# END-TO-END PERFORMANCE MEASUREMENTS FOR OVERLAY FLOW ENGINEERING IN THE INTERNET

By

Salim Ammir B. Mohamed

#### A THESIS

Submitted to Michigan State University in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

**Electrical Engineering** 

2012

### ABSTRACT

#### END-TO-END PERFORMANCE MEASUREMENTS FOR OVERLAY FLOW ENGINEERING IN THE INTERNET

By

#### Salim Ammir B. Mohamed

The heterogeneous and scalable best-effort IP routing forces packets to be forwarded in a manner that is insensitive to delay, loss and available bandwidth in specific sections of the Internet. This thesis presents short and long-term analysis of a targeted high-performance overlay flow-engineering framework through specific sections of the Internet. Relying on redundancy of undiscovered disjointed routes between end-hosts, this thesis devolves intensive characterization for 607,932 long-haul host-to-host routes in terms of delay, loss and bandwidth. In addition to per metric analysis, we used three benchmarks: traceroute, ping and iperf to construct new type of route examining and comparison of four distinct active and real Planetlab experiments in order to evaluate Internet performance, stability, symmetry and the significance of overlay routing. In our thesis, we developed the first component of the proposed overlay flow-engineering framework.

Based on the above experimental study, it was observed that for a large number of Internet flows there exist overlay routes, which can provide better delay, loss and available bandwidth compared to the default routes defined by the best-effort IP routing. Such overlay routes were identified by analyzing the measured host-to-host flow performance. An important implication of this observation is that it is possible to improve Internet flow performance by the way of overlay flow-engineering, which can ensure packet routing through high-performance sections of the Internet in a dynamic manner. Copyright by Salim Ammir B. Mohamed 2012 To my Mother, Father, Sister, and Brothers whose support is invaluable throughout the toughest time we pass in.

To everyone who looks toward us with respect and appreciation.

To the memory of sincere and brave relatives and friends whose smiles still linger in mind, Omar and others.

## ACKNOWLEDGMENTS

At the time I wish all superiority and creativity to my advisor Prof. Subir Biswas, I express my heartfelt gratitude to him for his enthusiasm, inspiration and great efforts in clarification, encouragement, sound advice, guidance and company.

# TABLE OF CONTENTS

LIST	OF TABLES	xiii
LIST	OF FIGURES	xiv
CHAI	PTER 1	1
INTR	ODUCTION	1
1.	Overview	1
2.	Testbeds	2
	2.1 Emulab	2
	2.2 Planetlab	3
3.	Active-Passive Measurements	5
4.	Data Sampling	5
5.	Second Order Metrics	5
6.	Overlay Routing	7
7.	Contribution	8
	7.1 Measurements Perspective	8
	7.2 Results Perspective	9
8.	Summary	9
CHA	PTER 2	11
RELA	ATED WORK AND MOTIVATION	11
1.	Overview	11
2.	Related Work	11
3.	Motivation	14

	3.1	The Objective	15
	3.2	The Missing Stage	16
	3.3	Measurement Stability	17
4.	Sumn	nary	17
CHAI	PTER 3		18
MOD	ELS AI	ND TOOLS	18
1.	Overv	iew	18
2.	Probi	ng Models	18
	2.1	Source-Based Model	19
	2.2	Destination-Based Model	20
3.	Delay	Estimation	20
	3.1	Delay Models	21
	3.2	Delay Tools	22
4.	Loss ]	Estimation	23
	4.1	Loss Tools	24
5.	Bandy	width Estimation	25
	5.1	One-Packet Model	25
	5.2	Packet-Pair Model	27
	5.3	Multi-Packet Model	30
	5.4	Bandwidth Tools	30
6.	Sumn	nary	36
CHAI	PTER 4		37
METH	HODOI	LOGY	37
1.	Overv	view	37

