 encapsulates IPv6 addresses. A tunnel with coexisting IPv4 and IPv6 addresses build up to the router.	Stateless with link-local or global prefix	Low efficiency of possible paths and complicated control plane
Teredo	Tunnel is built through NAT.	Stateless, port embedded in IPv6	Complicated implementation

Table 4.2: Different tunnelling techniques in IPv4-IPv6 transition

The manner in which a node determines the address of the tunnel endpoints is of significance. If a tunnel requires manual configuration of the tunnel endpoints, it is known as a configured tunnel. With this type of tunnel, the actual IPv4 address of the endpoint is not derived from the source or destination encoded IPv6 address, but is manually configured along with the static route information. This type of configuration is normally done when setting up router-to-router tunnelling configuration. Alternatively, automatic tunnels can be configured. No manual configurations are required, yet the nodes are able to determine the end points by using information related to logical tunnel interfaces, routes and source/destination IPv6 addresses. All tunnelling techniques, discussed next, can be classified as either configured or automatic tunnel.

#### 6to4

The 6to4 mechanism is an address assignment, router to router, host to router, and router to host automatic tunneling protocol that creates connectivity between IPv6 sites and hosts across IPv4 domains, by providing IPv6 unicast addresses to communicating hosts. The IPv4 domain that it connects through is treated as a single link (Bahaman, Hamid, & Prabuwno, 2012). Communication between 6to4 sites is tunnelled through directly, however, connectivity

to other IPv6 domains is attained by using 6to4 relay agents and 6to4 routers (Aazam, Syed, Shah, Khan, & Alam, 2011). So with minimal configuration, 6to4 can be implemented to allow IPv6 domains to communicate.

Tunneling techniques manipulate IP addresses to create appropriate node and tunnel endpoint addresses, as does the 6to4 mechanism. As described in RFC3056 (Carpenter & Moore, 2001), this mechanism uses the IANA-assigned IPv6 global address prefix `2002::/16` to indicate that a site is using a 6to4 tunnel (J.-L. Chen, Chang, & Lin, 2004a) (Elich, Velan, Jirsik, & Celeda, 2013). For 6to4 to send data from one site to another 6to4 site, the sending site assigns itself the IPv6 prefix `2002:a.b.c.d::/48`, where `a.b.c.d` is a globally unique v4 address of the interface on the 6to4 domains egress router (Carpenter & Moore, 2001) (Hei & Yamazaki, 2004) (Guo, Zhu, Chen, & He, 2012). The prepended number has precisely the same format as a typical /48 prefix, thus enabling the sending domain to use it like a usual valid /48 prefix. Thus, when a 6to4 domain needs to communicate with another similar domain, there is no need to create a tunnel since the attached prefixes take care of the addressing requirements. The destination address global routing IPv6 prefix contained in the IPv6 packet being sent will be used to determine the address of the tunnel endpoint. Also, to facilitate this communication, the edge routers do not need to run any specific IPv6 routing protocols. This is possible since IPv4 routing between the two 6to4 domains, which transmits via the Internet IPv4 cloud, takes care of inter 6to4 domain routing (Hadiya, Save, & Geetu, 2013). However, for a 6to4 domain to communicate with a non-6to4 domain, the process is slightly different. In this case, an IPv6 relay router is deployed, which essentially is an edge router configured with a minimum of one logical 6to4 interface and at least one native IPv6 interface (Carpenter, 2011). In this scenario, the relay router publicises the 6to4 `2002::/16` prefix within the native IPv6 domain, and IPv6 route information from that domain is advertised into its 6to4 connection. In order to establish the location of the relay router, IPv4 anycast is employed. Also, between the two dissimilar sites, the IPv6 exterior routing protocol must be implemented.

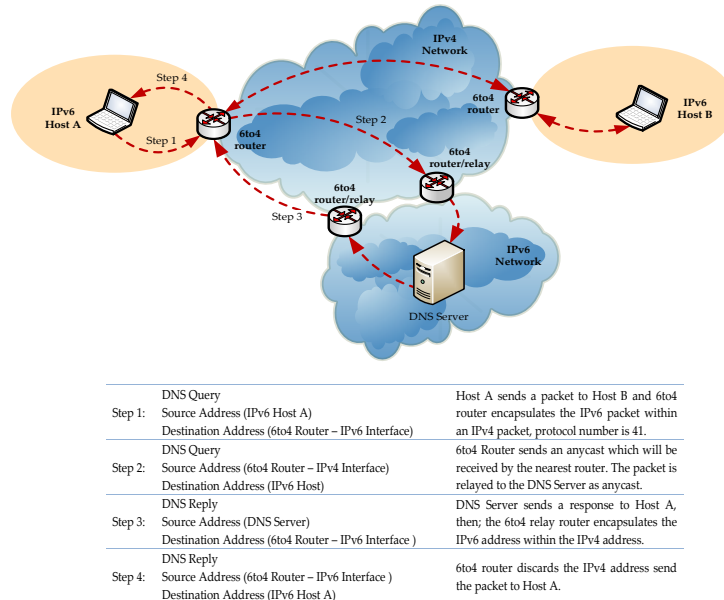


Figure 4.5: Architecture of 6to4 implementation

### Intra-Site Automatic Tunnel Addressing Protocol

Intra-Site Automatic Tunnel Addressing Protocol (ISATAP) is designed to tunnel traffic between routers within a site, however, it can also facilitate host to host, host to router, and router to host intra-site unicast IPv6 connectivity (Templin, Gleeson, & Thaler, 2008). It allows IPv4 hosts and IPv4/IPv6 dual-stacked hosts within a site to communicate with similar hosts, thus, creating an IPv6 infrastructure inside an IPv4 domain (S.-D. Lee, Shin, & Kim, 2006) (S. Hong, Ko, Ryu, & Kim, 2006). Within an ISATAP domain, the existing IPv4 infrastructure is seen as a virtual link layer for IPv6 communication, and other nodes in that site are viewed as potential IPv6 hosts and routers. ISATAP uses the underlying IPv4 domain as a virtual Non-Broadcast Multiple Access (NBMA) network, therefore, it does not require existing IPv4 infrastructure to support multicasting (Armitage, Schuler, Jork, & Harter, 1999). Furthermore, ISATAP can be implemented to facilitate communication to some outside IPv6 networks or hosts.

For an ISATAP host to communicate, first it has to create an interface identifier. To do this, private unicast IPv4 addresses are prepended with `::0:5efe` (Xiaodong, Mayan, & Yumei, 2009) (Aazam et al., 2011), while public unicast IPv4 addresses are prepended with `::200:5efe`. The interface identifier can then be combined with a 64bit prefix that is valid for IPv6 unicast addresses (Guo et al., 2012), which includes link local (`fe80::/64`), unique local, and global prefixes. For example, if two ISATAP nodes on one network with IPv4 addresses 10.5.6.1 and 10.5.6.2 are to communicate, they will automatically be configured with the ISATAP addresses of `fe80::5efe:10.5.6.1` and `fe80::5efe:10.5.6.2` respectively. So, the interface identifier part of the ISATAP address has an embedded IPV4 address, which is used to determine the destination IPv4 address after the ISATAP-addressed IPv6 packet has been tunnelled through the IPv4 domain. Needless to say, link local ISATAP addresses allow a host to communicate with another host only on the same network, however, to communicate beyond the local domain, ISATAP based global addresses have to be configured, and the host has to tunnel the packets to an ISATAP router. An ISATAP host is configured with a Potential Routers List (PRL), and the host randomly probes the routers in that list by sending ICMPv6 Router Solicitation (RS) packets to find out which routers are up and functioning. On receiving an RS, the router responds with a unicast Router Advertisement (RA) message. An ISATAP router is a dual-stacked device that is responsible for a few important things: (i) it advertises address prefixes to ISATAP hosts identifying the logical ISATAP subnet that the host is residing in, (ii) forwards packets between ISATAP hosts within a ISATAP subnet and also between to IPv6 hosts on other networks and (iii) is an IPv6 default router for local ISATAP hosts.

ISATAP is a commonly used automatic tunneling mechanism; however, there are a few design issues to be aware of. Firstly, there is heavy reliance on the IPv4 DNS server. When ISATAP is deployed on a network, an ISATAP host typically builds its PRL by consulting a DNS Server (Aazam, Khan, Alam, & Qayyum, 2010). This is normally done by resolving a name, such as `isatap.domain.com`, where `domain.com` is the local domain. This is seen as problematic since ISATAP, being a lower layer protocol, has to rely heavily

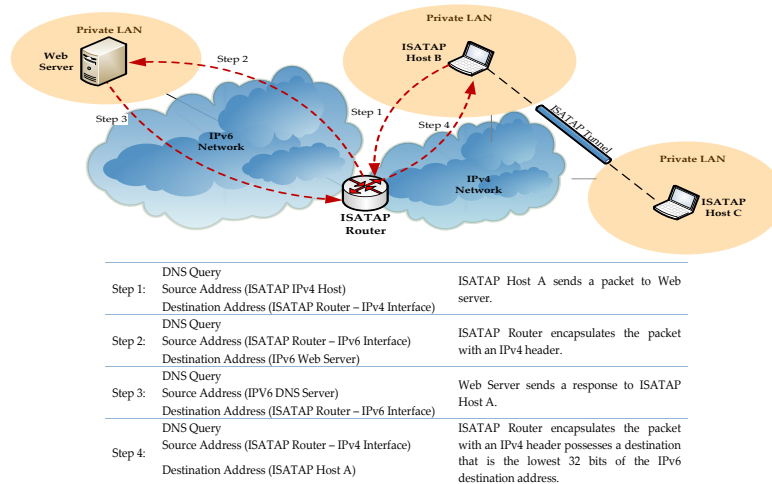


Figure 4.6: Architecture of ISATAP implementation

on higher layer IPv4 DNS services. Secondly, security breaches are possible at the edge of the network if the IPv4 virtual link to the network edge is not delimited carefully. If not configured securely, the external IPv4 host may pretend to be part of the ISATAP link, thus causing a security breach.

### Teredo

The Teredo transition mechanism, proposed by Microsoft, is a host to host automatic tunneling technique that addresses a fundamental problem suffered by both the tunneling techniques mentioned (6to4 and ISATAP). Both of these mechanisms require public IPv4 addresses configured at the tunnel end points, but with prevalent use of NAT to address the IPv4 shortage problem, this may not be possible (Elich et al., 2013). Teredo enables dual-stacked nodes located behind an IPv4 NAT to establish IPv6 communication by tunneling packets through the IPv4 infrastructure after encapsulating IPv6 packets in IPv4 based UDP messages (Aazam et al., 2010). The Teredo protocol was originally known as shipworm, based on a species of bivalve molluscs that burrow holes in wooden ship hulls, analogous to the mechanism's ability to penetrate through NAT. Teredo needs almost no manual configura-

tion from a user within a network, and can also be used by ISPs to perform large scale deployments (Huitema, 2006). Teredo clients, a server, and relays have to be configured on a network when deploying this mechanism. Teredo clients are IPv4/IPv6 dual stacked nodes behind a NAT IPv4 network, which at the start of a communication process perform a qualification procedure by exchanging appropriate messages with the Teredo server. The qualification procedure entails determining whether the client is located behind NAT, the actual type of NAT, and also its public IPv4 address. (S.-M. Huang, Wu, & Lin, 2005) The Teredo server uses UDP port 3544 for listening to requests from the clients, and responds by assigning an IPv6 address to the client. In addition to address assignment, the server also forwards the IPv4 encapsulated IPv6 packets it receives from the clients to the Teredo relay, and also on the reverse path (from Teredo relay to a Teredo client). The relays also manage advertising reachability of Teredo based services into the IPv6 network. Normally it is possible to co-locate a Teredo server and relay entities onto one device.

There are a few negatives associated with using Teredo as an IPv4/IPv6 tunneling solution. Due to the injection of information related to routing prefixes, the mechanism inefficiently uses IPv6 addresses. This is primarily due to the requirement that reachability of Teredo services has to be advertised to the IPv6 network, and in doing so a 32-bit prefix common to all Teredo servers and the IPv4 address of the server have to be advertised in the IPv6 domain. Secondly, Teredo services cannot traverse symmetric NATs. When a Teredo client attains an IPv6 address from a server, the mapped public IPv4 address and UDP port number are encoded in the IPv6 address. This encoded address is used by the Teredo relay when sending packets to the destination, but when traversing symmetric NATs different mapped port numbers are allocated for each pass-through flow, thus, causing a conflict. Finally, when compared with other automatic tunneling protocols, Teredo is a complex technology to implement on networks. Therefore, the Teredo automatic tunnelling protocol should only be considered as a last resort.



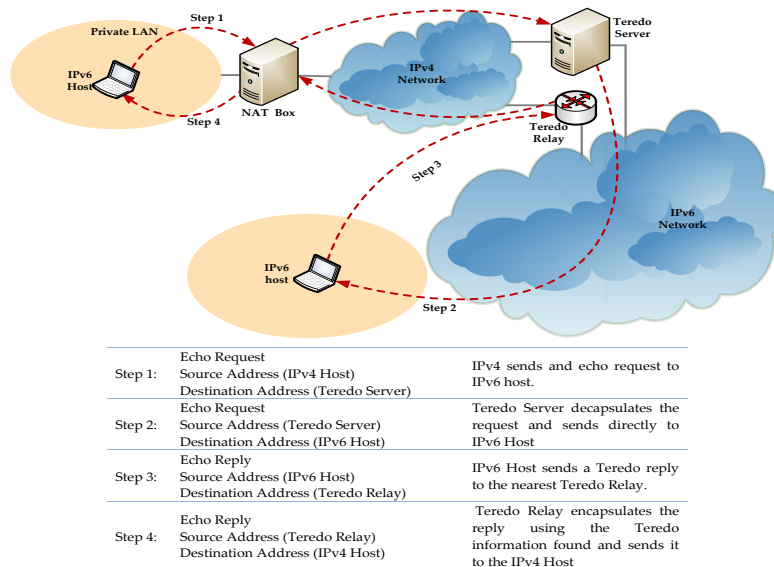


Figure 4.7: Architecture of Teredo implementation

### 6over4

6over4 is a tunnelling technique that allows IPv4 networks to communicate with IPv6 based domains. In addition to this, the mechanism can also be used to communicate between an isolated IPv6 host with another similar host over an inter-site IPv4 domain that is capable of supporting IPv4 multicasting. The mechanism uses the IPv4 network as a virtual data link layer to traverse IPv6 initiated traffic. Formally, 6over4 was known as IPv6 over IPv4, and is colloquially known as Virtual Ethernet. The mechanism allows a network to function with hosts that have either version the of IP stack, without requiring IPv6 hosts to be configured with IPv4 compatible addresses or with information related to tunnels. However, it does require interfaces on IPv6 capable routers and hosts to be enabled for 6over4 mode.

In relation to modifying IPv6 address, so that IPv6 traffic can traverse the IPv4 network, 6over4 uses a trivial technique for generating link-local IPv6 addresses. For any node wanting to communicate using a 6over4 tunnel over an IPv4 infrastructure, it has to set up a virtual IPv6 interface by creat-

ing an interface identifier. The interface identifier is created by prepending the IPv4 address of that interface with the suffix `fe80::/64`, creating a 128 bit IPv6 address. For example, a host with the IP address 192.0.2.142 will use `fe80::c000:28e` (where `c000:28e` is the hexadecimal representation of the IPv4 address) as its link local address. `6over4` treats the existing IPv4 network as a single link that has multicast capabilities. Therefore, the Neighbor Discovery process, such as router discovery and address resolution, works in a `6to4` regime. To facilitate multicasting, an IPv6 multicast address has to be translated into an IPv4 multicast address. This is done by encapsulating IPv6 multicast packets with the destination address `239.192.y.z`, where `y` and `z` are the last two bytes of the IPv6 multicast address.

For `6over4` to be implemented as a tunneling solution, it is mandatory that IPv4 multicasting is possible on the IPv4 infrastructure. However, there is limited availability of this service on v4 networks (D. Lee, Lough, Midkiff, Davis, & Benchoff, 1998). Also, not all operating systems support the use of `6over4`; hence, `6over4` is rarely used on networks as a transition mechanism.

## 4.4 Literature Analysis of Transition Mechanisms

This section presents a review of recurring issues related to transition mechanisms. There are multiple aspects of research within this topic, however, discussions herewith mainly concentrate on the network performance flank of common transition mechanisms. In relation to transition mechanism techniques, there are many strands that have been studied to enhance the mechanisms or to evolve some novel technique to interoperate IPv4 and IPv6 networks. Based on Teredo, a tunnelling protocol known as Escort has been developed, which addresses some of Teredo's fundamental problems (An, Luo, Li, Zhang, & Yan, 2009). Inherent in Escort are the fundamental advantages of Teredo, however, by incorporating Host Identity Protocol (HIP) architecture, Escort has introduced the concept of ID/Locator split (Henderson, Ahrenholz, & Kim, 2003) (Moskowitz & Nikander, 2006) (Farinacci, Fuller, Meyer, & Lewis, 2013). This allows Escort to decouple the name and locator roles currently filled by IP addresses, thereby, providing a more secure communication channel than Teredo. Therefore, this makes Escort a strong candidate as a mechanism for mobile and multihoming environments. This technique is similar in concept to Evolvable Locator/ID Separation Internet Architecture (ELISIA) (Zhang, Li, & Bao, 2013). Teredo Client Protection Algorithm (TCPA) has been mooted as a solution to protect Teredo clients from IPv6 routing header risks (Al-tamimi, Taib, & Budiarto, 2008). This protects from exploitation of source routing in IPv6 from both internal and external attacks. TCPA has been shown to be an efficient and logical alternative to plain Teredo. A technique known as SymTeredo (Punithavathani & Radley, 2014), which is a minor extension of Teredo, addresses the issue related to Teredo's inability to traverse symmetric NAT. This was also attempted in (S.-M. Huang, Wu, & Lin, 2006), however, the design violated the load-balancing design of Teredo and imposed heavy loads on the Teredo sever. With SymTeredo, only a slight modification to the Teredo mechanism is required to the relay and the client's components, allowing it to function even in a symmetric NAT environment.

Evident in literature are a few transition techniques that are novel in their approach in solving the IPv4-IPv6 coexistence problem. A framework known as Prefixing, Encapsulating and Translation (PET) has integrated concepts of tunnelling and translation techniques to support both traversing and interconnecting IPv4-IPv6 networks (P. Wu et al., 2010). In this technique, automatic translation spot election and translation context advertisements are used to build dynamic tunnels to connect networks. In another technique, which is based on hierarchical routing architecture (Sans & Gamess, 2013), encapsulation overhead in tunnel based mechanisms is improved and consumption of IPv4 addresses are reduced in translation based methods. Here, the concept is based on principles of IPv4 routing and the ability of nodes to establish whether a destination address is local, or on some remote network. Using this as a premise, a three-layer hierarchical schema is proposed where the host nodes are configured with an IPv4/IPv6 gateway, which is the router that handles routing different versions of IP traffic.

Another technique that addresses some of the problems with tunnelling techniques is the Dynamic Tunneling Transition Solution (DTTS) (Y. Wu & Zhou, 2011). This is based on a dynamic tunnelling technique and dual stack approach, enabling IPv4 applications to interact with similar applications in both IPv4 and IPv6 domains (K. Wang, Yeo, & Ananda, 2001). DTTS requires a pool of IPv4 addresses, thus, does not address the IPv4 shortage issue, however, end-to-end IP address transparency, scalable deployment and seamlessly running IPv4 applications in the IPv6 domain are some of DTTSs advantages. Some other transition techniques derived from already existing methods are highlighted in (Elich et al., 2013) (J.-L. Chen, Chang, & Lin, 2004b) and (Nordmark & Gilligan, 2005). Some researchers have also evaluated the intricacies of the current mechanisms with the view to improve their design characteristics.

Since the focus in this thesis is on performance evaluation of networks based on test-bed analysis, in Table 4.3, some of the key research that have taken a similar stance on transition mechanisms are presented.

Author(s)	Title	Research Focus
N. Chuangchunsong, S. Kamolphiwong, T. Kamolphiwong, R. Elz, and P. Pongpaibool (2014)	Performance Evaluation of IPv4/IPv6 Transition Mechanisms: IPv4-in-IPv6 Tunneling Techniques	4over6 and Dual Stack
Martin Elich, Petr Velan, Tomas Jirsik, and Pavel Celeda (2013)	An Investigation Into Teredo and 6to4 Transition Mechanisms: Traffic Analysis	Teredo and 6to4
Francisco Sans and Eric Gamess (2013)	Analytical Performance Evaluation of Native IPv6 and Several Tunneling Technics using Benchmarking Tools	ISATAP, 6to4 and Teredo
Rajesh Duvvuru and Sunil Kumar Singh (2013)	Minimizing Transmission Delay in IPv4 Network to IPv6 Network through ADSTM	Dual Stack
Jingtao Su and Xianwei Zhou (2013)	IVIT: A Core Stateless IPv4/IPv6 Translation Mechanism Combining Translation and Tunnel Technologies	NAT-PT
Fei Ren and Huachun Zhou (2012)	Implementation and Test of PMIPv6 Dual Stack Protocol	Dual Stack
Nazrulazhar Bahaman, Erman Hamid and Anton Satria Prabuwono (2012)	Network Performance Evaluation of 6to4 Tunneling	6to4
Anthony K. Tsetse, Alexander L. Wijesinha, Ramesh K. Karne and Alae Loukili (2012)	A 6to4 Gateway with Co-located NAT	6to4
Mohammad Aazam, Adeel M. Syed, Syed Atif, Imran Khan and Muhammad Alam (2011)	Evaluation of 6to4 and ISATAP on a Test LAN	6to4 and ISATAP
Yingjiao Wu and Xiaoqing Zhou (2011)	Research on the IPv6 Performance Analysis Based on Dual-Protocol Stack and Tunnel Transition	Dual Stack, 6to4 and ISATAP

Table 4.3: Recent research on performance evaluation of transition mechanisms

## 4.5 Test-bed Transition Mechanism Implementations

The overarching purpose of this research is to evaluate the behaviour of TCP and UDP traffic types on networks, and in this chapter the focus is on their behaviour across transition mechanism implementations. It is envisaged that by measuring common network performance metrics on test-bed implementations, the overall objective of the research will be achieved.

Next, the focus is on outlining the test-bed setups that were implemented in a laboratory environment. In total, seven different transition mechanisms have been tested.

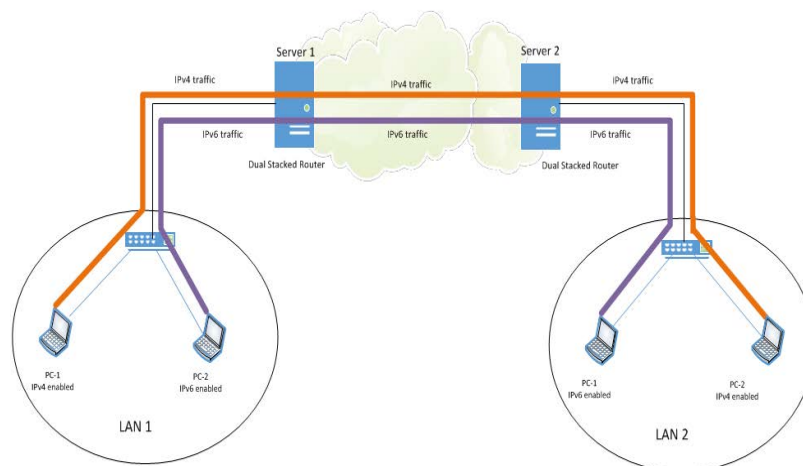


Figure 4.8: DSTM network diagram

The first network implemented was to test TCP and UDP behaviour across DSTM (Figure 4.8). As discussed earlier, this is the simplest of the transition mechanisms, allowing connectivity between IPv4 and IPv6 environments. Here, the routers were dual stacked with both versions of the protocols allowing them to facilitate communication between IPv4 hosts and other similar protocol hosts or IPv6 hosts. From an implementation point of view, this

was the simplest of all the networks set up, since the computers were either pure IPv4, or IPv6 (with appropriate gateways), and the two servers were dual stacked with routing enabled. For the purposes of measuring the performance metrics, two computers were chosen on the network as sender and receiver of different traffic types.

In the next series of experiments, various tunneling transition techniques were implemented. Configured Tunnel, 6to4, 6over4, ISATAP and Teredo were implemented, as shown in Figures 4.9, 4.10, 4.11 and 4.12 respectively.

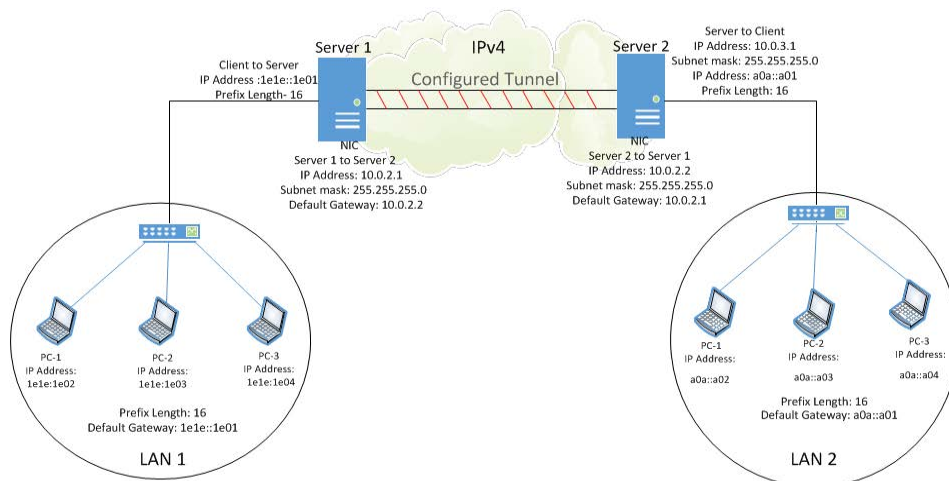


Figure 4.9: Configured tunnel network diagram

For configured tunnel, a dual stack gateway was configured as a tunnel endpoint on one end, while a dual stack router was implemented at the other end of the tunnel. This tunnel was established across the IPv4 infrastructure, and encapsulation and decapsulation of the traffic traversing the two IP environments was performed by the dual stack gateway and the router. Configured tunnel implementation was mainly done manually, accomplished using Command Line Interface (CLI).

In the 6to4 tunnel test-bed, the network setup was very similar to that of configured tunnel. A 6to4 gateway was configured on one router at one end of the tunnel, whilst the other tunnel end router was configured as a 6to4 relay agent. These two endpoints established a tunnel across the IPv4 network infrastructure to facilitate communication with IPv6 nodes. The IPv6 packets are encapsulated in IPv4 packets by the 6to4 gateway and then forwarded to the 6to4 relay router. At this point the packet is decapsulated and forwarded to the global IPv6 network.

The 6over4 technique uses 6in4 as the procedure for encapsulating IPv6 packets in IPv4 multicast infrastructure. It uses a multicast enabled IPv4 network as an intermediary between the different IP version infrastructures. After adding IPv4 stack to the nodes, a 6over4 tunnel was created between the routers, mainly via CLI using netsh and adds v6v4tunnel set of commands to add local addresses and remote addresses to the routers. Here, the local address is an IPv4 address that the tunnel uses to talk with the upstream version 6 router. The remote IPv4 address is that of the upstream router, which also has an IPv6 remote address.

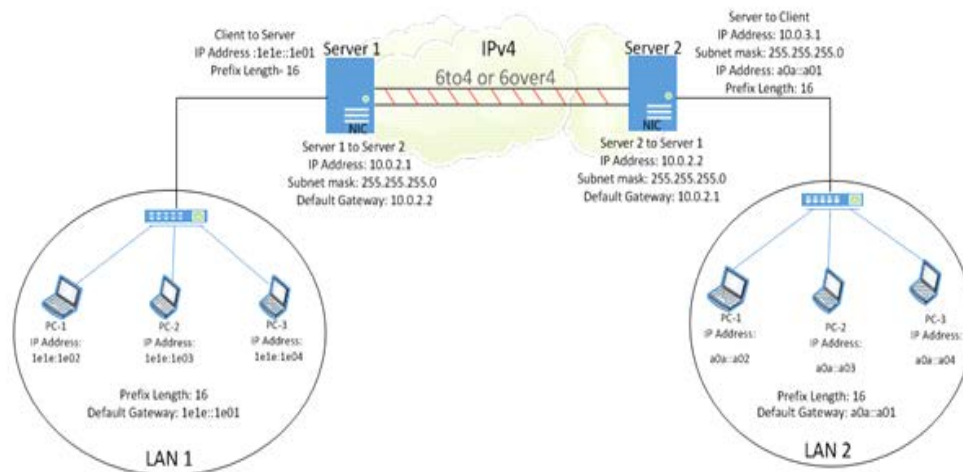


Figure 4.10: 6to4 network diagram



On completing the tests for 6over4, an ISATAP tunnel was implemented on the same setup. Since this is also one of the tunnelling transition mechanisms, the actual infrastructure did not require much change, however, there were configuration changes. After configuring the IPv4 and IPv6 stacks appropriately on all computers, the servers were configured as routers. Then *netsh* command was used on both the routers to establish forwarding and advertisements to establish a connection between them. This enabled an ISATAP tunnel to be created between the two IP environments.

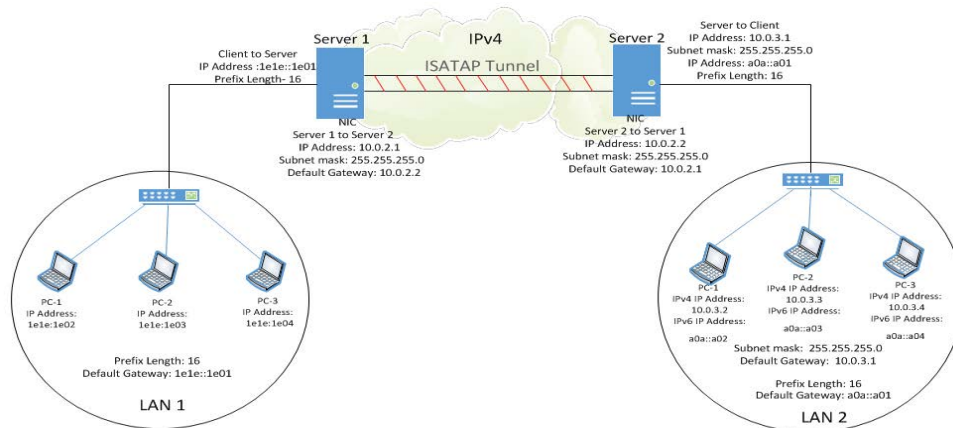


Figure 4.11: ISATAP network diagram

The last of the tunnelling transition mechanisms implemented was Teredo. Microsoft operating systems come pre-configured with Teredo, therefore, the implementation was straightforward. However, since various mechanisms were being compared for network performance, it was necessary to ensure that the implemented test beds were very similar (if not the same) in configuration for all setups. To test the performance of Teredo, a Teredo server was introduced on the IPv4 subnet. This server is responsible for the initial configuration of the tunnel and for address configuration of the Teredo clients. Also, a relay was configured that forwarded packets between IPv4 and the IPv6-only hosts on the pure IPv6 network.

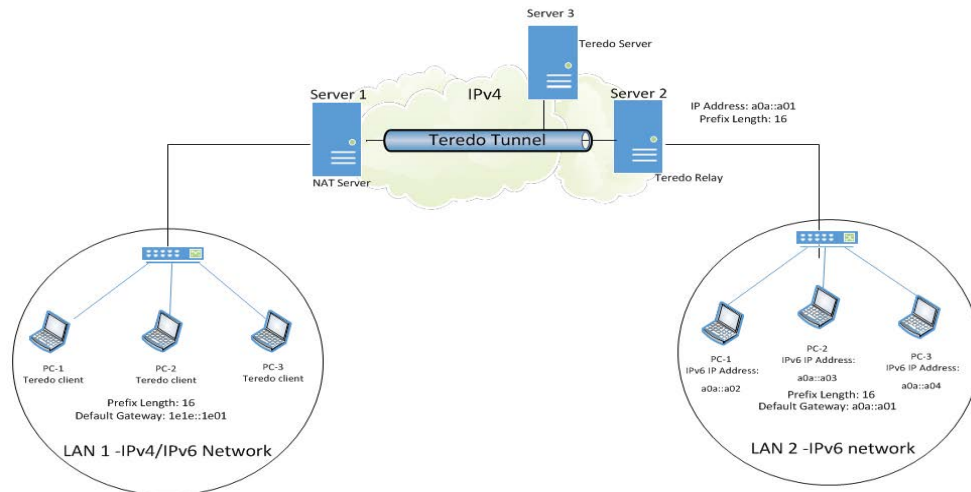


Figure 4.12: Teredo network diagram

Finally, NAT64, an IPv4-IPv6 translation based transition technique was implemented. This can be implemented either as stateless or stateful, but for this research it was implemented as the latter. In this, IPv6 addresses are translated in IPv4 (and vice versa) by creating bindings, or session states, while performing translation. Stateful NAT64 supports both IPv4 initiated and IPv6 initiated communications using static manually configured IP address mappings.

To implement this translation technique, a NAT64 server was deployed, which hosted DNS64 and commonly used domain services in the test-bed. In addition to this, Remote Access Management, DirectAccess and a Certification Authority were also implemented on the same server.

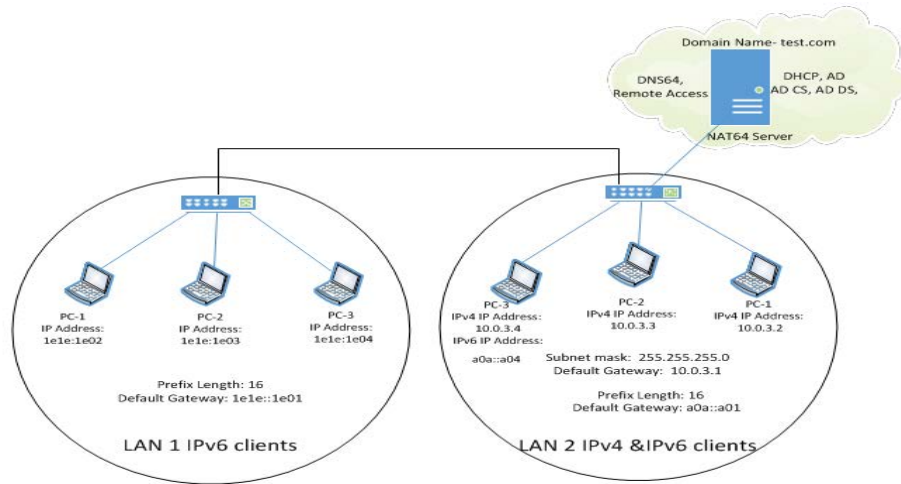


Figure 4.13: NAT64 network diagram

For each of the seven transition mechanisms implemented, a sender and a receiver computer were identified on the test-bed. These two nodes were configured with appropriate D-ITG components allowing TCP and UDP traffic, generated by D-ITG, to traverse between them. Four network performance related metrics (throughput, delay, jitter and Central Processing Unit (CPU) usage) were measured for various packet sizes.

## 4.6 Performance Metrics Measurement for various Transition Mechanisms

To compare behaviour of TCP and UDP traffic types in the context of transition mechanisms, a total of seven different mechanisms were implemented on the test-bed. For each mechanism, throughput, delay, jitter and CPU usage were measured.

At the beginning of the metrics measurement process, the first and second sets of measurements taken were for pure IPv4 and pure IPv6 networks respectively. Here, all nodes were stacked with either IPv4 or IPv6, and then appropriate performance metrics were measured. By doing so, a benchmark is established against which various transition mechanism metrics can be compared.

TCP throughput results are presented in Figure 4.14. Due to hardware limitations, the maximum bandwidth possible on the test-bed was 100Mbps (as was the case in all the tests), and it can be seen that the pure IPv4 network TCP throughput values are very close to the theoretical upper bound for most packet sizes. It can also be observed that IPv6 values are slightly lower than that of IPv4, for all packet sizes. For almost all packet sizes above 256Bytes, TCP throughput values range between 75-95 Bytes for the scenarios tested. However, there are differences in TCP behaviour amongst the transition mechanisms.

To gauge the difference in performance between the seven transition mechanisms and to compare that with pure IPv4 and IPv6 implementations, normalised data is represented in a colour heat map as shown in Figure 4.15. In this chart, green represents desirability, which in this case is higher TCP throughput values.

Collectively in Figures 4.14 and 4.15, it can be seen that there is a clear and significant distinction between TCP throughput values for different transition mechanisms. While pure IPv4/IPv6 have the highest values in this set,

dual stack stands out with the best throughput of all the seven mechanisms tested. Its throughput drop at the maximum is approximately 8%. Conversely, it can be seen that Teredo has the lowest throughput for almost all packet sizes. At its lowest point, the throughput drop is almost 30% (this is for packet size 256 and 384 bytes). In the tunnelling techniques tested, as opposed to dual stack and translation, the configured tunnel and 6to4 give slightly better TCP throughputs. ISATAP, apart from Teredo, has comparatively lower throughput values out of all the tunnelling techniques.

UDP throughput data is presented next. As seen in Figure 4.16 and the associated heat map in Figure 4.17, the highest values again are for pure IPv4 and IPv6 networks, these values being marginally higher than that for TCP. The transition mechanism UDP throughput values show differences of upto 35% between the highest and lowest values. The lowest values are again reported by Teredo while the highest of all transition mechanisms is by dual stack. This pattern is similar to that of TCP throughput values, however all values are marginally higher than TCP counterparts.

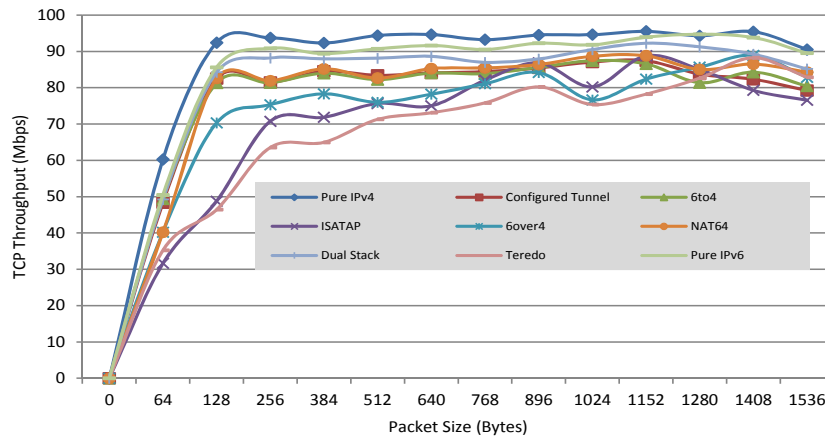


Figure 4.14: Graph of TCP throughput implementing different transition mechanisms

TCP Throughput: Transition Mechanisms													
	64	128	256	384	512	640	768	896	1024	1152	1280	1408	1536
Pure IPv4	0.63	0.97	0.98	0.97	0.99	0.99	0.98	0.99	0.99	1.00	0.99	1.00	0.95
Configured Tunnel	0.51	0.86	0.85	0.88	0.87	0.88	0.88	0.89	0.91	0.92	0.88	0.86	0.83
6to4	0.52	0.85	0.85	0.88	0.86	0.88	0.88	0.90	0.92	0.91	0.85	0.88	0.84
ISATAP	0.33	0.51	0.74	0.75	0.79	0.79	0.86	0.91	0.84	0.93	0.89	0.83	0.80
6over4	0.42	0.74	0.79	0.82	0.80	0.82	0.85	0.88	0.80	0.86	0.90	0.93	0.87
NAT64	0.42	0.86	0.86	0.89	0.87	0.89	0.89	0.90	0.93	0.93	0.89	0.91	0.88
Dual Stack	0.52	0.88	0.92	0.92	0.92	0.93	0.91	0.92	0.95	0.97	0.96	0.93	0.89
Teredo	0.37	0.49	0.67	0.68	0.75	0.77	0.79	0.84	0.79	0.82	0.87	0.92	0.87
Pure IPv6	0.53	0.90	0.95	0.94	0.95	0.96	0.95	0.97	0.96	0.98	0.99	0.98	0.94

Figure 4.15: Heat map of TCP throughput implementing different transition mechanisms

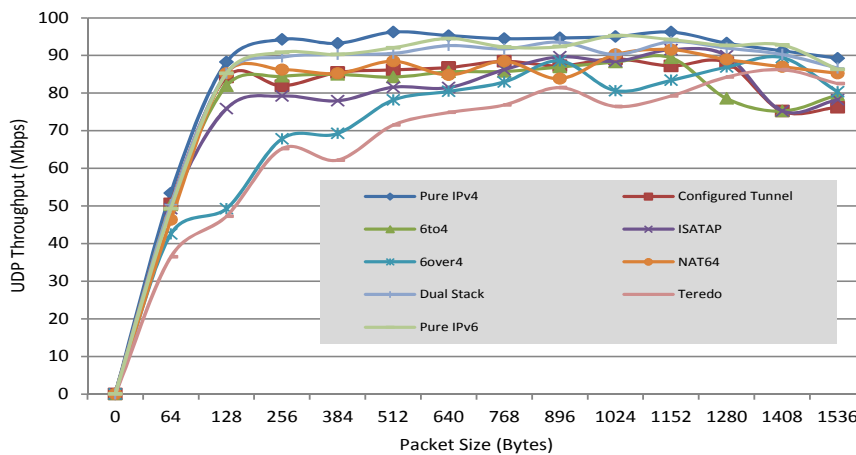


Figure 4.16: Graph of UDP throughput implementing different transition mechanisms

UDP Throughput: Transition Mechanisms													
	64	128	256	384	512	640	768	896	1024	1152	1280	1408	1536
Pure IPv4	0.56	0.92	0.98	0.97	1.00	0.99	0.98	0.98	0.99	1.00	0.97	0.95	0.93
Configured Tunnel	0.53	0.88	0.85	0.89	0.90	0.90	0.92	0.91	0.93	0.91	0.92	0.78	0.79
6to4	0.52	0.85	0.88	0.88	0.88	0.89	0.89	0.90	0.92	0.93	0.82	0.78	0.83
ISATAP	0.51	0.79	0.82	0.81	0.85	0.85	0.90	0.93	0.92	0.95	0.93	0.78	0.81
6over4	0.44	0.51	0.70	0.72	0.81	0.84	0.86	0.92	0.84	0.87	0.90	0.93	0.83
NAT64	0.48	0.89	0.90	0.89	0.92	0.88	0.92	0.87	0.94	0.95	0.92	0.90	0.89
Dual Stack	0.52	0.89	0.93	0.94	0.94	0.96	0.95	0.97	0.94	0.97	0.96	0.94	0.90
Teredo	0.38	0.49	0.68	0.65	0.74	0.78	0.80	0.85	0.79	0.82	0.88	0.90	0.86
Pure IPv6	0.51	0.89	0.94	0.94	0.96	0.98	0.96	0.96	0.99	0.98	0.96	0.96	0.90

Figure 4.17: Heat map of UDP throughput implementing different transition mechanisms

Comparing the transition mechanisms, 6over4 shows a slightly different behaviour to that with TCP traffic type. Here the values are lower, almost close to that of Teredo. The translation techniques tested all seems to have comparable values, apart from that of Teredo. Of these techniques, ISATAP is again recording comparatively marginally higher values for larger packet sizes, but definitely lower for packets smaller than 512 bytes.