2.	Probin	ng Utilities	37
3.	Probir	ng Accuracy	37
	3.1	Utilized Probing Schemes	38
	3.2	Probability of Collision	39
	3.3	Probing Algorithm	41
	3.4	Data Extraction and Analysis	43
	3.5	Validation	46
4.	Summ	nary	46
СНАР	TER 5		47
PACK	ET DE	LAY	47
1.	Overv	iew	47
2.	The D	Pelay Components	48
3.	The D	Pelay Experiments	49
4.	Tracer	route Analysis	50
	4.1.	Indirect Length Statistics	51
		4.1.1 Definition	51
		4.1.2 Result	51
	4.2	Indirect Length Stability	54
		4.2.1 Definition	54
		4.2.2 Result	54
	4.3	Indirect Link Utilization	61
5.	Ping A	Analysis	64
	5.1	Direct Delay Symmetry	64
		5.1.1 Definition	64

		5.1.2	Procedure	65
		5.1.3	Result	66
	5.2	Indire	ct Delay Symmetry	69
	5.3	Short	and Long-Term Route Symmetries	71
		5.3.1	Definition	71
		5.3.2	Procedure	71
		5.3.3	Result	72
	5.4	Indire	ct Length Statistics	81
	5.5	Direct	Delay and Distance	89
		5.5.1	Definition	89
		5.5.2	Procedure	90
		5.5.3	Result	91
	5.6	Indire	ct Delay and Distance	93
	5.7	The Si	gnificance of Indirect Delay Routing	96
		5.7.1	Overview	96
		5.7.2	Procedure	96
		5.7.3	Result	96
6.	Summ	nary		106
CHAI	PTER 6			107
PACK	KET LO	SS		107
1.	Overv	iew		107
2.	The Ir	nportan	ce of Measuring Loss	107
3.	The L	oss Exp	eriments	108
4.	Ping A	Analysis		109

4.1.	Direct	Loss Symmetry	110
	4.1.1	Definition	110
	4.1.2	Procedure	110
	4.1.3	Result	110
4.2	Indirec	ct Loss Symmetry	113
4.3	Short a	and Long-Term Route Symmetries	115
	4.3.1	Definition	115
	4.3.2	Result	115
4.4	Indirec	ct Length Statistics	123
	4.4.1	Definition	123
	4.4.2	Result	123
4.5	Indirec	ct Delay and Loss	128
	4.5.1	Definition	128
	4.5.2	Result	128
Iperf	Analysis		131
5.1	Short a	and Long-Term Route Symmetries	131
	5.1.1	Definition	131
	5.1.2	Result	132
5.2	Direct	Loss and Distance	137
5.3	Indirec	ct Loss and Distance	144
5.4	The Si	gnificance of Indirect Loss Routing	150
	5.4.1	Definition	150
	5.4.2	Result	151
Sumr	nary		156

5.

6.

CHAF	PTER 7		157
BANI	OWIDT	Н	157
1.	Overv	iew	157
2.	Bottle	neck and Available Bandwidths	158
3.	What'	s Missing?	159
4.	Bandv	vidth Impediments	160
5.	Achie	vable Bandwidth	160
	5.1	TCP Throughput	161
		5.1.1 Throughput Stability	164
		5.1.2 Direct Throughput Statistics	167
		5.1.3 Throughput and FTT	170
6.	UDP A	Available Bandwidth	173
	6.1	Practical Approaches	173
	6.2	The Affordable Transmission Rate and Loss	174
	6.3	The Horizontal Available Bandwidth	175
		6.3.1 Definition	175
		6.3.2 Result	176
	6.4	The Vertical Available Bandwidth	177
		6.4.1 Definition	177
		6.4.2 Result	178
	6.5	The Horizontal Available Bandwidth Symmetry	181
	6.6	The Maximum <i>AfTR</i> and Loss	182
	6.7	The Horizontal Available Bandwidth and Jitter	187
	6.8	The Short-Term AfTR and Loss	189