In Figures 4.18 and 4.19, TCP delay values are presented. Here a clear pattern emerges that shows there are significant differences between the mechanisms. This is very important when configuring delay sensitive applications on networks. As expected, networks without transition mechanisms have the lowest values, with the pure IPv6 network having values marginally lower than IPv4. All these values average around 100ms. NAT64 and dual stack values band together, averaging around 400ms, but in all cases, NAT64 values are marginally higher. The values reported by all other transition mechanisms are very similar, with an average around 1150ms. Teredo values are in the same band, however, its values have a great degree of fluctuations and are at the lower end of this band.

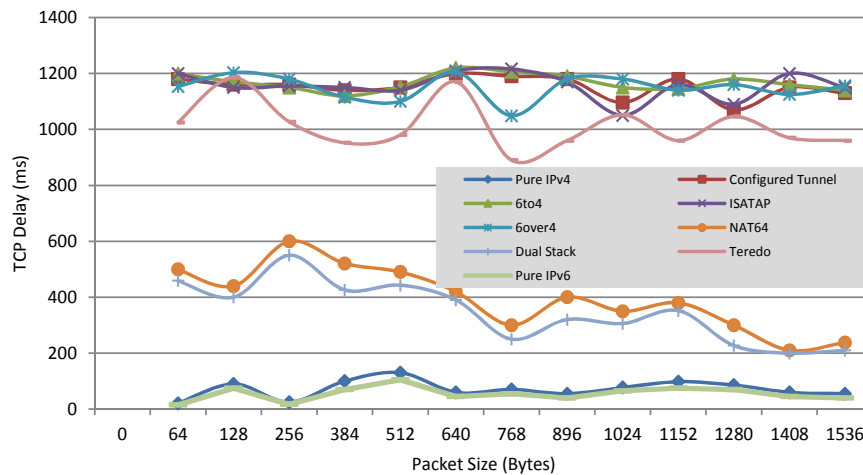


Figure 4.18: Graph of TCP delay implementing different transition mechanisms



#### 4.6. PERFORMANCE METRICS MEASUREMENT FOR VARIOUS TRANSITION MECHANISMS

TCP Delay: Transition Mechanisms													
	64	128	256	384	512	640	768	896	1024	1152	1280	1408	1536
Pure IPv4	0.02	0.07	0.02	0.08	0.11	0.05	0.06	0.05	0.06	0.08	0.07	0.05	0.05
Configured Tunnel	0.97	0.95	0.95	0.93	0.94	0.98	0.98	0.97	0.90	0.97	0.88	0.94	0.93
6to4	0.98	0.96	0.94	0.92	0.95	1.00	0.99	0.98	0.94	0.94	0.97	0.95	0.93
ISATAP	0.98	0.94	0.95	0.94	0.93	0.99	1.00	0.96	0.86	0.95	0.89	0.98	0.94
6over4	0.95	0.99	0.97	0.91	0.90	0.99	0.86	0.97	0.97	0.93	0.95	0.92	0.95
NAT64	0.41	0.36	0.49	0.43	0.40	0.34	0.25	0.33	0.29	0.31	0.25	0.17	0.20
Dual Stack	0.38	0.33	0.45	0.35	0.36	0.32	0.20	0.26	0.25	0.29	0.19	0.16	0.17
Teredo	0.84	0.97	0.84	0.78	0.80	0.96	0.73	0.79	0.86	0.79	0.86	0.80	0.79
Pure IPv6	0.01	0.06	0.01	0.06	0.09	0.04	0.05	0.03	0.05	0.06	0.06	0.04	0.03

Figure 4.19: Heat map of TCP delay implementing different transition mechanisms

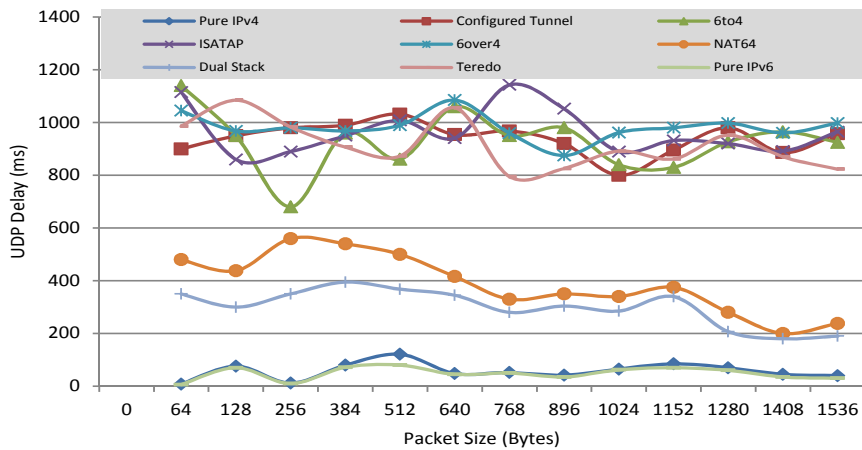


Figure 4.20: Graph of UDP delay implementing different transition mechanisms

The three bands that appear in the TCP graphs have a significant separation between them. Dual stack and NAT64, banded together, have values approximately two and half times more than networks without transition mechanisms, while all others have values 12-fold greater. Therefore, the choice of transition mechanism on networks that run delay sensitive applications is critical, otherwise there will be a significant impact on the quality of transmission.

Delay values attained for UDP traffic type are presented in Figures 4.20 and 4.21. Here again, a clear distinction can be seen, where groups of mechanisms are separated into three bands. The pattern is very similar to that of TCP delay, however all values are lower slightly lower than for TCP. At the bottom of the delay scale are the two networks without any transition mechanisms (average value less than 100ms), while in the middle are the dual stack and NAT64 lines averaging around 380ms. The top band is where all transition mechanisms are registering UDP delay values, and has an average of approximately 1000ms.

UDP Delay: Transition Mechanisms													
	64	128	256	384	512	640	768	896	1024	1152	1280	1408	1536
Pure IPv4	0.01	0.07	0.01	0.07	0.11	0.04	0.05	0.04	0.06	0.07	0.06	0.04	0.03
Configured Tunnel	0.79	0.83	0.86	0.87	0.90	0.83	0.85	0.80	0.70	0.78	0.86	0.78	0.84
6to4	1.00	0.83	0.59	0.84	0.75	0.93	0.83	0.86	0.73	0.73	0.81	0.84	0.81
ISATAP	0.98	0.75	0.78	0.83	0.88	0.82	1.00	0.92	0.78	0.82	0.80	0.78	0.84
6over4	0.91	0.85	0.86	0.85	0.87	0.95	0.84	0.77	0.84	0.86	0.87	0.84	0.87
NAT64	0.42	0.38	0.49	0.47	0.44	0.36	0.29	0.31	0.30	0.33	0.24	0.17	0.21
Dual Stack	0.31	0.26	0.31	0.35	0.32	0.30	0.24	0.27	0.25	0.30	0.18	0.16	0.17
Teredo	0.86	0.95	0.86	0.79	0.76	0.92	0.70	0.72	0.78	0.75	0.83	0.76	0.72
Pure IPv6	0.01	0.06	0.01	0.06	0.07	0.04	0.04	0.03	0.05	0.06	0.05	0.03	0.03

Figure 4.21: Heat map of UDP delay implementing different transition mechanisms

Jitter measurements for the different transition mechanisms are presented next. For the TCP traffic type (Figures 4.22 and 4.23) it can be seen that for lower packet sizes jitter values are good, however, larger packets generally give higher jitter values. This is the case for all transition mechanisms. For some packet sizes, pure IPv6 has the lowest jitter and for most scenarios the highest jitter value is for the packet size of 1152 bytes. TCP jitter values, in most cases, are below 1ms and distinct patterns cannot be identified to differentiate between the transition mechanisms.

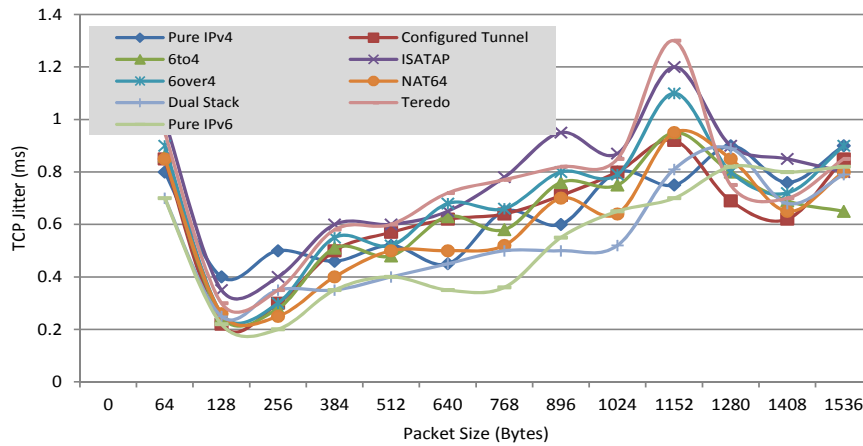


Figure 4.22: Graph of TCP jitter implementing different transition mechanisms

TCP Jitter: Transition Mechanisms													
	64	128	256	384	512	640	768	896	1024	1152	1280	1408	1536
Pure IPv4	0.62	0.31	0.38	0.35	0.40	0.35	0.50	0.46	0.62	0.58	0.69	0.58	0.69
Configured Tunnel	0.65	0.17	0.23	0.38	0.44	0.48	0.49	0.55	0.62	0.71	0.53	0.48	0.65
6to4	0.66	0.19	0.22	0.39	0.37	0.48	0.45	0.58	0.58	0.73	0.62	0.53	0.50
ISATAP	0.77	0.27	0.31	0.46	0.46	0.50	0.60	0.73	0.67	0.92	0.69	0.65	0.62
6over4	0.69	0.20	0.23	0.42	0.40	0.52	0.51	0.62	0.61	0.85	0.62	0.55	0.69
NAT64	0.65	0.20	0.19	0.31	0.38	0.38	0.40	0.54	0.49	0.73	0.65	0.50	0.62
Dual Stack	0.54	0.19	0.27	0.27	0.31	0.35	0.38	0.38	0.40	0.62	0.68	0.52	0.61
Teredo	0.73	0.23	0.27	0.45	0.46	0.55	0.59	0.63	0.65	1.00	0.58	0.54	0.65
Pure IPv6	0.54	0.17	0.15	0.27	0.31	0.27	0.28	0.42	0.50	0.54	0.63	0.62	0.63

Figure 4.23: Heat map of TCP jitter implementing different transition mechanisms

For UDP traffic, values are presented in Figures 4.24 and 4.25. The pattern here is very similar to that of TCP delay. Again, no clear distinction can be made between the mechanisms however, for most packet sizes, dual stack and NAT64 delay values band together with that of the pure IPv4/IPv6 networks. ISATAP and Teredo together are registering marginally higher values than all the other mechanisms.

It can be seen that all TCP and UDP values approximately range between 0.2ms and 1ms. UDP jitter values are marginally lower for all transition mechanisms. Also, NAT64 and dual stack have comparable values with that of networks with pure IP stacks. In the context of the graphs presented, the difference between the high values and that at the lower end of the band is approximately double, that is, for a given packet size both TCP and UDP jitter can be almost double between the different transition mechanisms.

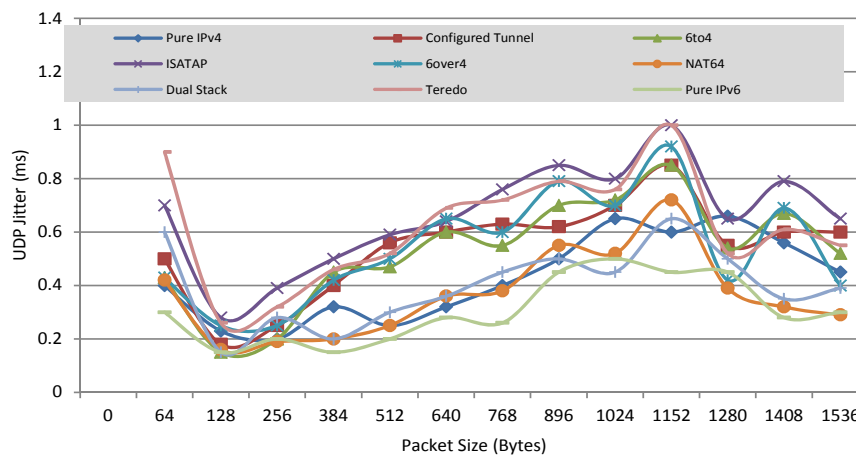


Figure 4.24: Graph of UDP jitter implementing different transition mechanisms

UDP Jitter: Transition Mechanisms													
	64	128	256	384	512	640	768	896	1024	1152	1280	1408	1536
Pure IPv4	0.40	0.23	0.20	0.32	0.25	0.32	0.40	0.50	0.65	0.60	0.66	0.56	0.45
Configured Tunnel	0.50	0.18	0.25	0.40	0.56	0.60	0.63	0.62	0.70	0.85	0.55	0.60	0.60
6to4	0.43	0.15	0.20	0.45	0.47	0.60	0.55	0.70	0.72	0.85	0.54	0.67	0.52
ISATAP	0.70	0.28	0.39	0.50	0.59	0.64	0.76	0.85	0.80	1.00	0.65	0.79	0.65
6over4	0.43	0.25	0.25	0.42	0.50	0.65	0.60	0.79	0.70	0.92	0.42	0.69	0.40
NAT64	0.42	0.18	0.19	0.20	0.25	0.36	0.38	0.55	0.52	0.72	0.39	0.32	0.29
Dual Stack	0.60	0.15	0.28	0.20	0.30	0.36	0.45	0.50	0.45	0.65	0.50	0.35	0.39
Teredo	0.90	0.26	0.32	0.46	0.52	0.69	0.72	0.79	0.76	1.00	0.52	0.61	0.55
Pure IPv6	0.30	0.15	0.20	0.15	0.20	0.28	0.26	0.45	0.50	0.45	0.45	0.28	0.30

Figure 4.25: Heat map of UDP jitter implementing different transition mechanisms

The final metrics measured was CPU usage on the sender (CPU1) and receiver (CPU2) nodes, results of which are presented in Figures 4.26 to 4.33. In all cases IPv4 and IPv6 pure networks used the lowest resources, while CPU usage ranged between 10 to 40% with transition mechanism. Although this metrics does not really distinguish between the different TCP and UDP characteristics on transition mechanisms, it does show that on networks with transition mechanisms, CPU usage increases drastically. This is evident in all cases, with both TCP and UDP portraying similar characteristics.

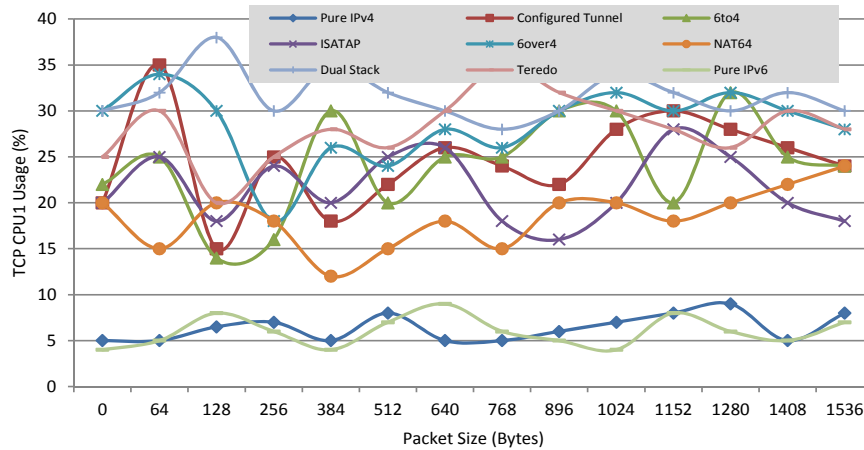


Figure 4.26: Graph of TCP CPU1 (sender) usage percentage implementing different transition mechanisms

TCP CPU1 Usage: Transition Mechanisms													
	64	128	256	384	512	640	768	896	1024	1152	1280	1408	1536
Pure IPv4	0.13	0.17	0.18	0.13	0.21	0.13	0.13	0.16	0.18	0.21	0.24	0.13	0.21
Configured Tunnel	0.92	0.39	0.66	0.47	0.58	0.68	0.63	0.58	0.74	0.79	0.74	0.68	0.63
6to4	0.66	0.37	0.42	0.79	0.53	0.66	0.66	0.79	0.79	0.53	0.84	0.66	0.63
ISATAP	0.66	0.47	0.63	0.53	0.66	0.68	0.47	0.42	0.53	0.74	0.66	0.53	0.47
6over4	0.89	0.79	0.47	0.68	0.63	0.74	0.68	0.79	0.84	0.79	0.84	0.79	0.74
NAT64	0.39	0.53	0.47	0.32	0.39	0.47	0.39	0.53	0.53	0.47	0.53	0.58	0.63
Dual Stack	0.84	1.00	0.79	0.92	0.84	0.79	0.74	0.79	0.89	0.84	0.79	0.84	0.79
Teredo	0.79	0.53	0.66	0.74	0.68	0.79	0.92	0.84	0.79	0.74	0.68	0.79	0.74
Pure IPv6	0.13	0.21	0.16	0.11	0.18	0.24	0.16	0.13	0.11	0.21	0.16	0.13	0.18

Figure 4.27: Heat map of TCP CPU1 (sender) usage percentage implementing different transition mechanisms

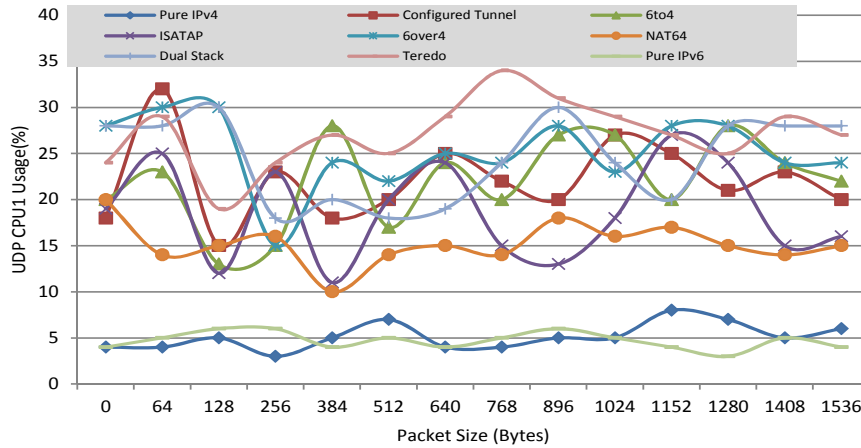


Figure 4.28: Graph of UDP CPU1 (sender) usage percentage implementing different transition mechanisms

UDP CPU1 Usage: Transition Mechanisms													
	64	128	256	384	512	640	768	896	1024	1152	1280	1408	1536
Pure IPv4	0.12	0.15	0.09	0.15	0.21	0.12	0.12	0.15	0.15	0.24	0.21	0.15	0.18
Configured Tunnel	0.94	0.44	0.68	0.53	0.59	0.74	0.65	0.59	0.79	0.74	0.62	0.68	0.59
6to4	0.68	0.38	0.44	0.82	0.50	0.71	0.59	0.79	0.79	0.59	0.82	0.71	0.65
ISATAP	0.74	0.35	0.68	0.32	0.59	0.71	0.44	0.38	0.53	0.79	0.71	0.44	0.47
6over4	0.88	0.88	0.44	0.71	0.65	0.74	0.71	0.82	0.68	0.82	0.82	0.71	0.71
NAT64	0.41	0.44	0.47	0.29	0.41	0.44	0.41	0.53	0.47	0.50	0.44	0.41	0.44
Dual Stack	0.82	0.88	0.53	0.59	0.53	0.56	0.71	0.88	0.71	0.59	0.82	0.82	0.82
Teredo	0.85	0.56	0.71	0.79	0.74	0.85	1.00	0.91	0.85	0.79	0.74	0.85	0.79
Pure IPv6	0.15	0.18	0.18	0.12	0.15	0.12	0.15	0.18	0.15	0.12	0.09	0.15	0.12

Figure 4.29: Heat map UDP of CPU1 (sender) usage percentage implementing different transition mechanisms



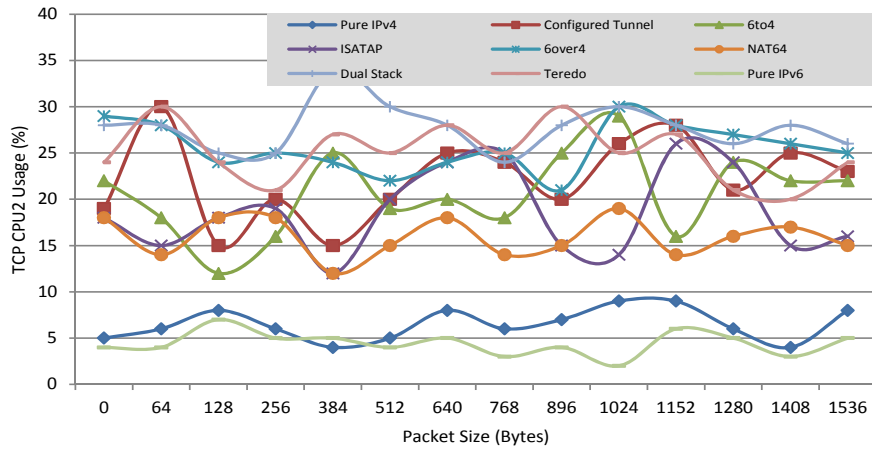


Figure 4.30: Graph of TCP CPU2 (receiver) usage percentage implementing different transition mechanisms

TCP CPU2 Usage: Transition Mechanisms													
	64	128	256	384	512	640	768	896	1024	1152	1280	1408	1536
Pure IPv4	0.18	0.24	0.18	0.12	0.15	0.24	0.18	0.21	0.26	0.26	0.18	0.12	0.24
Configured Tunnel	0.88	0.44	0.59	0.44	0.59	0.74	0.71	0.59	0.76	0.82	0.62	0.74	0.68
6to4	0.53	0.35	0.47	0.74	0.56	0.59	0.53	0.74	0.85	0.47	0.71	0.65	0.65
ISATAP	0.44	0.53	0.56	0.35	0.59	0.71	0.74	0.44	0.41	0.76	0.71	0.44	0.47
Gover4	0.82	0.71	0.74	0.71	0.65	0.71	0.74	0.62	0.88	0.82	0.79	0.76	0.74
NAT64	0.41	0.53	0.53	0.35	0.44	0.53	0.41	0.44	0.56	0.41	0.47	0.50	0.44
Dual Stack	0.82	0.74	0.74	1.00	0.88	0.82	0.71	0.82	0.88	0.82	0.76	0.82	0.76
Teredo	0.88	0.71	0.62	0.79	0.74	0.82	0.74	0.88	0.74	0.79	0.62	0.59	0.71
Pure IPv6	0.12	0.21	0.15	0.15	0.12	0.15	0.09	0.12	0.06	0.18	0.15	0.09	0.15

Figure 4.31: Heat Map of TCP CPU2 (receiver) Usage Percentage Implementing Different Transition Mechanisms

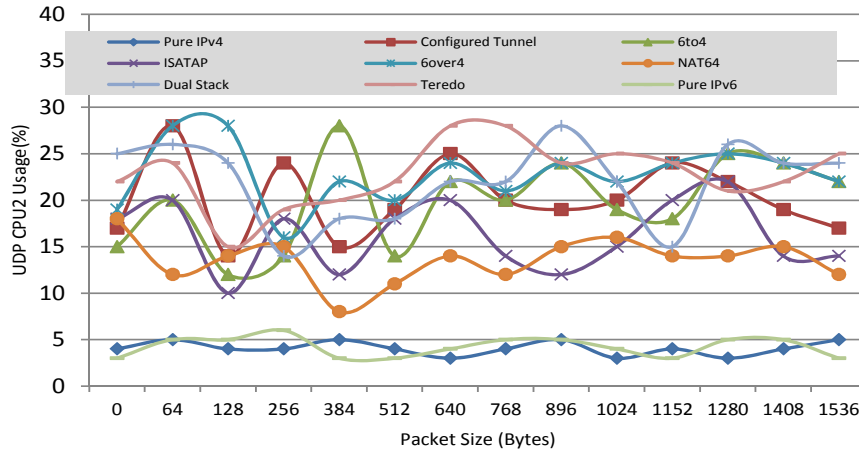


Figure 4.32: Graph of UDP CPU2 (receiver) usage percentage implementing different transition mechanisms

UDP CPU2 Usage: Transition Mechanisms													
	64	128	256	384	512	640	768	896	1024	1152	1280	1408	1536
Pure IPv4	0.18	0.14	0.14	0.18	0.14	0.11	0.14	0.18	0.11	0.14	0.11	0.14	0.18
Configured Tunnel	1.00	0.50	0.86	0.54	0.68	0.89	0.71	0.68	0.71	0.86	0.79	0.68	0.61
6to4	0.71	0.43	0.50	1.00	0.50	0.79	0.71	0.86	0.68	0.64	0.89	0.86	0.79
ISATAP	0.71	0.36	0.64	0.43	0.64	0.71	0.50	0.43	0.54	0.71	0.79	0.50	0.50
6over4	1.00	1.00	0.57	0.79	0.71	0.86	0.75	0.86	0.79	0.86	0.89	0.86	0.79
NAT64	0.43	0.50	0.54	0.29	0.39	0.50	0.43	0.54	0.57	0.50	0.50	0.54	0.43
Dual Stack	0.93	0.86	0.50	0.64	0.64	0.79	0.79	1.00	0.79	0.54	0.93	0.86	0.86
Teredo	0.86	0.54	0.68	0.71	0.79	1.00	1.00	0.86	0.89	0.86	0.75	0.79	0.89
Pure IPv6	0.18	0.18	0.21	0.11	0.11	0.14	0.18	0.18	0.14	0.11	0.18	0.18	0.11

Figure 4.33: Heat map of UDP CPU2 (receiver) usage percentage implementing different transition mechanisms

## 4.7 Results Evaluation

In the preceding section, results attained from the transition mechanism test-beds have been presented and general trends have been explained. This gives an interesting insight into the behaviour of TCP and UDP traffic as they traverse networks with different transition mechanisms. In relation to the four performance metrics measured in each scenario, there are a number of specific observations of interest. These are discussed herewith.

There is a slight difference in throughput between IPv4 and IPv6. As discussed in Chapter 2, there are many advantages that the new version offers over the predecessor, however, the throughput values attained in all tests show that network performance slightly downgrades on networks implemented with IPv6. This is the case for both TCP and UDP, with the difference being just slight.

The throughput values attained in the test-bed tests show that there are significant differences between the various transition mechanisms. All mechanisms definitely have an impact on the throughput, but to varying degrees. For TCP traffic, Teredo and ISATAP mechanisms register the highest throughput drop (almost 30%) while the rest show approximately 8%. For UDP traffic type, Teredo and 6over4 have the greatest impact on performance degradation, where the maximum drop is approximately 35%. In all cases, transition mechanisms significantly impact network performance, irrespective of the traffic type. Of the seven mechanisms installed, five were tunnelling techniques, and in relation to their throughput values, differences have also been identified for both the protocols.

TCP and UDP delay values are good differentiators of the various transition mechanisms. As mentioned earlier, this is a significant metric since there are many delay sensitive applications currently being used on networks. It is seen that networks that do not have transition mechanisms have low delay, however, for all transition mechanisms tested, there is a remarkable increase in the values, irrespective of the actual traffic type. However, generally UDP

delay values are slightly lower. Some mechanisms increase delay at least three-fold, while others are registering a 10-12-fold increase. Dual stack and 6to4 increase the value the least while all the others have a significant increase.

Jitter measurements do not show any clear distinction between the TCP and UDP protocols, however, there is some indication that the actual transition mechanism implemented does impact traffic jitter to a certain extent. In the band of all values, there is a difference of approximately 0.2ms for most packet sizes. This is the case for both TCP and UDP, however, in UDP jitter values, NAT64 and dual stack stand out from the rest for a few packet sizes. The values for these are comparable with IPv4 and IPv6 networks without transition mechanisms.

Similar to jitter, CPU usage on the sending and the receiving nodes does not distinctly show a difference between the mechanisms, however, it does show that they add substantial overheads to the communicating nodes. Increases of between 10 to 30% is evident, while that on networks with the transition mechanisms is around 5%.

The test-bed results presented in this chapter gives an insight into the network performance of seven transition mechanisms. Of interest in this chapter has been the difference between the transition mechanisms with TCP and UDP traffic traversing the test-bed - this has been highlighted in the discussions. Since all performance metrics highlight the difference between the mechanisms and the protocols, they can be used as a criterion for creating an ordinal ranking of the mechanisms. This will be discussed later in the thesis, in Chapter 7.

## Chapter 5

---

# TCP and UDP Behaviour in Wireless Environments

### 5.1 Introduction and Motivation

In the last decade, the world has become increasingly mobile and consequently traditional ways of creating networks have proven to be inadequate. Traditional cabled networks reduce mobility, whereas there is no such limitation with wireless connectivity. There is also an increased drive towards the concept of Bring Your Own Device (BYOD) and connecting seamlessly to different network infrastructures. Wireless networking technologies are slowly encroaching on the realms of traditional, fixed, wired networks. Although this technology is versatile, it currently dominates only in the *last mile data delivery* on network infrastructures. Consequently, a wired-to-wireless network is a common occurrence.

In this chapter, TCP and UDP end-to-end network performance issues are examined as data traverses a heterogeneous network comprising of wired and wireless links. On a test-bed, the last mile connection wireless is set up and common performance related metrics are measured and analysed. This is all done specifically within the IEEE 802.11 WLAN's three latest standards. A description of wireless techniques are presented, which is followed by a discussion relating to wireless standards. Wireless security protocols are presented followed by an outline of the experimental test bed configuration and a presentation of the test bed results.

## 5.2 Wireless Preamble

The Internet has been growing at a phenomenal rate and become a huge mesh of interconnected sub-networks, one that is increasingly becoming a heterogeneous environment. The unprecedented growth of computer networks and the heterogeneity in its components has fundamentally become a concern for a legacy protocol like TCP (Niehenke, 2014). What began as a network created using just copper wires, has now evolved into a global network, that in addition to using typical wires, is employing state-of-the-art fibre optics and numerous wireless technologies such as infrared, satellites and radio waves. Subsequently, today there is a new class of connected end-users who remains linked to networks without the need for a physical medium (wire) between them and a network's gateway. This is the mobile and wireless network revolution.

Wireless technologies continue to play an ever increasing role in all aspects of network communication. This field has also attracted a great deal of interest from researchers and developers, thus has expanded rapidly in the last two decades (Petersen & Carlsen, 2008); (Hengstler & Aghajan, 2006). This is evident with the continuous development of new IEEE802.11 standards and the dominant penetration of wireless broadband Internet access in all aspects of our lives at home and in business establishments. In relation to indoor wireless local area networks (WLANs), the predominant use of this is currently for the last mile connection, that is, it enables users at the edge of the Internet to remain connected to the infrastructure via some network but without the use of traditional cables (Dimitrakopoulos, Demestichas, & Koenig, 2010) (Aslam, Guinard, McGibney, Rea, & Pesch, 2011). Mostly radio frequencies are used on WLANs as the foundation technology.

Nowadays, a typical network is one that has both wired and wireless parts. Generally, the network backbone and core are wired, and the last hop portion is normally wireless. Hence data travelling on networks often traverses a wireless link prior to arriving at a destination node. This is convenient for the end users, however, it is a well-known fact that wireless communication

channels, when compared with their counterparts, are notoriously unreliable (Chan & Baciú, 2012). This is mainly due to the general characteristics of the associated wireless technologies, including the use of radio waves as the communication link (Wells, 2009). In addition to this, it has been shown in numerous studies that due to variability in technologies and unpredictability in conditions, wireless transmissions are the bottleneck when it comes to performance issues on a network (Chan & Baciú, 2012) (Xylomenos, Polyzos, Mahonen, & Saaranen, 2001). For this particular reason, researching the interaction between TCP/UDP and IEEE802.11 WLAN technologies has been a focus of much research in recent times (Huston, 2001) (Xylomenos & Polyzos, 1999) (Balakrishnan, Padmanabhan, Seshan, & Katz, 1997). However, the bulk of these studies does not necessarily look at the last-mile connection issues related to network performance, but are mainly focusing on the sending-side (that is, primarily the servers) in communication channels.

In a heterogeneous environment, as data moves between wired and wireless sectors, it is important to realise that network performance will be of concern. Communication protocols, like TCP and UDP, establish end to end connectivity between the sender and the receiver, irrespective of the type of communication media in the path. This non-discrimination leads to the protocol not being able to distinguish between wired and wireless paths. This is good news because this transparency implies communication protocols can function without the knowledge of the intricacies of the actual transmission path. However, data losses in wireless paths is much greater than that on wired links, and communication protocols are not able to differentiate between the implications of such an unreliable medium. Thus in the event that the wired and wireless links have differing reliability, the communication protocols react to both networks the same. This matters since a protocol, like TCP, is designed to deal with wired losses by reducing its sending rates, but for a wireless network this throttling back is not necessary since it may be the unreliability of the link (and not congestion). So instead of wireless data losses that can generally be recovered locally using numerous error checking and recovery techniques, a solution that reduces data transfer rates unnecessarily is applied by default.

### 5.3 Wireless Standards

Networking standards by IEEE are generally designated with standard numbers starting with 802, and the 11 family of the standards relates to wireless local area networks. IEEE802.11 standard is a set of both Media Access Control (MAC) and Physical Layer (PHY) specifications that are used to implement WLANs using radio frequency bands (Saliga, 2000). Back in 1997, the first standard of wireless IEEE802.11 was released. This supported a maximum speed of only 2Mbps, which is insufficient for most applications currently being used, and for this reason, 802.11 wireless products cease to exist. Initially, it was deployed only in vertical applications, such as inventory management, point of sale and transportation management, but later was used to experiment with enterprise type networks.

RFID		Sensors		ZigBee Alliance 802.15.4
<b>Wide Area Network (WAN)</b>				
802.16e Nomadic	802.20 Mobile	802.21 Handoff	802.22 WRAN	2, 2.5, 3, 4G Cellular
<b>Wide Area Network (WAN)</b>				
802.16 WiMax				
<b>Local Area Network (LAN)</b>				
802.11 Wi-Fi (a,b,g,n,ac)				
<b>Personal Area Network (PAN)</b>				
802.15 Bluetooth	8020.15.3		802.15.4 ZigBee	

Table 5.1: Wireless networks and their corresponding IEEE 802.11 protocols

The original standard used either infrared signals or the Industrial Scientific Medical (ISM) frequency band at 2.4GHz and also defined the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) as its media access mechanism (T.-S. Ho & Chen, 1996). CSMA/CA technique, allows maximum capacity on a channel to about 65% of the bandwidth, mainly due to the overheads of error correction and error checking. All WLAN technologies today operate using the ISM band, which has been defined as three unli-



censed bands in the range 902-928MHz (Ultra High Frequency (UHF) band), 2.400-4.835GHz (S-Band) and 5.725-5.850GHz (C-Band). A major weakness of 802.11 was that it offered a lot of choice for manufacturers and designers of products, thus interoperability between different vendor items was always a challenge.

In 1999, two new standards for WLANs were released, one of which was IEEE802.11a. This higher bandwidth standard permitted data rates up to 54Mbps and used the 5GHz radio frequency range (Kapp, 2002). The actual release of products based on IEEE802.11a happened in 2001, mainly due to lack of development in radio frequency bands. It used Orthogonal Frequency-Division Multiplexing (OFDM) for signal generation. In the main, OFDM divides the 5GHz sending band into 53 sub bands, with 48 sub bands for sending and 4 for control information (Kapp, 2002) (Doufexi et al., 2002) ("Supplement to IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High-Speed Physical Layer in the 5 GHz Band", 1999). The use of sub bands reduces the effects of signal interference and enhances transmission security.

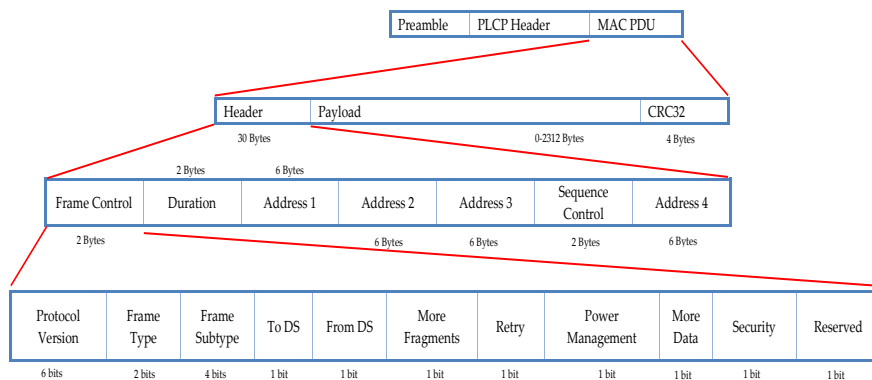


Figure 5.1: The 802.11 frame format structure

### 5.3.1 IEEE802.11b

IEEE802.11b rectified all the amendments to the original standard and was released in 1999, together with the IEEE802.11a standard. However, since this standard used the unprotected 2.4GHz frequency range and a lot of development had already been undertaken in this frequency range, products based on this standard were released into the market much sooner than the actual release date of the standard (Bousquet, Messier, & Magierowski, 2007) (“Supplement to IEEE Standard for Information Technology- Telecommunications and Information Exchange Between Systems- Local and Metropolitan Area Networks- Specific Requirements- Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band”, 2000). Unlike (IEEE802.11a, IEEE802.11b) uses the same frequency range as that of the original IEEE802.11, but has an increased maximum data rate of 11Mbps (Hoefel, 2008). This much needed increased bandwidth and substantial price reduction (Celebi, Dericiogullari, & Bitirim, 2007) in IEEE802.11b devices allowed developers to utilise typical office applications over wireless. This led to a wide acceptance of WLAN technology and at that stage 802.11b became the definitive standard for WLAN.

802.11 Protocol	Year Introduced	Frequency (GHz)	Bandwidth (MHz)	Modulation Technique	Theoretical Maximum Data Rate (Mbps)
802.11a	1999	5	20	OFDM	54
802.11b	1999	2.4	22	DSSS	11
802.11g	2003	2.4	20	OFDM, DSSS	54
802.11n	2009	2.4, 5	20, 40	OFDM	300
802.11ac	2013	5	20, 40, 80, 160	OFDM	600

Table 5.2: Different wireless standards and their specifications

IEEE802.11b uses High-Rate Direct Sequence Spread Spectrum (HR-DSSS) for radio wave signal generation. This method (Jemai, Piesiewicz, & Kurner, 2005) is very similar to traditional DSSS, however, the actual encoding mechanism uses Complementary Coding Key (CCK) (Pursley & Royster, 2007) (Pursley & Royster, 2009). CCK takes 4 or 8 bits of data and encodes it into one CCK unit. Thus the HR-DSSS CCK modulation mechanism can revert to low data rates, if necessary, to allow backward compatibility. In the main, IEEE802.11b allows a data transfer rate of 11Mbps, but can scale back to either 1, 2, or 5.5Mbps (Saliga, 2000). At lower transmission speeds less complex and more redundant methods of data encoding are used, so data is less susceptible to corruption due to the vulnerabilities of wireless transmission technology (de Carvalho, Veiga, Marques, Pacheco, & Reis, 2010). Irrespective of all these, from today's stand point, 802.11b has a low maximum speed (Garg & Kappes, 2003) and its use of an unregulated frequency band implies interference from home and office appliances.

### 5.3.2 IEEE802.11g

In 2003, IEEE ratified 802.11g as the third wireless modulation standard. This attempted to combine the best of 802.11a and 802.11b, by offering support for a maximum throughput of 54Mbps using 2.4GHz frequency for increased coverage range. Being backward compatible with its predecessor, allowed the use of 802.11g equipment on the 802.11b infrastructure during the migration phase (T. Wang & Refai, 2005). Dual-band devices (that support both 802.11a and 802.11b) started appearing in the market as dual-band/tri-mode, supporting all 802.11a, b and g in one single device (Celebi et al., 2007) (Drilo & Flatz, 2003). As was the case with the previous version, actual products with this standard came onto the market well before the ratification date. IEEE802.11g wireless signal has a great maximum speed and is not easily susceptible to typical wireless obstructions. It uses OFDM and CCK technology for mandatory modulation schemes, giving it an upper mandatory data rate of 24Mbps, although it also uses other schemes to maintain compatibility (Drilo & Flatz, 2003) (M.-J. Ho, Wang, Shelby, & Haisch, 2003) (Issac, Hamid, & Tan, 2006). However, since it also has optional components, it can provide optional higher data rates of 36, 48, and 54Mbps. Although 802.11g

is backward compatible, the presence of any 802.11b device on the network significantly, negatively impacts on the network speed.

For this standard to maintain backward compatibility, 802.11g uses a DSSS modulation technique. The mechanism is complex, but in the main it uses four different physical layers, three of which are described as Extended Rate Physical (ERP) and one which is DSSS-OFDM (S.-C. Wang, Chen, Lee, & Helmy, 2005). All of these four coexist at the sender end during initial frame exchange, and depending on what is supported at the other end of the communication link, the sender is able to use any one of the four to initiate backward compatibility mode communication.

For its time, 802.11g was compelling because it gave a five-fold increase in WLAN speed over its predecessor and offered great range and coverage (S.-C. Wang et al., 2005). Its backward compatibility was also a bonus.

### 5.3.3 IEEE802.11n

The previous 802.11 standards were again improved, and in 2009 the IEEE802.11n WLAN standard was introduced, a standard that took seven years of development by the IEEE TGN task group. This amendment improved WLANs (Lator, Torfs, & Blondia, 2012) (Lim, Kim, & Suh, 2012) (Paul & Ogunfunmi, 2008) significantly since it added Multiple-Input Multiple-Output (MIMO) to its mechanism, together with Spatial-Division Multiplexing (SDM), Space-Time Block Coding (STBC) and transmitter beam forming (Lim et al., 2012) (Paul & Ogunfunmi, 2008) (Perahia, 2008). These effectively enhanced the maximum achievable speed from 54Mbps to 600Mbps, using four spatial transmission streams at a channel width of 40MHz. In addition to MIMO, frame aggregation and security improvements were also introduced, whilst still enabling transmission using 2.4GHz and 5GHz frequency bands (Lim et al., 2012) (Perahia, 2008). These techniques theoretically increase speed more than ten-fold over the maximum data rate possible with the 802.11a/g standards.

The increase in speed and range is evident in the 802.11n standard. This has

been achieved by enhancements in both the PHY and MAC layers. Frame aggregation, enables the media to send multiple MAC frames in one PHY packet, reducing the associated overheads. Another key enhancement is related to the introduction of a Reverse Direction (RD) protocol. This effectively allows the transmitting device currently holding the transmission channel to seamlessly transfer control to another device, without requiring the second device to initiate a data transfer. For delay sensitive applications, such as multimedia streaming and Voice-over WLAN (VoWLAN) (Lim et al., 2012), 802.11n has inbuilt QoS features, allowing it to prioritise network traffic (Kolahi, Cao, & Chen, 2013). If handheld devices are using the 802.11n standard, its Power Save Multi-Poll (PSMP) feature conserves battery markedly. In addition, 802.11n has an extended channel switch announcement, allowing an access point to switch between different supported channels (Perahia, 2008). There has also been an improvement in radio resource management which enables 802.11n based access points to function more efficiently in an environment where there are multiple access points functioning together on one network. There is also improved handoff between base stations, which enables better use of VoIP technology on mobile phones on wireless networks.

IEEE802.11n has had resounding market success. Initially, devices appeared with draft standards, however, when the 802.11n proper was released, over 100 devices were released in the first few months, three times as many as with the 802.11a, b, or g standards (Perahia, 2008). This was the first time wireless networks became capable of hundreds of megabits of data transfer, as opposed to just a few megabits per second.

#### **5.3.4 IEEE802.11ac**

The latest wireless standard released in early 2014, is IEEE802.11ac, which took approximately two years to develop, frame format shown in Figure 5.2. This standard provides Very High Throughput (VHT), with data rates in excess of 1Gbps and uses the 5.8 ISM radio band. It also claims that a data rate of up to 7Gbps is possible (L. Verma, Fakharzadeh, & Choi, 2013). For the first time, Multi-User MIMO (MU-MIMO) technology has been em-

ployed (L. Verma et al., 2013). This enables increased total capacity in situations where a single access point is sharing wireless channels with a number of other wireless stations. MU-MIMO allows simultaneous transmission of data from an access point to the connecting devices this is in contrast to the time sharing schemas used in CSMA technology that were employed in the earlier standard (L. Verma et al., 2013) (Ong et al., 2011). It is expected that the use of 802.11ac will become dominant in the market space and that by 2015, there will be close to one billion devices in the world using this VHT standard.

There are a number of key improvements that really differentiates 802.11ac from its predecessors. IEEE802.11n supported the use of only two channels (20 and 40 MHz) while the new standard supports 20, 40 and 80, and has the option to support 160MHz as well (Hoefel, 2013) (L. Verma et al., 2013). The 160MHz channel is available with both contiguous and non-contiguous types allowing flexible channel assignment (Hoefel, 2013). Larger channel sizes are desired since they increase data transfer rates. In relation to spatial streams, 802.11n was capable of handling a maximum of four streams, while 802.11ac has increased this to eight streams. With this, beamforming has also been enhanced, enabling the wireless antennas to focus the transmission of radio frequencies to where they are required, unlike earlier uni/omnidirectional type antennas. Since beamforming is based on explicit channel measurements, both the transmitter and the receiver must support the technology. In addition to supporting the traditional 64 Quadrature Amplitude Modulation (QAM), the new standard also supports 256QAM as an optional mode (C.-W. Huang et al., 2012). There are also three optional features that have been inherited from 802.11n: Low-Density Parity-Check (LDPC) code, STBC and Short Guard Interval (SGI) of 400ns. These technical enhancements together have increased data transfer rates approximately ten fold from what was offered by earlier wireless standards.

Wireless standards continue to evolve, and there is a juggling act between the maximum amount of data it can transmit and the size of the coverage area. Higher frequency (5GHz versus 2.4GHz) implies greater bandwidth, and greater attenuation. Radio waves attenuate exponentially over distance, therefore, since 802.11ac is using 5GHz frequency, it will undoubtedly deliver increased data rates, however, the coverage area may be significantly smaller than that expected when using 2.4GHz devices. Wireless Gigabit (WiGig) is the next big evolutionary step in developing wireless standards (Vaughan-Nichols, 2010) (Hansen, 2011). The next proposed standard IEEE802.11ad, will be based on this and will be using a 60GHz frequency band. This is just the beginning of an exciting technical roadmap progression that will take WiGig to new heights using proven techniques, as well as new ones, yet to be discovered.

Enhancing data transfer rates and coverage area is important, however equally important is securing data that is being transmitted on connection media. On a wireless network this is more of an issue than with wired transmissions, giving rise to the development of security protocols.

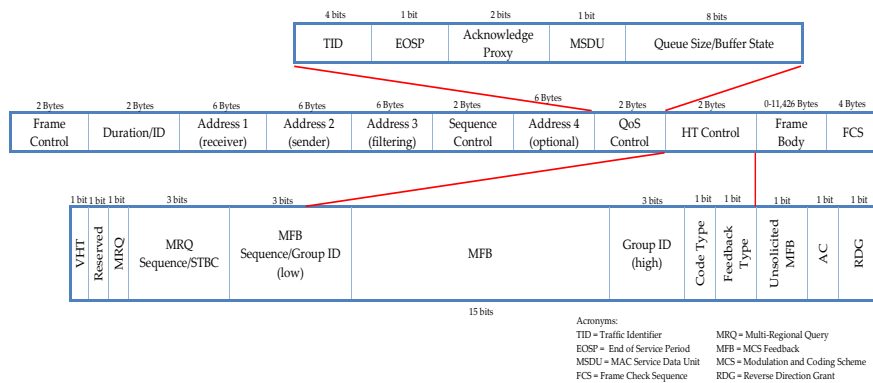


Figure 5.2: The 802.11ac frame format structure

## 5.4 Wireless Security Protocols

Wireless communication security is more of a concern than wired since there is no inherent physical protection between devices. The physical connection has been replaced by logical associations using radio, which by nature uses broadcast for transmission. In such an environment common security threats, such as eavesdropping, injecting bogus messages, jamming, replaying attacks and Denial of Service (DoS) can be easily mounted (Nisbet, 2012). To protect from these and to give confidentiality, authenticity and integrity to data travelling via wireless, wireless security protocols have been developed. Wireless security protocols are presented in Table 5.3.

	WEP	WPA	WPA2
<b>IV Length</b>	24 bits	48 bits	48 bits
<b>Authentication</b>	N/A	IEEE 802.1X/EAP/PSK	IEEE 802.1X/EAP/PSK
<b>Cryptographic Algorithm</b>	RC4	RC4	AES
<b>Data Integrity</b>	CRC32	MIC	CCM
<b>Encryption Method</b>	WEP	TKIP	CCMP
<b>Key Size</b>	40 – 104 bits	128 bits	128 bits
<b>Keys for Packets</b>	No	Yes	Yes
<b>Key Management</b>	Manual key rotation	Per packet key rotation	Per packet key rotation (TKIP) Per session key rotation (AES-CCMP)

Table 5.3: Different wireless security mechanisms and specifications

### 5.4.1 Wired Equivalent Privacy

Wired Equivalent Privacy (WEP) was the first IEEE802.11 specification that attempted to secure wireless communication. Developed in 1999 by the Wi-Fi alliance, its aim was to make wireless networks at least as secure as a wired local area network, that is, it was never intended to achieve strong security (Lashkari, Mansoor, & Danesh, 2009) (Lashkari, Towhidi, & Hosseini, 2009) (Maple, Jacobs, & Reeve, 2006). In essence, WEP aimed to provide access control to the network, message confidentiality, and integrity. For controlling access, the connecting station first needs to authenticate itself with the access



point that it wants to establish communication with. This is achieved using a simple challenge-response protocol between the two, at the end of which communication is established based on the success or failure of the authentication. Actual authentication is one way only, that is, the access point is not authenticated to the connecting station, but the station is to the access point. In relation to confidentiality, WEP uses the Rivest Cipher 4 (RC4) stream cipher. Here, for each message sent via wireless, RCA is initialised and a pseudo-random byte sequence (key stream) is generated (Lashkari, Mansoor, & Danesh, 2009) (Lashkari, Towhidi, & Hosseini, 2009). This sequence is XORed with the message to generate the encrypted message. In this schema (Figure 5.3), it is essential that each message is encrypted with a different key stream. To ensure that data integrity is maintained, WEP does this using protection based on an encrypted Cyclic Redundancy Check (CRC)(Schweber, 1992) value operation (Lashkari, Towhidi, & Hosseini, 2009) (Borsc & Shinde, 2005). Here, an Integrity Check Value (ICV) is generated and appended to the message prior to the encryption before transmission.

The introduction of WEP was the first attempt to secure WLAN communications. However, soon there were a number of vulnerabilities found in its mechanism. WEP goals were ill-defined from the offset, thus, there were flaws just waiting to be exploited. Having only one way authentication meant that a connecting station could associate with a rogue access point; using the same shared key for both authentication and encryption was another major weakness. During authentication between the connecting nodes, no sessions are established so once the process completes, it is possible for a potential attacker to spoof the MAC address of the connecting station and continue communication, that is, the connection station can be impersonated (Reddy, Sai Ramani, Rijutha, Ali, & Reddy, 2010) (Maple et al., 2006). WEPs confidentiality mechanism is also flawed since it uses weak RC4 keys for the seed values; therefore the beginning RC4 output is not really random and reveals a lot of secret information about the key that is used. WEP is susceptible to passive attacks (based on traffic statistical analysis), active attacks (based on traffic injections and impersonation) and also dictionary-building brute force attacks.

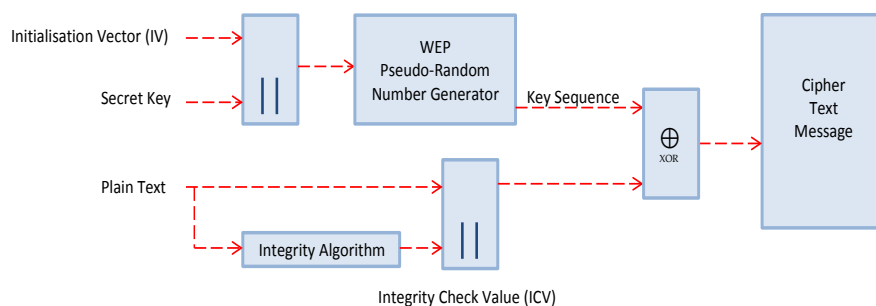


Figure 5.3: WEP encryption block diagram

### 5.4.2 Wi-Fi Protected Access

Due to a number of apparent vulnerabilities in WEP, Wi-Fi Protected Access (WPA) was formally adopted in 2003 as the replacement for its predecessor. WPA, also referred to as the draft IEEE802.11i, has the facility to be installed on WLANs with just a firmware upgrade on WEP capable Network Interface Cards (NICs), however, it does require totally new access points, since there are major changes required in them (Maple et al., 2006) (Bohn, Grob, Nubgen, & Schwann, 2006). Commonly WPA was implemented as WPA-Pre-Shared Key (PSK), which uses keys of length 256bits, a significant change from the 64 and 128bit keys that were used in WEP implementations.

There were a number of major changes implemented in WPA. Encrypted Message Integrity Checks (MIC) were incorporated in WPA to ascertain if an attacker had captured or altered data packets between the communicating node and the access point (MIC has replaced CRC). This reduces the chance of a DoS and spoofing type attacks. The encryption algorithm has been improved to the Temporal Key Integrity Protocol (TKIP) (Figure 5.4), which supplies each connecting host with a much longer unique key that gets rotated at a configurable interval (Lashkari, Mansoor, & Danesh, 2009) (Selim, El Badawy, & Salam, 2006). TKIP uses a per-packet key, which dynamically generates a 128bit key for each packet, resulting in negating attacks that normally compromised WEP (Lashkari, Mansoor, & Danesh, 2009).

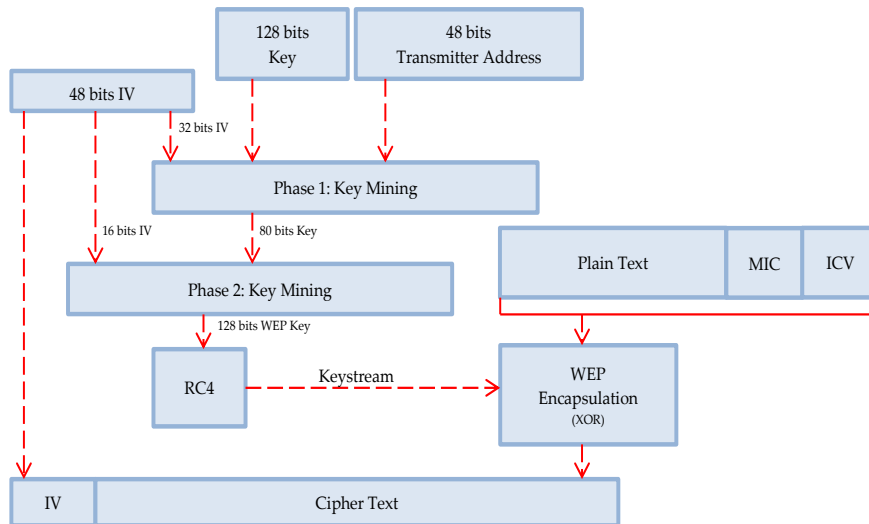


Figure 5.4: TKIP encryption block diagram

For authentication enhancement, WPA can be used together with a RADIUS server to implement WPA-Enterprise. In this scenario, WPA uses 802.1X+EAP for authentication while the RADIUS server eliminates the need for using pre-shared keys (Y. Wu, Zhu, Kong, & Yi, 2009). Such a setup is desirable since it can be easily integrated with the Windows login process.

Despite significant improvements in WPA over WEP, the new WLAN security protocol has already been exploited. Different approaches have been employed to circumvent WPA security, showing that there are numerous vulnerabilities in its design (ref 1, ref 2, ref 3). Eventually in 2010 it was proved that nearly all traffic going towards a WPA enabled WLAN client can be decrypted by using fragmentation and injection of an arbitrary amount of packets in the data stream.

### 5.4.3 Wi-Fi Protected Access 2

WPA2 is the most recent of the wireless encryption algorithms and has been in existence since 2004. WPA, the predecessor, was always thought of as an intermediate measure until a new protocol was made available, hence WPA2. Also known as IEEE802.11i-2004, it provides much stronger data protection and network layer access control than WEP and WPA (Li, Kolahi, Safdari, & Argawe, 2011). One of its major significant changes has been the mandatory use of the Advanced Encryption Standard (AES) algorithm and counter cipher mode with block Chaining Message Authentication Code Protocol (CCMP), which together has replaced TKIP (Kolahi, Singla, Ehsan, & Dong, 2011). WPA2 provides wireless security ranging from small home installations to government grade security on large implementations.

WPA2 by default provides support for WPA mechanism. However, in addition to this, WPA2 has a number of other additional benefits and features. It provides strong authentication and encryption support for both infrastructure and ad-hoc implementations. In comparison to this, WPA services for strong encryption, is limited to infrastructure networks only. There is also support for key caching - this mechanism reduces overheads on network nodes that are roaming between different access points. Also, WPA2 supports pre-authentication, that is, it capable of completing an authentication exchange between a wireless node and an access point prior to initiation of roaming.

The security provided by implementing WPA2 is robust. This has been achieved by using an authentication services that uses a four-way handshake mechanism to authenticate wireless stations and nodes during the first stage of the communication process.

## 5.5 Some Key Research

There are numerous researches that have been undertaken in the area of wireless networks. This is mainly due to the diversity in the topic. In relation to IEEE802.11, some recent research are mentioned in Table 5.4. In all these research undertakings, some aspect of wireless networks have been performance analysed on test-bed implementation. This approach is similar to what has been used in this thesis undertaking.

Author(s)	Title	Research Focus
José A. R. Pacheco de Carvalho, H. Veiga, Cláudia F. F. P. Ribeiro Pacheco, and A. D. Reis (2014)	Experimental Performance Studies of Laboratory WPA IEEE 802.11b,g PTMP Links	IEEE 802.11b, IEEE 802.11g, WPA
Upendra Singh and Poonam Jindal (2014)	Performance Analysis of Secure Wireless Local Area Network using Test-bed	IEEE 802.11n, WPA, and WEP
Roger Pierre Fabris Hoefel (2014)	IEEE 802.11ac: A Performance Evaluation with Lattice-Based MMSE and Zero Forcing MIMO OFDM Receivers	IEEE 802.11ac
Vincent Picard and Eric Lafond (2014)	Performance Evaluation of Next Generation Wi-Fi (802.11ac) for Mobile Offloading	IEEE 802.11ac
Saad Saleh, Zawar Shah, and Adeel Baig (2013)	Capacity Analysis of Combined IPTV and VoIP Over IEEE 802.11n	IEEE 802.11n
M. Hasbullah Mazlan, Sharifah H.S. Ariffin, Mohammed Balfaqih, S. Norhaizum M. Hasnan, and Shariq Haseeb (2012)	Latency Evaluation of Authentication Protocols in Centralized 802.11 Architecture	WPA2
Naeem Khademi, Michael Welzl, and Stein Gjessing (2012)	Experimental Evaluation of TCP Performance in Multi-rate 802.11 WLANs	IEEE 802.11g
J. A. R. Pacheco de Carvalho, Cláudia F. F. P., Nuno Marques, and H. Veiga (2011)	Comparative Performance Studies of Laboratory Wi-Fi IEEE 802.11 b, g WEP Point-to-Point Links	IEEE 802.11b, IEEE 802.11g, and WEP
J. A. R. Pacheco de Carvalho, H. Veiga, N. Marques, C. F. F. Ribeiro Pacheco, and A. D. Reis (2011)	Performance Measurements of IEEE 802.11 b, g Laboratory WEP and WPA Point-to-Point Links using TCP, UDP and FTP	IEEE 802.11b, IEEE 802.11g, WEP, WPA
Emilija Miletic, Klaus Tittelbach-Helmrich, and Goran Panic (2011)	Performance Investigation on an MIMO capable 802.11aCompliant MAC Protocol Implementation	IEEE 802.11a

Table 5.4: Recent research on performance evaluation of wireless standards

## 5.6 Wireless Test-bed Setup

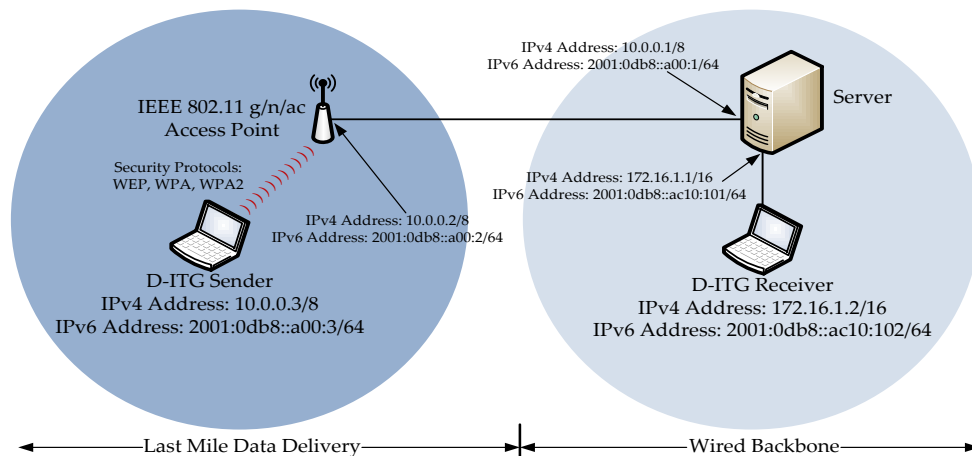


Figure 5.5: Wireless network diagram

Figure 5.5 shows the network that was implemented in a laboratory environment in which wireless networks were tested. In this, there is a wired connection between the wireless access point and a server, and a wireless connection between a client node and the access point. Since D-ITG was used to generate data and to measure various network performance metrics, another node was connected to the server to act as D-ITG receiver.

The access point and the wireless nodes were configured with various wireless standards and encryption techniques. On completing each configuration and testing it for functionality, D-ITG session was initiated and various performance related metrics were collected. With wireless networks, unlike with transition mechanism, TCP and UDP drop rates were also measured. However, all TCP drop rates registered were zero. Data attained from the wireless test-bed is presented and discussed in the next section.

## 5.7 Performance Metrics Measurements from Wireless Test-beds

To compare behaviour of TCP and UDP traffic types on wireless implementations, a test-bed of wireless devices was implemented. Various wireless standards, encryptions and IP versions were implemented and for each combination, throughput, and delay were measured. The first and the second set of measurements that were taken were for pure IPv4 and pure IPv6 wired networks respectively. Here, all nodes were stacked with either IPv4 or IPv6, and then appropriate performance metrics were measured. Later the wired network was removed, and on the same test-bed various wireless combinations were implemented. This was done with a *wireless infrastructure* type network using an access point. The test-bed used in these evaluations was capable of transferring data at gigabit speeds.

In Figure 5.6, TCP throughput values are presented. To clearly see the difference between the wireless options, wired results have been omitted in this graph. However, it is worth noting wired values attained plateaued at approximately 450Mbps. Evaluating just the networks with different variations of wireless, it is evident that the IEEE802.11ac network, without encryption, gives the highest throughput values. Whilst the theoretical upper bound for this is significantly higher (discussed earlier in section 5.3.4), the values attained on the test-bed are heavily compromised - the maximum attained here just average 120Mbps. It is also noted that in most of the other scenarios, the wireless throughput values registered are all under 50Mbps. Again, this is significantly lower than the theoretical values for the different standards. To see the difference between all values recorded under 50Mbps, an extrapolated TCP throughput graph is presented in Figure 5.7.

In the TCP extracted graph, together with the TCP heat map in Figure 5.8, there are a few interesting observations. In all throughputs under 50Mbps, IEEE802.11n with WPA2 as the encryption protocol is the best performer. This is true for both versions of the Internet protocol. Following that, IEEE802.11ac with WPA2 peaks at around 40Mbps. It is again noticed that in both pairs

mentioned above, IPv6 values are slightly lower than IPv4. All other values average below 10Mbps mostly with IEEE802.11g WPA2 registering the lowest values. It is also observed that IEEE802.11ac with WPA2 v6 values significantly fluctuate for different packet sizes.

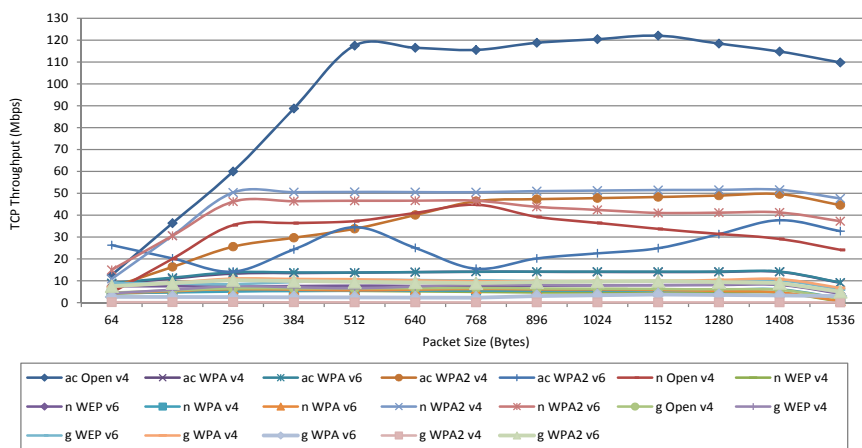


Figure 5.6: Graph of TCP throughput for various network scenarios

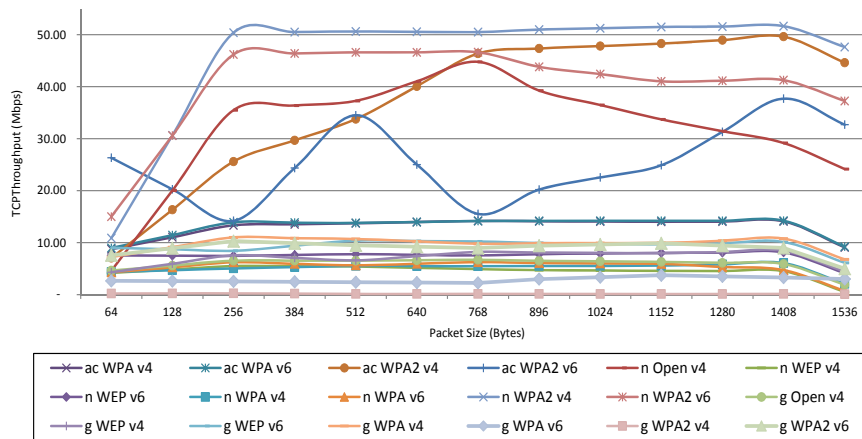


Figure 5.7: Extracted graph of TCP throughput (up to 60 Mbps)



		TCP Throughput: Wireless												
		64	128	256	384	512	640	768	896	1024	1152	1280	1408	1536
Wired	Pure IPv4	0.05	0.12	0.18	0.23	0.28	0.38	0.49	0.64	0.72	0.80	0.89	0.99	0.98
	Pure IPv6	0.05	0.11	0.18	0.28	0.38	0.34	0.30	0.55	0.68	0.81	0.90	1.00	0.99
Wireless IEEE 802.11ac	ac Open v4	0.10	0.30	0.49	0.73	0.96	0.95	0.95	0.97	0.99	1.00	0.97	0.94	0.90
	ac WPA v4	0.07	0.09	0.11	0.11	0.11	0.11	0.12	0.12	0.12	0.11	0.12	0.12	0.07
	ac WPA v6	0.07	0.09	0.11	0.11	0.11	0.11	0.12	0.12	0.12	0.12	0.12	0.12	0.08
	ac WPA2 v4	0.06	0.13	0.21	0.24	0.28	0.33	0.38	0.39	0.39	0.40	0.40	0.41	0.37
	ac WPA2 v6	0.22	0.17	0.12	0.20	0.28	0.20	0.13	0.17	0.18	0.20	0.26	0.31	0.27
Wireless IEEE 802.11n	n Open v4	0.09	0.39	0.69	0.70	0.72	0.79	0.87	0.76	0.71	0.65	0.61	0.56	0.47
	n WEP v4	0.08	0.09	0.11	0.11	0.11	0.10	0.09	0.09	0.09	0.09	0.09	0.09	0.01
	n WEP v6	0.15	0.15	0.14	0.15	0.15	0.15	0.15	0.15	0.15	0.16	0.16	0.16	0.08
	n WPA v4	0.08	0.09	0.10	0.10	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.12	0.04
	n WPA v6	0.08	0.10	0.12	0.11	0.11	0.12	0.12	0.12	0.12	0.11	0.10	0.09	0.01
	n WPA2 v4	0.21	0.59	0.98	0.98	0.98	0.98	0.98	0.99	0.99	1.00	1.00	1.00	0.92
	n WPA2 v6	0.29	0.59	0.89	0.90	0.90	0.90	0.90	0.85	0.82	0.79	0.80	0.80	0.72
Wireless IEEE 802.11g	g Open v4	0.42	0.51	0.59	0.59	0.59	0.60	0.61	0.59	0.58	0.57	0.55	0.54	0.17
	g WEP v4	0.39	0.54	0.69	0.64	0.60	0.67	0.75	0.74	0.73	0.72	0.74	0.76	0.40
	g WEP v6	0.82	0.79	0.77	0.85	0.94	0.94	0.93	0.90	0.89	0.88	0.90	0.92	0.56
	g WPA v4	0.66	0.83	1.00	0.99	0.97	0.93	0.89	0.90	0.90	0.90	0.94	0.98	0.62
	g WPA v6	0.24	0.24	0.23	0.22	0.22	0.21	0.20	0.27	0.30	0.34	0.32	0.30	0.27
	g WPA2 v4	0.02	0.02	0.02	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
	g WPA2 v6	0.69	0.81	0.93	0.91	0.86	0.84	0.81	0.86	0.88	0.90	0.96	0.82	0.45

Figure 5.8: Heat map of TCP throughput for various network scenarios

Next, UDP throughput values attained in all test-bed experiments are presented. In Figure 5.9, it is observed that the lines mostly follow a pattern similar to that of TCP, however, all values are slightly higher. IEEE802.11ac, without encryption, is registering the highest values, and these values are almost 35% higher than TCP. All the other wireless scenarios register values at almost one-third that of the IEEE802.11ac, ranging to a maximum of 60Mbps. Within this range, there are distinct patterns that can be identified, as can be clearly seen in the extracted graph, presented in Figure 5.10.

There is a significant difference between the UDP throughput values registered in different scenarios, as shown in Figure 5.10 and emphasised in the heat map, Figure 5.11. As was the case for TCP, IEEE802.11n with WPA2 on IPv4 has the highest throughput values, maximizing to approximately 55Mbps. This is almost one-third that of a network with IEEE802.11ac without any encryption. Following closely is WPA2 on the IPv4 network implemented with IEEE802.11ac. This is registering a drop of almost 10% from that of IEEE802.11n. Following that, in the band 30-40Mbps, are again the WPA2 lines for both IEEE802.11n and ac standards, but for the IPv6 networks. The graph shows that in relation to WPA2, there is a performance drop of at least 40% between IPv4 and IPv6, with the latter version having lower values. IEEE802.11n IPv4 without any encryption values are comparable with those discussed above, since they all fall in the same band in the graph. However, these are all at the lower end of that band.

For both TCP and UDP traffic types there are patterns evident in the graphs. TCP shows values slightly lower than UDP, while IPv6 values in all cases are also lower than its counterpart version. The latest standard in wireless definitely gives the best throughput on a network without encryption, however, with encryption throughput, values drop to that comparable with earlier editions of the wireless standard. In all cases there is a remarkable throughput drop between wired and wireless networks, therefore, since throughput values are different in most scenarios, wireless techniques can be ranked, which will be done later in the thesis.

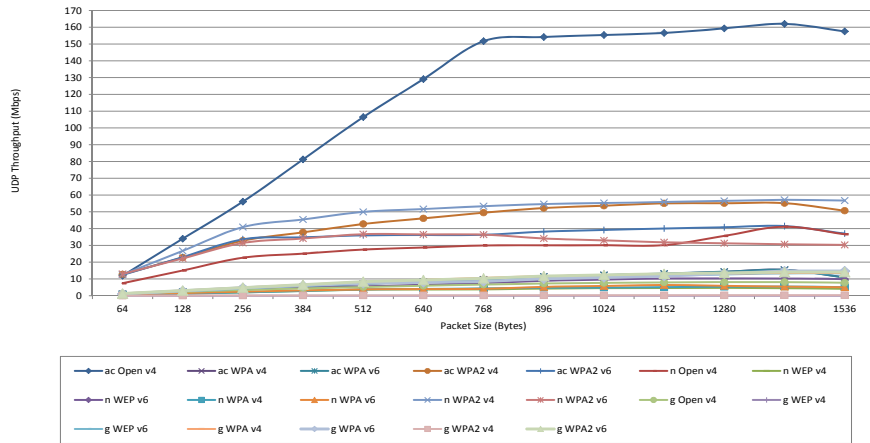


Figure 5.9: Graph of UDP throughput for various network scenarios

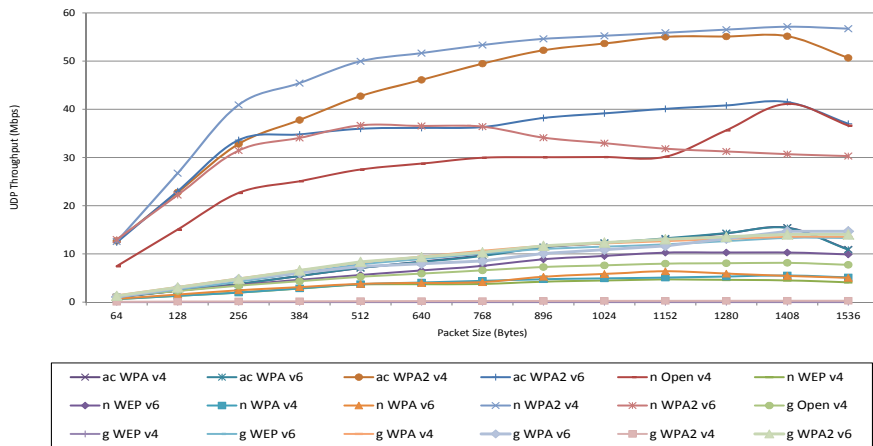


Figure 5.10: Extracted graph of UDP throughput (up to 60 Mbps)

		UDP Throughput: Wireless												
		64	128	256	384	512	640	768	896	1024	1152	1280	1408	1536
Wired	Pure IPv4	0.05	0.11	0.18	0.27	0.36	0.46	0.55	0.69	0.76	0.82	0.84	0.85	0.85
	Pure IPv6	0.05	0.12	0.20	0.27	0.34	0.42	0.50	0.67	0.75	0.84	0.92	1.00	0.99
Wireless IEEE 802.11ac	ac Open v4	0.07	0.21	0.35	0.50	0.66	0.80	0.94	0.95	0.96	0.97	0.98	1.00	0.97
	ac WPA v4	0.01	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.08	0.09	0.10	0.07
	ac WPA v6	0.01	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.08	0.09	0.10	0.07
	ac WPA2 v4	0.08	0.14	0.20	0.23	0.26	0.28	0.31	0.32	0.33	0.34	0.34	0.34	0.31
	ac WPA2 v6	0.08	0.14	0.21	0.21	0.22	0.22	0.22	0.24	0.24	0.25	0.25	0.26	0.23
Wireless IEEE 802.11n	n Open v4	0.13	0.26	0.40	0.44	0.48	0.50	0.52	0.53	0.53	0.53	0.62	0.72	0.64
	n WEP v4	0.01	0.02	0.03	0.05	0.06	0.06	0.07	0.07	0.08	0.08	0.08	0.08	0.07
	n WEP v6	0.02	0.04	0.06	0.08	0.10	0.12	0.13	0.16	0.17	0.18	0.18	0.18	0.17
	n WPA v4	0.01	0.02	0.03	0.05	0.07	0.07	0.08	0.08	0.09	0.09	0.09	0.10	0.09
	n WPA v6	0.01	0.03	0.04	0.05	0.07	0.07	0.07	0.09	0.10	0.11	0.10	0.10	0.09
	n WPA2 v4	0.22	0.47	0.72	0.80	0.87	0.90	0.93	0.96	0.97	0.98	0.99	1.00	0.99
	n WPA2 v6	0.23	0.39	0.55	0.60	0.64	0.64	0.64	0.60	0.58	0.56	0.55	0.54	0.53
Wireless IEEE 802.11g	g Open v4	0.08	0.16	0.23	0.29	0.36	0.40	0.45	0.50	0.52	0.54	0.55	0.55	0.53
	g WEP v4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	g WEP v6	0.07	0.18	0.29	0.41	0.53	0.61	0.68	0.75	0.78	0.82	0.86	0.91	0.91
	g WPA v4	0.09	0.21	0.33	0.45	0.56	0.64	0.72	0.79	0.83	0.86	0.89	0.92	0.92
	g WPA v6	0.07	0.20	0.32	0.41	0.50	0.54	0.58	0.65	0.74	0.79	0.80	1.00	1.00
	g WPA2 v4	0.00	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
	g WPA2 v6	0.09	0.21	0.33	0.45	0.57	0.64	0.70	0.80	0.84	0.87	0.91	0.96	0.96

Figure 5.11: Heat map of UDP throughput for various network scenarios

The next metric measured on various wireless networks was jitter, values for which are presented herewith. In Figure 5.12, all values for TCP jitter are shown and it highlights that IEEE802.11g, with a WPA2 encryption protocol on a network with IPv4, has significantly higher jitter values than all the other wireless combinations implemented. While all values are below 0.01ms in all scenarios, IEEE802.11g with WPA2 on the IPv4 network has, at a maximum, 8-fold higher values. This is mostly the case for larger packet sizes. In Figure 5.13, the extracted values up, to the maximum of 0.006ms, are presented and here a differentiation can be made between most of the other wireless scenarios. For smaller packet sizes all values are close to zero, but for large packet sizes, values range from 0 to 0.004ms. The IEEE802.11ac TCP jitter values attained are comparable with all the others. IEEE802.11g with WPA on an IPv6 infrastructure stands out in this band, with all values significantly higher than the rest. These differences are emphasised in the associated heat map shown in Figure 5.14

In relation to UDP jitter measured on wireless networks, the patterns here are very similar to that of TCP. As shown in Figure 5.15, for all cases but one, the values attained are close to zero, similar to that of TCP. As is the case with TCP, IEEE802.11g with WPA2 encryption protocol on a network with IPv4 has the highest values, significantly higher than the rest. At its peak, the UDP jitter is almost 6 times higher than all the other scenarios. When values close to zero are extrapolated (Figure 5.16) it can be seen that although all UDP jitter recorded are under 0.005ms, there still exists a pattern of separation between the wireless combinations. This is also highlighted in the associated heat map in Figure 5.17. It can be seen that in this minor range of values there are three distinct bands. Tightly grouped together at the top are IEEE802.11g WEP with IPv6 and two IEEE802.11n WPA lines, followed by a combination of WPA and WEP implemented in different combinations. At the bottom of the band are mostly the lines associated with IEEE802.11ac, showing that the new version of the wireless standard has significant improvement in reducing wireless UDP jitter on networks.

Wireless jitter, like throughput, is an important performance metrics. Jitter

is an issue on wireless networks and this has been emphasised in the results attained. Since different combinations of wireless standards and encryptions on the two versions of IP networks produce different jitter values, it will be used to rank the wireless combinations later in this thesis.

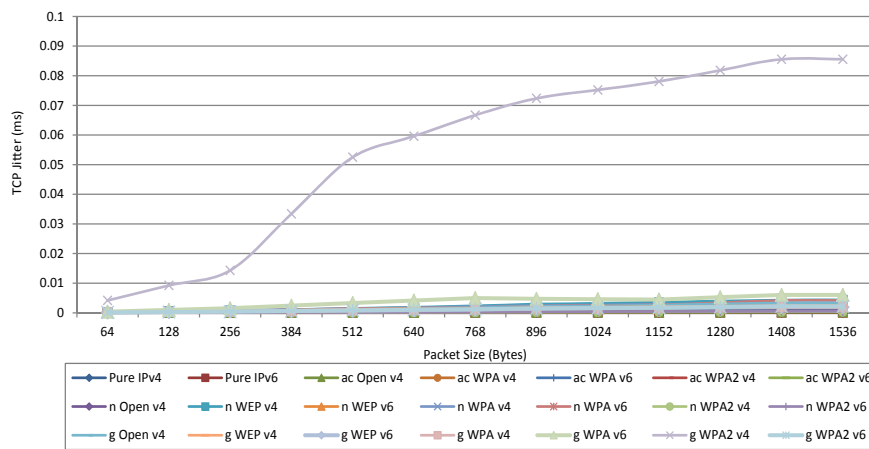


Figure 5.12: Graph of TCP jitter for various network scenarios

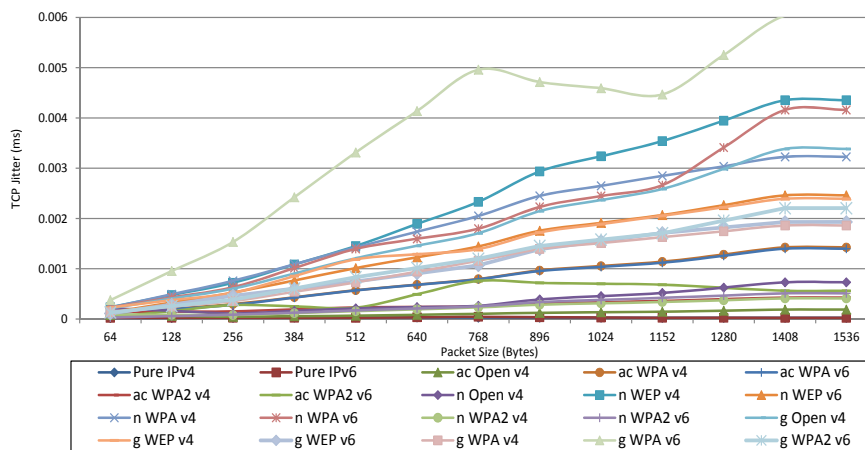


Figure 5.13: Extracted graph of TCP jitter (up to 0.006 ms)

5.7. PERFORMANCE METRICS MEASUREMENTS FROM WIRELESS TEST-BEDS

		TCP Jitter: Wireless												
		64	128	256	384	512	640	768	896	1024	1152	1280	1408	1536
Wired	Pure IPv4	0.48	0.51	0.54	0.63	0.72	0.66	0.61	0.59	0.58	0.57	0.55	0.54	0.54
	Pure IPv6	0.50	0.53	0.57	0.54	0.52	0.76	1.00	0.77	0.66	0.54	0.53	0.52	0.52
Wireless IEEE 802.11ac	ac Open v4	0.03	0.03	0.02	0.02	0.03	0.03	0.04	0.05	0.06	0.06	0.07	0.08	0.08
	ac WPA v4	0.05	0.08	0.12	0.18	0.24	0.28	0.33	0.40	0.43	0.47	0.53	0.59	0.59
	ac WPA v6	0.05	0.08	0.12	0.17	0.23	0.28	0.32	0.39	0.43	0.46	0.52	0.57	0.57
	ac WPA2 v4	0.06	0.06	0.06	0.08	0.09	0.10	0.10	0.13	0.14	0.15	0.16	0.18	0.18
	ac WPA2 v6	0.02	0.06	0.11	0.10	0.09	0.20	0.31	0.30	0.29	0.28	0.26	0.23	0.23
Wireless IEEE 802.11n	n Open v4	0.05	0.04	0.03	0.04	0.05	0.05	0.06	0.09	0.10	0.12	0.14	0.17	0.17
	n WEP v4	0.05	0.11	0.17	0.25	0.33	0.43	0.54	0.67	0.74	0.81	0.91	1.00	1.00
	n WEP v6	0.03	0.08	0.12	0.18	0.23	0.28	0.33	0.40	0.44	0.48	0.52	0.57	0.57
	n WPA v4	0.05	0.11	0.17	0.25	0.33	0.40	0.47	0.56	0.61	0.65	0.70	0.74	0.74
	n WPA v6	0.05	0.10	0.15	0.23	0.32	0.37	0.41	0.51	0.56	0.61	0.78	0.96	0.96
	n WPA2 v4	0.02	0.02	0.02	0.03	0.04	0.04	0.05	0.07	0.07	0.08	0.09	0.09	0.09
	n WPA2 v6	0.01	0.01	0.02	0.03	0.04	0.05	0.06	0.08	0.09	0.10	0.11	0.12	0.12
Wireless IEEE 802.11g	g Open v4	0.00	0.00	0.01	0.01	0.01	0.02	0.02	0.03	0.03	0.03	0.03	0.04	0.04
	g WEP v4	0.00	0.00	0.01	0.01	0.01	0.02	0.02	0.02	0.02	0.02	0.03	0.03	0.03
	g WEP v6	0.00	0.00	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0.02	0.02	0.02	0.02
	g WPA v4	0.00	0.00	0.00	0.01	0.01	0.01	0.01	0.02	0.02	0.02	0.02	0.02	0.02
	g WPA v6	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.06	0.05	0.05	0.06	0.07	0.07
	g WPA2 v4	0.05	0.11	0.17	0.39	0.61	0.70	0.78	0.85	0.88	0.91	0.96	1.00	1.00
	g WPA2 v6	0.00	0.00	0.00	0.01	0.01	0.01	0.01	0.02	0.02	0.02	0.02	0.03	0.03

Figure 5.14: Heat map of TCP jitter for various network scenarios

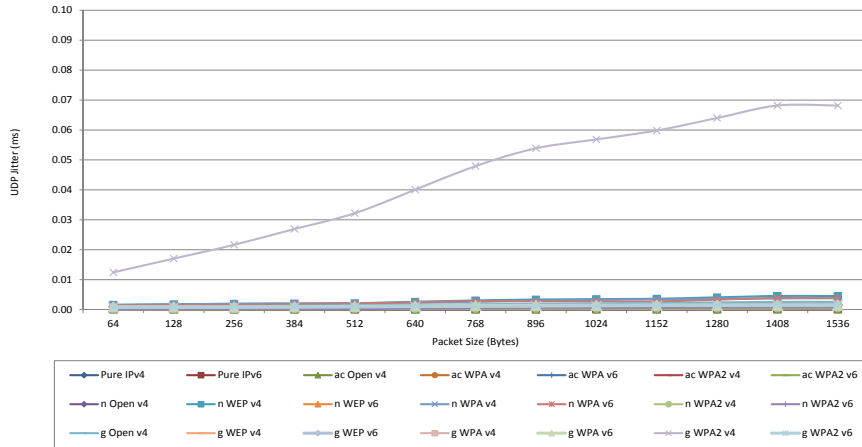


Figure 5.15: Graph of UDP jitter for various network scenarios

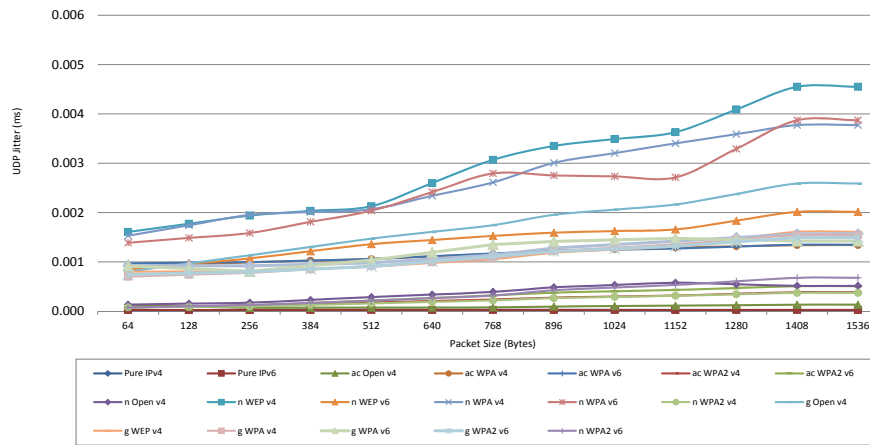


Figure 5.16: Extracted graph of UDP jitter (up to 0.006 ms)



		UDP Jitter: Wireless												
		64	128	256	384	512	640	768	896	1024	1152	1280	1408	1536
Wired	Pure IPv4	0.67	0.71	0.76	0.88	1.00	0.92	0.85	0.82	0.80	0.79	0.77	0.76	0.76
	Pure IPv6	0.79	0.76	0.73	0.77	0.82	0.82	0.82	0.79	0.77	0.76	0.77	0.79	0.79
Wireless IEEE 802.11ac	ac Open v4	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
	ac WPA v4	0.06	0.07	0.07	0.07	0.07	0.08	0.08	0.09	0.09	0.09	0.09	0.09	0.09
	ac WPA v6	0.07	0.07	0.07	0.07	0.07	0.08	0.08	0.09	0.09	0.09	0.09	0.09	0.09
	ac WPA2 v4	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0.02	0.02	0.02	0.03	0.03
	ac WPA2 v6	0.01	0.01	0.01	0.01	0.02	0.02	0.02	0.03	0.03	0.03	0.03	0.04	0.04
Wireless IEEE 802.11n	n Open v4	0.03	0.03	0.04	0.05	0.06	0.07	0.09	0.11	0.12	0.13	0.12	0.11	0.11
	n WEP v4	0.35	0.39	0.43	0.45	0.47	0.57	0.67	0.74	0.77	0.80	0.90	1.00	1.00
	n WEP v6	0.19	0.21	0.24	0.27	0.30	0.32	0.34	0.35	0.36	0.36	0.40	0.44	0.44
	n WPA v4	0.34	0.38	0.43	0.44	0.45	0.51	0.57	0.66	0.70	0.75	0.79	0.83	0.83
	n WPA v6	0.30	0.33	0.35	0.40	0.45	0.53	0.61	0.61	0.60	0.60	0.72	0.85	0.85
	n WPA2 v4	0.02	0.02	0.02	0.03	0.03	0.04	0.05	0.06	0.06	0.07	0.08	0.08	0.08
	n WPA2 v6	0.02	0.02	0.03	0.04	0.05	0.06	0.07	0.09	0.11	0.12	0.13	0.15	0.15
Wireless IEEE 802.11g	g Open v4	0.01	0.01	0.02	0.02	0.02	0.02	0.03	0.03	0.03	0.03	0.03	0.04	0.04
	g WEP v4	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0.02	0.02	0.02	0.02	0.02
	g WEP v6	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
	g WPA v4	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0.02	0.02	0.02	0.02	0.02
	g WPA v6	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
	g WPA2 v4	0.18	0.25	0.32	0.39	0.47	0.59	0.70	0.75	0.83	0.88	0.94	1.00	1.00
	g WPA2 v6	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0.02	0.02	0.02	0.02	0.02

Figure 5.17: Heat map of UDP jitter for various network scenarios

TCP and UDP delay values measured on wireless networks are presented here. In Figure 5.18, it can be seen that all TCP delay values are below approximately 10ms and they are distinctly separated into three different bands. The group that has registered the highest delay is comprised of the IEEE802.11n standard only. In this, the IEEE802.11n with WPA on both versions of IP and IEEE802.11n with WPA2 on IPv4 networks are present, registering a reading averaging 10ms. At half this delay value is the next band averaging around 5ms. In this there are no specific patterns that can be identified, but two lines are for WPA2 and two are with IEEE802.11n. IEEE802.11ac with WPA2 on an IPv6 network is also present. In this band two scenarios that register very close to the same values are for networks with WPA2 on IPv6 for the two newer standards IEEE802.11n and ac. The rest of the scenarios are all registering values close to zero. To differentiate between the TCP values, and to highlight the existence of the three bands, a heat map is presented Figure 5.19. This emphasises that there are clear bands of delays in the scenarios tested.

TCP delay is a significant factor on wireless networks. Wired networks tested showed almost insignificant delays (results not shown in graphs), whilst some wireless scenarios report almost 10 times higher TCP delay values.

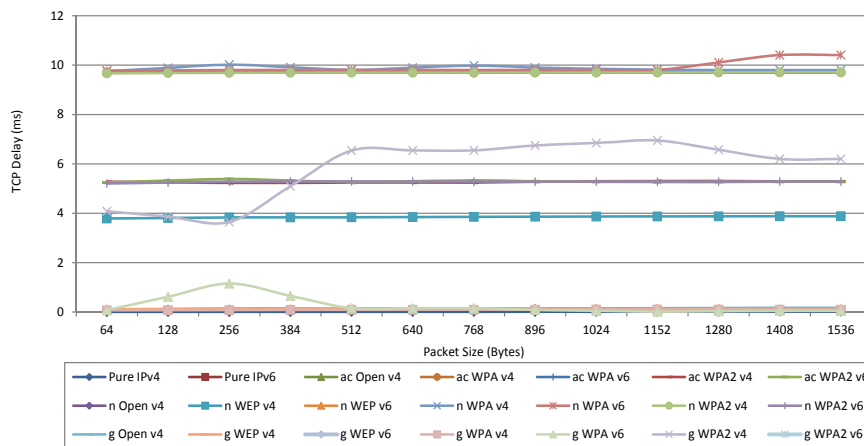


Figure 5.18: Graph of TCP delay for various network scenarios

5.7. PERFORMANCE METRICS MEASUREMENTS FROM WIRELESS TEST-BEDS

		TCP Delay: Wireless												
		64	128	256	384	512	640	768	896	1024	1152	1280	1408	1536
Wired	Pure IPv4	0.48	0.28	0.07	0.35	0.64	0.61	0.59	0.61	0.63	0.64	0.82	1.00	1.00
	Pure IPv6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Wireless IEEE 802.11ac	ac Open v4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	ac WPA v4	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	ac WPA v6	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	ac WPA2 v4	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.55	0.55	0.55	0.55	0.55
	ac WPA2 v6	0.54	0.55	0.56	0.55	0.54	0.55	0.55	0.55	0.54	0.54	0.54	0.54	0.54
Wireless IEEE 802.11n	n Open v4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	n WEP v4	0.36	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37
	n WEP v6	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
	n WPA v4	0.94	0.95	0.96	0.95	0.94	0.95	0.96	0.95	0.95	0.94	0.94	0.94	0.94
	n WPA v6	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.97	1.00	1.00
	n WPA2 v4	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93
	n WPA2 v6	0.50	0.50	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51
Wireless IEEE 802.11g	g Open v4	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
	g WEP v4	0.02	0.02	0.02	0.02	0.02	0.02	0.01	0.02	0.02	0.02	0.02	0.02	0.02
	g WEP v6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	g WPA v4	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
	g WPA v6	0.01	0.05	0.17	0.09	0.01	0.02	0.02	0.01	0.01	0.01	0.00	0.01	0.01
	g WPA2 v4	0.59	0.56	0.52	0.73	0.94	0.94	0.94	0.97	0.98	1.00	0.95	0.89	0.89
	g WPA2 v6	0.94	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93

Figure 5.19: Heat map of TCP delay for various network scenarios

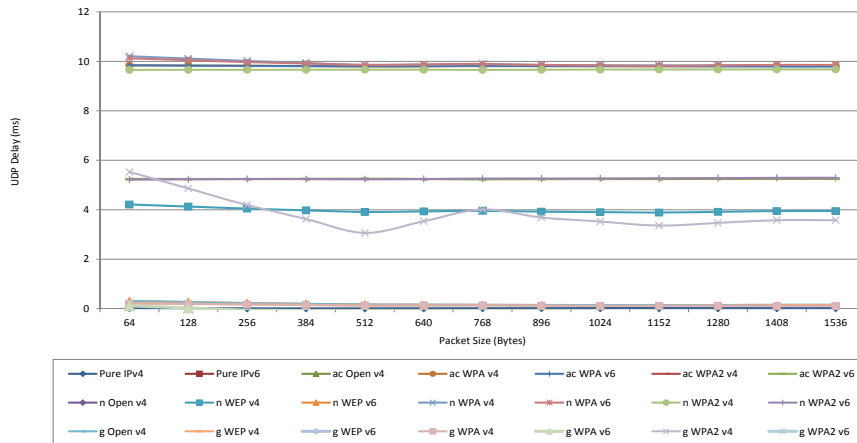


Figure 5.20: Graph of UDP delay for various network scenarios

UDP delay results are presented in Figure 5.20, with subsequent heat map in Figure 5.21. As with TCP, there are significant differences between the scenarios tested and again the results can be divided into three bands. At the maximum, the values attained are approximately 10ms, so the values for UDP delay are comparable with those of TCP. In this upper most band there are five lines comprising of IEEE802.11n and ac, with WPA and WPA2 encryption mechanisms. In this, both IPv4 and IPv6 exist, however none of the lines represent data in which WEP has been used as the encryption mechanism. In between 4 and 6ms, there is a mix of lines, mostly with IEEE802.11n and ac wireless standards. Following this are the lines registering UDP delays at almost 0ms, and in this all the lines are associated with IEEE802.11g wireless standard.

The delay values attained on individual test-beds show that there are significant differences between some of the combinations of wireless standards, encryptions methodology and IP version. The actual difference between TCP and UDP values is negligible, nonetheless, together both TCP and UDP delay values can be used as a performance metric to rank various wireless scenarios.

5.7. PERFORMANCE METRICS MEASUREMENTS FROM WIRELESS TEST-BEDS

		UDP Delay: Wireless												
		64	128	256	384	512	640	768	896	1024	1152	1280	1408	1536
Wired	Pure IPv4	1.00	0.78	0.56	0.63	0.71	0.82	0.92	0.93	0.94	0.94	0.95	0.96	0.96
	Pure IPv6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Wireless IEEE 802.11ac	ac Open v4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	ac WPA v4	1.00	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	ac WPA v6	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.99
	ac WPA2 v4	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53
	ac WPA2 v6	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53
Wireless IEEE 802.11n	n Open v4	0.06	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	n WEP v4	0.41	0.40	0.40	0.39	0.38	0.39	0.39	0.38	0.38	0.38	0.38	0.39	0.39
	n WEP v6	0.03	0.03	0.02	0.02	0.02	0.02	0.02	0.01	0.01	0.01	0.01	0.01	0.01
	n WPA v4	1.00	0.99	0.98	0.97	0.96	0.97	0.97	0.97	0.96	0.96	0.96	0.96	0.96
	n WPA v6	0.99	0.98	0.98	0.97	0.97	0.97	0.97	0.97	0.96	0.96	0.96	0.97	0.97
	n WPA2 v4	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95
	n WPA2 v6	0.51	0.51	0.51	0.51	0.51	0.51	0.52	0.52	0.52	0.52	0.52	0.52	0.52
Wireless IEEE 802.11g	g Open v4	0.06	0.05	0.04	0.04	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03
	g WEP v4	0.04	0.04	0.03	0.03	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.03	0.03
	g WEP v6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	g WPA v4	0.04	0.03	0.03	0.03	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
	g WPA v6	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
	g WPA2 v4	1.00	0.88	0.76	0.66	0.55	0.64	0.73	0.67	0.64	0.61	0.63	0.65	0.65
	g WPA2 v6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Figure 5.21: Heat map of UDP delay for various network scenarios

The final metric measured in the wireless environment is drop rate. This performance metric is not an issue of concern on wired networks, however, in case of wireless links, it can be a major factor impacting on network performance. This is presented below in Figure 5.22 and the subsequent colour chart in Figure 5.23. Here, only UDP drop rate is presented since for TCP data type, all values are approximately zero.

The drop rate graph and the heat map show that the packet drop rate is only of concern when the transmission packet size is small, say under 512 Bytes. For larger packets, in most cases, the values cannot be differentiated. However, for all packet sizes there are two scenarios that stand out. IEEE802.11ac with WPA for both versions of the IP is registering values much higher than all other values where for smaller packets, drop rate reaches almost 70%. The difference in drop rates between the two versions of IP in IEEE802.11ac with WPA is negligible, however, as the packet size increase, drop rate decreases to almost 10%.

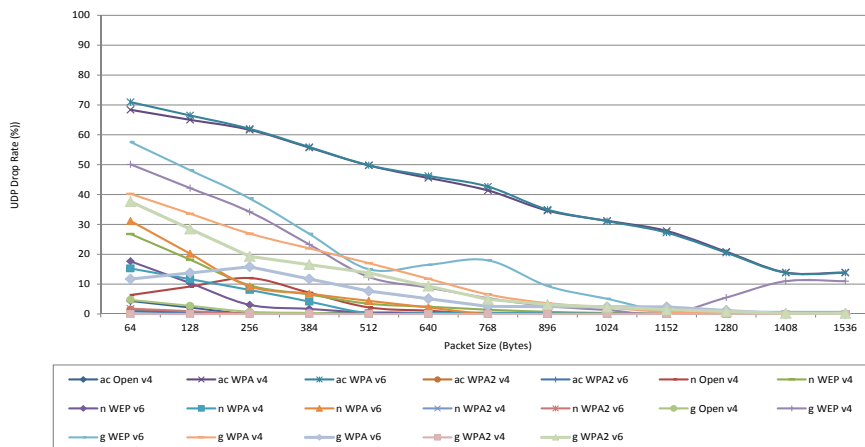


Figure 5.22: Graph of UDP drop rate for various network scenarios

		UDP Drop Rate: Wireless												
		64	128	256	384	512	640	768	896	1024	1152	1280	1408	1536
Wireless IEEE 802.11ac	ac Open v4	0.05	0.02	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	ac WPA v4	0.71	0.67	0.64	0.58	0.52	0.47	0.43	0.36	0.32	0.29	0.22	0.14	0.14
	ac WPA v6	0.74	0.69	0.64	0.58	0.52	0.48	0.44	0.36	0.32	0.28	0.21	0.14	0.14
	ac WPA2 v4	0.02	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	ac WPA2 v6	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Wireless IEEE 802.11n	n Open v4	0.20	0.29	0.39	0.23	0.07	0.04	0.00	0.00	0.00	0.00	0.01	0.02	0.02
	n WEP v4	0.86	0.58	0.31	0.21	0.11	0.08	0.05	0.02	0.01	0.00	0.00	0.00	0.00
	n WEP v6	0.57	0.33	0.10	0.06	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	n WPA v4	0.49	0.37	0.26	0.13	0.01	0.01	0.01	0.01	0.00	0.00	0.00	0.00	0.00
	n WPA v6	1.00	0.65	0.30	0.22	0.14	0.07	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	n WPA2 v4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	n WPA2 v6	0.06	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Wireless IEEE 802.11g	g Open v4	0.08	0.05	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	g WEP v4	0.87	0.73	0.59	0.40	0.21	0.15	0.09	0.05	0.02	0.00	0.10	0.19	0.19
	g WEP v6	1.00	0.84	0.67	0.47	0.26	0.29	0.31	0.16	0.09	0.01	0.01	0.01	0.01
	g WPA v4	0.70	0.58	0.47	0.38	0.30	0.20	0.11	0.06	0.04	0.01	0.01	0.00	0.00
	g WPA v6	0.20	0.24	0.27	0.20	0.13	0.09	0.04	0.04	0.04	0.04	0.02	0.00	0.00
	g WPA2 v4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	g WPA2 v6	0.65	0.49	0.34	0.29	0.24	0.16	0.08	0.05	0.04	0.03	0.01	0.00	0.00

Figure 5.23: Heat map of UDP drop rate for various network scenarios

## 5.8 Results Evaluation

In this chapter, the focus has been on evaluating TCP and UDP behaviour on networks that have some implementation of different wireless standards and encryption techniques. Results and data have been presented using line graphs and heat maps presented in the previous section. For wireless, the metrics measured are throughput, jitter, delay, and UDP packet drop rate. Whilst the intricacies of each scenario have been discussed already, here, the key observations will be highlighted.

On wireless networks, one of the primary differentiating performance metrics is throughput. Throughput values attained are all a fraction of what can be achieved on wired networks. It is also observed that there is a major difference between theoretical bandwidth and the maximum throughput achieved on test beds. In the case of wired networks, this difference is not that big (a 10 to 15% drop), but for wireless there is almost a 10-fold difference. As expected, IEEE802.11ac gives the highest throughput for both TCP and UDP traffic types. When comparing TCP and UDP throughput values, UDP, in the most extreme case, is giving an almost 35% higher reading than TCP. Implementing wireless encryption standards, reduce network performance and in this thesis the maximum reported drop in this scenario is almost 10%. This has been the case for WPA2 where implemented on IEEE802.11ac. Also it can be seen that WPA2 on networks gives the best TCP and UDP throughput values when compared to the other security mechanisms. The difference between IPv4 and IPv6 is heavily emphasised on wireless networks, where it can be seen that there is a performance drop of approximately 40%, in favour of IPv4.

Jitter plays a critical role on networks, and its impact on compromising network performance can be immense. In this thesis, it can be seen that most jitter values are acceptable and fall below 0.01ms on wireless infrastructures, however, one case (a combination of IEEE802.11g with WPA2) is registering values almost 8-fold higher than the rest. This clearly shows that from a network performance perspective, the choice of the actual wireless standard



and the security encryption mechanism is vital.

Undoubtedly, delay on wireless networks is generally a concern, and this has been emphasised on the test bed networks. Here, it can be seen that delay happens to be the key differentiator between the wireless scenarios, highlighting that some combinations of wireless and security standards can give 10 times higher values than others. The actual difference between TCP and UDP is insignificant, however in both, a clear distinction can be made between the wireless scenarios. Also noteworthy is that a majority of the scenarios are registering jitter readings close to zero.

On wireless networks, unlike on wired, packet drop rate can heavily impact on network performance. In the tests conducted, TCP traffic type did not register a significant drop rates, however UDP did. At the most there is a UDP drop rate of 70% and this is very interesting since it is for the newest version of wireless standard (but with an older version of security mechanism). This again highlights that the choice of combination of wireless standard and the security enhancement is critical for getting the best network performance.

The wireless networks implemented on test-beds have highlighted some significant characteristics of such networks. As mentioned earlier in the thesis, wireless usage on LANs is currently mainly for *last mile data delivery*, therefore, enhancing network performance on this critical part of the communication link is of the uttermost importance. The network performance metrics measured in these scenarios does distinguish between the combinations of wireless standards and security mechanisms - this will be used later in the thesis to rank various wireless networks based on the common metrics that have been measured.



## Chapter 6

---

# TCP/UDP Behaviour in Virtual Private Networks

### 6.1 Introduction and Motivation

The existence of a network-centric world has revolutionised the way people conduct their businesses. It is now common practice for employees to connect to a company's internal network from diverse locations, and for individuals to use networks to retrieve critical and confidential data from distant locations, for example, Internet banking from a home computer or mobile phone (Ortiz, 1997). Alliances and partnerships among organisations have become a crucial business strategy, thus requiring a secure communication channel between various sites. The openness and availability of the Internet has facilitated this explosive growth, however, data confidentiality, integrity, and availability are always a paramount concern.

Virtual Private Network (VPN) technology is the solution that can secure communication between various sites or within one site. This chapter provides details about VPN, first by presenting some commonly used VPN protocols and then outlining the literature related to its network performance issues. This is followed by details of the test-bed that was implemented to measure performance metrics related to TCP/UDP traffic types on selected VPN protocols, and the results attained from it.

### 6.2 VPN Preamble

VPN is defined as a communication environment constructed by controlled segmentation of a shared communication infrastructure to emulate the char-

acteristics of a private network. Proposed as a standard in 1994 in Hanks, Li, Farinacci, and Traina (1994a) and Hanks, Li, Farinacci, and Traina (1994b) (and later in Farinacci, Li, Hanks, Meyer, and Traina (2000) and Dommett (2000)), it is a TCP/IP based technology that can extend a private network beyond one location across any public network, such as the Internet. VPN enables communicating nodes to share information securely as if they were directly connected on one cohesive private network (Pena & Evans, 2000) (Metz, 2003). The extended VPN secured network can be used to access network based resources in the same way as the resources are available within a network, that is, the entire setup will be transparent from a network users perspective (Wood, Stoss, Chan-Lizardo, Papacostas, & Stinson, 1988). It is also worth noting that a VPN does not establish a link between different sites to create an integrated network, but is solely implemented for the purposes of inexpensively securing data transfer on an already existing network infrastructure.

There are various types of VPNs that can be implemented. On an one site network in an organisation, VPN can be deployed to secure access to local network resources (J. Zeng & Ansari, 2003) (Berger, 2006). Here, data sent via VPN will reside on that particular network and will traverse only the nodes nominated to communicate using the security mechanism. The nodes that are not part of the secured channel will continue communicating as per normal. In addition to this, a VPN can also be employed to allow secure remote access to a corporate network (this is known as client VPN) (Chandra & Nair, 2007). VPN used in this way will enable users to connect to a network when they are offsite but need to connect to their corporate private network to access data securely. This is similar to an archaic dial-up connection to network; however, with VPN the connection is secure and easier to implement since all that is required is a TCP/IP connection between the users node and the network. Another major use of VPN is to create what is commonly termed site-to-site VPN (FitzGerald, 2011). This facilitates secure communication in geographically disparate offices by encrypting/decrypting data as it leave/arrives at various sites. The intermediary network between the sites is generally the Internet or some form of third party line, details of which

may not necessarily be known. For a site-to-site VPN, dedicated equipment at the edge of the networks may be necessary and generally site-to-site VPNs can either be policy based (where some traffic destined for the other network triggers a session to be established) or may be router based (where traffic is routed via a tunnel to some destination router). Finally, another type of VPN which commonly used is similar to the client VPN, however, it does not require preconfigured software at the client end of communication. Instead a standard browser that supports active contents, acts as a VPN client a setup beneficial to remote clients since there is no configuration required and they can simply logon via a webpage to initiate the session . This technique is an OSI layer-4 VPN solution typically used to carry out bank transactions, while the other techniques mentioned are layer-3 solutions.

### 6.3 VPN Protocols

With the availability of numerous off-the-shelf utilities, data in motion on a network can easily be breached, even by novice users. To protect such data, a proven solution is implementing VPN between the transmission end points (Chandra & Nair, 2007). This will secure moving data by encrypting it so that any unauthorised party eavesdropping will be not be able to make sense of what is being transmitted. In addition to encrypting with VPN, authentication between communicating entities can also be achieved.

VPN protocols are generally categorised as per their functionality at different layers of the OSI model, in particular the layer at which data is exchanged between the sites (Knight & Lewis, 2004) (Joha, Shatwan, & Ashibani, 2007). While the focus of this research is to evaluate TCP/UDP traffic behaviour across VPN, the emphasis will just be on layer three protocols. Nonetheless, layer-2 protocols are briefly discussed next, prior to delving into the intricacies of layer-3 VPN. Brief advantages and disadvantages of VPN protocols are presented in Table 6.1.

	Description	Advantages	Disadvantage
Layer 2 VPNs	Forwards Layer 2 information (MAC, DLCI, VLAN ID)	ISP dependent	ISP dependent
		Low latency	Hardware overhead cost
		No packet size overhead	Security issues
		Multiprotocol support	
		Easy to implement	
Layer 3 VPNs	Forwards Layer 3 information (IP addresses)	Secured	Complicated to implement
		Scalable	Increased latency
		Lower cost	Overhead cost for further tunnelling

Table 6.1: Comparison of layer 2 and layer 3 VPNs

### 6.3.1 Layer 2 VPN Protocols

There are numbers of Layer 2 VPN (L2VPN) that has been in use for a number of years (Andersson & Rosen, 2006). Encapsulating packets at layer 2 enables the VPN tunnel to transport non-IP protocols, however, nowadays most networks are IP based (Metz, 2004) (S. Kim, Ryu, Park, & Kim, 2006). So VPN tunnels based on layer can theoretically transport any kind of packet, mainly using Point-to-Point Protocol (PPP)(Simpson, 1994) to connect the tunnel endpoints. Table 6.2 summarises the two common Layer 2 VPN protocols (Y. Wang, Yao, Zhao, & Zhou, 2001) (Townesley et al., 1999).

Layer 2 VPN Protocol	Description
L2TP	Combination of PPTP and L2F which sends encapsulated packets over ATM, HDLC, and Frame Relay. Paired with IPSec to provide encryption such as, 3DES and AES.
PPP	Provides authentication (PAP and CHAP), error detection, and link quality mechanism. Encapsulates data (HDLC frame), and maintains and terminates link setup.

Table 6.2: Different types of layer 2 VPN protocols

### 6.3.2 Layer 3 VPN Protocols

There are a number of commonly used Layer 3 VPNs as shown in Tables 6.3. These are generally used to by consumers on edge devices (Knight & Lewis, 2004) that generally connects a customer's end node to that of a service provider's network infrastructure. With VPNs, data can be encrypted and/or authenticated ( (Mohapatra, Metz, & Cui, 2007) and (Metz, 2003)), some common algorithms for these are presented in Table 6.4.

Layer 3 VPN Protocol	Encryption Strength	Algorithms	Compatible Platforms	Ports
PPTP	128 bits MPPE protocol	RSA and RC4	Windows, Linux, Mac OSX, Android, and iOS.	TCP port 1723
IPSec	256 bits key	3DES and AES	Windows, Linux, Mac OSX, Android, and iOS.	UDP ports 500, 1701, and 4500
SSTP	256 bits encryption with 2048 bits key	AES and RC4	Windows and Linux	TCP port 443
OpenVPN	128 bits encryption with 1024 bits keys, 256 bit encryption for control channel	3DES, AES, Blowfish. and RC5	Windows, Linux, Mac OSX, Android, and iOS.	TCP port 443 UDP port 53 (can be changed)

Table 6.3: Different types of layer 3 VPN protocols

Encryption Algorithm	Year Published	Key Length	Block Size	Rounds
AES (Rijndael)	1998	128, 192 or 256 selectable	128 bits	10, 12 or 14 depending on key size
Blowfish	1993	32-448 bits	64 bits	16
DES	1977	56 bits (+8 parity bits)	64 bits	16
3DES	1998	168, 112 or 56 bits	64 bits	48 DES-equivalent
RC4	1987	40-2048		1
RC5	1994	0-2040 bits	32, 64 or 128 bits	1-255

Table 6.4: Comparison of data encryption algorithms of block cipher

### Point to Point Tunnelling Protocol

The Point to Point Tunnelling Protocol (PPTP) specification was initially published in 1999, RFC 2637 (Hamzeh et al., 1999) and was originally developed by the PPTP forum, a vendor consortium of Microsoft, U.S. Robotics and several remote access vendor companies. This tunnelling protocol is inbuilt in Microsoft operating systems, thus, allowing any client running such an operating system to seamlessly connect securely to a corporate VPN (Joha et al., 2007). The actual PPTP specification does not detail any encryption or authentication mechanisms, however, for both of these, PPTP relies on PPP (Simpson, 1994) (Jaha, Ben-Shatwan, & Ashibani, 2008). Thus, PPTP implementations with commonly used Microsoft operating systems have various levels of authentication and encryption natively embedded into the Windows PPTP stack. Initially Microsoft used the Data Encryption Standard (DES) in the Microsoft-Challenge Handshake Authentication Protocol (MS-CHAP) for authenticating remote access clients, later, the protocol was changed to Protected Extensible Authentication Protocol (PEAP)/EAP with MSCHAPv2. In the Linux domain, PoPToP daemons (and modules of PPP and Microsoft Point-to-Point Encryption (MPPE)) are generally used in the server to provide support for PPTP. At the client end, there used to be limited support initially, however, new Linux distributions (since SuSE Linux 10) provide full support for this VPN protocol.

PPTP is designed to use two different types of packets to establish secure communication between different nodes. Generic routing encapsulation (GRE) is used to carry VPN payload, achieved by adding the GRE header to the original data packets. In the GRE header, control bit information, sequence and tunnel number information are present. This is very similar to the Layer 2 Tunnelling Protocol (L2TP) header (Berger, 2006). In addition to using the GRE packet, PPTP uses a second type of packet that contains the PPTP control message, which is just a generic TCP packet using port 1723. The second packet contains information related to session control, like connection request and response, related parameters and information related to transmission errors. Both GRE and the PPTP message lack facilities to provide authentication and encryption, therefore PPTP has to be combined with other



methods so that security enhancements can be achieved.

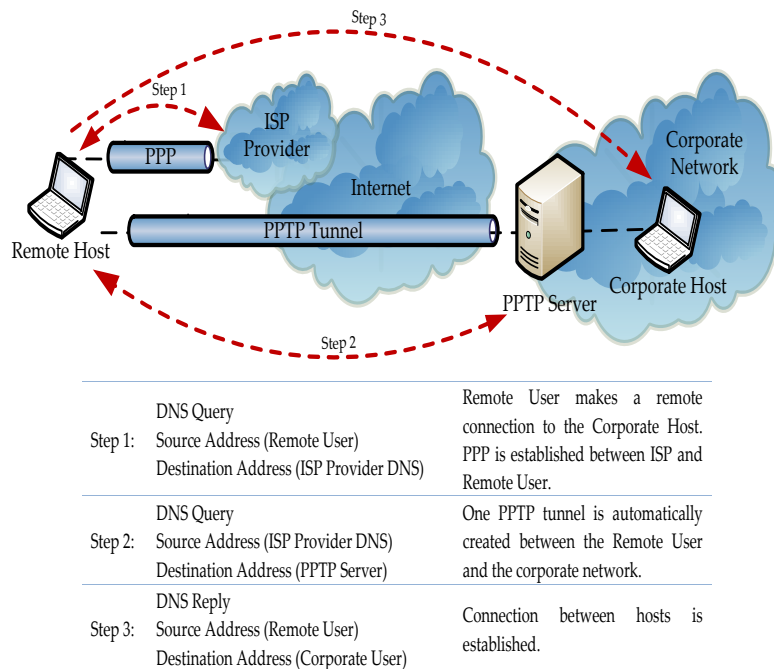


Figure 6.1: PPTP implementation via remote access to the corporate network

There have been many significant security vulnerabilities seen in relation to PPTP. These vulnerabilities are mainly related not to the underlying architecture of the protocol itself, but to its dependability on other protocols like PPP and MPPE. Also, there are known problems when PPP and MPPE are integrated, which also contributes to PPTP problems. MS-CHAP version 1 and 2 have been used with PPTP for authentication, but in recent times it has been shown that both versions can be breached. Version 1 is fundamentally insecure and there exist numerous tools that can be employed to extract the passwords, since the hashing algorithm it employs is weak. MS-CHAP version 2 can be exploited as well since it uses 3DES, which is vulnerable to dictionary brute force attacks. Such attacks can be mounted on captured

challenge-response datastream by trying out all possible combinations of 256 bit key. MPPE, generally used in conjunction with MS-CHAP, uses RC4 and can also be easily cryptanalysed since it uses the same key for encrypting data in both directions in the communication flow. This can be achieved by XORing the sniffed datastream from both directions together. PPTP lacks two factor authentications, but relies on a simple username/password combination. This, when compared to newer VPN protocols is a weak point.

### **Internet Protocol Security**

IPsec comprises of a suite of IETF protocols and algorithms that together can enhance end-to-end security in an IP based network. The IPsec protocol suite contains protocols that can be employed to establish mutual authentication between nodes at the beginning of a secure session and also to negotiate details of the cryptographic keys that will be used during the secured session (Kent & Seo, 2005) (Hamed, Al-Shaer, & Marrero, 2005). It provides IP layer transient data security by enabling a system to select required security related protocols and algorithms to use for the services. In the main, IPsec is a network layer protocol that can authenticate data origin, check data integrity, perform encryption and protect against some common network data attacks. IPsec can be implemented either in a network host, a security gateway (a router or firewall which is IPsec enabled) or in any independent network device. Security enhancement that IPsec offers on the infrastructure is based on the requirements defined in the Security Policy Database (SPD) (Lian & Wen-Mei, 2007). This database can be maintained by a network user, an administrator or some application operating within the constraints of that network. So IPsec compares the information in the packet header with that in the SPD, and either protects it using the IPsec mechanism, discards it, or allows it to bypass the protection offered by the security mechanism.

The architecture of IPsec is such that it is able to provide cryptographically-based security enhancements for both IPv4 and IPv6 infrastructures. Its various security related services operate at the IP layer and are also able to provide data protection at the upper layers of the protocol. Security Associations (SAs) are at the core of this protocol. These define a number of IPsec re-

lated entities including authentication protocols, keys, algorithms and cryptographic synchronisation related information. There are two modes in which IPsec can operate, transport, and tunnel (Varadarajan & Crosby, 2014). A host that is using IPsec has to support both modes of operation, while security gateways in the data transmission path must only support tunnel mode. Depending on the implementation and the combinations chosen, different security implementations can be reached with VPN, however, IPsec in tunnel mode is mainly used to establish VPN.

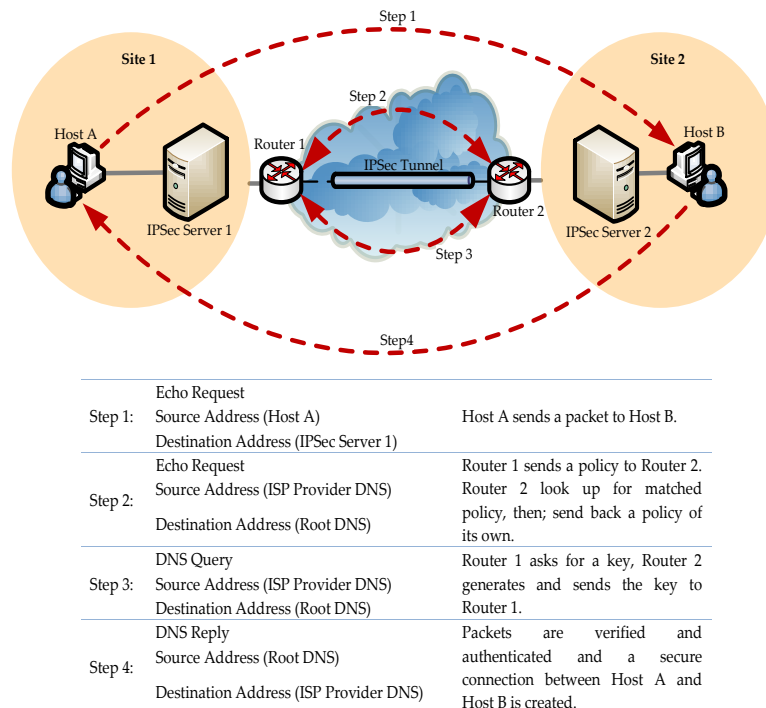


Figure 6.2: IPsec implementation via site-to-site connection

There are various kinds of VPN that can be established with IPsec. A Host-based VPN between two networked hosts can be established in either tunnel or transport mode, regardless of the type of connection that exists as long as

a TCP/IP session is maintained between them. A gateway-based VPN can be established between two security gateways so that it can facilitate a secure connection between all nodes on the two, respective networks. This is more like a network-to-network data security mechanism where data encryption and decryption happens at the respective network gateways. A host-to-gateway VPN scenario is generally referred to as a remote access VPN. Here, a remote host that has access via an IP connection to the Internet, is able to establish a secure connection to some corporates internal network. Thus, with configurations available on various hosts and gateways, together with the availability of the transport and tunnel modes, several configurations of IPsec-based VPNs are possible. But there are only two commonly implemented configurations : a network-to-network VPN securing data between two networks, and a client-to-network configuration, where data between a remote client accessing some trusted network via the Internet is safeguarded.

IPsec leverages security protocols to safeguard data that moves in its established VPN tunnel. In the main, it uses Authentication Header (AH) and Encapsulated Security Payload (ESP), which are both security protocols for data in motion (RFC 1826 and 1827 respectively for the two protocols (Atkinson, 1995a) (Atkinson, 1995b)) (Adeyinka, 2008). Configuring AH and/or ESP determines if IPsec in transport or tunnel mode is implemented. In addition to these, cryptographic keys are also used, which may simply be PSK or can be issued a more complex means like a Certification Authority (CA). The choice of what actually gets used and the combinations in any particular context are determined by a number of factors including security and systems requirements of the nodes, users, applications, and organisations. Whatever combination protocols and cryptographic keys are employed, to the end user, the process of using the VPN created by IPsec is mainly transparent.

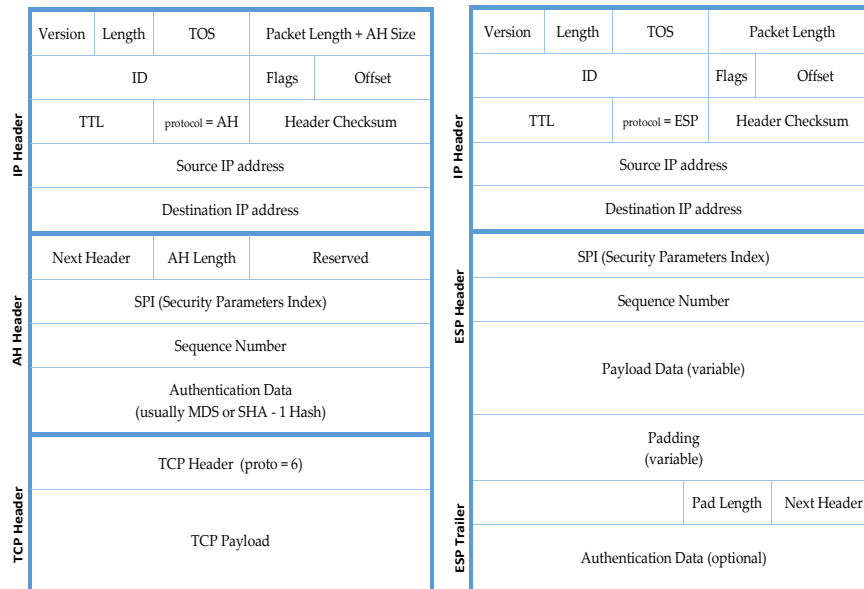


Figure 6.3: IPsec AH and ESP packet structure

**Authentication Header Protocol** : AH is used in IPsec to provide IP datagram integrity and an origination authentication check of the IP header and upper level protocol data (Kent, 2005a). Whilst doing this, it can also provide protection against replay attacks by enabling the receiver of the data to check increments to the sequence numbers upon establishment of security associations between the sending and receiving entities. Thus, AH is primarily responsible for providing authentication services (and not confidentiality) when IPsec is used as a VPN protocol.

AH, when implemented, adds an additional header 24Bytes long to the IP datagram. IANA has defined AH as protocol 51, so in the IP header field preceding the AH, a value of 51 will be added to the 8-bit protocol field (Kurose & Ross, 2008). Following this, the security protocol used is identified in the Next Header field. The actual length of the message (payload size) information follows in the Payload Length field, which is then followed by

a Reserved field (not being used). Next, Security Parameter Index (SPI) information is inscribed, which identifies the context in which SAs will be interpreted. Sequence Number related to the actual session is in the next field and finally, Authentication Data along with Integrity Value Check (IVC) information exists (Hirschler & Treytl, 2012). At the beginning of the session establishment, AH is generated at the sender end using Hashed Message Authentication Code (HMAC), which gets decoded at the receiver end. HMAC generally uses common cryptographic hash functions, like Message Digest 5 (MD5) and Secure Hash Algorithm 1 (SHA1) or may optionally be using DES for security transformations (Younglove, 2001).

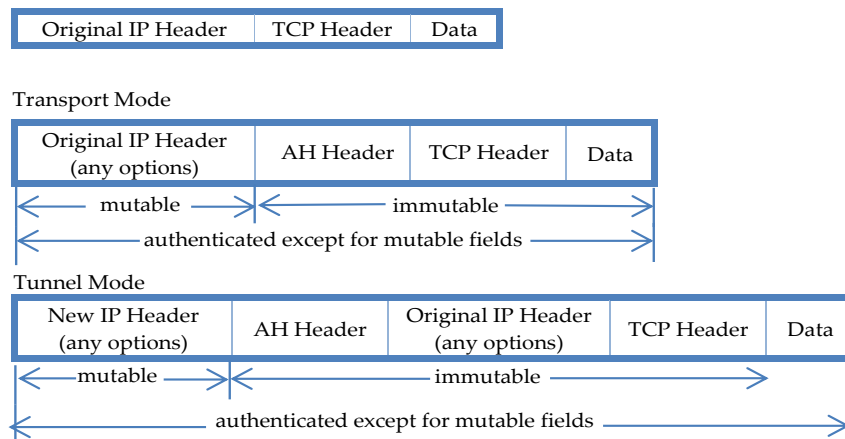


Figure 6.4: An IPsec IPv4 packet after applying AH

Depending on the mode of IPsec implementation, AH will be placed in different locations during the communication process. In transport mode, AH is placed after the IP header and before any other upper layer protocol, while in the tunnel mode the IP header and options precede the AH header with the original packet behind the header (Figure 6.4). For IPsec to be applied properly, different sequences of steps have to be executed depending on whether the data is inbound or outbound. For outbound traffic, SA that calls the AH processing is determined first, followed by the header being inserted after

the IP header. The sequence number for application to the header is then generated, which is followed by message integrity calculations and information. The message integrity information is applied to the end of the AH and just before the upper layer protocols. For an inbound process, since fragmentation may have occurred during data transit, the data is reassembled prior to processing of the AH by the IPsec mechanism. This is followed by a series of verification steps in which the actual association of the datagram to the SA is determined to ensure that the communication received is really from the valid SA. While verifying, SPI in the AH and the IP address is looked up first. This is followed by verification of the sequence numbers, as a mechanism for anti-replay attacks, and then the data is authenticated, which is finally forwarded to the destination based on the information in the IP header.

**Encapsulated Security Payload** : ESP is often used when data confidentiality is a concern. When IPsec ESP is implemented, malicious attackers will not be able to read and make sense of the data as it moves between the source and destination. In addition to enhancing confidentiality, ESP also provides a mix of other security related services, including authenticity and integrity related checks. If required it can be implemented in encryption-only or authentication-only configurations, however, security is heavily compromised if only one is implemented. When IPsec is implemented in transport mode it does not provide integrity and authentication for data in motion, however in tunnel mode ESP protects the entire inner IP packet while the outer header remains unprotected. ESP uses IP protocol number 50 and operates on top of the IP layer.

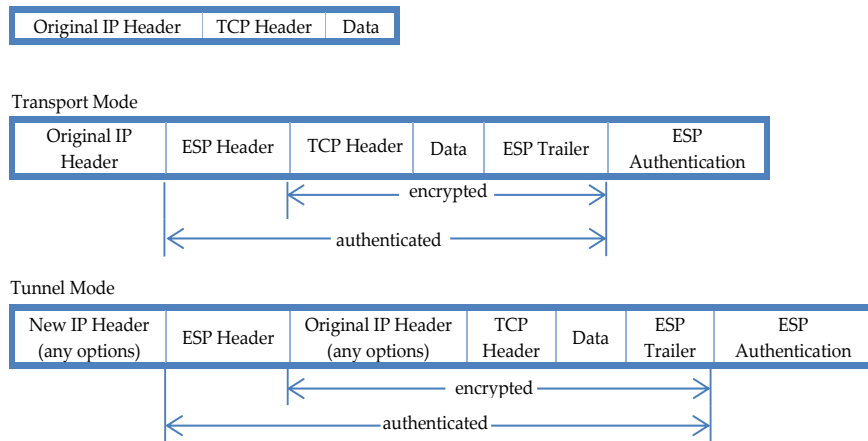


Figure 6.5: An IPsec IPv4 packet after applying ESP

In ESP, there are several fields in the header similar to AH, but packages in the fields have very different configurations. Instead of having just a header, there are now three components divided into ESP header, ESP trailer, and ESP authentication data (Kent, 2005b) (Hirschler & Treytl, 2012). The SPI field is the first field of size 32 bits (Figure 6.5), followed by a value that is used to replay resistance and origin authentication (W. Liu, Jiang, & Zhang, 2006). The actual payload is next, which is a field of variable length depending on the size of the actual data being transmitted. At the beginning of the payload there may be an Initialisation Vector (IV), which is present if required by the encryption algorithm (Younglove, 2001). The actual encryption is not really part of the IPsec process but solely part of the used algorithm. Then follows padding, which provides limited communication flow confidentiality and also maintains encryption boundaries as required by some encryption algorithms. The length of padding used (an optional field) is a value mentioned in Pad Length field; this field has to exist even if the actual pad is not being used. This field is followed by another value field (Next Header), which notifies the recipient of the header type contained within the encrypted payload. The final field is the authentication data, which contains the message digest calculated against the ESP packet.



The use and management of keys when IPsec is used as a VPN protocol also needs some consideration. Internet Key Exchange (IKE) has the responsibility to create and manage keys used during the communications process, accommodating for the multiple variations in its implementation (Haddad, Berenjkoub, & Gazor, 2004). In addition, IKE also plays a major role at the beginning of the communication process where it is responsible for session establishment between the VPN nodes (Mao, Zhu, & Qin, 2012). During the entire process, secure keys have to be used; otherwise, irrespective of the encryption and the authentication algorithms used, the process can be significantly compromised. The simplest way to assign keys is manually. This suits in small environments where there are limited dynamic changes happening in relation to VPN implementation. Automatic key management is desired since there is no manual intervention required, allowing VPN to be deployed seamlessly on large scale implementations. To automatically assign keys for IPsec VPN, three primary protocols are combined to provide the required key requirements for the process - they are Internet Security Association and Key Management Protocol (ISAKMP), Oakley and Secure Key Exchange Mechanism (SKEME) (Maughan, Schertler, Schneider, & Turner, 1998) (Orman, 1998). The ISAKMP framework is used for authenticating communicating partners and to generate the required keys, while Oakley, provides a mechanism for an exchange protocol that is required by the ISAKMP process (Sierra, Hernandez, Ribagorda, & Jayaram, 2002). SKEME is a Diffie-Hellman algorithm and incorporates Public Key Infrastructure (PKI) into the process of issuing keys during IPsec authentication process (Krawczyk, 1996). SKEME is a versatile key exchange mechanism that provides anonymity, repudiability and quick key refreshment to the IPsec process.

### **Secure Socket Tunnelling Protocol**

The Secure Socket Tunnelling Protocol (SSTP) is the newest of the protocols that can be used to establish a VPN, allowing remote users and networks to securely connect to a corporate network. In the main, it uses a Secure Socket Layer (SSL) 3.0 layer protocol to send either PPP or L2TP traffic between the tunnel nodes. Data encryption and traffic integrity checks for SSTP are

done by SSL and its mechanism allows SSTP medium traffic to pass through commonly used firewalls, NAT and proxy implementations. Thus SSTP is able to solve the typical VPN problem of not being able to traverse traffic through network perimeter security enhancements.

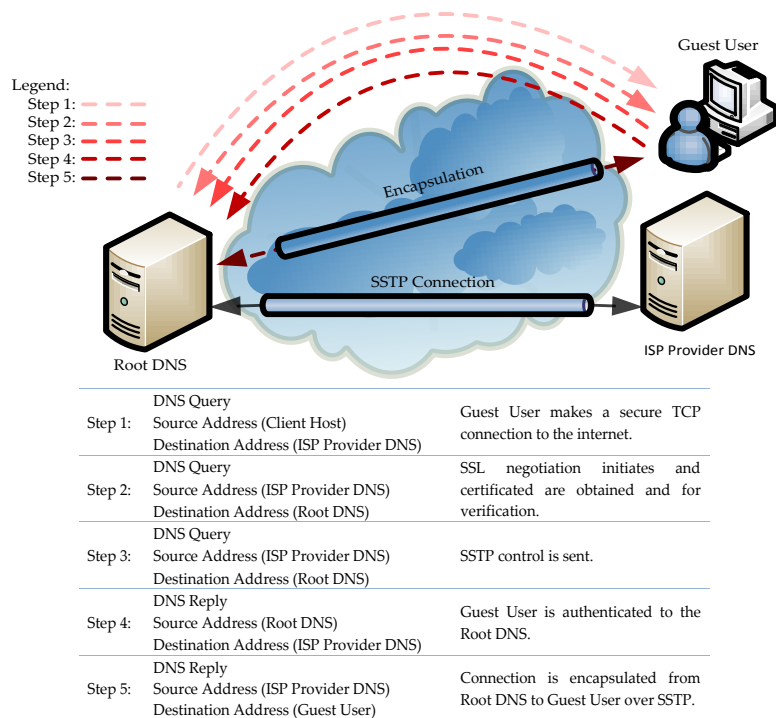


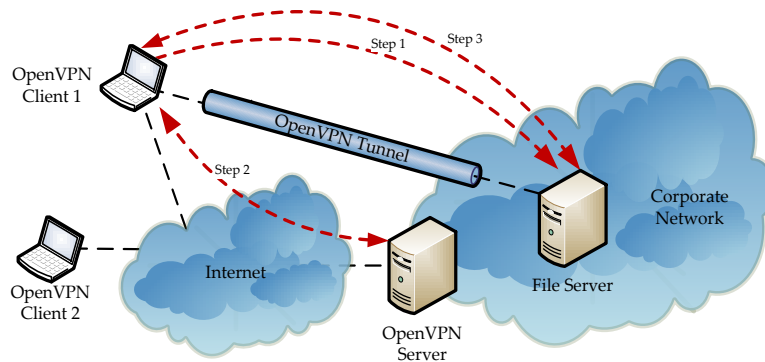
Figure 6.6: SFTP implementation via secure connection to the internet

For SFTP to establish a VPN tunnel between the client and the server, first a TCP connection between the client and the server is established, where communication using a dynamically allocated TCP on the client and TCP port 443 on the server are used (Figure 6.6). The client then sends a SSL-Hello message to create a session with the SFTP server. The server on receiving the SSL-Hello message, sends a certificate, which gets validated, the encryption method is determined and a SSL session is generated which then gets

encrypted with the SSTP servers public key. On completing this, the encrypted form of the SSL session key is sent by the client to the server, which the server decrypts with its private key. Afterwards, all communication between the SSTP client and the server is encrypted with the agreed encryption method using the SSL session key. The SSTP client also sends over a request for a HTTP session, and then negotiates an SSTP tunnel and a PPP connection with the SSTP server. In this negotiation, credentials are negotiated for the PPP authentication method and finally settings are configured for an either IPv4 or IPv6 traffic type. This completes the setup of SSTP, allowing the nodes to send IPv4 or IPv6 traffic over the negotiated PPP link.

SSTP was originally introduced with Microsoft Windows Vista SP1 where it provided a mechanism to encapsulate PPP traffic over a Secure HTTP (HTTPS) link. Nowadays, SSTP is commonly available in other server and router based operating systems, however, its dominance is mainly in the Microsoft environment. It is similar in functionality to Open VPN, however, when implemented on a vendor platform (unlike open source), its performance and security benefits are tremendously enhanced. Therefore SSTP is currently the favoured VPN protocol to enhance moving data security when safeguarding information is of utmost importance, such as credit card numbers and other confidential data.

Open VPN



	Echo Request	
Step 1:	Source Address (OpenVPN Client) Destination Address (File Server)	OpenVPN Client 1 attempts to access File Server.
Step 2:	Echo Request Source Address (OpenVPN Server) Destination Address (OpenVPN Client)	OpenVPN Server creates an encrypted UDP connection over internet going to File Server.
Step 3:	Echo Reply Source Address (File Server) Destination Address (OpenVPN Client)	A VPN tunnel and connection between File server and OpenVPN client are established.

Figure 6.7: OpenVPN implementation via secure remote connection

Open VPN is an open source tool used to build site-to-site VPNs with SSL and TLS protocols using pre-shared keys. It is a new and a desirable solution to make secure VPN since it combines several advantages of prior technologies (Kotuliak, Rybar, & Truchly, 2011). When it is needed to secure communication between server and client (Figure 6.7), regardless of protocol, then open VPN is an effective solution to secure data that is in motion. Open VPN can be implemented at layer two or three of OSI model and uses a single TCP and UDP port when establishing connection channel (Qu, Li, & Dang, 2012). Being client and server architecture, it must be installed on both VPN extremities. With open VPN a TCP or UDP port on the firewall needs to be opened (Yang, Ding, Wen, & Zhang, 2010). It can provide multiple incoming connections on same port and communicate through proxy servers. Therefore, there is no need to make changes to firewall configuration.

## SSL/TLS

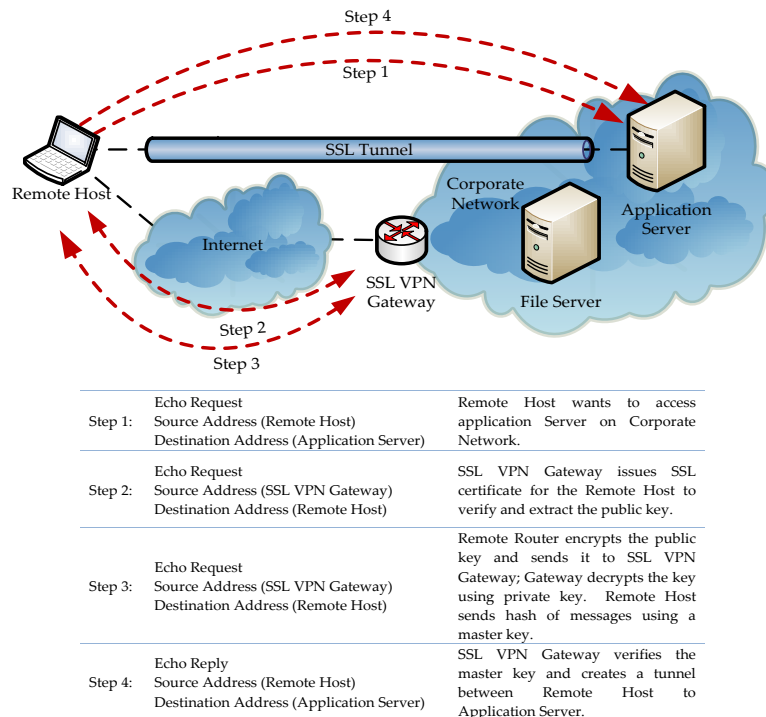


Figure 6.8: SSL implementation via secure remote connection

SSL provides a secure transport connection between applications. It was designed to provide encrypted and authenticated channels for web servers and clients (S. Sun, 2011) (Lakbabi, Orhanou, & El Hajji, 2012). SSL session is an association between a client and a server. The session state includes security algorithms and parameters. Connections of the same session share session state (Badra & Hajjeh, 2006). A session can include multiple secure connections between servers and clients. SSL protocol uses public key encryption for authentication (Kuihe & Xin, 2007) (Mao et al., 2012). SSL connection begins with handshake (Figure 6.8), during which applications exchange digital certificates, agree on the encryption algorithms, and generate encryption

keys used for remaining session (Kotuliak et al., 2011) (Farrell, 2010). Within SSL, three protocols exist: handshake protocol, record protocol and the alert protocol. At the beginning of the session, client authenticates with the server using handshake. Following that, all messages/data encryption is handled in record protocol phase. If there are any questionable packets that appear, then alert protocol comes into action.

TLS is suitable for stream-oriented applications. It uses secure remote password protocol for authentication (Otrok, Haraty, & El-Kassar, 2006). This protocol does not require trusted third party authentication and client certificates, like some of the other mechanisms do (Apostolopoulos, Peris, & Saha, 1999). Similar to SSL, TLS utilises the handshake and record protocol (Rescorla & Modadugu, 2006). These protocols are used to establish the shared key and using key to protect communication respectively.

## 6.4 Key VPN Research

There are many avenues that has been researched in the area on VPNs. Many of them has been already highlighted in this chapter. In Table 6.5 some other key researches have been presented. The listed outputs emphasise VPN related work undertaken using test-beds analysis. This is the same approach that has been taken in this thesis.

Author(s)	Title	Research Focus
Alexander Uskov and Hayk Avagyan (2014)	The Efficiency of Block Ciphers in Galois/Counter Mode in IPsec-Based Virtual Private Networks	IPSec
Cătălin Cioponea, Mihai Bucicoiu and Daniel Rosner (2013)	Analysis of VoIP Encryption Performance Using Dedicated Hardware	IPSec, 3DES DES and AES
Onur Demir and Tolga Aktas (2013)	Evaluation of Two Models for Securing SIP for Home Network Communications	IPSec and TLS
Antonio De Rubertis, Luca Mainetti, Vincenzo, and Stefano Pascali (2013)	Performance Evaluation of End-to-End Security Protocols in an Internet of Things	IPSec and TLS
Merima Kulin, Tarik Kazaz and Sasa Mrdovic (2012)	SIP Server Security with TLS: Relative Performance Evaluation	TLS
Alexander V. Uskov (2012)	Information Security of IPsec-Based Mobile VPN: Authentication and Encryption Algorithms Performance	IPSec AH, and ESP
Eman M. Mohamed, Sherif EI-Etriby and Hatem S. Abdul-kader (2012)	Randomness Testing of Modern Encryption Techniques in Cloud Environment	RC4, AES, 3DES and DES
Dherik Barison, Rodrigo Sanches Miani, Bruno Bogaz Zarpelão, Gean Davis Breda and Leonardo de Souza (2012)	Evaluation of Quality in Encrypted VoIP Calls	OpenVPN
Junhua Qu, Tao Li and Fangfang Dang (2012)	Performance Evaluation and Analysis of OpenVPN on Android	OpenVPN
O P Verma, Ritu Agarwal, Dhiraj Dafouti and Shobha Tyagi (2011)	Performance Analysis Of Data Encryption Algorithms	AES, DES and 3DES

Table 6.5: Recent research on performance evaluation of VPN protocols

## 6.5 VPN Test-bed Setup

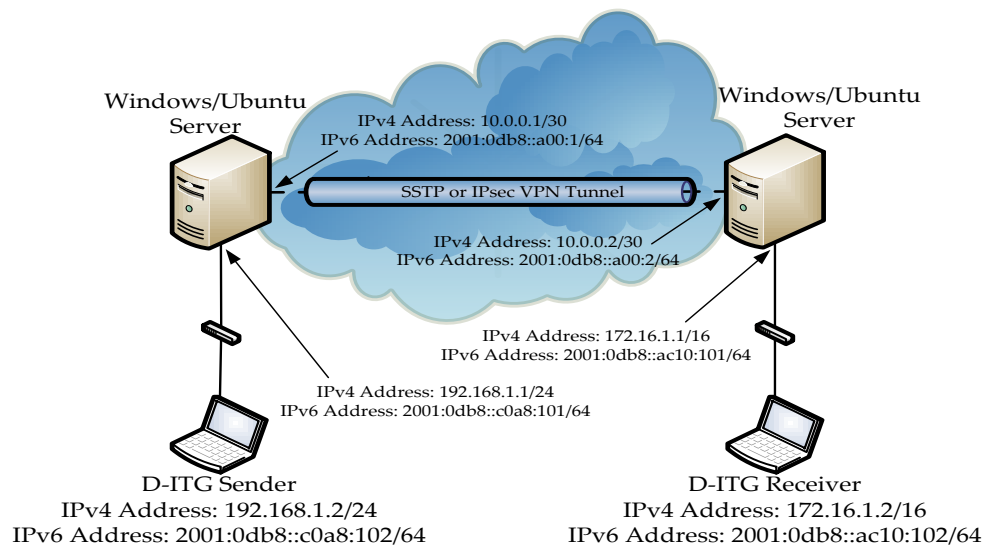


Figure 6.9: VPN network diagram

The test-bed used to evaluate VPN mechanisms is presented in Figure 6.9. Two servers and two computers were connected in the configuration shown in the figure. The two end nodes were purely setup for the purposes of generating and sending traffic through the tunnel, while the two servers acted as end points for the VPN tunnel. Each server was configured as a router since data sent had to traverse three networks

The two servers were then configured with various combinations of VPN techniques and encryption algorithms. Also, for VPN network performance analysis, the base operating systems on the servers were taken into consideration, therefore, three different operating systems were configured on them. On successfully configuring a particular configuration, D-ITG session was initiated and performance related data was collected. This is presented and discussed in the next section.



## 6.6 Performance Metrics Measurements from IPSec and SSTP VPN Protocols

To compare behaviour of TCP and UDP traffic type on networks with VPN, two VPN protocols were implemented with different combinations of selected associated algorithms. These combinations were implemented on networks with either IPv4 or IPv6, similar to the tests conducted in the previous two chapters. However, unlike in the previous two contexts, having VPN as the base operating system on the end servers has been taken into consideration as well. Therefore, all VPN tests have been conducted on three different operating systems (Windows Server 2008, Windows Server 2012 and Linux Ubuntu 12). This has been done to see if the operating system, in addition, to the actual VPN tunnel type, has an impact on the network performance of TCP and UDP traffic types. For comparative purposes, the first test in each combination was performed on a network without VPN, but with both versions of IP, one at a time.

The first performance metric presented is TCP throughput. In Figure 6.10 and the associated heat map in Figure 6.11 it can be seen that all scenarios record low throughput for small packet sizes, but thereafter the values increase in all cases, but by different degrees. Three different bands are evident, where the uppermost band has all the scenarios representing network without VPNs. All other lines underneath are for those with VPNs, therefore, this shows that VPNs do reduce network throughput when TCP traffic traverses them. The second band is comprised of lines representing six scenarios, the common factor between them is that all are for VPNs using SSTP as the tunnelling protocol. These lines are distinctly separate from networks without VPN and at the most register a throughput drop of approximately 40%. IPSec lines are in the third band with values averaging around 75Mbps for most packet sizes. This is a drop in throughput of almost five-fold from the same network without any VPN protocol. Also, no clear distinction can be made between the IP versions or the operating systems or the different algorithms in each category.

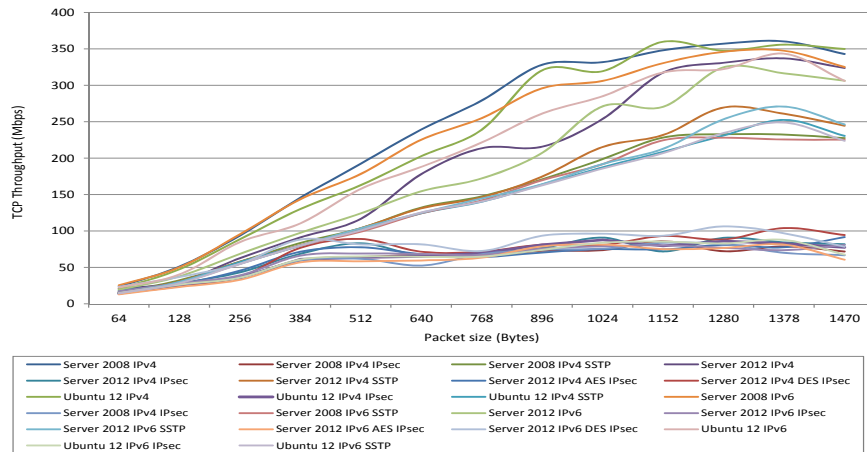


Figure 6.10: Graph of TCP throughput with VPN implemented on different operating systems

UDP throughput values follow a very similar pattern to that of TCP. This is evident in Figure 6.12 and emphasised in Figure 6.13. Here, the three bands are seen again, with the networks without VPN having the greatest throughput. The values are low for small packets and then increment steadily as the packet size increase. Differentiation in each band between the scenarios is not that obvious, but, the three bands themselves can be clearly distinguished. As in the case of TCP, the middle band has SSTP related lines on different operating systems and IP versions. The values average almost 20% lower than a network without VPN. The lowest band represents all other scenarios averaging almost 75Mbps for most packet sizes. This, again, is approximately five-fold lower than the pure networks.

Comparing TCP and UDP throughput values, it can be seen that the difference between the two traffic types is not that significant, however, in all cases the degree to which throughput varies is dependant on the actual VPN protocol implemented. SSTP has much higher throughput readings than IPsec, and both the graphs and the heat maps show that there are some differences. Therefore, throughput on VPN implementations is plausible as a tool for ordinal ranking of the technologies.

6.6. PERFORMANCE METRICS MEASUREMENTS FROM IPSEC AND SSTP VPN PROTOCOLS

TCP Throughput: VPN													
	64	128	256	384	512	640	768	896	1024	1152	1280	1378	1470
Server 2008 IPv4	0.07	0.14	0.26	0.40	0.53	0.66	0.77	0.91	0.92	0.97	0.99	1.00	0.95
Server 2008 IPv4 IPsec	0.04	0.07	0.10	0.17	0.18	0.19	0.18	0.20	0.20	0.24	0.20	0.22	0.20
Server 2008 IPv4 SSTP	0.05	0.09	0.16	0.23	0.29	0.37	0.41	0.47	0.55	0.63	0.65	0.65	0.63
Server 2012 IPv4	0.06	0.08	0.17	0.25	0.32	0.49	0.59	0.60	0.70	0.88	0.92	0.94	0.90
Server 2012 IPv4 IPsec	0.05	0.08	0.12	0.19	0.23	0.18	0.19	0.21	0.25	0.20	0.25	0.23	0.23
Server 2012 IPv4 SSTP	0.04	0.09	0.16	0.22	0.29	0.36	0.41	0.49	0.60	0.64	0.75	0.72	0.68
Server 2012 IPv4 AES IPsec	0.05	0.08	0.13	0.20	0.22	0.19	0.18	0.20	0.21	0.21	0.22	0.22	0.25
Server 2012 IPv4 DES IPsec	0.04	0.07	0.10	0.21	0.25	0.20	0.20	0.23	0.22	0.26	0.25	0.29	0.26
Ubuntu 12 IPv4	0.06	0.13	0.25	0.36	0.45	0.56	0.66	0.89	0.89	1.00	0.96	0.99	0.97
Ubuntu 12 IPv4 IPsec	0.04	0.07	0.10	0.17	0.18	0.18	0.19	0.23	0.24	0.22	0.24	0.23	0.22
Ubuntu 12 IPv4 SSTP	0.04	0.08	0.16	0.22	0.28	0.35	0.39	0.45	0.52	0.58	0.64	0.70	0.64
Server 2008 IPv6	0.07	0.14	0.27	0.40	0.50	0.63	0.71	0.82	0.85	0.92	0.96	0.96	0.90
Server 2008 IPv4 IPsec	0.04	0.07	0.10	0.17	0.17	0.15	0.19	0.21	0.21	0.23	0.23	0.19	0.19
Server 2008 IPv6 SSTP	0.04	0.08	0.15	0.22	0.28	0.35	0.39	0.47	0.53	0.62	0.63	0.63	0.63
Server 2012 IPv6	0.05	0.10	0.19	0.27	0.34	0.43	0.48	0.58	0.75	0.75	0.90	0.88	0.85
Server 2012 IPv6 IPsec	0.05	0.08	0.11	0.18	0.19	0.19	0.19	0.21	0.22	0.22	0.21	0.20	0.22
Server 2012 IPv6 SSTP	0.04	0.08	0.15	0.22	0.29	0.35	0.40	0.46	0.53	0.59	0.70	0.75	0.68
Server 2012 IPv6 AES IPsec	0.04	0.06	0.09	0.16	0.16	0.17	0.18	0.22	0.23	0.21	0.21	0.23	0.17
Server 2012 IPv6 DES IPsec	0.07	0.10	0.15	0.25	0.23	0.23	0.20	0.26	0.27	0.26	0.30	0.27	0.22
Ubuntu 12 IPv6	0.06	0.11	0.24	0.31	0.44	0.52	0.61	0.73	0.79	0.88	0.89	0.95	0.85
Ubuntu 12 IPv6 IPsec	0.04	0.07	0.10	0.17	0.18	0.18	0.18	0.21	0.23	0.24	0.23	0.24	0.19
Ubuntu 12 IPv6 SSTP	0.04	0.08	0.15	0.22	0.28	0.35	0.39	0.45	0.52	0.57	0.65	0.69	0.62

Figure 6.11: Heat map of TCP throughput with VPN implemented on different operating systems

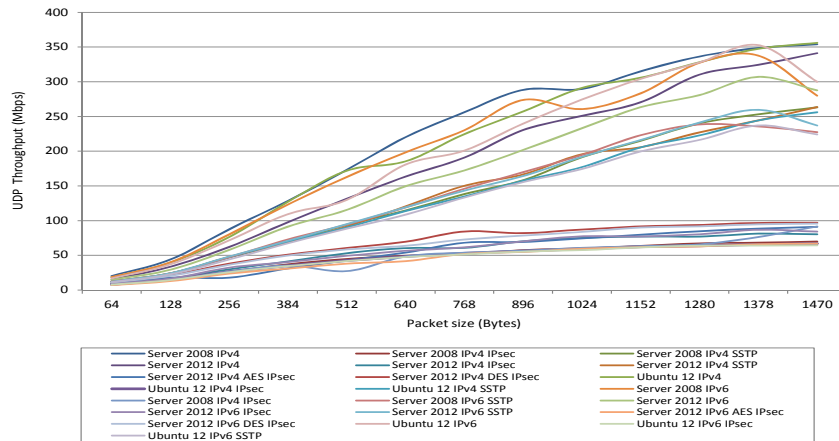


Figure 6.12: Graph of UDP throughput with VPN implemented on different operating systems

The next network performance metric presented is delay. In previous chapters it can be seen that delay is a good differentiating entity in the other network scenarios and in the case of VPN protocols it is the same. Delay is a sensitive metric to measure since it is time dependant, therefore, the synchronisation of clocks and associated accuracies plays a vital role in getting sensible data. Since delay values are small (measured in fraction of a millisecond), and precision of inbuilt computer clocks are questionable at this minute level, expensive external time synchronisation devices are necessary where precise delay values are to be observed. In this thesis the focus has been to compare delay attained in one scenario to that in another, and for this purpose measurements have been taken that rely on the accuracy of the inbuilt computer clock (due to non-availability of external time synchronisation devices). For this reason, some of the readings presented in delay graphs are negative values. This is acceptable for the purposes of a comparative study, where the focus is to evaluate the difference between various scenarios. In the case of transition mechanisms and wireless, no negative results for delay were measured, however this is not the case for VPN.

6.6. PERFORMANCE METRICS MEASUREMENTS FROM IPSEC AND SSTP VPN PROTOCOLS

UDP Throughput: VPN													
	64	128	256	384	512	640	768	896	1024	1152	1280	1378	1470
Server 2008 IPv4	0.06	0.12	0.25	0.36	0.49	0.62	0.72	0.81	0.81	0.88	0.95	0.98	0.99
Server 2008 IPv4 IPsec	0.02	0.04	0.07	0.10	0.13	0.14	0.15	0.16	0.17	0.18	0.19	0.19	0.20
Server 2008 IPv4 SSTP	0.04	0.07	0.13	0.19	0.26	0.32	0.39	0.44	0.54	0.60	0.67	0.71	0.74
Server 2012 IPv4	0.04	0.09	0.17	0.27	0.37	0.46	0.54	0.65	0.70	0.76	0.87	0.91	0.96
Server 2012 IPv4 IPsec	0.03	0.05	0.08	0.12	0.15	0.17	0.17	0.20	0.21	0.22	0.22	0.23	0.23
Server 2012 IPv4 SSTP	0.03	0.07	0.13	0.19	0.26	0.34	0.42	0.47	0.55	0.58	0.64	0.69	0.74
Server 2012 IPv4 AES IPsec	0.03	0.05	0.05	0.09	0.12	0.15	0.19	0.19	0.21	0.22	0.24	0.25	0.26
Server 2012 IPv4 DES IPsec	0.03	0.06	0.11	0.14	0.17	0.20	0.24	0.23	0.24	0.26	0.26	0.27	0.27
Ubuntu 12 IPv4	0.05	0.11	0.21	0.36	0.48	0.52	0.63	0.72	0.82	0.86	0.92	0.98	1.00
Ubuntu 12 IPv4 IPsec	0.02	0.04	0.07	0.10	0.12	0.14	0.15	0.16	0.17	0.17	0.18	0.18	0.19
Ubuntu 12 IPv4 SSTP	0.03	0.06	0.13	0.20	0.25	0.32	0.38	0.44	0.50	0.58	0.63	0.69	0.72
Server 2008 IPv6	0.05	0.11	0.23	0.35	0.46	0.56	0.65	0.77	0.73	0.80	0.92	0.95	0.95
Server 2008 IPv4 IPsec	0.02	0.04	0.07	0.10	0.08	0.14	0.15	0.16	0.17	0.18	0.18	0.21	0.26
Server 2008 IPv6 SSTP	0.03	0.07	0.13	0.20	0.26	0.33	0.41	0.48	0.55	0.63	0.67	0.66	0.64
Server 2012 IPv6	0.04	0.08	0.16	0.26	0.32	0.42	0.48	0.57	0.65	0.74	0.79	0.86	0.81
Server 2012 IPv6 IPsec	0.03	0.05	0.09	0.11	0.14	0.16	0.17	0.20	0.22	0.22	0.23	0.24	0.24
Server 2012 IPv6 SSTP	0.03	0.07	0.14	0.20	0.27	0.33	0.40	0.46	0.54	0.61	0.68	0.73	0.67
Server 2012 IPv6 AES IPsec	0.02	0.04	0.07	0.09	0.11	0.12	0.15	0.15	0.17	0.17	0.18	0.19	0.19
Server 2012 IPv6 DES IPsec	0.03	0.06	0.10	0.14	0.16	0.18	0.20	0.22	0.24	0.25	0.26	0.27	0.27
Ubuntu 12 IPv6	0.05	0.10	0.20	0.31	0.36	0.51	0.56	0.67	0.77	0.85	0.92	0.99	0.56
Ubuntu 12 IPv6 IPsec	0.02	0.04	0.07	0.10	0.12	0.13	0.14	0.16	0.16	0.17	0.18	0.18	0.18
Ubuntu 12 IPv6 SSTP	0.03	0.06	0.12	0.19	0.25	0.30	0.37	0.44	0.49	0.56	0.61	0.67	0.63

Figure 6.13: Heat map of UDP throughput with VPN implemented on different operating systems

TCP delay results shown in Figure 6.14 distinctly show that there is significant difference between some of the VPN technologies implemented, but also show that some technologies cannot be differentiated based on TCP delay. This is emphasised in the associated heat map presented in Figure 6.15. The graph and the heat map also show that there are three clusters into which the VPN scenarios are grouped. All lines have zero gradient, indicating that TCP delay values do not change with packet size in the VPN implementations. The lowest delays are registered by two VPN scenarios, both SSTP with IP versions 4 and 6. Also observed is that these are implemented in the latest version of the operating system. So here it can be seen that the newest of the VPN protocols tested gives the most favourable TCP delay. Most of the scenarios tested are grouped together into the middle cluster, where the difference between the lines is not that obvious. The worst reported TCP delay, with values almost double that of the best readings, are that of the IPsec VPN technique when implemented on both IPv4 and IPv6. Again, as in the case of the best values, the worst values are registered in the latest version of the operating systems tested.

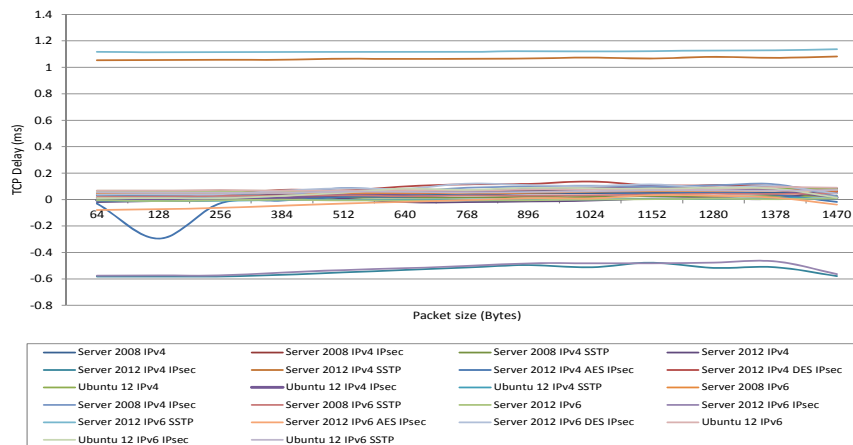


Figure 6.14: Graph of TCP delay with VPN implemented on different operating systems

6.6. PERFORMANCE METRICS MEASUREMENTS FROM IPSEC AND SSTP VPN PROTOCOLS

TCP Delay: VPN													
	64	128	256	384	512	640	768	896	1024	1152	1280	1378	1470
Server 2008 IPv4	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.04	-0.04	-0.04	-0.04	-0.05	-0.04
Server 2008 IPv4 IPsec	0.00	0.00	0.00	-0.01	-0.02	-0.04	-0.06	-0.06	-0.07	-0.06	-0.09	-0.08	-0.05
Server 2008 IPv4 SSTP	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02
Server 2012 IPv4	0.02	0.01	0.00	-0.04	-0.01	0.02	0.02	0.01	0.01	-0.01	-0.01	-0.01	-0.07
Server 2012 IPv4 IPsec	0.50	0.50	0.50	0.49	0.48	0.46	0.44	0.43	0.44	0.41	0.45	0.44	0.50
Server 2012 IPv4 SSTP	-0.91	-0.91	-0.91	-0.91	-0.92	-0.92	-0.92	-0.92	-0.93	-0.92	-0.91	-0.93	-0.93
Server 2012 IPv4 AES IPsec	0.03	0.26	0.02	0.01	-0.02	-0.02	-0.03	-0.05	-0.05	-0.04	-0.05	-0.03	0.02
Server 2012 IPv4 DES IPsec	-0.04	-0.03	-0.03	-0.06	-0.06	-0.09	-0.10	-0.10	-0.12	-0.09	-0.09	-0.09	-0.05
Ubuntu 12 IPv4	-0.06	-0.06	-0.06	-0.06	-0.06	-0.06	-0.06	-0.07	-0.07	-0.07	-0.07	-0.07	-0.07
Ubuntu 12 IPv4 IPsec	0.00	0.00	0.00	-0.01	-0.03	-0.04	-0.05	-0.07	-0.07	-0.09	-0.07	-0.06	-0.01
Ubuntu 12 IPv4 SSTP	0.01	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.01	-0.01
Server 2008 IPv6	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.05	-0.05	-0.06	-0.06	-0.06
Server 2008 IPv6 IPsec	-0.01	-0.01	-0.01	-0.03	-0.08	-0.06	-0.08	-0.09	-0.09	-0.08	-0.09	-0.10	-0.02
Server 2008 IPv6 SSTP	-0.02	-0.02	-0.02	-0.03	-0.03	-0.02	-0.03	-0.03	-0.03	-0.03	-0.04	-0.04	-0.04
Server 2012 IPv6	0.00	0.01	0.01	0.00	0.01	0.01	0.01	0.01	0.00	0.00	-0.01	0.00	-0.01
Server 2012 IPv6 IPsec	0.50	0.50	0.49	0.48	0.46	0.45	0.43	0.42	0.42	0.42	0.41	0.40	0.49
Server 2012 IPv6 SSTP	-0.96	-0.96	-0.96	-0.96	-0.96	-0.96	-0.96	-0.97	-0.97	-0.97	-0.97	-0.97	-0.98
Server 2012 IPv6 AES IPsec	0.07	0.06	0.05	0.04	0.03	0.01	0.00	-0.01	-0.01	-0.03	-0.02	-0.01	0.03
Server 2012 IPv6 DES IPsec	-0.03	-0.03	-0.03	-0.06	-0.07	-0.07	-0.10	-0.09	-0.09	-0.10	-0.08	-0.09	-0.04
Ubuntu 12 IPv6	-0.06	-0.06	-0.06	-0.06	-0.06	-0.06	-0.06	-0.07	-0.07	-0.07	-0.07	-0.08	-0.08
Ubuntu 12 IPv6 IPsec	-0.01	-0.01	-0.01	-0.02	-0.04	-0.07	-0.06	-0.08	-0.07	-0.07	-0.08	-0.07	-0.01
Ubuntu 12 IPv6 SSTP	-0.05	-0.04	-0.04	-0.04	-0.05	-0.05	-0.05	-0.05	-0.05	-0.06	-0.06	-0.06	-0.06

Figure 6.15: Heat map of TCP delay with VPN implemented on different operating systems

As in the case of TCP delay, UDP delay, presented in Figure 6.16 and associated heat map in Figure 6.17 shows that there is a clear differentiation between a handful of the VPN technologies. It also shows that in most cases UDP traffic delay is the same and therefore to some extent, UDP delay may not be a good differentiating metric in the majority of VPN scenarios. Similar to TCP delay results, UDP delay also shows that the values attained are independent of the actual packet transmission size.

Further, it can be seen that there are two scenarios in which there are highly favourable delays. For UDP unlike TCP, IPsec, for both versions of IP, is superior. SSTP, which registered the most favourable TCP delay values, is showing the highest UDP delay values. This is the case for both IP versions. So overall, delay as a performance metric is a good differentiator in VPN implementations, identifying the extreme cases of the best and worst VPN scenarios for TCP and UDP traffic. Overall, the TCP and UDP delay values attained in all scenarios are comparable.

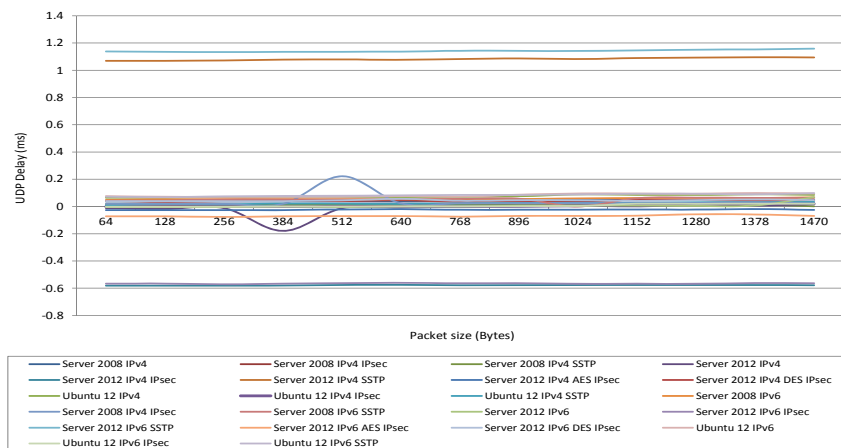


Figure 6.16: Graph of UDP delay with VPN implemented on different operating systems



6.6. PERFORMANCE METRICS MEASUREMENTS FROM IPSEC AND SSTP VPN PROTOCOLS

UDP Delay: VPN													
	64	128	256	384	512	640	768	896	1024	1152	1280	1378	1470
Server 2008 IPv4	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.04	0.04	0.04	0.04	0.04	0.04
Server 2008 IPv4 IPsec	0.00	0.00	0.01	0.00	0.00	0.00	0.01	0.02	0.02	0.02	0.02	0.02	0.02
Server 2008 IPv4 SSTP	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.01	0.01	0.01	0.00
Server 2012 IPv4	-0.01	-0.02	-0.01	-0.16	-0.01	-0.01	0.00	0.00	0.00	0.00	0.01	0.01	0.01
Server 2012 IPv4 IPsec	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50
Server 2012 IPv4 SSTP	0.92	0.92	0.93	0.93	0.93	0.93	0.93	0.94	0.93	0.94	0.94	0.95	0.95
Server 2012 IPv4 AES IPsec	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02
Server 2012 IPv4 DES IPsec	0.04	0.03	0.03	0.04	0.03	0.04	0.03	0.03	0.03	0.04	0.04	0.04	0.04
Ubuntu 12 IPv4	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.08	0.07	0.07	0.08	0.07
Ubuntu 12 IPv4 IPsec	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.02	0.02	0.02
Ubuntu 12 IPv4 SSTP	0.01	0.02	0.01	0.02	0.02	0.02	0.02	0.02	0.02	0.03	0.03	0.03	0.03
Server 2008 IPv6	0.04	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.05	0.06
Server 2008 IPv4 IPsec	0.02	0.02	0.02	0.02	0.19	0.02	0.02	0.02	0.02	0.02	0.02	0.03	0.02
Server 2008 IPv6 SSTP	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.06	0.05	0.05
Server 2012 IPv6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.06
Server 2012 IPv6 IPsec	-0.49	-0.49	-0.49	-0.49	-0.49	-0.49	-0.49	-0.49	-0.49	-0.49	-0.49	-0.49	-0.49
Server 2012 IPv6 SSTP	0.98	0.98	0.98	0.98	0.98	0.98	0.99	0.99	0.99	0.99	0.99	1.00	1.00
Server 2012 IPv6 AES IPsec	-0.06	-0.06	-0.07	-0.06	-0.06	-0.06	-0.06	-0.06	-0.06	-0.06	-0.05	-0.05	-0.06
Server 2012 IPv6 DES IPsec	0.03	0.03	0.03	0.03	0.04	0.05	0.04	0.04	0.04	0.04	0.04	0.04	0.04
Ubuntu 12 IPv6	0.07	0.06	0.06	0.06	0.06	0.07	0.06	0.07	0.08	0.08	0.08	0.09	0.08
Ubuntu 12 IPv6 IPsec	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.02	0.02	0.02
Ubuntu 12 IPv6 SSTP	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.08	0.08	0.08	0.08	0.08

Figure 6.17: Heat map of UDP delay with VPN implemented on different operating systems

The final metric measured on networks with various VPN technologies is jitter. Most jitter values registered are low, as evident for TCP in Figure 6.18 and for UDP in Figure 6.21. In both, the maximum value recorded was 0.0004ms. This shows that jitter is not a significant issue on networks with VPN. However, even in these minuscule values, there are patterns that can be used to see the differences between some of the VPN scenarios implemented. This is highlighted in the TCP and UDP heat maps shown in Figures 6.19 and 6.20 respectively. In all graphs it can be seen that the lowest jitter values are registered by networks without any VPN implementations. Also in this band are the SSTP lines, that is, jitter on networks with SSTP is comparable with networks without any VPN. For all the other scenarios, there is a difference in jitter between the small packets and the large packets, where for larger packets jitter values increase steadily. This is the case for both TCP and UDP. A greater level of distinction between the scenarios is seen in the case of UDP - this is mostly true for large packet sizes. So overall, jitter as a performance metric on VPN networks is an average differentiator of TCP/UDP behaviour on VPN networks and can be used as a mechanism to ordinal rank different implementations.

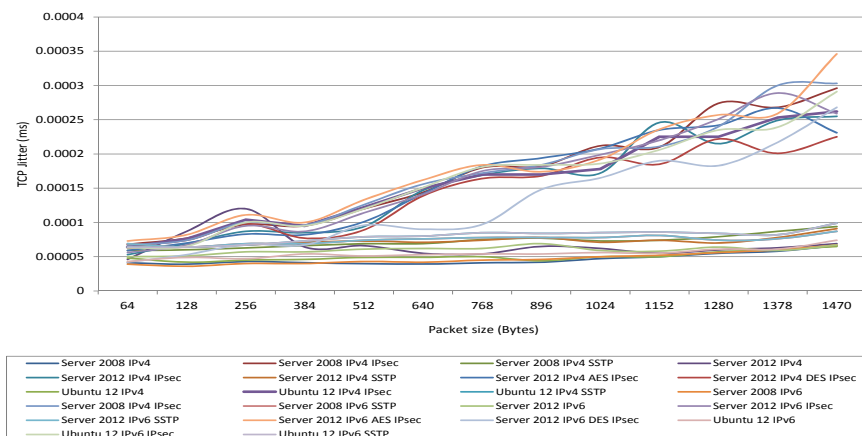


Figure 6.18: Graph of TCP jitter with VPN implemented on different operating systems

6.6. PERFORMANCE METRICS MEASUREMENTS FROM IPSEC AND SSTP VPN PROTOCOLS

TCP Jitter: VPN													
	64	128	256	384	512	640	768	896	1024	1152	1280	1378	1470
Server 2008 IPv4	0.12	0.11	0.12	0.12	0.12	0.11	0.12	0.12	0.14	0.14	0.16	0.17	0.19
Server 2008 IPv4 IPsec	0.20	0.22	0.28	0.27	0.35	0.42	0.52	0.53	0.61	0.61	0.79	0.77	0.86
Server 2008 IPv4 SSTP	0.17	0.17	0.18	0.19	0.20	0.20	0.22	0.22	0.21	0.21	0.23	0.25	0.27
Server 2012 IPv4	0.13	0.25	0.35	0.18	0.19	0.16	0.16	0.19	0.18	0.16	0.17	0.18	0.20
Server 2012 IPv4 IPsec	0.15	0.20	0.25	0.25	0.27	0.42	0.49	0.52	0.50	0.71	0.62	0.72	0.74
Server 2012 IPv4 SSTP	0.18	0.18	0.19	0.21	0.21	0.21	0.21	0.23	0.21	0.21	0.20	0.23	0.26
Server 2012 IPv4 AES IPsec	0.17	0.20	0.24	0.24	0.29	0.41	0.53	0.56	0.60	0.68	0.70	0.77	0.67
Server 2012 IPv4 DES IPsec	0.19	0.21	0.28	0.22	0.26	0.40	0.47	0.49	0.56	0.53	0.64	0.58	0.65
Ubuntu 12 IPv4	0.14	0.12	0.13	0.13	0.14	0.14	0.14	0.13	0.14	0.14	0.16	0.17	0.19
Ubuntu 12 IPv4 IPsec	0.18	0.22	0.30	0.27	0.35	0.43	0.49	0.49	0.51	0.65	0.65	0.73	0.76
Ubuntu 12 IPv4 SSTP	0.18	0.18	0.19	0.21	0.23	0.23	0.25	0.24	0.25	0.25	0.24	0.24	0.29
Server 2008 IPv6	0.11	0.10	0.12	0.12	0.12	0.12	0.13	0.13	0.14	0.15	0.16	0.17	0.19
Server 2008 IPv6 IPsec	0.19	0.21	0.29	0.28	0.36	0.45	0.50	0.53	0.60	0.61	0.69	0.87	0.88
Server 2008 IPv6 SSTP	0.20	0.18	0.20	0.20	0.21	0.22	0.23	0.23	0.23	0.23	0.21	0.22	0.25
Server 2012 IPv6	0.14	0.15	0.16	0.16	0.18	0.18	0.18	0.20	0.17	0.17	0.18	0.17	0.19
Server 2012 IPv6 IPsec	0.16	0.19	0.27	0.25	0.33	0.41	0.51	0.52	0.58	0.64	0.73	0.84	0.75
Server 2012 IPv6 SSTP	0.20	0.18	0.20	0.20	0.21	0.22	0.23	0.23	0.23	0.23	0.21	0.22	0.25
Server 2012 IPv6 AES IPsec	0.21	0.24	0.32	0.29	0.38	0.47	0.53	0.50	0.55	0.68	0.74	0.75	1.00
Server 2012 IPv6 DES IPsec	0.12	0.15	0.20	0.19	0.28	0.03	0.03	0.43	0.48	0.55	0.53	0.63	0.77
Ubuntu 12 IPv6	0.12	0.14	0.14	0.16	0.15	0.15	0.16	0.16	0.16	0.16	0.18	0.18	0.21
Ubuntu 12 IPv6 IPsec	0.18	0.19	0.29	0.27	0.35	0.44	0.52	0.53	0.54	0.60	0.68	0.69	0.84
Ubuntu 12 IPv6 SSTP	0.18	0.18	0.19	0.21	0.23	0.23	0.25	0.24	0.25	0.25	0.24	0.24	0.29

Figure 6.19: Heat map of TCP jitter with VPN implemented on different operating systems

UDP Jitter: VPN													
	64	128	256	384	512	640	768	896	1024	1152	1280	1378	1470
Server 2008 IPv4	0.16	0.15	0.15	0.15	0.15	0.14	0.15	0.02	0.18	0.18	0.19	0.20	0.21
Server 2008 IPv4 IPsec	0.36	0.37	0.47	0.49	0.55	0.63	0.73	0.74	0.82	0.83	0.87	0.93	0.94
Server 2008 IPv4 SSTP	0.26	0.27	0.27	0.29	0.28	0.28	0.28	0.29	0.28	0.28	0.28	0.28	0.29
Server 2012 IPv4	0.21	0.19	0.21	0.19	0.19	0.20	0.20	0.20	0.21	0.22	0.21	0.21	0.21
Server 2012 IPv4 IPsec	0.32	0.40	0.42	0.43	0.48	0.52	0.62	0.62	0.66	0.71	0.80	0.80	0.86
Server 2012 IPv4 SSTP	0.26	0.27	0.27	0.28	0.28	0.27	0.26	0.28	0.27	0.29	0.29	0.29	0.29
Server 2012 IPv4 AES IPsec	0.32	0.36	0.38	0.42	0.47	0.52	0.55	0.62	0.66	0.61	0.65	0.67	0.71
Server 2012 IPv4 DES IPsec	0.28	0.30	0.34	0.04	0.42	0.45	0.50	0.52	0.56	0.61	0.61	0.65	0.67
Ubuntu 12 IPv4	0.18	0.17	0.17	0.16	0.16	0.18	0.18	0.18	0.18	0.19	0.20	0.20	0.02
Ubuntu 12 IPv4 IPsec	0.37	0.44	0.46	0.50	0.57	0.62	0.66	0.73	0.83	0.86	0.91	0.96	0.90
Ubuntu 12 IPv4 SSTP	0.27	0.28	0.26	0.27	0.28	0.28	0.28	0.27	0.28	0.26	0.28	0.28	0.28
Server 2008 IPv6	0.17	0.16	0.16	0.16	0.16	0.16	0.17	0.17	0.20	0.20	0.20	0.21	0.61
Server 2008 IPv4 IPsec	0.37	0.42	0.49	0.47	0.62	0.64	0.70	0.76	0.80	0.84	0.88	0.86	0.70
Server 2008 IPv6 SSTP	0.26	0.27	0.27	0.28	0.27	0.28	0.28	0.29	0.27	0.28	0.28	0.27	0.32
Server 2012 IPv6	0.23	0.22	0.21	0.22	0.21	0.22	0.20	0.21	0.23	0.22	0.22	0.24	0.26
Server 2012 IPv6 IPsec	0.34	0.39	0.40	0.47	0.51	0.57	0.62	0.64	0.71	0.75	0.76	0.81	1.85
Server 2012 IPv6 SSTP	0.26	0.27	0.27	0.28	0.27	0.28	0.28	0.29	0.27	0.28	0.28	0.27	0.32
Server 2012 IPv6 AES IPsec	0.41	0.46	0.50	0.53	0.65	0.75	0.72	0.78	0.82	0.83	0.90	1.00	0.94
Server 2012 IPv6 DES IPsec	0.30	0.31	0.34	0.39	0.43	0.49	0.51	0.55	0.59	0.62	0.67	0.69	0.74
Ubuntu 12 IPv6	0.20	0.18	0.19	0.18	0.21	0.18	0.20	0.19	0.19	0.20	0.20	0.20	0.38
Ubuntu 12 IPv6 IPsec	0.42	0.43	0.48	0.53	0.58	0.62	0.70	0.80	0.87	0.88	0.91	0.99	1.00
Ubuntu 12 IPv6 SSTP	0.27	0.28	0.26	0.27	0.28	0.28	0.28	0.27	0.28	0.26	0.28	0.28	0.28

Figure 6.20: Heat map of UDP jitter with VPN implemented on different operating systems

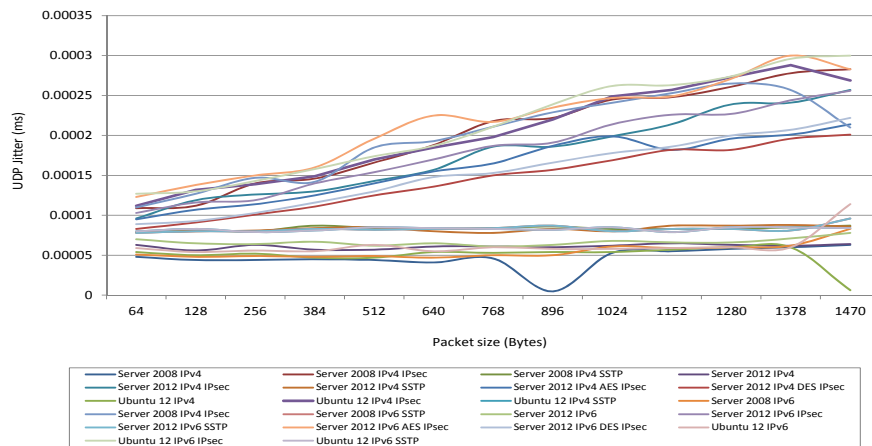


Figure 6.21: Graph of UDP jitter with VPN implemented on different operating systems

Finally, to see the effect different operating systems have on network performance in a given scenario, graphs in Figures 6.22-6.24 are presented. In each of the graphs shown, the only variable is the operating system. All the scenarios presented show that the operating system definitely affects network performance metrics, but the impact each has, is to a different degree. The set of throughput graphs (Figure 6.22) collectively show that operating systems can vary throughput by almost 50% in some network scenarios. Linux Ubuntu is generally outperformed by a Microsoft operating system in all cases, but to different degrees. Delay graphs presented in 6.23 represent only IPv4 implementations, and in all cases there is a clear distinction between the operating systems. In some of the graphs the difference between the operating systems is substantial. No one particular vendor operating system is a winner, however, distinction between the operating systems is noteworthy. The same is the case in IPv6 delays (Figure 6.24) where the difference between the operating systems is again emphasised. So overall, it is evident that operating systems do affect network performance metrics, however, the degree to which they impact depends on the actual metrics and the chosen operating system.

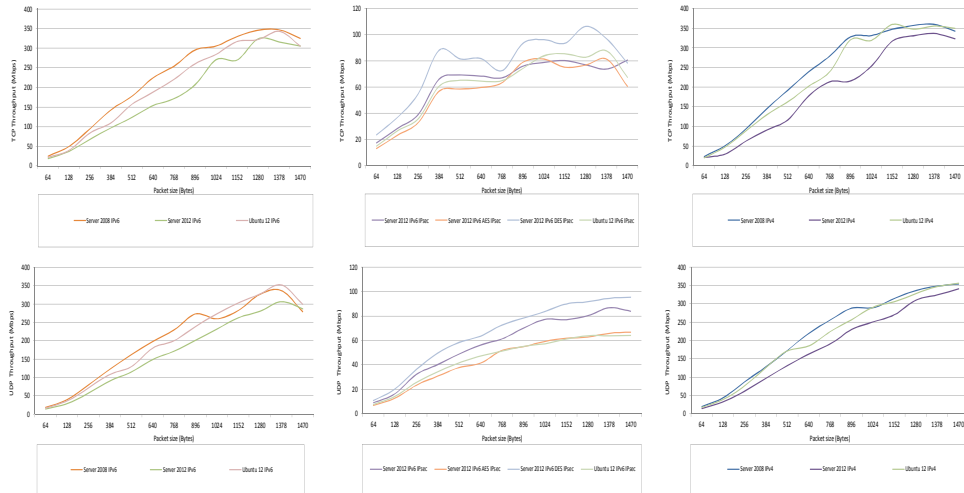


Figure 6.22: Collective graphs of throughput emphasizing characteristics of various operating systems

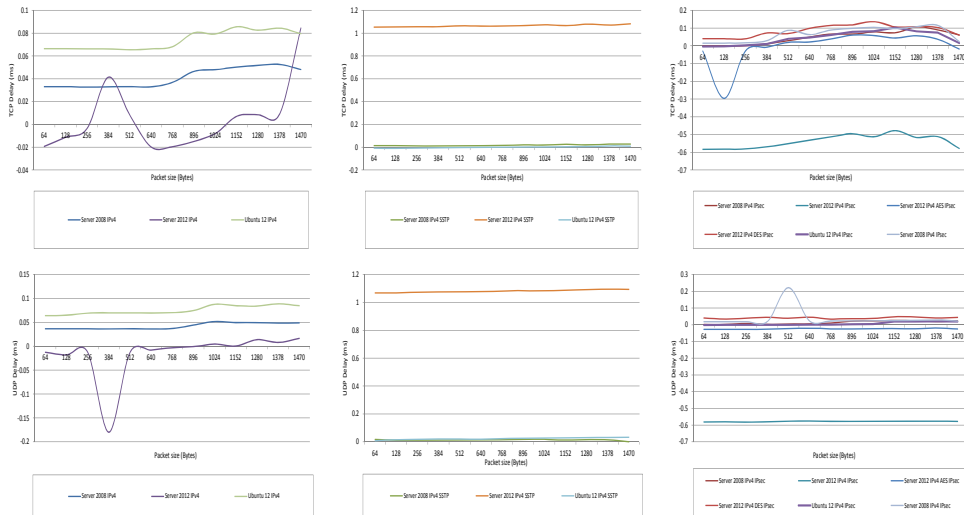


Figure 6.23: Collective graphs of IPv4 delay emphasizing characteristics of various operating systems

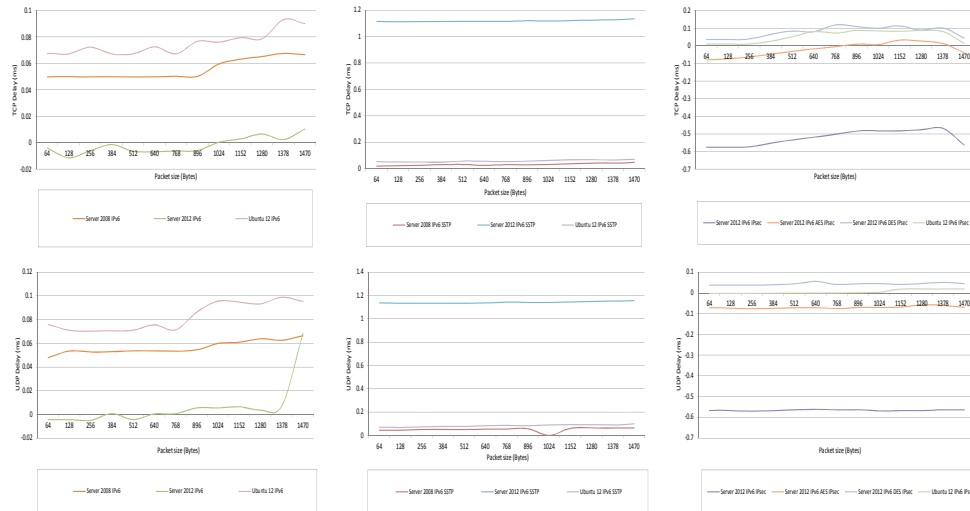


Figure 6.24: Collective graphs of IPv6 delay emphasizing characteristics of various operating systems

## 6.7 Results Evaluation

The network performance of TCP and UDP traffic types on VPN implementations, along with related, performance metrics, have been presented. In addition to ascertaining the difference between the two VPN tunnel techniques (and some of the associated algorithms), there has been an emphasis on finding out if the base operating system used on VPN servers have an impact on the network performance. Some of the key overall observations are highlighted herewith.

It is cleared that VPNs do have an impact on throughput. When comparing SSTP and IPsec, the former is registering a throughput value drop by approximately 40% (from networks with VPN), while the latter has an almost five-fold drop. This also highlights the difference between the two VPN protocols and shows that the difference is significant. TCP and UDP seem to follow similar trends and the difference between them is minimal. Also, it has been observed that there are three bands that can be identified in each of

the throughput graphs.

Delay proved to be a good indicator of difference between the VPN scenarios tested. This metric shows clusters of VPN scenarios in both TCP and UDP, however, it also shows a clear distinction between individual clusters. Negative delay has been reported in VPN tests and this is mainly due to the precision level of the inbuilt computer levels. For TCP traffic, the lowest delay is registered by SSTP with both versions of IP, while for UDP traffic it is seen that IPsec registered the lowest values. The degree to which variance in delays occurs is significant, however, network behaviour based on delay is similar for both TCP and UDP traffic types.

Jitter values registered on VPN networks are interesting since all values are comparatively small. However, even with the upper bound of approximately 0.0004ms, there are some patterns that can be identified in the presented data. In the case of TCP, there exist two bands but this is just the case for larger packet sizes. For smaller packets, there is no clear distinction between the scenarios. In the case of UDP, again there are two clusters of data, however the grouping here is different to that of TCP.

Undoubtedly various VPN technologies impact on network performance, but to different degrees. From the graphs in this chapter, it is also evident that the operating systems also impact on network performance. However, taking into consideration all the different performance metrics, there is no one clear operating system that outperforms the others. The results obtained show that the degree to which an operating system impacts on performance metrics is operating system dependent.

The test-bed results presented in this chapter have clearly shown that VPN technology selection is critical when it comes to ensuring network performance is at its optimum. The actual encryption and authentication algorithm of each SSTP or IPsec does not have any drastic impact on the test-bed, however, the base operating system does.



### 7.1 Preamble

The focus of this thesis is to evaluate the network performance of TCP and UDP traffic types on networks in different implementations. To achieve this, various networks were configured on test-beds in three different contexts. In total, 50 different networks were configured, comprising of: two with IPv4 and IPv6, seven with various transition mechanisms, nineteen with different wireless standards and encryption techniques, and twenty-two with VPN protocols having different algorithms on multiple operating systems. Depending on the actual implementation, various network performance metrics (throughput, jitter, delay, CPU usage and drop rate) were measured to ascertain the network behaviour of TCP and UDP traffic types. In doing so, the goal was to evaluate traffic behaviour in the context of that particular network and to rank various techniques for network performance within that context.

In Chapters 4, 5, and 6, data from various test-beds have been presented as graphs and heat maps. The results attained in each context have been explained, and key findings have been discussed. Such findings have also been presented in various individual research papers published to date (*List of Publications*).

In this discussion, based on the observations so far, and the behaviour of network performance metrics, networking techniques within the contexts will be ranked. This ranking is established by evaluating the desirability of each performance metric, primarily by analysing all the graphs and heat maps. This will lead to identification of key contributions in this thesis.

## 7.2 Networks with Transition Mechanisms

There are many transition mechanisms that are currently being used, and of these, seven have been subjected to test-bed analysis in this thesis. In the doctoral work of Grayeli (2013), a selection of transition mechanisms have been ranked based on various performance metrics. That research undertaking is grounded purely in an OPNET simulation environment. In that work, six transition mechanisms have been ranked using an ordinal value system based on delay, jitter, and throughput. In this thesis, a similar ranking will be done (for networks in all three contexts), however, novelty in the work comes from the difference in primary data collection technique - in Grayeli (2013), simulation data was used, while in this thesis the primary data source is test-bed analysis. Earlier in Chapter 3, merits of different approaches have been discussed.

The four metrics measured on networks with transition mechanism are throughput, jitter, delay, and CPU usage. Based on the data gathered, these metrics have been ranked. This ranking shows that when it comes to differentiating transition mechanisms based on the criteria of network performance, some metrics are more significant than others. In the case of networks with transition mechanisms, delay is the best differentiator out of the four metrics. The rankings of performance metrics (lower number being more effective) that can be used to evaluate various transition mechanisms' network performance are presented in Table 7.1.

Transition Mechanisms	
Rank	Metrics
1	TCP and UDP Delay
2	TCP Throughput
3	UDP Throughput
4	TCP and UDP CPU Usage
5	UDP Jitter
6	TCP Jitter

Table 7.1: Transition mechanism performance metric's ranking

Table 7.1 shows that TCP/UDP delay and jitter are better differentiators of network performance on infrastructures with transition mechanisms than the other two metrics that were measured in this context. Therefore, based on this information, in Tables 7.2 and 7.3, the transition mechanisms are presented in an ordinal sequence. A lower numbered rank shows that a mechanism outperforms one that has a higher number.

TCP Delay		UDP Delay	
Rank	Transition Mechanisms	Rank	Transition Mechanisms
1	Dual Stack	1	Dual Stack
2	NAT64	2	NAT64
3	Teredo	3	Configured Tunnel
4	6over4	4	Teredo
5	Configured Tunnel	5	6over4
6	ISATAP	6	6to4
7	6to4	7	ISATAP

Table 7.2: Transition mechanism ranking as per TCP and UDP delay

TCP Throughput		UDP Throughput	
Rank	Transition Mechanisms	Rank	Transition Mechanisms
1	Dual Stack	1	Dual Stack
2	NAT64	2	NAT64
3	6to4	3	Configured Tunnel
4	Configured Tunnel	4	ISATAP
5	ISATAP	5	6over4
	6over4		6to4
	Teredo		Teredo

Table 7.3: Transition mechanism ranking as per TCP and UDP throughput

In all cases, it can be seen that dual stack and NAT64 are ranked the highest. The other five mechanisms' ordinal ranking changes depending on the actual performance metric, while TEREDO, in two of the tables, is at the bottom. The tables also emphasises that the ranking is different if the actual traffic type is taken into consideration. This shows that TCP and UDP traffic behave differently on networks with transition mechanisms.

### 7.3 Networks with Wireless Implementations

Networks that have wireless implementation can be based on different IEEE-802.11 standards. There is also a choice of various encryption techniques and for this particular reason, 19 different networks with various combinations of wireless standards and encryption techniques were subject to test-bed analysis in this thesis. In the doctoral work of Taank (2008), wired-to-wireless networks have been evaluated on test-beds, where the focus has been to systematically evaluate such networks end-to-end in order to establish rules that can assist with optimal deployment of TCP traffic type. In this thesis, the aim in evaluating wireless networks is similar, however, both TCP and UDP traffic types have been taken into consideration and various wireless combinations have been ranked based on different network performance metrics.

On networks with wireless networks, there were four performance metrics measured, throughput, delay, jitter, and drop rate. This was done for both TCP and UDP, however, drop rate for only UDP has been reported since all values for TCP were zero. Based on the observations and the behaviour of the metrics on various networks, in Table 7.4, the performance metrics are presented, ranked in order of significance when it comes to differentiating network behaviour on wireless networks. As in the case of transition mechanisms, it can be seen that delay is the best differentiator. Drop rate, which only registers significant values on wireless networks generally (and not wired implementations), is not a good differentiator between wireless combinations. This is also the case for wireless jitter.

Wireless	
Rank	Metrics
1	TCP and UDP Delay
2	TCP and UDP Throughput
3	TCP and UDP Jitter
4	UDP Drop Rate

Table 7.4: Wireless performance metrics ranking

Table 7.4 shows that delay and throughput are the best performance metrics when in come to using them to differentiate between various wireless and security protocols. Undoubtedly, with the enhancement of technology, newer wireless standards generally outperform their predecessors in terms of data transfer rates. So obviously throughput will be a great differentiator. Based on this and UDP/TCP delay, wireless combinations are ranked in Tables 7.5 and 7.6.

TCP Delay			UDP Delay		
Rank	Wireless Standard	Authentication Rank	Rank	Wireless Standard	Authentication Rank
1	IEEE 802.11g	WPA	1	IEEE 802.11g	WPA
		WEP			WEP
		WPA2			WPA2
2	IEEE 802.11n	WEP	2	IEEE 802.11n	WEP
		WPA2			WPA2
		WPA			WPA
3	IEEE 802.11ac	WPA2	3	IEEE 802.11ac	WPA2
		WPA			WPA

Table 7.5: Wireless standard rankings as per TCP and UDP delay

TCP Throughput			UDP Throughput		
Rank	Wireless Standard	Authentication Rank	Rank	Wireless Standard	Authentication Rank
1	IEEE 802.11ac	WPA2	1	IEEE 802.11ac	WPA2
		WPA			WPA
		WPA2			WPA2
2	IEEE 802.11n	WEP	2	IEEE 802.11n	WEP
		WPA			WPA
		WEP			WPA
3	IEEE 802.11g	WPA	3	IEEE 802.11g	WPA2
		WPA2			WEP

Table 7.6: Wireless standard rankings as per TCP and UDP throughput

Here, it is seen that TCP and UDP delay give similar rankings, while the list is slightly different in the case of throughput. In case of delay, IEEE802.11g is superior, however, IEEE802.11ac obviously has the greatest throughput. WPA2, the latest wireless security protocol's network performance is better than its predecessors.

## 7.4 Networks with VPNs

There are many VPN protocols, and of these IPsec and SSTP are the latest ones. Both are being widely used on current network infrastructures to secure data in motion. Also, each has a number of algorithms that can be chosen during configuration for authentication and encryption. When evaluating various VPNs for network performance, base operating systems on VPN servers were also factored in, and for these reasons a total of 22 various combinations were implemented on the test-bed for performance evaluation.

In regards to VPNs, the performance metrics measured were throughput, delay, and jitter. Their significance in relation to ranking networks that have VPN is presented in Table 7.7. Here, as in the case of the other two network contexts, delay is the best performance metric for differentiating various VPN techniques.

VPN	
Rank	Metrics
1	TCP and UDP Delay
2	TCP and UDP Throughput
3	TCP and UDP Jitter

Table 7.7: VPN performance metrics ranking

The test-bed analysis in this thesis did not distinctly differentiate between all encryption and authentication algorithms within each VPN protocol. Whilst the initial goal was to do so, results attained show that there were hardly any differences between most of the actual algorithms. Where distinctions are identifiable, those results have been presented. The actual difference between the VPN techniques, and the effects of the actual operating systems have been identified clearly.

In Tables 7.8 and 7.9, the VPN protocols have been ranked using delay and throughput as the performance metrics. In the tables, the operating systems have been listed and then within each set, VPN protocols have been ranked.

TCP Delay			UDP Delay		
Rank	Operating System	Encryption Rank	Rank	Operating System	Encryption Rank
1	Server 2012	SSTP	1	Server 2012	SSTP
		IPsec DES			IPsec AES
		IPsec AES			IPsec
		IPsec			IPsec
2	Server 2008	SSTP	2	Server 2008	SSTP
		IPsec			IPsec
3	Ubuntu	SSTP	3	Ubuntu	IPsec
		IPsec			SSTP

Table 7.8: VPN protocol rankings as per TCP and UDP delay

TCP Throughput			UDP Throughput		
Rank	Operating System	Encryption Rank	Rank	Operating System	Encryption Rank
1	Server 2012	SSTP	1	Server 2012	SSTP
		IPsec DES			IPsec DES
		IPsec AES			IPsec
		IPsec			IPsec AES
2	Ubuntu	SSTP	2	Server 2008	SSTP
		IPsec			IPsec
3	Server2008	SSTP	3	Ubuntu	SSTP
		IPsec			IPsec

Table 7.9: VPN protocol rankings as per TCP and UDP throughput

Overall, it can be seen that SSTP outperforms IPsec on all networks that have the latest version of the Microsoft Server implemented. On the other two operating systems, the top ranked VPN technology varies. It is also observed that TCP and the UDP ranking lists are different for both performance metrics, which indicates that the two traffic types perform differently on VPN networks.

## 7.5 Thesis Contributions

In this thesis, the network behaviour of TCP and UDP traffic has been evaluated in three different contexts. In each, various networks have been implemented to evaluate protocol behaviour. Resulting from this comprehensive research, the key contributions of this thesis are:

1. **Key Performance Metric Identification:** There are various performance metrics that can be measured on networks to gauge behaviour of traffic types on infrastructures. Whilst accurately measuring all of them may be important for different purposes, they are not all necessary when it comes to evaluating different techniques in a network context. In this thesis, it has been shown that in the contexts of networks with wireless, or networks with transition mechanisms, or with VPN technologies, TCP/UDP delay followed by throughput are the only key differentiators between different techniques.
2. **Ranking Transition Mechanisms:** A number of transition mechanisms have been subject to evaluation in this thesis. It has been seen that their ordinal ranking, based on TCP/UDP delay and throughput, are different. Irrespective of the performance metric being used to rank, dual stack and NAT64 are always the best performers. Detailed ranking based on delay and throughput are presented in Tables 7.2 and 7.3 respectively.
3. **Ranking Wireless Standards with Security Protocols:** IEEE802.11 technology is associated with WLANs and its usage is immense. The three wireless standards tested with different combinations of encryption protocols show that the new standard most certainly outperforms that of others in data rate transfers. However, it is also observed that the actual throughput is significantly lower than the theoretical bandwidth (a 10-fold drop is evident). Depending on the actual performance metric, ordinal ranking is different. Detailed ranking based on delay and throughput are presented in Tables 7.5 and 7.6 respectively.



4. **Ranking VPN Technologies:** Ranking VPN technologies shows that in most cases, the actual security and encryption algorithms implemented do not affect network performance. However, the choice of the actual VPN techniques does impact on the network performance. It is evident that SSTP has superior network performance in most cases, when compared with IPsec. The base operating systems on the VPN server does impact network performance. Detailed ranking based on delay and throughput are presented in Tables 7.8 and 7.9 respectively.

5. **Significant Observations on Networks with Transition Mechanisms:** When networks are implemented with transition mechanisms, network performance will downgrade, but to different degrees. This performance downgrading depends on the choice of the actual mechanism and in the case of ISATAP and Teredo, throughput can be compromised by almost 30%. The case is similar for delay, whereby a 10-12 fold increase in delay may be seen with some of the mechanisms. Jitter values attained show that irrespective of the transition mechanism being implemented, jitter will increase, but in most cases, this increase will be about the same. Detailed findings related to performance on networks with transition mechanisms are presented in Section 4.7.

6. **Significant Observations on Networks with Wireless:** On wireless networks, the difference in throughput between TCP and UDP is almost 35% in favour of UDP. Further, it is seen that wireless encryptions reduce network performance further, and that the maximum drop reported in this thesis is almost 10%. Also, it is observed that the newer of the wireless standards and encryption protocol gives the best network performance. Detailed findings related to performance on networks with transition mechanisms are presented in Section 5.7.

7. **Significant Observations on Networks with VPN Technologies:** VPN technologies dramatically impact on network performance. It can be seen that when comparing networks with and without VPNs, there can be an almost five-fold drop in the data transfer rate. Also, there is a significant difference between SSTP and IPsec network performance, however, effects of the actual authentication and encryption algorithms in each VPN

technology is indistinguishable. Detailed findings related to performance on networks with transition mechanisms are presented in Section 6.5.

**8. IPv4 and IPv6:** The difference between the network performance of IPv4 and IPv6 can be highly significant, depending on the network technologies implemented. While in some situations, the difference is insignificant, on networks with wireless a performance difference of almost 40% was observed, in favour of IPv4. In all the test-bed evaluations, IPv4 has always outperformed IPv6, but to different degrees, depending on the network details.

**9. Impact of Operating System Choice on Networks:** Operating systems do impact on network performance, and the degree to which they have an impact depends on the chosen operating system and the network configuration. In the case of VPN technologies, it can be seen that there is no one clear superior operating system. In all the VPN scenarios, all operating systems gave different performance metrics.

**10. TCP and UDP** TCP and UDP traffic types, in some situations register similar performance metrics, while in others, the variation between them can be significant. In extreme cases, the difference between the two is almost 35%. Also, it is observed that on wired networks the TCP packet drop rate is minuscule, while that of UDP is at a maximum 70%.

## Chapter 8

---

## Conclusions

This thesis has provided many interesting insights into the network behaviour of TCP and UDP traffic types on various networks. The contributions of the thesis are important, and they are all geared towards understanding network performance characteristics of infrastructures. The technologies at the centre of the three network scenarios around which test-bed evaluations have been conducted, are topical environments that currently exist on today's network implementations. Therefore, it is anticipated that contributions of the thesis will be useful to both academia and network practitioners.

In Chapter 4, the focus was on networks with various transition mechanism techniques. Many such mechanisms were implemented, out of which some very useful and interesting behaviours are evident. Dual stack and NAT64 outperform the other mechanisms, however, all mechanisms impacted negatively on network performance. The degree to which they downgrade performance metrics depends on the choice of the actual mechanism, and it has been observed that two mechanisms can negatively impact network performance by almost 30%. Further, there are varying degree of impact on network delay and jitter, however all transition mechanisms do have a significant impact.

In Chapter 5, evaluation of wireless networks has shown that the choice of the wireless standard, and combining it with the appropriate security mechanism, have an impact on both TCP and UDP traffic types. There is strong evidence that TCP and UDP traffic types behave differently on wireless networks, which then leads to significantly different values for network perfor-

mance metrics. This chapter has also successfully highlighted that there is almost a ten-fold throughput drop from the theoretical bandwidth values on wireless networks.

Chapter 6 focused on, VPN implementations on test-beds and the difference between the two commonly used VPN techniques is evident. It has also been observed that when comparing networks with and without VPN, there can be an almost five-fold performance drop, depending on the combination of VPN technology and the algorithms. The differentiation between the various algorithms within each VPN technique has been negligible.

In Chapter 7, the results are discussed, first in the context of each of the scenarios in which network performance have been analysed, and then the discussion leads onto identification of the key thesis contributions. Here, ten thesis contributions have been mentioned, highlighting ranking of various techniques and significant observations.

In addition to exploiting TCP and UDP in three network environments, network performance metrics that can be used as good differentiators within a network context have also been highlighted. It is observed that delay and throughput are the best metrics for performance differentiators in the three scenarios. Also, based on these two metrics, various networking techniques have been ranked. The novelty of this work lies in the fact that the rankings have been based on data collected primarily from test-bed experiments.

---

## References

- Aazam, M., Khan, I., Alam, M., & Qayyum, A. (2010, June). Comparison of IPv6 tunneled traffic of teredo and ISATAP over test-bed setup. In *Proceedings of the International Conference on Information and Emerging Technologies (ICIET)*. (p. 1-4). doi: 10.1109/ICIET.2010.5625689
- Aazam, M., Syed, A., Shah, S., Khan, I., & Alam, M. (2011, March). Evaluation of 6to4 and ISATAP on a test LAN. In *Proceedings of the IEEE Symposium on Computers Informatics (ISCI)*. (p. 46-50). doi: 10.1109/ISCI.2011.5958881
- Abed, G., Ismail, M., & Jumari, K. (2011, Sept). A comparison and analysis of congestion window for HS-TCP, full-TCP, and TCP-Linux in long term evolution system model. In *Proceedings of the IEEE Conference on Open Systems (ICOS)*. (p. 358-362). doi: 10.1109/ICOS.2011.6079287
- Adeyinka, O. (2008, May). Analysis of problems associated with IPSec VPN technology. In *Proceedings of the Canadian Conference on Electrical and Computer Engineering (CCECE)*. (p. 1903-1908). doi: 10.1109/CCECE.2008.4564875
- Ahmad, K. (2001). *Sourcebook of ATM and IP Internetworking* (1st ed.). New York, NY, USA: John Wiley & Sons, Inc.
- Ahmed, A., Zaidi, S., & Ahmed, N. (2004, June). Performance evaluation of transmission control protocol in mobile ad hoc networks. In *Proceedings of the International Networking and Communication Conference (INCC)*. (p. 13-18). doi: 10.1109/INCC.2004.1366569
- Ahvar, E., & Fathy, M. (2007). Performance evaluation of routing protocols for high density ad hoc networks based on energy consumption by GlomoSim simulator. *World Academy of Science, Engineering and Technology*, 29.
- Akhtar, N., & Siddiqui, F. (2011, June). UDP packet monitoring with stanford data stream manager. In *Proceedings of the International Conference on Recent Trends in Information Technology (ICRTIT)*. (p. 533-537). doi: 10.1109/ICRTIT.2011.5972403
- Alcock, S., & Nelson, R. (2011, May). Passive detection of TCP congestion events. In *Proceedings of the 18th International Conference on Telecommunications (ICT)*. (p. 499-504). doi: 10.1109/CTS.2011.5898976

- Alexander, S., & Droms, R. (1997, March). *DHCP Options and BOOTP Vendor Extensions* (No. 2132). RFC 2132 (Draft Standard). IETF. Retrieved from <http://www.ietf.org/rfc/rfc2132.txt> (Updated by RFCs 3442, 3942, 4361, 4833, 5494)
- Ali, Q., Abdulmaowjod, A., & Mohammed, H. (2010, Nov). Simulation and performance study of wireless sensor network (WSN) using MATLAB. In *Proceedings of the 1st International Conference on Energy, Power and Control (EPC-IQ)*. (p. 307-314).
- Allman, M., Paxson, V., & Blanton, E. (2009, September). *TCP Congestion Control* (No. 5681). RFC 5681 (Draft Standard). IETF. Retrieved from <http://www.ietf.org/rfc/rfc5681.txt>
- Almquist, P. (1992, July). *Type of Service in the Internet Protocol Suite* (No. 1349). RFC 1349 (Proposed Standard). IETF. Retrieved from <http://www.ietf.org/rfc/rfc1349.txt> (Obsoleted by RFC 2474)
- Alrshah, M., & Othman, M. (2013, Nov). Performance evaluation of parallel TCP, and its impact on bandwidth utilization and fairness in high-BDP networks based on test-bed. In *Proceedings of the IEEE Malaysia International Conference on Communications (MICC)*. (p. 23-28). doi: 10.1109/MICC.2013.6805793
- Alsadeh, A., Rafiee, H., & Meinel, C. (2012, Oct). Cryptographically Generated Addresses (CGAs): Possible attacks and proposed mitigation approaches. In *Proceedings of the 12th IEEE International Conference on Computer and Information Technology (CIT)*. (p. 332-339). doi: 10.1109/CIT.2012.84
- Al-tamimi, B., Taib, A., & Budiarto, R. (2008, Oct). Protecting teredo clients from source routing exploits. In *Proceedings of the 1st International Conference on Distributed Framework and Applications (DFMA)*. (p. 126-133). doi: 10.1109/ICDFMA.2008.4784425
- Altman, E., & Jimnez, T. (2012). NS simulator for beginners. *Synthesis Lectures on Communication Networks*, 5(1), 1-184. doi: 10.2200/S00397ED1V01Y201112CNT010
- An, H., Luo, W., Li, X., Zhang, X., & Yan, B. (2009, Jan). A new IPv6 tunneling protocol: Escort. In *Proceedings of the International Symposium on Computer Network and Multimedia Technology (CNMT)*. (p. 1-6). doi: 10.1109/CNMT.2009.5374781
- AnalogX. (2013, December). *Internet traffic report*. Retrieved from <http://www.internettrafficreport.com/>
- Andersson, L., & Rosen, E. (2006, September). *Framework for Layer 2 Virtual Private Networks (L2VPNs)* (No. 4664). RFC 4664 (Informational). IETF. Retrieved from <http://www.ietf.org/rfc/rfc4664.txt>
- Aoun, C., & Davies, E. (2007, July). *Reasons to Move the Network Address Translator - Protocol Translator (NAT-PT) to Historic Status* (No. 4966). RFC 4966 (Informational). IETF. Retrieved from <http://www.ietf.org/rfc/rfc4966.txt>
- Apostolopoulos, G., Peris, V., & Saha, D. (1999, Mar). Transport layer security: How much does it really cost? In *Proceedings of the 18th IEEE Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*. (Vol. 2, p. 717-725). doi: 10.1109/INFOCOM.1999.751458
- Arkko, J., & Baker, F. (2011, May). *Guidelines for Using IPv6 Transition Mechanisms dur-*

- ing IPv6 Deployment* (No. 6180). RFC 6180 (Informational). IETF. Retrieved from <http://www.ietf.org/rfc/rfc6180.txt>
- Arkko, J., Kempf, J., Zill, B., & Nikander, P. (2005, March). *SEcure Neighbor Discovery (SEND)* (No. 3971). RFC 3971 (Proposed Standard). IETF. Retrieved from <http://www.ietf.org/rfc/rfc3971.txt> (Updated by RFCs 6494, 6495, 6980)
- Armitage, G., Schuster, P., Jork, M., & Harter, G. (1999, January). *IPv6 over Non-Broadcast Multiple Access (NBMA) networks* (No. 2491). RFC 2491 (Proposed Standard). IETF. Retrieved from <http://www.ietf.org/rfc/rfc2491.txt>
- Arpaci, M., & Copeland, J. (2000). An adaptive queue management method for congestion avoidance in TCP/IP networks. In *Proceedings of the IEEE Conference on Global Telecommunications (GLOBECOM)*. (Vol. 1, p. 309-315). doi: 10.1109/GLOCOM.2000.892022
- Aslam, M., Guinard, A., McGibney, A., Rea, S., & Pesch, D. (2011, May). Wi-design, Wi-manage, why bother? In *Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management (IM)*. (p. 730-744). doi: 10.1109/INM.2011.5990597
- Atkinson, R. (1995a, August). *IP Authentication Header* (No. 1826). RFC 1826 (Proposed Standard). IETF. Retrieved from <http://www.ietf.org/rfc/rfc1826.txt> (Obsoleted by RFC 2402)
- Atkinson, R. (1995b, August). *IP Encapsulating Security Payload (ESP)* (No. 1827). RFC 1827 (Proposed Standard). IETF. Retrieved from <http://www.ietf.org/rfc/rfc1827.txt> (Obsoleted by RFC 2406)
- Audet, F., & Jennings, C. (2007, January). *Network Address Translation (NAT) Behavioral Requirements for Unicast UDP* (No. 4787). RFC 4787 (Best Current Practice). IETF. Retrieved from <http://www.ietf.org/rfc/rfc4787.txt> (Updated by RFC 6888)
- Aura, T. (2005, March). *Cryptographically Generated Addresses (CGA)* (No. 3972). RFC 3972 (Proposed Standard). IETF. Retrieved from <http://www.ietf.org/rfc/rfc3972.txt> (Updated by RFCs 4581, 4982)
- Avallone, S., Guadagno, S., Emma, D., Pescape, A., & Ventre, G. (2004, Sept). D-ITG distributed internet traffic generator. In *Proceedings of the 1st IEEE International Conference on Quantitative Evaluation of Systems, (QEST)*. (p. 316-317). doi: 10.1109/QEST.2004.1348045
- Badra, M., & Hajjeh, I. (2006, June). Enabling VPN and secure remote access using TLS protocol. In *Proceedings of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. (p. 308-314). doi: 10.1109/WIMOB.2006.1696366
- Bagnulo, M., Garcia-Martinez, A., & Van Beijnum, I. (2012, July). The NAT64/DNS64 tool suite for IPv6 transition. *IEEE Communications Magazine*, 50(7), 177-183. doi: 10.1109/MCOM.2012.6231295
- Bagnulo, M., Matthews, P., & van Beijnum, I. (2011, April). *Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers* (No. 6146). RFC 6146 (Proposed Standard). IETF. Retrieved from <http://www.ietf.org/rfc/rfc6146.txt>
- Bahaman, N., Hamid, E., & Prabuwno, A. (2012, May). Network performance evaluation of 6to4 tunneling. In *Proceedings of the International Conference on Innovation Management*

- and Technology Research (ICIMTR)*. (p. 263-268). doi: 10.1109/ICIMTR.2012.6236400
- Balakrishnan, H., Padmanabhan, V., Seshan, S., & Katz, R. (1997, Dec). A Comparison of Mechanisms for Improving TCP Performance Over Wireless Links. *Journal of IEEE/ACM Transactions on Networking*, 5(6), 756-769. doi: 10.1109/90.650137
- Barr, R., Haas, Z., & Van Renesse, R. (2004). Jist: Embedding simulation time into a virtual machine. In *Eurosim Congress on Modelling and Simulation*.
- Barrenetxea, G., Ingelrest, F., Schaefer, G., Vetterli, M., Couach, O., & Parlange, M. (2008, April). Sensorscope: Out-of-the-box environmental monitoring. In *Proceedings of the IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. (p. 332-343). doi: 10.1109/IPSN.2008.28
- Benko, P., Malicsko, G., & Veres, A. (2004, March). A large-scale, passive analysis of end-to-end TCP performance over GPRS. In *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*. (Vol. 3, p. 1882-189). doi: 10.1109/INFCOM.2004.1354598
- Berger, T. (2006, April). Analysis of current VPN technologies. In *Proceedings of the 1st International Conference on Availability, Reliability and Security (ARES)*. (p. 8). doi: 10.1109/ARES.2006.30
- Bhanumathi, V., & Dhanasekaran, R. (2010, Feb). TCP variants - a comparative analysis for high bandwidth - delay product in mobile adhoc network. In *Proceedings of the 2nd International Conference on Computer and Automation Engineering (ICCAE)*. (Vol. 2, p. 600-604). doi: 10.1109/ICCAE.2010.5451683
- Bohn, S., Grob, S., Nubgen, R., & Schwann, P. (2006, Jan). An automated system interoperability test bed for WPA and WPA2. In *Proceedings of the IEEE Symposium on Radio and Wireless*. (p. 615-618). doi: 10.1109/RWS.2006.1615232
- Borman, D. (2012, July). *TCP Options and Maximum Segment Size (MSS)* (No. 6691). RFC 6691 (Informational). IETF. Retrieved from <http://www.ietf.org/rfc/rfc6691.txt>
- Borsc, M., & Shinde, H. (2005, Jan). Wireless security privacy. In *Proceedings of the IEEE International Conference on Personal Wireless Communications (ICPWC)*. (p. 424-428). doi: 10.1109/ICPWC.2005.1431380
- Botta, A., Dainott, A., & Pescapè, A. (2012). A tool for the generation of realistic network workload for emerging networking scenarios. *Computer Networks*, 56(15), 3531-3547.
- Boucadair, M., Grimault, J.-L., Levis, P., Villefranque, A., & Morand, P. (2009, Aug). Anticipate ipv4 address exhaustion: A critical challenge for Internet survival. In *Proceedings of the 1st International Conference on Evolving Internet (INTERNET)*. (p. 27-32). doi: 10.1109/INTERNET.2009.11
- Bousquet, J., Messier, G., & Magierowski, S. (2007, April). IEEE 802.11b SDMA performance in realistic environments. In *Proceedings of the 65th IEEE Vehicular Technology Conference (VTC)*. (p. 539-543). doi: 10.1109/VETECS.2007.122
- Braden, R. (1989, October). *Requirements for Internet Hosts - Communication Layers* (No. 1122). RFC 1122 (INTERNET STANDARD). IETF. Retrieved from <http://www.ietf.org/rfc/rfc1122.txt> (Updated by RFCs 1349, 4379, 5884,



- 6093, 6298, 6633, 6864)
- Bradner, S. (1991, July). *Benchmarking Terminology for Network Interconnection Devices* (No. 1242). RFC 1242 (Informational). IETF. Retrieved from <http://www.ietf.org/rfc/rfc1242.txt> (Updated by RFC 6201)
- Bradner, S., & Mankin, A. (1995, January). *The Recommendation for the IP Next Generation Protocol* (No. 1752). RFC 1752 (Proposed Standard). IETF. Retrieved from <http://www.ietf.org/rfc/rfc1752.txt>
- Bradner, S., & McQuaid, J. (1999, March). *Benchmarking Methodology for Network Interconnect Devices* (No. 2544). RFC 2544 (Informational). IETF. Retrieved from <http://www.ietf.org/rfc/rfc2544.txt> (Updated by RFCs 6201, 6815)
- Breslau, L., Estrin, D., Fall, K., Floyd, S., Heidemann, J., Helmy, A., ... Yu, H. (2000, May). Advances in network simulation. *Computer*, 33(5), 59-67. doi: 10.1109/2.841785
- Brownlee, N., & Claffy, K. (2002, Oct). Understanding internet traffic streams: Dragonflies and tortoises. *Communications Magazine, IEEE*, 40(10), 110-117. doi: 10.1109/MCOM.2002.1039865
- Bui, V., & Zhu, W. (2007, Oct). Improving multipath live streaming performance with markov decision processes. In *Proceedings of the IEEE International Symposium on Communications and Information Technologies (ISCIT)*. (p. 580-585). doi: 10.1109/ISCIT.2007.4392085
- Bush, R., & Meyer, D. (2002, December). *Some Internet Architectural Guidelines and Philosophy* (No. 3439). RFC 3439 (Informational). IETF. Retrieved from <http://www.ietf.org/rfc/rfc3439.txt>
- Caceres, R., & Iftode, L. (1995, Jun). Improving the performance of reliable transport protocols in mobile computing environments. *IEEE Journal on Selected Areas in Communications*, 13(5), 850-857. doi: 10.1109/49.391749
- Cai, J., Zhang, Z., & Song, X. (2010, Nov). An analysis of UDP traffic classification. In *Proceedings of the 12th IEEE International Conference on Communication Technology (ICCT)*. (p. 116-119). doi: 10.1109/ICCT.2010.5689203
- Callon, R., & Haskin, D. (1997, September). *Routing Aspects of IPv6 Transition* (No. 2185). RFC 2185 (Informational). IETF. Retrieved from <http://www.ietf.org/rfc/rfc2185.txt>
- Cardwell, N., Savage, S., & Anderson, T. (2000, Mar). Modeling TCP latency. In *Proceedings of the 19th IEEE Annual Joint Conference of Computer and Communications Societies (INFOCOM)*. (Vol. 3, p. 1742-1751). doi: 10.1109/INFCOM.2000.832574
- Carpenter, B. (2011, August). *Advisory Guidelines for 6to4 Deployment* (No. 6343). RFC 6343 (Informational). IETF. Retrieved from <http://www.ietf.org/rfc/rfc6343.txt>
- Carpenter, B., & Moore, K. (2001, February). *Connection of IPv6 Domains via IPv4 Clouds* (No. 3056). RFC 3056 (Proposed Standard). IETF. Retrieved from <http://www.ietf.org/rfc/rfc3056.txt>
- Celebi, B., Dericiogullari, B., & Bitirim, Y. (2007, March). Performance evaluation of IEEE 802.11b, IEEE 802.11g and GPRS/EDGE based on query retrieval time. In *Proceedings of the 3rd International Conference on Wireless and Mobile Communications (ICWMC)*. (p. 66-

- 66). doi: 10.1109/ICWMC.2007.66
- Cerf, V. (2004, Sept). On the evolution of internet technologies. *Proceedings of the IEEE*, 92(9), 1360-1370. doi: 10.1109/JPROC.2004.832974
- Cerf, V., & Kahn, R. (1974, May). A protocol for packet network intercommunication. *Journal of IEEE Transactions on Communications*, 22(5), 637-648. doi: 10.1109/TCOM.1974.1092259
- Chakraborty, K., Dutta, N., & Biradar, S. (2009, Dec). Simulation of IPv4-to-IPv6 dual stack transition mechanism (DSTM) between IPv4 hosts in integrated IPv6/IPv4 network. In *Proceedings of the 4th IEEE International Conference on Computers and Devices for Communication (CODEC)*. (p. 1-4).
- Chan, E., & Baciu, G. (2012). Introduction to wireless localization. , 1(1), 1-10.
- Chandra, S., & Nair, A. (2007, Oct). VPN for remote digital evidence acquisition. In *Proceedings of the IEEE Region 10 Conference (TENCON)*. (p. 1-4). doi: 10.1109/TENCON.2007.4429020
- Chang, J.-M., Chao, H.-C., Chen, J.-L., & Lai, C.-F. (2012, Dec). An efficient service discovery system for dual-stack cloud file service. *Journal of IEEE Systems*, 6(4), 584-592. doi: 10.1109/JSYST.2011.2177131
- Chang, X. (1999). Network simulations with OPNET. In *Proceedings of the Winter Simulation Conference*. (Vol. 1, p. 307-314). doi: 10.1109/WSC.1999.823089
- Charalambous, C., Frost, V., & Evans, J. (1999, Jul). Performance evaluation of TCP extensions on ATM over high bandwidth delay product networks. *IEEE Communications Magazine*, 37(7), 57-63. doi: 10.1109/35.774881
- Chen, J., Jia, J., & Li, X. (2011, May). A new design of embedded IPv4/IPv6 dual-stack protocol. In *Proceedings of the International Conference on Network Computing and Information Security (NCIS)*. (Vol. 2, p. 163-167). doi: 10.1109/NCIS.2011.131
- Chen, J.-L., Chang, Y.-C., & Lin, C.-H. (2004a, Feb). Performance investigation of IPv4/IPv6 transition mechanisms. In *Proceeding of the 6th IEEE International Conference on Advanced Communication Technology*. (Vol. 2, p. 545-550). doi: 10.1109/ICACT.2004.1292930
- Chen, J.-L., Chang, Y.-C., & Lin, C.-H. (2004b, Feb). Performance investigation of IPv4/IPv6 transition mechanisms. In *Proceedings of the 6th International Conference on Advanced Communication Technology*. (Vol. 2, p. 545-550). doi: 10.1109/ICACT.2004.1292930
- Chen, X., Zhai, H., Wang, J., & Fang, Y. (2004, Jan). TCP performance over mobile ad hoc networks. *Canadian Journal of Electrical and Computer Engineering*, 29(1/2), 129-134. doi: 10.1109/CJECE.2004.1425806
- Chowdhury, I., Lahiry, J., & Hasan, S. (2009, Dec). Performance analysis of datagram congestion control protocol (DCCP). In *Proceedings of the 12th IEEE International Conference on Computers and Information Technology (ICCIT)*. (p. 454-459). doi: 10.1109/ICCIT.2009.5407282
- Chuangchunsong, N., Kamolphiwong, S., Kamolphiwong, T., Elz, R., & Pongpailool, P. (2014, Feb). Performance evaluation of IPv4/IPv6 transition mechanisms: IPv4-in-IPv6 tunneling techniques. In *Proceedings of the IEEE International Conference on Information Networking (ICOIN)*. (p. 238-243). doi: 10.1109/ICOIN.2014.6799698

- Chuanhuang, Z., & Haonan, H. (2012, Aug). Analysis of the NAT-PT gateway. In *Proceedings of the International Conference on Computer Science Service System (CSSS)*. (p. 46-49). doi: 10.1109/CSSS.2012.20
- Cioponea, C., Bucicoiu, M., & Rosner, D. (2013, Jan). Analysis of VoIP encryption performance using dedicated hardware. In *Proceedings of the 11th IEEE Roedunet International Conference (RoEduNet)*. (p. 1-4). doi: 10.1109/RoEduNet.2013.6511751
- Cong, L., & Miki, T. (2000). New acknowledgement mechanism for TCP/IP over ATM. In *Proceedings of the International Conference on Communication Technology (ICCT)*. (Vol. 1, p. 436-443). doi: 10.1109/ICCT.2000.889244
- Constantine, B., Forget, G., Geib, R., & Schrage, R. (2011, August). *Framework for TCP Throughput Testing* (No. 6349). RFC 6349 (Informational). IETF. Retrieved from <http://www.ietf.org/rfc/rfc6349.txt>
- Courtney, M. (2012, August). 'super IP' to the rescue! *Journal of Engineering Technology*, 7(7), 76-78.
- Crepaldi, R., Friso, S., Harris, A., Mastrogiovanni, M., Petrioli, C., Rossi, M., ... Zorzi, M. (2007, May). The design, deployment, and analysis of signetlab: A sensor network testbed and interactive management tool. In *Proceedings of the 3rd International Conference on Testbeds and Research Infrastructure for the Development of Networks and Communities (TridentCom)*. (p. 1-10). doi: 10.1109/TRIDENTCOM.2007.4444656
- Day, J., & Zimmermann, H. (1983, Dec). The OSI reference model. *Proceedings of the IEEE*, 71(12), 1334-1340. doi: 10.1109/PROC.1983.12775
- de Carvalho, J., Veiga, H., Marques, N., Pacheco, C., & Reis, A. (2010, Sept). A contribution to performance measurements of IEEE 802.11 a, b, g laboratory WEP point-to-point links using TCP, UDP and FTP. In *Proceedings of the International Conference on Applied Electronics (AE)*. (p. 1-5).
- Deering, S., & Hinden, R. (1998, December). *Internet Protocol, Version 6 (IPv6) Specification* (No. 2460). RFC 2460 (Draft Standard). IETF. Retrieved from <http://www.ietf.org/rfc/rfc2460.txt> (Updated by RFCs 5095, 5722, 5871, 6437, 6564, 6935, 6946, 7045, 7112)
- Demichelis, C., & Chimento, P. (2002, November). *IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)* (No. 3393). RFC 3393 (Proposed Standard). IETF. Retrieved from <http://www.ietf.org/rfc/rfc3393.txt>
- Demir, O., & Aktas, T. (2013, June). Evaluation of two models for securing SIP for home network communications. In *Proceedings of the 16th IEEE International Symposium on Wireless Personal Multimedia Communications (WPMC)*. (p. 1-4).
- De Rubertis, A., Mainetti, L., Mighali, V., Patrono, L., Sergi, I., Stefanizzi, M., & Pascali, S. (2013, Sept). Performance evaluation of end-to-end security protocols in an internet of things. In *Proceedings of the 21st IEEE International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. (p. 1-6). doi: 10.1109/SoftCOM.2013.6671893
- Dimitrakopoulos, G., Demestichas, P., & Koenig, W. (2010, June). Introduction of cognitive systems in the wireless world - research achievements and future challenges for end-

- to-end efficiency. In *Proceedings of the Future Network and Mobile Summit*. (p. 1-9).
- Ding, Y., Savolainen, T., Korhonen, J., & Kojo, M. (2012, June). Speeding up IPv6 transition: Discovering NAT64 and learning prefix for IPv6 address synthesis. In *Proceedings of the IEEE International Conference on Communications (ICC)*. (p. 6862-6868). doi: 10.1109/ICC.2012.6364825
- Dommetry, G. (2000, September). *Key and Sequence Number Extensions to GRE* (No. 2890). RFC 2890 (Proposed Standard). IETF. Retrieved from <http://www.ietf.org/rfc/rfc2890.txt>
- Doufexi, A., Armour, S., Butler, M., Nix, A., Bull, D., McGeehan, J., & Karlsson, P. (2002, May). A comparison of the HIPERLAN/2 and IEEE 802.11a wireless LAN standards. *IEEE Communications Magazine*, 40(5), 172-180. doi: 10.1109/35.1000232
- Drilo, B., & Flatz, L. (2003, Oct). Comparison of IEEE 802.11g optional standard elements in WLAN hotspot scenario. In *Proceedings of the 17th International Conference on Applied Electromagnetics and Communications (ICECom)*. (p. 147-151). doi: 10.1109/ICE-COM.2003.1290976
- Droms, R. (1997, March). *Dynamic Host Configuration Protocol* (No. 2131). RFC 2131 (Draft Standard). IETF. Retrieved from <http://www.ietf.org/rfc/rfc2131.txt> (Updated by RFCs 3396, 4361, 5494, 6842)
- Droms, R. (1999, Jul). Automated configuration of TCP/IP with DHCP. *Journal of IEEE Internet Computing*, 3(4), 45-53. doi: 10.1109/4236.780960
- Duke, M., Braden, R., Eddy, W., & Blanton, E. (2006, September). *A Roadmap for Transmission Control Protocol (TCP) Specification Documents* (No. 4614). RFC 4614 (Informational). IETF. Retrieved from <http://www.ietf.org/rfc/rfc4614.txt> (Updated by RFC 6247)
- Duvvuru, R., & Singh, S. (2013, Sept). Minimizing transmission delay in IPv4 network to IPv6 network through ADSTM. In *Proceedings of the 15th IEEE International Conference on Advanced Computing Technologies (ICACT)*. (p. 1-5). doi: 10.1109/ICACT.2013.6710491
- Edwan, T., Guan, L., Oikonomou, G., & Phillips, I. (2010, June). Higher order delay functions for delay-loss based TCP congestion control. In *Proceedings of the 6th Conference on Wireless Advanced (WiAD)*. (p. 1-6). doi: 10.1109/WIAD.2010.5544874
- Elich, M., Velan, P., Jirsik, T., & Celeda, P. (2013, Oct). An investigation into TEREDO and 6to4 transition mechanisms: Traffic analysis. In *Proceedings of the 38th IEEE Conference on Local Computer Networks Workshops (LCN Workshops)*. (p. 1018-1024). doi: 10.1109/LCNW.2013.6758546
- Fabris Hoefel, R. (2014, April). IEEE 802.11ac: A performance evaluation with lattice-based MMSE and zero forcing MIMO OFDM receivers. In *Proceedings of the IEEE Wireless Telecommunications Symposium (WTS)*. (p. 1-7). doi: 10.1109/WTS.2014.6835024
- Farinacci, D., Fuller, V., Meyer, D., & Lewis, D. (2013, January). *The Locator/ID Separation Protocol (LISP)* (No. 6830). RFC 6830 (Experimental). IETF. Retrieved from <http://www.ietf.org/rfc/rfc6830.txt>
- Farinacci, D., Li, T., Hanks, S., Meyer, D., & Traina, P. (2000, March). *Generic Routing Encapsulation (GRE)* (No. 2784). RFC 2784 (Proposed Standard). IETF. Retrieved from

- <http://www.ietf.org/rfc/rfc2784.txt> (Updated by RFC 2890)
- Farrell, S. (2010, Jan). Why didn't we spot that? [practical security]. *Journal of IEEE Internet Computing*, 14(1), 84-87. doi: 10.1109/MIC.2010.21
- Fenner, W. (1997, November). *Internet Group Management Protocol, Version 2* (No. 2236). RFC 2236 (Proposed Standard). IETF. Retrieved from <http://www.ietf.org/rfc/rfc2236.txt> (Updated by RFC 3376)
- FitzGerald, J. (2011). *Business Data Communications and Networking* (11th ed.). Wiley Global Education.
- Floyd, S., Mahdavi, J., Mathis, M., & Podolsky, M. (2000, July). *An Extension to the Selective Acknowledgement (SACK) Option for TCP* (No. 2883). RFC 2883 (Proposed Standard). IETF. Retrieved from <http://www.ietf.org/rfc/rfc2883.txt>
- Forouzan, B. (2001). *Data Communications and Networking* (2nd ed.). New York, NY, USA: McGraw-Hill.
- Forouzan, B. (2003). *TCP/IP Protocol Suite* (2nd ed.). New York, NY, USA: McGraw-Hill.
- Fukushima, M., & Goto, S. (1999). Analysis of TCP flags in congested network. In *Journal of Internet Workshop (IWS)* (p. 151-156). doi: 10.1109/IWS.1999.811007
- Fuller, V., & Li, T. (2006, August). *Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan* (No. 4632). RFC 4632 (Best Current Practice). IETF. Retrieved from <http://www.ietf.org/rfc/rfc4632.txt>
- Garg, S., & Kappes, M. (2003, March). An experimental study of throughput for UDP and VoIP traffic in IEEE 802.11b networks. In *Proceedings of the IEEE Conference on Wireless Communications and Networking (WCNC)*. (Vol. 3, p. 1748-1753). doi: 10.1109/WCNC.2003.1200651
- Göktürk, E. (2005). Emulating ad hoc networks: Differences from simulations and emulation specific problems. *New Trends in Computer Networks*, 1, 329-338.
- Göktürk, E. (2007). A stance on emulation and testbeds, and a survey of network emulators and testbeds. *Proceedings of (ECMS)*.
- Goode, R. (1998, Oct). Next generation Internet Protocol-testbed experience. In *Proceedings of the IEEE Conference on Military Communications (MILCOM)*. (Vol. 1, p. 297-301). doi: 10.1109/MILCOM.1998.722592
- Google. (2014, October). *Ipv6 adoption*. Retrieved from <https://www.google.com/intl/en/ipv6/statistics.html>
- Gopinath, T., Kumar, A., & Sharma, R. (2013, April). Performance evaluation of TCP and UDP over wireless ad-hoc networks with varying traffic loads. In *Proceedings of the International Conference on Communication Systems and Network Technologies (CSNT)*. (p. 281-285). doi: 10.1109/CSNT.2013.66
- Goth, G. (2012, March). The end of IPv4 is nearly here - really. *Journal of IEEE Internet Computing*, 16(2), 7-11. doi: 10.1109/MIC.2012.37
- Grayeli, P. (2013). *Performance Modeling and Analysis of IPv6 Transition Mechanisms over MPLS* (Unpublished doctoral dissertation). The George Washington University.
- Guo, Z., Zhu, Z., Chen, R., & He, L. (2012, Sept). Analysis and research on transition proposal from IPv4 to IPv6 in metropolitan area network. In *Proceedings of the 8th IEEE*

- International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM)*. (p. 1-4). doi: 10.1109/WiCOM.2012.6478426
- Haddad, H., Berenjkoub, M., & Gazor, S. (2004, May). A proposed protocol for internet key exchange (IKE). In *Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering*. (Vol. 4, p. 2017-2020). doi: 10.1109/CCECE.2004.1347630
- Hadiya, D., Save, R., & Geetu, G. (2013, Dec). Network performance evaluation of 6to4 and configured tunnel transition mechanisms: An empirical test-bed analysis. In *Proceedings of the 6th International Conference on Emerging Trends in Engineering and Technology (ICETET)*. (p. 56-60). doi: 10.1109/ICETET.2013.14
- Halsall, F. (1996). *Data Communications, Computer Networks and Open Systems* (4th ed.). Addison-Wesley.
- Hamed, H., Al-Shaer, E., & Marrero, W. (2005, Nov). Modeling and verification of IPsec and VPN security policies. In *Proceedings of the 13th IEEE International Conference on Network Protocols (ICNP)*. (p. 10). doi: 10.1109/ICNP.2005.25
- Hamzeh, K., Pall, G., Verthein, W., Taarud, J., Little, W., & Zorn, G. (1999, July). *Point-to-Point Tunneling Protocol (PPTP)* (No. 2637). RFC 2637 (Informational). IETF. Retrieved from <http://www.ietf.org/rfc/rfc2637.txt>
- Hanks, S., Li, T., Farinacci, D., & Traina, P. (1994a, October). *Generic Routing Encapsulation (GRE)* (No. 1701). RFC 1701 (Informational). IETF. Retrieved from <http://www.ietf.org/rfc/rfc1701.txt>
- Hanks, S., Li, T., Farinacci, D., & Traina, P. (1994b, October). *Generic Routing Encapsulation over IPv4 networks* (No. 1702). RFC 1702 (Informational). IETF. Retrieved from <http://www.ietf.org/rfc/rfc1702.txt>
- Hansen, C. (2011, December). WiGiG: Multi-gigabit wireless communications in the 60 GHz band. *Journal of IEEE Wireless Communications*, 18(6), 6-7. doi: 10.1109/MWC.2011.6108325
- Hartman, S. (2006, December). *Desired Enhancements to Generic Security Services Application Program Interface (GSS-API) Version 3 Naming* (No. 4768). RFC 4768 (Informational). IETF. Retrieved from <http://www.ietf.org/rfc/rfc4768.txt>
- Hei, Y., & Yamazaki, K. (2004). Traffic analysis and worldwide operation of open 6to4 relays for IPv6 deployment. In *Proceedings of the International Symposium on Applications and the Internet*. (p. 265-268). doi: 10.1109/SAINT.2004.1266125
- Heidemann, J., Bulusu, N., Elson, J., Intanagonwiwat, C., Lan, K., Xu, Y., ... Govindan, R. (2001). Effects of detail in wireless network simulation. In *Proceedings of the SCS Multiconference on Distributed Simulation*. (p. 3-11).
- Heller, C., Heidinger, E., Schneelee, S., Fischer, W., & Klose, P. (2010, Oct). Power-over-ethernet for avionic networks. In *Proceedings of the 29th IEEE/AIAA Digital Avionics Systems Conference (DASC)*. (p. 1-11). doi: 10.1109/DASC.2010.5655327
- Henderson, T., Ahrenholz, J., & Kim, J. (2003, March). Experience with the host identity protocol for secure host mobility and multihoming. In *Proceedings of the IEEE Conference on Wireless Communications and Networking (WCNC)*. (Vol. 3, p. 2120-2125). doi: 10.1109/WCNC.2003.1200714

- Hengartner, U., Bolliger, J., & Gross, T. (2000, Mar). TCP vegas revisited. In *Proceedings of the IEEE 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*. (Vol. 3, p. 1546-1555). doi: 10.1109/INFOCOM.2000.832553
- Hengstler, S., & Aghajan, H. (2006, Oct). Application development in vision-enabled wireless sensor networks. In *Proceedings of the International Conference on Systems and Networks Communications (ICSNC)*. (p. 30-30). doi: 10.1109/ICSNC.2006.21
- Hinden, R., & Deering, S. (2003, April). *Internet Protocol Version 6 (IPv6) Addressing Architecture* (No. 3513). RFC 3513 (Proposed Standard). IETF. Retrieved from <http://www.ietf.org/rfc/rfc3513.txt> (Obsoleted by RFC 4291)
- Hinden, R., & Deering, S. (2006, February). *IP Version 6 Addressing Architecture* (No. 4291). RFC 4291 (Draft Standard). IETF. Retrieved from <http://www.ietf.org/rfc/rfc4291.txt> (Updated by RFCs 5952, 6052, 7136, 7346, 7371)
- Hirschler, B., & Treytl, A. (2012, March). Internet protocol security and power line communication. In *Proceedings of the 16th IEEE International Symposium on Power Line Communications and Its Applications (ISPLC)*. (p. 102-107). doi: 10.1109/ISPLC.2012.6201308
- Ho, M.-J., Wang, J., Shelby, K., & Haisch, H. (2003, Oct). IEEE 802.11g OFDM WLAN throughput performance. In *Proceedings of the 58th IEEE Conference on Vehicular Technology Conference (VTC)*. (Vol. 4, p. 2252-2256). doi: 10.1109/VETECE.2003.1285930
- Ho, T.-S., & Chen, K.-C. (1996, Oct). Performance analysis of IEEE 802.11 CSMA/CA medium access control protocol. In *Proceedings of the 7th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. (Vol. 2, p. 407-411). doi: 10.1109/PIMRC.1996.567426
- Hodzic, E., & Mrdovic, S. (2012, Oct). IPv4/IPv6 transition using DNS64/NAT64: Deployment issues. In *Proceedings of the 9th International Symposium on Telecommunications (BIHTEL)*. (p. 1-6). doi: 10.1109/BIHTEL.2012.6412066
- Hoefel, R. (2008, May). IEEE WLANS: 802.11, 802.11e MAC and 802.11a, 802.11b, 802.11g PHY cross layer link budget model for cell coverage estimation. In *Proceedings of the Canadian Conference on Electrical and Computer Engineering (CCECE)*. (p. 1877-1882). doi: 10.1109/CCECE.2008.4564870
- Hoefel, R. (2013, Nov). Multi-user OFDM MIMO in IEEE 802.11ac WLAN: A simulation framework to analysis and synthesis. In *Proceedings of the IEEE Conference on Latin-America Communications (LATINCOM)*. (p. 1-6). doi: 10.1109/LatinCom.2013.6759823
- Hong, R. (2011, May). Research and application of TCP/IP protocol in embedded system. In *Proceedings of the 3rd IEEE International Conference on Communication Software and Networks (ICCSN)*. (p. 584-587). doi: 10.1109/ICCSN.2011.6014961
- Hong, S., Ko, N., Ryu, H. Y., & Kim, N. (2006, July). New IPv6 transition mechanism based on end-to-end tunnel. In *Proceedings of the Joint IEEE International Conference on Optical Internet and Next Generation Network006 (COIN-NGNCON)*. (p. 168-170). doi: 10.1109/COINNGNCON.2006.4454599
- Hong, Y.-G., Shin, M.-K., & Kim, H.-J. (2003, Sept). Application translation for IPv6 at NAT-PT. In *Proceedings of the 9th Asia-Pacific Conference on Communications (APCC)*. (Vol. 1,

- p. 203-207). doi: 10.1109/APCC.2003.1274343
- Hornig, C. (1984, April). *A Standard for the Transmission of IP Datagrams over Ethernet Networks* (No. 894). RFC 894 (INTERNET STANDARD). IETF. Retrieved from <http://www.ietf.org/rfc/rfc894.txt>
- Hou, H., Zhao, Q., & Ma, Y. (2010, Oct). Design and implementation of a solution to smooth IPv6 transition. In *Proceedings of the IEEE International Conference on Advanced Intelligence and Awareness Internet (AIAI)*. (p. 157-161). doi: 10.1049/cp.2010.0743
- Housley, R., Curran, J., Huston, G., & Conrad, D. (2013, August). *The Internet Numbers Registry System* (No. 7020). RFC 7020 (Informational). IETF. Retrieved from <http://www.ietf.org/rfc/rfc7020.txt>
- Huang, C.-W., Antognetti, P., Lam, L., Quaglietta, T., Doherty, M., & Vaillancourt, W. (2012, June). A highly integrated dual-band SiGe power amplifier that enables 256 QAM 802.11ac WLAN radio front-end designs. In *Proceedings of the IEEE Conference on Radio Frequency Integrated Circuits (RFIC)*. (p. 225-228). doi: 10.1109/RFIC.2012.6242269
- Huang, S.-M., Wu, Q., & Lin, Y.-B. (2005, March). Tunneling IPv6 through NAT with TEREDO mechanism. In *Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA)*. (Vol. 2, p. 813-818). doi: 10.1109/AINA.2005.333
- Huang, S.-M., Wu, Q., & Lin, Y.-B. (2006, May). Enhancing TEREDO IPv6 tunneling to traverse the symmetric NAT. *IEEE Communications Letters*, 10(5), 408-410. doi: 10.1109/LCOMM.2006.1633339
- Huitema, C. (2006, February). *Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)* (No. 4380). RFC 4380 (Proposed Standard). IETF. Retrieved from <http://www.ietf.org/rfc/rfc4380.txt> (Updated by RFCs 5991, 6081)
- Huston, G. (2001, Mar). TCP in a wireless world. *Journal of IEEE Internet Computing*, 5(2), 82-84. doi: 10.1109/4236.914651
- Huston, G. (2014). *IPv4 address report*. Retrieved from <http://www.potaroo.net/tools/ipv4/>
- Imran, M., Said, A., & Hasbullah, H. (2010, June). A survey of simulators, emulators and testbeds for wireless sensor networks. In *Proceedings of the IEEE International symposium on Information Technology (ITSim)*. (Vol. 2, p. 897-902). doi: 10.1109/IT-SIM.2010.5561571
- Issac, B., Hamid, K., & Tan, C. (2006, June). Analysis of single and mixed 802.11 networks and mobility architecture. In *Proceedings of the International Conference on Computing Informatics (ICOI)*. (p. 1-6). doi: 10.1109/ICOI.2006.5276532
- Jacob, L., Srijiith, K. N., Duo, H., & Ananda, A. (2002). Effectiveness of TCP SACK, TCP HACK and TCP Trunk over satellite links. In *Proceedings on the IEEE International Conference on Communications (ICC)*. (Vol. 5, p. 3038-3043). doi: 10.1109/ICC.2002.997397
- Jacobson, V., Braden, R., & Borman, D. (1992, May). *TCP Extensions for High Performance* (No. 1323). RFC 1323 (Proposed Standard). IETF. Retrieved from <http://www.ietf.org/rfc/rfc1323.txt> (Obsoleted by RFC 7323)
- Jaha, A., Ben-Shatwan, F., & Ashibani, M. (2008, Sept). Performance evaluation for remote access VPNs on windows server 2003. In *Proceedings of the 2nd IEEE International Confer-*



- ence on Next Generation Mobile Applications, Services and Technologies (NGMAST). (p. 582-587). doi: 10.1109/NGMAST.2008.17
- Jankiewicz, E., Chan, K., & Green, D. (2006, Oct). An approach to IPv6 transition in wireless networks. In *Proceedings of the IEEE Conference on Military Communications (MILCOM)*. (p. 1-7). doi: 10.1109/MILCOM.2006.302468
- Jankiewicz, E., Loughney, J., & Narten, T. (2011, December). *IPv6 Node Requirements* (No. 6434). RFC 6434 (Informational). IETF. Retrieved from <http://www.ietf.org/rfc/rfc6434.txt>
- Javed, U., Suchara, M., He, J., & Rexford, J. (2009, Jan). Multipath protocol for delay-sensitive traffic. In *Proceedings of the 1st IEEE International Conference on Communication Systems and Networks and Workshops (COMSNETS)*. (p. 1-8). doi: 10.1109/COMSNETS.2009.4808885
- Jemai, J., Piesiewicz, R., & Kurner, T. (2005, May). Calibration of an indoor radio propagation prediction model at 2.4 GHz by measurements of the IEEE 802.11b preamble. In *Proceedings of the 61st IEEE Conference on Vehicular Technology Conference (VTC)*. (Vol. 1, p. 111-115). doi: 10.1109/VETECS.2005.1543260
- Joha, A., Shatwan, F., & Ashibani, M. (2007, Sept). Performance evaluation for remote access VPN on windows server 2003 and fedora core 6. In *Proceedings of the 8th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services (TELSIKS)*. (p. 587-592). doi: 10.1109/TELSKS.2007.4376082
- Johnson, D., Stack, T., Fish, R., Flickinger, D., Stoller, L., Ricci, R., & Lepreau, J. (2006, April). Mobile emulab: A robotic wireless and sensor network testbed. In *Proceedings of the 25th IEEE International Conference on Computer Communications*. (p. 1-12). doi: 10.1109/INFOCOM.2006.182
- Judd, G., & Steenkiste, P. (2005). Using emulation to understand and improve wireless networks and applications. In *Proceedings of the 2nd Conference on Symposium on Networked Systems Design and Implementation (NSDI)*. (p. 203-216). Berkeley, CA, USA: USENIX Association. Retrieved from <http://dl.acm.org/citation.cfm?id=1251203.1251218>
- Jurca, D., & Frossard, P. (2007, April). Video packet selection and scheduling for multipath streaming. *Multimedia, IEEE Transactions*, 9(3), 629-641. doi: 10.1109/TMM.2006.888017
- Kapp, S. (2002, Jul). 802.11a. more bandwidth without the wires. *Journal of IEEE Internet Computing*, 6(4), 75-79. doi: 10.1109/MIC.2002.1020329
- Karafillis, P., Fouli, K., ParandehGheibi, A., & Medard, M. (2013, March). An algorithm for improving sliding window network coding in TCP. In *Proceedings of the 47th Annual Conference on Information Sciences and Systems (CISS)*. (p. 1-5). doi: 10.1109/CISS.2013.6552263
- Kent, S. (2005a, December). *IP Authentication Header* (No. 4302). RFC 4302 (Proposed Standard). IETF. Retrieved from <http://www.ietf.org/rfc/rfc4302.txt>
- Kent, S. (2005b, December). *IP Encapsulating Security Payload (ESP)* (No. 4303). RFC 4303 (Proposed Standard). IETF. Retrieved from

- <http://www.ietf.org/rfc/rfc4303.txt>
- Kent, S., & Atkinson, R. (1998, November). *Security Architecture for the Internet Protocol* (No. 2401). RFC 2401 (Proposed Standard). IETF. Retrieved from <http://www.ietf.org/rfc/rfc2401.txt> (Obsoleted by RFC 4301, updated by RFC 3168)
- Kent, S., & Seo, K. (2005, December). *Security Architecture for the Internet Protocol* (No. 4301). RFC 4301 (Proposed Standard). IETF. Retrieved from <http://www.ietf.org/rfc/rfc4301.txt> (Updated by RFC 6040)
- Khademi, N., Welzl, M., & Gjessing, S. (2012, June). Experimental evaluation of TCP performance in multi-rate 802.11 WLANs. In *Proceedings of the IEEE International Symposium on World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. (p. 1-9). doi: 10.1109/WoWMoM.2012.6263696
- Khalifa, I., & Trajkovic, L. (2004, May). An overview and comparison of analytical TCP models. In *Proceedings of the International Symposium on Circuits and Systems (SCAS)*. (Vol. 5, p. 469-472). doi: 10.1109/ISCAS.2004.1329680
- Kim, D., Cai, H., Na, M., & Choi, S. (2008, June). Performance measurement over mobile WiMAX/IEEE 802.16e network. In *Proceedings of the International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. (p. 1-8). doi: 10.1109/WOW-MOM.2008.4594817
- Kim, K., Sung, H., & Lee, H. (1997, Apr). Performance analysis of the TCP/IP protocol under UNIX operating systems for high performance computing and communications. In *Proceedings of the High Performance Computing on the Information Superhighway (HPC)*. (p. 499-504). doi: 10.1109/HPC.1997.592198
- Kim, S., Ryu, H., Park, J., & Kim, T. (2006, Feb). Design and implementation of martini based layer 2 VPN. In *Proceedings of the 8th International Conference on Advanced Communication Technology (ICACT)*. (Vol. 3, p. 4-1500). doi: 10.1109/ICACT.2006.206268
- Knight, P., & Lewis, C. (2004, June). Layer 2 and 3 virtual private networks: taxonomy, technology, and standardization efforts. *IEEE Communications Magazine*, 42(6), 124-131. doi: 10.1109/MCOM.2004.1304248
- Kohler, E., Morris, R., & Poletto, M. (2002). Modular components for network address translation. In *Proceedings of the IEEE Conference on Open Architectures and Network Programming* (p. 39-50). doi: 10.1109/OPNARC.2002.1019227
- Kolahi, S., Cao, Y., & Chen, H. (2013, July). Evaluation of IPv6 with IPSec in IEEE 802.11n wireless LAN using fedora 15 operating system. In *Proceedings of the IEEE Symposium on Computers and Communications (ISCC)*. (p. 203-206). doi: 10.1109/ISCC.2013.6754946
- Kolahi, S., Narayan, S., Nguyen, D., & Sunarto, Y. (2011, March). Performance monitoring of various network traffic generators. In *Proceedings of the 13th IEEE International Conference on Computer Modelling and Simulation (UKSim)*. (p. 501-506). doi: 10.1109/UK-SIM.2011.102
- Kolahi, S., Singla, H., Ehsan, M., & Dong, C. (2011, Feb). The influence of WPA2 security on the UDP performance of IPv4 and IPv6 using 802.11n WLAN in windows 7-windows

- 2008 environment. In *Proceedings of the Baltic Congress on Future Internet Communications (BCFIC Riga)*. (p. 50-53). doi: 10.1109/BCFIC-RIGA.2011.5733211
- Kotsiolis, A., Antonopoulos, C., & Koubias, S. (2010, July). Performance evaluation of TCP algorithms on hybrid wired/wireless LAN test-bed. In *Proceedings of the IEEE International Conference on Data Communication Networking (DCNET)*. (p. 1-7).
- Kotuliak, I., Rybar, P., & Truchly, P. (2011, Oct). Performance comparison of IPSec and TLS based VPN technologies. In *Proceedings of the 9th International Conference on Emerging eLearning Technologies and Applications (ICETA)*. (p. 217-221). doi: 10.1109/ICETA.2011.6112567
- Krawczyk, H. (1996, Feb). SKEME: a versatile secure key exchange mechanism for internet. In *Proceedings of the IEEE Symposium on Network and Distributed System Security*. (p. 114-127). doi: 10.1109/NDSS.1996.492418
- Krop, T., Bredel, M., Hollick, M., & Steinmetz, R. (2007). Jist/mobnet: Combined simulation, emulation, and real-world testbed for ad hoc networks. In *Proceedings of the 2nd ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization (WinTECH)*. (p. 27-34). New York, NY, USA: ACM. Retrieved from <http://doi.acm.org/10.1145/1287767.1287774> doi: 10.1145/1287767.1287774
- Kuihe, Y., & Xin, C. (2007, Aug). Implementation of improved VPN based on SSL. In *Proceedings of the 8th International Conference on Electronic Measurement and Instruments (ICEMI)*. (p. 15-19). doi: 10.1109/ICEMI.2007.4350641
- Kulin, M., Kazaz, T., & Mrdovic, S. (2012, Oct). SIP server security with TLS: Relative performance evaluation. In *Proceedings of the 9th IEEE International Symposium on Telecommunications (BIHTEL)*. (p. 1-6). doi: 10.1109/BIHTEL.2012.6412062
- Kurose, J., & Ross, K. (2008). *Computer Networking* (4th ed.). Addison Wesley.
- Ladid, L. (2001). IPv6 on everything: The new internet IPv6 helps network architects address the ip address shortage, security, QoS, multicast and management. In *Proceedings of the 2nd IEEE International Conference on 3G Mobile Communication Technologies*. (p. 317-322). doi: 10.1049/cp:20010064
- Lakbabi, A., Orhanou, G., & El Hajji, S. (2012, Dec). Vpn ipsec and ssl technology security and management point of view. In *Proceedings of the Next Generation Networks and Services (NGNS)*. (p. 202-208). doi: 10.1109/NGNS.2012.6656108
- Lam, K., Chapin, J., & Chan, V. (2011, March). Performance analysis and optimization of multipath TCP. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*. (p. 695-700). doi: 10.1109/WCNC.2011.5779217
- Lashkari, A., Mansoor, M., & Danesh, A. (2009, May). Wired Equivalent Privacy (WEP) versus Wi-Fi protected access (WPA). In *Proceedings of the International Conference on Signal Processing Systems*. (p. 445-449). doi: 10.1109/ICSPS.2009.87
- Lashkari, A., Towhidi, F., & Hosseini, R. (2009, April). Wired equivalent privacy (WEP). In *Proceedings of the International Conference on Future Computer and Communication (ICFCC)*. (p. 492-495). doi: 10.1109/ICFCC.2009.32
- Leavitt, N. (2011, Sept). IPv6: Any closer to adoption? *Computer*, 44(9), 14-16. doi:

- 10.1109/MC.2011.284
- Lee, D., Carpenter, B., & Brownlee, N. (2010, May). Observations of UDP to TCP ratio and port numbers. In *Proceedings of the 5th IEEE International Conference on Internet Monitoring and Protection (ICIMP)*. (p. 99-104). doi: 10.1109/ICIMP.2010.20
- Lee, D., Lough, D., Midkiff, S., Davis, I., N.J., & Benchoff, P. (1998, Jan). The next generation of the internet: Aspect of the internet protocol version 6. *Network, IEEE*, 12(1), 28-33. doi: 10.1109/65.660004
- Lee, S., Shin, M.-K., Kim, Y.-J., Nordmark, E., & Durand, A. (2002, October). *Dual Stack Hosts Using "Bump-in-the-API" (BIA)* (No. 3338). RFC 3338 (Experimental). IETF. Retrieved from <http://www.ietf.org/rfc/rfc3338.txt> (Obsoleted by RFC 6535)
- Lee, S.-D., Shin, M.-K., & Kim, H.-J. (2006, Feb). The implementation of ISATAP router. In *Proceedings of the 8th International Conference on Advanced Communication Technology (ICACT)*. (Vol. 2, p. 3-1163). doi: 10.1109/ICACT.2006.206177
- Leerujikul, G., & Ahmed, K. (2001). TCP over satellite link with SACK enhancement. In *Proceedings of the IEEE Pacific Rim Conference on Communications, Computers and signal Processing (PACRIM)*. (Vol. 2, p. 350-353). doi: 10.1109/PACRIM.2001.953641
- Letor, N., Torfs, W., & Blondia, C. (2012, Nov). Multimedia multicast performance analysis for 802.11n network cards. In *Proceedings of the IFIP Wireless Days (WD)*. (p. 1-3). doi: 10.1109/WD.2012.6402854
- Li, P., Kolahi, S., Safdari, M., & Argawe, M. (2011, March). Effect of WPA2 security on IEEE802.11n bandwidth and round trip time in peer-peer wireless local area networks. In *Proceedings of the IEEE Workshops of International Conference on Advanced Information Networking and Applications (WAINA)*. (p. 777-782). doi: 10.1109/WAINA.2011.76
- Lian, L., & Wen-Mei, G. (2007, Sept). Building IPsec VPN in IPv6 based on openswan. In *Proceedings of the IFIP International Conference on Network and Parallel Computing Workshops*. (p. 784-787). doi: 10.1109/NPC.2007.50
- Lim, W.-S., Kim, D.-W., & Suh, Y.-J. (2012, May). Design of efficient multicast protocol for IEEE 802.11n WLANs and cross-layer optimization for scalable video streaming. *Journal of IEEE Transactions on Mobile Computing*, 11(5), 780-792. doi: 10.1109/TMC.2011.95
- Liu, P., Meng, M., Ye, X., & Gu, J. (2002). An UDP-based protocol for Internet robots. In *Proceedings of the 4th World Congress on Intelligent Control and Automation*. (Vol. 1, p. 59-65). doi: 10.1109/WCICA.2002.1022068
- Liu, T., Guan, X., Zheng, Q., & Qu, Y. (2009, September). A new worm exploiting IPv6 and IPv4-ipv6 dual-stack networks: experiment, modeling, simulation, and defense. *IEEE Network*, 23(5), 22-29. doi: 10.1109/MNET.2009.5274918
- Liu, W., Jiang, Z., & Zhang, H. (2006, Nov). A secure mobile-IPv6 network model. In *Proceedings of the IET International Conference on Wireless, Mobile and Multimedia Networks*. (p. 1-4).
- Luglio, M., Sanadidi, M., Gerla, M., & Stepanek, J. (2004, Feb). On-board satellite "split TCP" proxy. *IEEE Journal on Selected Areas in Communications*, 22(2), 362-370. doi: 10.1109/JSAC.2003.819987
- Mao, H., Zhu, L., & Qin, H. (2012, Sept). A comparative research on SSL VPN and IPsec VPN.

- In *Proceedings of the 8th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM)*. (p. 1-4). doi: 10.1109/WiCOM.2012.6478270
- Maple, C., Jacobs, H., & Reeve, M. (2006, April). Choosing the right wireless LAN security protocol for the home and business user. In *Proceedings of the 1st International Conference on Availability, Reliability and Security (ARES)*. (p. 8). doi: 10.1109/ARES.2006.42
- Martin, J., Li, B., Pressly, W., & Westall, J. (2010, March). WiMAX performance at 4.9 Ghz. In *Proceedings of the IEEE Aerospace Conference*. (p. 1-8). doi: 10.1109/AERO.2010.5446943
- Mathis, M., Mahdavi, J., Floyd, S., & Romanow, A. (1996, October). *TCP Selective Acknowledgment Options* (No. 2018). RFC 2018 (Proposed Standard). IETF. Retrieved from <http://www.ietf.org/rfc/rfc2018.txt>
- Maughan, D., Schertler, M., Schneider, M., & Turner, J. (1998, November). *Internet Security Association and Key Management Protocol (ISAKMP)* (No. 2408). RFC 2408 (Proposed Standard). IETF. Retrieved from <http://www.ietf.org/rfc/rfc2408.txt> (Obsoleted by RFC 4306)
- Mayer, M., Chan, K., Grillo, B., & Thomas, E. (2007, Oct). IPv6 - from concept to field trials. In *Proceedings of the IEEE Conference on Military Communications (MILCOM)*. (p. 1-7). doi: 10.1109/MILCOM.2007.4455066
- Mazlan, M., Ariffin, S., Balfaqih, M., Hasnan, S., & Haseeb, S. (2012, Oct). Latency evaluation of authentication protocols in centralized 802.11 architecture. In *Proceedings of the IET International Conference on Wireless Communications and Applications (ICWCA)*. (p. 1-6). doi: 10.1049/cp.2012.2113
- McDonald, D. (1994, December). *A Convention for Human-Readable 128-bit Keys* (No. 1751). RFC 1751 (Informational). IETF. Retrieved from <http://www.ietf.org/rfc/rfc1751.txt>
- Metz, C. (2003, Jan). The latest in virtual private networks: part i. *Journal of IEEE Internet Computing*, 7(1), 87-91. doi: 10.1109/MIC.2003.1167346
- Metz, C. (2004, May). The latest in VPNs: part ii. *Journal of IEEE Internet Computing*, 8(3), 60-65. doi: 10.1109/MIC.2004.1297275
- Meyer, M. (1999). TCP performance over GPRS. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*. (p. 1248-1252). doi: 10.1109/WCNC.1999.796937
- Michalski, M. (2012, July). The configurations for experimental study of the network performance. In *Proceedings of the 8th IEEE International Symposium on Communication Systems, Networks Digital Signal Processing (CSNDSP)*. (p. 1-6). doi: 10.1109/CSNDSP.2012.6292790
- Miletic, E., Tittelbach-Helmrich, K., & Panic, G. (2011, Dec). Performance investigation on an MIMO-capable 802.11a compliant MAC protocol implementation. In *Proceedings of the 7th IEEE International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*. (p. 190-193). doi: 10.1109/MSN.2011.57
- Mohamed, E., El-Etriby, S., & Abdul-kader, H. (2012, May). Randomness testing of modern encryption techniques in cloud environment. In *Proceedings of the 8th IEEE International Conference on Informatics and Systems (INFOS)*. (p. 1-6).

- Mohapatra, P., Metz, C., & Cui, Y. (2007, April). Layer 3 VPN services over IPv6 backbone networks: Requirements, technology, and standardization efforts. *IEEE Communications Magazine*, 45(4), 32-37. doi: 10.1109/MCOM.2007.343609
- Moskowitz, R., & Nikander, P. (2006, May). *Host Identity Protocol (HIP) Architecture* (No. 4423). RFC 4423 (Informational). IETF. Retrieved from <http://www.ietf.org/rfc/rfc4423.txt>
- Muskinja, N., Tovornik, B., & Terbuc, M. (2003, Dec). Use of TCP/IP protocol in industrial environment. In *Proceedings of the IEEE International Conference on Industrial Technology*. (Vol. 2, p. 896-900). doi: 10.1109/ICIT.2003.1290778
- Nagle, J. (1984, January). *Congestion Control in IP/TCP Internetworks* (No. 896). RFC 896. IETF. Retrieved from <http://www.ietf.org/rfc/rfc896.txt>
- Narten, T. (1999, Jul). Neighbor discovery and stateless autoconfiguration in IPv6. *Journal of IEEE Internet Computing*, 3(4), 54-62. doi: 10.1109/4236.780961
- Narten, T., Nordmark, E., Simpson, W., & Soliman, H. (2007, September). *Neighbor Discovery for IP version 6 (IPv6)* (No. 4861). RFC 4861 (Draft Standard). IETF. Retrieved from <http://www.ietf.org/rfc/rfc4861.txt> (Updated by RFCs 5942, 6980, 7048)
- Neigus, N. (1972, December). *Network logical map* (No. 432). RFC 432. IETF. Retrieved from <http://www.ietf.org/rfc/rfc432.txt>
- Nguyen, C., Vialatte, M.-C., & Rieu, C. (1989, Oct). Osi application layer standards analysis for a distributed application implementation. In *Proceedings of the 14th IEEE Conference on Local Computer Networks*. (p. 225-233). doi: 10.1109/LCN.1989.65266
- Nguyen, T.-H., Park, M., Youn, Y., & Jung, S. (2013, July). An improvement of TCP performance over wireless networks. In *Proceedings of the 5th International Conference on Ubiquitous and Future Networks (ICUFN)*. (p. 214-219). doi: 10.1109/ICUFN.2013.6614814
- Nichols, K., Blake, S., Baker, F., & Black, D. (1998, December). *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers* (No. 2474). RFC 2474 (Proposed Standard). IETF. Retrieved from <http://www.ietf.org/rfc/rfc2474.txt> (Updated by RFCs 3168, 3260)
- Niehenke, E. (2014, March). Wireless communications: Present and future: Introduction to focused issue articles. *IEEE Microwave Magazine*, 15(2), 26-35. doi: 10.1109/MMM.2013.2296207
- Nisbet, A. (2012, Jan). A tale of four cities: Wireless security and growth in New Zealand. In *Proceedings of the International Conference on Computing, Networking and Communications (ICNC)*. (p. 1167-1171). doi: 10.1109/ICCNC.2012.6167391
- Nordmark, E., & Gilligan, R. (2005, October). *Basic Transition Mechanisms for IPv6 Hosts and Routers* (No. 4213). RFC 4213 (Proposed Standard). IETF. Retrieved from <http://www.ietf.org/rfc/rfc4213.txt>
- Nossier, B. (2004, Sept). Performance evaluation of TCP congestion control for different network operating systems. In *Proceedings of the IEEE International Conference on Electrical, Electronic and Computer Engineering (ICEEC)*. (p. 112-115). doi: 10.1109/ICEEC.2004.1374394
- Obata, H., Tamehiro, K., & Ishida, K. (2011, March). Experimental evaluation of TCP-STAR

- for Satellite Internet over WINDS. In *Proceedings of the 10th International Symposium on Autonomous Decentralized Systems (ISADS)*. (p. 605-610). doi: 10.1109/ISADS.2011.86
- Omprakash, P., & Sabitha, R. (2011, April). Performance analysis of TCP over WiMAX. In *Proceedings of the 3rd International Conference on Electronics Computer Technology (ICECT)*. (Vol. 6, p. 348-352). doi: 10.1109/ICECTECH.2011.5942113
- Ong, E., Knecht, J., Alanen, O., Chang, Z., Huovinen, T., & Nihtila, T. (2011, Sept). IEEE 802.11ac: Enhancements for very high throughput WLANs. In *Proceedings of the 22nd IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*. (p. 849-853). doi: 10.1109/PIMRC.2011.6140087
- Orman, H. (1998, November). *The OAKLEY Key Determination Protocol* (No. 2412). RFC 2412 (Informational). IETF. Retrieved from <http://www.ietf.org/rfc/rfc2412.txt>
- Ortiz, S. (1997, Nov). Virtual private networks: leveraging the Internet. *Journal of Computer*, 30(11), 18-20. doi: 10.1109/2.634834
- Othman, A.-K., Zakaria, M., & Ab Hamid, K. (2007, Dec). TCP performance measurement in different GPRS network scenarios. In *Proceedings of the Asia-Pacific Conference on Applied Electromagnetics (APACE)*. (p. 1-5). doi: 10.1109/APACE.2007.4603985
- Otrok, H., Haraty, R., & El-Kassar, A. N. (2006, July). Improving the secure socket layer protocol by modifying its authentication function. In *Proceedings of the World Automation Congress (WAC)*. (p. 1-6). doi: 10.1109/WAC.2006.375755
- Pacheco de Carvalho, J., Pacheco, F., Reis, A., Marques, N., & Veiga, H. (2011, June). Comparative performance studies of laboratory Wi-Fi IEEE 802.11 b,g WEP point-to-point links. In *Proceedings of the 6th IEEE Iberian Conference on Information Systems and Technologies (CISTI)*. (p. 1-6).
- Pacheco de Carvalho, J., Veiga, H., Pacheco, F., & Reis, A. (2014, July). Experimental performance studies of laboratory WPA IEEE 802.11b, g PTMP links. In *Proceedings of the 9th IEEE International Symposium on Communication Systems, Networks Digital Signal Processing (CSNDSP)*. (p. 575-579). doi: 10.1109/CSNDSP.2014.6923894
- Park, Z., Lee, J., & Kim, M. (2003, July). Design of an extended TCP for preventing DOS attacks. In *Proceedings of the 7th Korea-Russia International Symposium on Science and Technology (KORUS)*. (Vol. 2, p. 385-389).
- Partridge, C., & Pink, S. (1993, Aug). A faster UDP [user datagram protocol]. *Journal of IEEE/ACM Transactions on Networking*, 1(4), 429-440. doi: 10.1109/90.251895
- Paul, T., & Ogunfunmi, T. (2008, First). Wireless LAN comes of age: Understanding the IEEE 802.11n amendment. *Journal of IEEE Circuits and Systems Magazine*, 8(1), 28-54. doi: 10.1109/MCAS.2008.915504
- Pena, C., & Evans, J. (2000). Performance evaluation of software virtual private networks (VPN). In *Proceedings of the 25th Annual IEEE Conference on Local Computer Networks (LCN)*. (p. 522-523). doi: 10.1109/LCN.2000.891094
- Peng, F., Cardona, A., Shafiee, K., & Leung, V. (2012, Sept). TCP performance evaluation over GEO and LEO Satellite links between performance enhancement proxies. In *Proceedings of the IEEE Vehicular Technology Conference (VTC Fall)*. (p. 1-5). doi: 10.1109/VTC-

- Fall.2012.6399248
- Peng, W., Zhou, Y., Wang, C., & Yang, Y. (2009, Nov). Research on IPSec-based NAT-PT transition mechanism. In *Proceedings of the IEEE International Conference on Network Infrastructure and Digital Content (ICNIDC)*. (p. 222-226). doi: 10.1109/ICNIDC.2009.5360823
- Perahia, E. (2008, July). IEEE 802.11n development: History, process, and technology. *IEEE Communications Magazine*, 46(7), 48-55. doi: 10.1109/MCOM.2008.4557042
- Petersen, S., & Carlsen, S. (2008, March). Wireless sensor networks: Introduction to installation and integration on an offshore oil and gas platform. In *Proceedings of the 19th Australian Conference on Software Engineering (ASWEC)*. (p. 53-53). doi: 10.1109/ASWEC.2008.4483192
- Picard, V., & Lafond, E. (2014, Jan). Performance evaluation of next generation Wi-Fi (802.11ac) for mobile offloading. In *Proceedings of the 11th IEEE Consumer Communications and Networking Conference (CCNC)*. (p. 515-516). doi: 10.1109/CCNC.2014.6940508
- Postel, J. (1980, August). *User Datagram Protocol* (No. 768). RFC 768 (INTERNET STANDARD). IETF. Retrieved from <http://www.ietf.org/rfc/rfc768.txt>
- Postel, J. (1981a, September). *Internet Control Message Protocol* (No. 792). RFC 792 (INTERNET STANDARD). IETF. Retrieved from <http://www.ietf.org/rfc/rfc792.txt> (Updated by RFCs 950, 4884, 6633, 6918)
- Postel, J. (1981b, September). *Internet Protocol* (No. 791). RFC 791 (INTERNET STANDARD). IETF. Retrieved from <http://www.ietf.org/rfc/rfc791.txt> (Updated by RFCs 1349, 2474, 6864)
- Postel, J. (1981c, September). *Transmission Control Protocol* (No. 793). RFC 793 (INTERNET STANDARD). IETF. Retrieved from <http://www.ietf.org/rfc/rfc793.txt> (Updated by RFCs 1122, 3168, 6093, 6528)
- Postel, J. (1983, November). *The TCP Maximum Segment Size and Related Topics* (No. 879). RFC 879. IETF. Retrieved from <http://www.ietf.org/rfc/rfc879.txt> (Updated by RFC 6691)
- Postel, J., & Reynolds, J. (1988, February). *Standard for the transmission of IP datagrams over IEEE 802 networks* (No. 1042). RFC 1042 (INTERNET STANDARD). IETF. Retrieved from <http://www.ietf.org/rfc/rfc1042.txt>
- Punithavathani, D., & Radley, S. (2014, Third). Performance analysis for wireless networks: An analytical approach by multifarious sym teredo. *The Scientific World Journal*, 2014(3), 1407-1424. doi: 10.1155/2014/304914
- Pursley, M., & Royster, T. (2007, June). IEEE 802.11b complementary code keying and complementary signals derived from biorthogonal sequences. In *Proceedings of the IEEE International Conference on Communications (ICC)*. (p. 902-907). doi: 10.1109/ICC.2007.153
- Pursley, M., & Royster, T. (2009, February). Properties and performance of the IEEE 802.11b complementary-code-key signal sets. *Journal of IEEE Transactions on Communications*, 57(2), 440-449. doi: 10.1109/TCOMM.2009.02.060642
- Qu, J., Li, L., & Dang, F. (2012, Aug). Performance evaluation and analysis of open VPN on android. In *Proceedings of the 4th International Conference on Computational and Informa-*



- tion Sciences (ICCIS). (p. 1088-1091). doi: 10.1109/ICCIS.2012.203
- Qureshi, B., Othman, M., & Hamid, N. A. W. (2009, Feb). Progress in various TCP variants (February 2009). In *Proceedings of the 2nd International Conference on Computer, Control and Communication (IC4)*. (p. 1-6). doi: 10.1109/IC4.2009.4909200
- Rafiee, H., & Meinel, C. (2013a, Aug). A secure, flexible framework for DNS authentication in IPv6 autoconfiguration. In *Proceedings of the 12th IEEE International Symposium on Network Computing and Applications (NCA)*. (p. 165-172). doi: 10.1109/NCA.2013.37
- Rafiee, H., & Meinel, C. (2013b, July). Ssas: A simple secure addressing scheme for IPv6 autoconfiguration. In *Proceedings of the 11th Annual International Conference on Privacy, Security and Trust (PST)*. (p. 275-282). doi: 10.1109/PST.2013.6596063
- Raicu, I., & Zeadally, S. (2003, Feb). Impact of IPv6 on end-user applications. In *Proceedings of the 10th IEEE International Conference on Telecommunications (ICT)*. (Vol. 2, p. 973-980). doi: 10.1109/ICTEL.2003.1191571
- Ramakrishnan, K., Floyd, S., & Black, D. (2001, September). *The Addition of Explicit Congestion Notification (ECN) to IP* (No. 3168). RFC 3168 (Proposed Standard). IETF. Retrieved from <http://www.ietf.org/rfc/rfc3168.txt> (Updated by RFCs 4301, 6040)
- Raychaudhuri, D., Seskar, I., Ott, M., Ganu, S., Ramachandran, K., Kremo, H., ... Singh, M. (2005, March). Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols. In *Proceedings of the IEEE International Conference on Wireless Communications and Networking* (Vol. 3, p. 1664-1669). doi: 10.1109/WCNC.2005.1424763
- Reddy, S., Sai Ramani, K., Rijutha, K., Ali, S., & Reddy, C. (2010, June). Wireless hacking - a WiFi hack by cracking WEP. In *Proceedings of the 2nd International Conference on Education Technology and Computer (ICETC)*. (Vol. 1, p. 189-193). doi: 10.1109/ICETC.2010.5529269
- Ren, F., & Zhou, H. (2012, July). Implementation and test of PMIPv6 dual stack protocol. In *Proceedings of the 6th IEEE International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*. (p. 305-310). doi: 10.1109/IMIS.2012.67
- Renbo, W., & Xiong, W. (2009, May). Reduced TCP/IP protocol implement in VHDL. In *Proceedings of the International Workshop on Intelligent Systems and Applications (ISA)*. (p. 1-5). doi: 10.1109/IWISA.2009.5073030
- Rescorla, E., & Modadugu, N. (2006, April). *Datagram Transport Layer Security* (No. 4347). RFC 4347 (Proposed Standard). IETF. Retrieved from <http://www.ietf.org/rfc/rfc4347.txt> (Obsoleted by RFC 6347, updated by RFC 5746)
- Rooney, T. (2011). *IP Address Management Principles and Practive* (1st ed.). New Jersey, NJ, USA: John Wiley & Sons, Inc.
- Safa, H., Karam, M., Assi, R., & Mcheick, H. (2011, March). A transport layer interoperability model for mobile ad-hoc environment. In *Proceedings of the International Conference on Communications and Information Technology (ICCIT)*. (p. 130-133). doi: 10.1109/IC-CITECHNOL.2011.5762663
- Saini, T., & Dhaka, M. (2009, Dec). Performance simulation of Tahoe, Reno, New Reno and

- SACK over terrestrial and geostationary satellite links. In *Proceedings of the International Conference on Methods and Models in Computer Science (ICM2CS)*. (p. 1-5). doi: 10.1109/ICM2CS.2009.5397980
- Saleh, S., Shah, Z., & Baig, A. (2013, Oct). Capacity analysis of combined IPTV and VoIP over IEEE 802.11n. In *Proceedings of the 38th IEEE Conference on Local Computer Networks (LCN)*. (p. 785-792). doi: 10.1109/LCN.2013.6761333
- Saliga, S. (2000, June). An introduction to IEEE 802.11 wireless LANs. In *Proceedings of the IEEE Symposium on Radio Frequency Integrated Circuits (RFIC)*. (p. 11-14). doi: 10.1109/RFIC.2000.854406
- Sans, F., & Gamess, E. (2013, Oct). Analytical performance evaluation of native IPv6 and several tunneling technics using benchmarking tools. In *Proceedings of the XXXIX Latin American Conference on Computing (CLEI)*. (p. 1-9). doi: 10.1109/CLEI.2013.6670610
- Schweber, W. (1992). *Data Communications*. McGraw-Hill Education (India) Pvt Limited.
- Selim, G., El Badawy, H., & Salam, M. (2006, Feb). New protocol design for wireless networks security. In *Proceedings of the 8th International Conference Advanced Communication Technology (ICACT)*. (Vol. 1, p. 4-776). doi: 10.1109/ICACT.2006.206078
- Shah, S., Bilal, K., Khan, A., & Rehman, A. (2007, Nov). TCP congestion control: A hybrid approach. In *Proceedings of the International Conference on Emerging Technologies (ICET)*. (p. 89-93). doi: 10.1109/ICET.2007.4516322
- Shakkottai, S., Srikant, R., Brownlee, N., Broido, A., & Others. (2004). The RTT distribution of TCP flows in the internet and its impact on TCP-based flow control.
- Sharma, C., & Tyagi, B. (2013, Feb). Performance evaluation of TCP variants under different node speeds using OPNET simulator. In *Proceedings of the 3rd IEEE International Advance Computing Conference (IACC)*. (p. 302-307). doi: 10.1109/IAdCC.2013.6514240
- Shekhar, H., & Ramanatha, K. (2010, June). Performance evaluation of datagram congestion control protocol in mobile ad hoc networks. In *Proceedings of the 2nd IEEE International Conference on Communication Systems, Networks and Applications (ICCSNA)*. (Vol. 1, p. 71-76). doi: 10.1109/ICCSNA.2010.5588774
- Shen, S., Lee, X., Sun, Z., & Jiang, S. (2011, June). Enhance IPv6 dynamic host configuration with cryptographically generated addresses. In *Proceedings of the 5th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*. (p. 487-490). doi: 10.1109/IMIS.2011.126
- Shi, W., Huang, C., Wang, Q., Chen, Y., Huang, Y., & Cheng, Y. (2007, Feb). A novel IPv4/IPv6 translation mechanism based on NAT-PT. In *Proceedings of the 9th International Conference on Advanced Communication Technology*. (Vol. 2, p. 1037-1041). doi: 10.1109/ICACT.2007.358535
- Shukla, A., & Brecht, T. (2006, Nov). TCP connection management mechanisms for improving Internet server performance. In *Proceedings of the 1st IEEE Workshop on Hot Topics in Web Systems and Technologies (HOTWEB)*. (p. 1-12). doi: 10.1109/HOTWEB.2006.355264
- Sierra, J., Hernandez, J., Ribagorda, A., & Jayaram, N. (2002). Migration of internet security protocols to the ipsec framework. In *Proceedings of the 36th IEEE Annual International Carnahan Conference on Security Technology*. (p. 134-143). doi:

- 10.1109/CCST.2002.1049239
- Sikdar, B., Kalyanaraman, S., & Vastola, K. (2003, Dec). Analytic models for the latency and steady-state throughput of TCP Tahoe, Reno, and SACK. *Journal of IEEE/ACM Transactions on Networking*, 11(6), 959-971. doi: 10.1109/TNET.2003.820427
- Simpson, W. (1994, July). *The Point-to-Point Protocol (PPP)* (No. 1661). RFC 1661 (INTERNET STANDARD). IETF. Retrieved from <http://www.ietf.org/rfc/rfc1661.txt> (Updated by RFC 2153)
- Singh, A., Xiang, M., Konsgen, A., & Goerg, C. (2013, July). Performance and fairness comparison of extensions to dynamic window coupling for multipath TCP. In *Proceedings of the 9th IEEE International Conference on Wireless Communications and Mobile Computing (IWCMC)*. (p. 947-952). doi: 10.1109/IWCMC.2013.6583684
- Singh, U., & Jindal, P. (2014, Feb). Performance analysis of secure wireless local area network using test-bed. In *Proceedings of the 4th IEEE International Conference on Advanced Computing Communication Technologies (ACCT)*. (p. 386-389). doi: 10.1109/ACCT.2014.55
- Sobeih, A., Chen, W.-P., Hou, J., Kung, L.-C., Li, N., Lim, H., ... Zhang, H. (2005, April). J-Sim: A simulation environment for wireless sensor networks. In *Proceedings of the 38th IEEE annual symposium on simulation*. (p. 175-187). doi: 10.1109/ANSS.2005.27
- Socolofsky, T., & Kale, C. (1991, January). *TCP/IP tutorial* (No. 1180). RFC 1180 (Informational). IETF. Retrieved from <http://www.ietf.org/rfc/rfc1180.txt>
- Srisuresh, P., & Egevang, K. (2001, January). *Traditional IP Network Address Translator (Traditional NAT)* (No. 3022). RFC 3022 (Informational). IETF. Retrieved from <http://www.ietf.org/rfc/rfc3022.txt>
- Srisuresh, P., & Holdrege, M. (1999, August). *IP Network Address Translator (NAT) Terminology and Considerations* (No. 2663). RFC 2663 (Informational). IETF. Retrieved from <http://www.ietf.org/rfc/rfc2663.txt>
- Staalhagen, L. (1996, Jan). A comparison between the OSI reference model and the B-ISDN protocol reference model. *Journal of IEEE Network*, 10(1), 24-33. doi: 10.1109/65.484229
- Stallings, W. (1996, Jul). IPv6: The new internet protocol. *Communications Magazine, IEEE*, 34(7), 96-108. doi: 10.1109/35.526895
- Steffann, S., van Beijnum, I., & van Rein, R. (2013, November). *A Comparison of IPv6-over-IPv4 Tunnel Mechanisms* (No. 7059). RFC 7059 (Informational). IETF. Retrieved from <http://www.ietf.org/rfc/rfc7059.txt>
- Stevens, W. (1997, January). *TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms* (No. 2001). RFC 2001 (Proposed Standard). IETF. Retrieved from <http://www.ietf.org/rfc/rfc2001.txt> (Obsoleted by RFC 2581)
- Su, J., & Zhou, X. (2013, Nov). IVIT: A core stateless IPv4/IPv6 transition mechanism combining translation and tunnel technologies. In *Proceedings of the IEEE International Conference on Cyberspace Technology (CCT)*. (p. 252-257). doi: 10.1049/cp.2013.2134
- Sun, S. (2011, July). The advantages and the implementation of SSL VPN. In *Proceedings of the 2nd IEEE International Conference on Software Engineering and Service Science (ICSESS)*. (p. 548-551). doi: 10.1109/ICSESS.2011.5982375
- Sun, X. (2012, June). TCP congestion control algorithm research. In *Proceedings of the 8th*

- International Conference on Information Science and Digital Content Technology (ICIDT)*. (Vol. 3, p. 703-706).
- Supplement to IEEE standard for information technology - telecommunications and information exchange between systems - local and metropolitan area networks - specific requirements. part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: High-speed physical layer in the 5 GHz band. (1999). *IEEE Std 802.11a-1999*, i-. doi: 10.1109/IEEESTD.1999.90606
- Supplement to IEEE standard for information technology- telecommunications and information exchange between systems- local and metropolitan area networks- specific requirements- part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: Higher-speed physical layer extension in the 2.4 GHz band. (2000). *IEEE Std 802.11b-1999*, i-90. doi: 10.1109/IEEESTD.2000.90914
- Taank, R. K. (2008). *An Evaluation of TCP over Wired-to-Wireless Networks* (Unpublished doctoral dissertation). Aston University.
- Taggart, E., D. Burger, & Rudolph, D. (2009). *Next generation Internet: IPv4 address exhaustion, mitigation strategies and implications for the U.S.* [White Paper]. Retrieved from <http://www.ieeeusa.org/policy/whitepapers/IEEEUSAWP-IPv62009.pdf>
- Tandjaoui, D., & Badache, N. (2004, April). Performance evaluation of micromobility environments for various TCP protocols. In *Proceedings of the 13th IEEE Workshop on Local and Metropolitan Area Networks (LANMAN)*. (p. 157-162). doi: 10.1109/LANMAN.2004.1338424
- Tanenbaum, A., & Wetherall, D. (2011). *Computer Networks* (5th ed.). USA: Prentice Hall.
- Templin, F., Gleeson, T., & Thaler, D. (2008, March). *Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)* (No. 5214). RFC 5214 (Informational). IETF. Retrieved from <http://www.ietf.org/rfc/rfc5214.txt>
- Thomson, S., Narten, T., & Jinmei, T. (2007, September). *IPv6 Stateless Address Autoconfiguration* (No. 4862). RFC 4862 (Draft Standard). IETF. Retrieved from <http://www.ietf.org/rfc/rfc4862.txt>
- Tippanagoudar, V., Mahgoub, I., & Badi, A. (2007, May). Implementation of the sensor-MAC protocol for the JiST/SWANS simulator. In *Proceedings of the IEEE International Conference on Computer Systems and Applications*. (p. 225-232). doi: 10.1109/AICCSA.2007.370887
- Townsley, W., Valencia, A., Rubens, A., Pall, G., Zorn, G., & Palter, B. (1999, August). *Layer Two Tunneling Protocol "L2TP"* (No. 2661). RFC 2661 (Proposed Standard). IETF. Retrieved from <http://www.ietf.org/rfc/rfc2661.txt>
- Tsetse, A., Wijesinha, A., Karne, R., & Loukili, A. (2012, May). A 6to4 gateway with co-located NAT. In *Proceedings of the IEEE International Conference on Electro/Information Technology (EIT)*. (p. 1-6). doi: 10.1109/EIT.2012.6220769
- Tsiknas, K., & Stamatelos, G. (2012, April). Performance evaluation of TCP in IEEE 802.16 networks. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*. (p. 2951-2955). doi: 10.1109/WCNC.2012.6214309

- Tsirsis, G., & Srisuresh, P. (2000, February). *Network Address Translation - Protocol Translation (NAT-PT)* (No. 2766). RFC 2766 (Historic). IETF. Retrieved from <http://www.ietf.org/rfc/rfc2766.txt> (Obsoleted by RFC 4966, updated by RFC 3152)
- Ullah, I., & Khan, R. (2008, Dec). Congestion control algorithm for high speed networks with end system awareness. In *Proceedings of the IEEE International Multitopic Conference (INMIC)*. (p. 318-322). doi: 10.1109/INMIC.2008.4777756
- Uskov, A. (2012, June). Information security of IPsec-based mobile VPN: Authentication and encryption algorithms performance. In *Proceedings of the 11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. (p. 1042-1048). doi: 10.1109/TrustCom.2012.187
- Uskov, A., & Avagyan, H. (2014, June). The efficiency of block ciphers in galois/counter mode in IPsec-based virtual private networks. In *Proceedings of the IEEE International Conference on Electro/Information Technology (EIT)*. (p. 173-178). doi: 10.1109/EIT.2014.6871757
- Varadarajan, P., & Crosby, G. (2014, March). Implementing IPsec in wireless sensor networks. In *Proceedings of the 6th IEEE International Conference on New Technologies, Mobility and Security (NTMS)*. (p. 1-5). doi: 10.1109/NTMS.2014.6814024
- Varshini, S., & Chaurasia, L. (2010, Dec). TCP agents and QoS measurements: Performance analysis on multi-source scenario. In *Proceedings of the annual IEEE India Conference (INDICON)*. (p. 1-5). doi: 10.1109/INDCON.2010.5712714
- Vaughan-Nichols, S. J. (2010, Nov). Gigabit Wi-Fi is on its way. *Journal of Computer*, 43(11), 11-14. doi: 10.1109/MC.2010.318
- Verma, L., Fakharzadeh, M., & Choi, S. (2013, December). Wifi on steroids: 802.11ac and 802.11ad. *Journal of IEEE Wireless Communications*, 20(6), 30-35. doi: 10.1109/MWC.2013.6704471
- Verma, O., Agarwal, R., Dafouti, D., & Tyagi, S. (2011, April). Performance analysis of data encryption algorithms. In *Proceedings of the 3rd IEEE International Conference on Electronics Computer Technology (ICECT)*. (Vol. 5, p. 399-403). doi: 10.1109/ICECTECH.2011.5942029
- Waddington, D., & Chang, F. (2002, Jun). Realizing the transition to IPv6. *IEEE Communications Magazine*, 40(6), 138-147. doi: 10.1109/MCOM.2002.1007420
- Waghmare, S., Parab, A., Nikose, P., & Bhosale, S. J. (2011, April). Comparative analysis of different TCP variants in a wireless environment. In *Proceedings of the 3rd International Conference on Electronics Computer Technology (ICECT)*. (Vol. 4, p. 158-162). doi: 10.1109/ICECTECH.2011.5941878
- Wang, Liu, K., & Hu, F. (2005, Nov). Simulation of wireless sensor networks localization with omnet. In *Proceedings of the 2nd International Conference on Mobile Technology, Applications and Systems*. (p. 1-6). doi: 10.1109/MTAS.2005.207141
- Wang, K., Yeo, A.-K., & Ananda, A. (2001). DTTS: a transparent and scalable solution for IPv4 to IPv6 transition. In *Proceedings of the 10th International Conference on Computer Communications and Networks* (p. 248-253). doi: 10.1109/ICCCN.2001.956257

- Wang, S.-C., Chen, Y.-M., Lee, T.-H., & Helmy, A. (2005, April). Performance evaluations for hybrid IEEE 802.11b and 802.11g wireless networks. In *Proceedings of the 24th IEEE International Performance, Computing, and Communications Conference (IPCCC)*. (p. 111-118). doi: 10.1109/PCCC.2005.1460529
- Wang, T., & Refai, H. (2005, March). Network performance analysis on IEEE 802.11g with different protocols and signal to noise ratio values. In *Proceedings of the 2nd IFIP International Conference on Wireless and Optical Communications Networks (WOCN)*. (p. 29-33). doi: 10.1109/WOCN.2005.1435983
- Wang, Y., Yao, S., Zhao, Y., & Zhou, M. (2001). CPN modeling and analysis of L2TP. In *Proceedings of the International Conference on Computer Networks and Mobile Computing*. (p. 281-288). doi: 10.1109/ICCNMC.2001.962609
- Wei, X., Zhang, J.-W., & Zhang, G.-D. (2009, May). Application research on IPv4/IPv6 dual stack technology. In *Proceedings of the International Conference on Signal Processing Systems*. (p. 826-828). doi: 10.1109/ICSPS.2009.200
- Wells, J. (2009, May). Faster than fiber: The future of multi-G/s wireless. *Journal of IEEE Microwave Magazine*, 10(3), 104-112. doi: 10.1109/MMM.2009.932081
- Welsh, E., Fish, W., & Frantz, J. (2003, May). GNOMES: A testbed for low power heterogeneous wireless sensor networks. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*. (Vol. 4, p. 836-839). doi: 10.1109/ISCAS.2003.1206350
- Wennstrom, A., Brunstrom, A., & Rendon, J. (2004, Sept). Impact of GPRS buffering on TCP performance. *Electronics Letters*, 40(20), 1279-1281. doi: 10.1049/el:20045967
- Werner-Allen, G., Swieskowski, P., & Welsh, M. (2005, April). Motelab: A wireless sensor network testbed. In *Proceedings of the 4th IEEE International Symposium on Information Processing in Sensor Networks (IPSN)*. (p. 483-488). doi: 10.1109/IPSN.2005.1440979
- Wing, D. (2010, July). Network address translation: Extending the Internet address space. *Journal of IEEE Internet Computing*, 14(4), 66-70. doi: 10.1109/MIC.2010.96
- Wood, D., Stoss, V., Chan-Lizardo, L., Papacostas, G. S., & Stinson, M. E. (1988, Jun). Virtual private networks. In *Proceedings of the International Conference on Private Switching Systems and Networks*. (p. 132-136).
- Wu, P., Cui, Y., Wu, J., Liu, J., & Metz, C. (2013, Third). Transition from IPv4 to IPv6: A state-of-the-art survey. *Communications Surveys Tutorials, IEEE*, 15(3), 1407-1424. doi: 10.1109/SURV.2012.110112.00200
- Wu, P., Cui, Y., Xu, M., Wu, J., Li, X., Metz, C., & Wang, S. (2010, Dec). PET: Prefixing, encapsulation and translation for IPv4-IPv6 coexistence. In *Proceedings of the IEEE Conference on Global Telecommunications (GLOBECOM)*. (p. 1-5). doi: 10.1109/GLOCOM.2010.5683446
- Wu, Y., & Zhou, X. (2011, Aug). Research on the IPv6 performance analysis based on dual-protocol stack and tunnel transition. In *Proceedings of the 6th International Conference on Computer Science Education (ICCSE)*. (p. 1091-1093). doi: 10.1109/ICCSE.2011.6028824
- Wu, Y., Zhu, J., Kong, S., & Yi, P. (2009, Nov). Experimental analysis of secure authentication protocols for WLAN-based mesh networks. In *Proceedings of the 4th International Conference on Computer Sciences and Convergence Information Technology (ICCIT)*. (p. 1062-

- 1065). doi: 10.1109/ICCIT.2009.40
- Xia, H., Bataineh, K., Hassoun, M., & Kryzak, J. (1999, Jul). A mixed-signal behavioral level implementation of 1000base-x physical layer for gigabit ethernet. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*. (Vol. 1, p. 431-434). doi: 10.1109/ISCAS.1999.777902
- Xiaodong, Z., Mayan, & Yumei, Z. (2009, Dec). Research on the next-generation internet transition technology. In *Proceedings of the 2nd IEEE International Symposium on Computational Intelligence and Design (ISCID)*. (Vol. 2, p. 380-382). doi: 10.1109/ISCID.2009.241
- Xiaohong, L. (2013, June). The research of network transitional technology from IPv4 to IPv6. In *Proceedings of the 4th IEEE International Conference on Digital Manufacturing and Automation (ICDMA)*. (p. 1507-1509). doi: 10.1109/ICDMA.2013.361
- Xiaorong, F., Jun, L., & Shizhun, J. (2013, Dec). Security analysis for IPv6 neighbor discovery protocol. In *Proceedings of the 2nd International Symposium on Instrumentation and Measurement, Sensor Network and Automation (IMSNA)*. (p. 303-307). doi: 10.1109/IMSNA.2013.6743275
- Xu, S., & Saadawi, T. (2001). Evaluation for TCP with delayed ACK option in wireless multi-hop networks. In *Proceedings of the IEEE VTS 54th Vehicular Technology Conference (VTC)*. (Vol. 1, p. 267-271). doi: 10.1109/VTC.2001.956599
- Xu, S., Saadawi, T., & Lee, M. (2000). Comparison of TCP Reno and Vegas in wireless mobile ad hoc networks. In *Proceedings of the 25th Annual IEEE Conference on Local Computer Networks (LCN)*. (p. 42-43). doi: 10.1109/LCN.2000.891005
- Xu, W., Xu, Y., Wu, X., & Ou, K. (2011, Jan). Modeling TCP sack steady state performance in lossy networks. In *Proceedings of the International Conference on Information Networking (ICOIN)*. (p. 278-283). doi: 10.1109/ICOIN.2011.5723193
- Xylomenos, G., & Polyzos, G. (1999, Mar). TCP and UDP performance over a wireless LAN. In *Proceedings of the 18th IEEE Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*. (Vol. 2, p. 439-446). doi: 10.1109/INFOCOM.1999.751376
- Xylomenos, G., Polyzos, G., Mahonen, P., & Saaranen, M. (2001, Apr). TCP performance issues over wireless links. *IEEE Communications Magazine*, 39(4), 52-58. doi: 10.1109/35.917504
- Yang, Y., Ding, J., Wen, Q., & Zhang, H. (2010, Oct). Research and design of the PMI-based access control model for openVPN. In *Proceedings of the International Conference on Advanced Intelligence and Awareness Internet (AIAD)*. (p. 77-80). doi: 10.1049/cp.2010.0724
- Younglove, R. (2001, Feb). IP security: what makes it work? *Computing Control Engineering Journal*, 12(1), 44-46. doi: 10.1049/cce:20010107
- Yue, Z., Zhang, X., Ren, Y., Li, J., & Zhong, Q. (2012, June). The performance evaluation and comparison of TCP-based high-speed transport protocols. In *Proceedings of the 3rd IEEE International Conference on Software Engineering and Service Science (ICSESS)*. (p. 509-512). doi: 10.1109/ICSESS.2012.6269516
- Zander, S., Andrew, L., Armitage, G., & Huston, G. (2013, Oct). Estimating IPv4 address space usage with capture-recapture. In *Proceedings of the 38th IEEE Conference on Local Computer Networks Workshops (LCN Workshops)*. (p. 1010-1017). doi:

- 10.1109/LCNW.2013.6758545
- Zeadally, S., Wasseem, R., & Raicu, I. (2004, June). Comparison of end-system IPv6 protocol stacks. *Communications, IEEE*, 151(3), 238-242. doi: 10.1049/ip-com:20040283(410) 151
- Zeng, H., Peng, X., Li, M., Xu, H., & Jin, S. (2009, Dec). Research on an effective approach against DDoS attacks. In *Proceedings of the International Conference on Research Challenges in Computer Science (ICRCCS)*. (p. 21-23). doi: 10.1109/ICRCCS.2009.15
- Zeng, J., & Ansari, N. (2003, April). Toward IP virtual private network quality of service: a service provider perspective. *IEEE Communications Magazine*, 41(4), 113-119. doi: 10.1109/MCOM.2003.1193984
- Zhang, H., Li, X., & Bao, C. (2013, July). An evolvable locator/ID separation internet architecture (ELISIA). In *Proceedings of the 8th IEEE International Conference on Networking, Architecture and Storage (NAS)*. (p. 141-150). doi: 10.1109/NAS.2013.24
- Zhou, L., van Renesse, R., & Marsh, M. (2002). Implementing IPv6 as a peer-to-peer overlay network. In *Proceedings of the 21st IEEE Symposium on Reliable Distributed Systems*. (p. 347-351). doi: 10.1109/RELDIS.2002.1180208