7.	The P	ing Bulk Transfer Rate	192
8.	Summ	ary	195
CHAF	PTER 8		196
CON	CLUSIC	ON AND FUTURE WORK	196
1.	Concl	usion	196
	1.1	Packet Delay Analysis	196
	1.2	Packet Loss Analysis	196
	1.3	Bandwidth Analysis	197
2.	Future	Work	197
APPE	NDICE	S	201
APPE	NDIX I		202
APPE	NDIX I	I	210
APPE	NDIX I	П	219
APPE	NDIX I	V	227
BIBL	IOGRA	РНҮ	235

## LIST OF TABLES

3.1	The Source-Based Model Advantages and Disadvantages	20
3.2	The Destination-Based Model Advantages and Disadvantages	20
3.3	The Definitions of Bandwidth Parameters	25
3.4	The Bottleneck Bandwidth Scenarios	28
4.1	The Parameters of the Probing Schemes	38
4.2	The Parameters of the Probing Algorithm	42
4.3	The Statistics of the Probing Algorithm	43
6.1	The Ordered Probing Rates	109
Ι	Traceroute Experiment	202
II	Ping Experiment	210
III	Iperf-TCP Experiment	219
IV	Iperf-UDP Experiment	227

# **LIST OF FIGURES**

1.1	The Planetlab Terminology	4
1.2	The Abstraction of Communication Overlay	7
1.3	The RON Overlay Inefficiency	8
2.1	The General Abstraction of the Study	16
3.1	The Source-Based Model	19
3.2	The Destination-Based Model	19
3.3	The One-Packet Delay Model Diagram	26
3.4	The Packet-Pair Model Diagram	28
3.5	Unloaded Route Scenario	29
3.6	Cross Traffic Scenario	29
3.7	Multiple Queues Scenario	29
4.1	The State Diagram of the Probing Algorithm	41
5.1	The First Run Length Statistics	52
5.2	The Second Run Length Statistics	52
5.3	The Third Run Length Statistics	53
5.4	The Forth Run Length Statistics	53
5.5	The Stability of Two Hops Length	54
5.6	The Stability of Three Hops Length	55
5.7	The Stability of Four Hops Length	55
5.8	The Stability of Five Hops Length	56
5.9	The Stability of Six Hops Length	56

5.10	The Stability of Seven Hops Length	57
5.11	The Stability of Eight Hops Length	57
5.12	The Stability of Nine Hops Length	58
5.13	The Stability of Ten Hops Length	58
5.14	The First Run Lengths Histogram	59
5.15	The Second Run Lengths Histogram	60
5.16	The Third Run Lengths Histogram	60
5.17	The Forth Run Lengths Histogram	61
5.18	The First Run Link Utilization	62
5.19	The Second Run Link Utilization	62
5.20	The Third Run Link Utilization	63
5.21	The Forth Run Link Utilization	63
5.22	Ping Runs Timesheet	64
5.23	Direct Delay-Symmetry Example	66
5.24	The 0.05MByte Runs Direct DSY Stability	67
5.25	The 0.1MByte Runs Direct DSY Stability	67
5.26	The 0.25MByte Runs Direct DSY Stability	68
5.27	The 0.5MByteRuns Direct DSY Stability	68
5.28	The 0.05MByte Runs Indirect DSY Stability	69
5.29	The 0.1MByte Runs Indirect DSY Stability	70
5.30	The 0.25MByte Runs Indirect DSY Stability	70
5.31	The 0.5MByte Runs Indirect DSY Stability	71
5.32	The Terminologies of Route Symmetry	72

5.33	The First 0.05MByte Run Short-Term Indirect RSY	73
5.34	The Second 0.05MByte Run Short-Term Indirect RSY	74
5.35	The First 0.1MByte Run Short-Term Indirect RSY	74
5.36	The Second 0.1MByte Run Short-Term Indirect RSY	75
5.37	The First 0.25MByte Run Short-Term Indirect RSY	75
5.38	The Second 0.25MByte Run Short-Term Indirect RSY	76
5.39	The First 0.5MByte Run Short-Term Indirect RSY	76
5.40	The Second 0.5MByte Run Short-Term Indirect RSY	77
5.41	The First 0.05-0.1MByte Runs Long-Term Indirect RSY	78
5.42	The First 0.05-0.25MByte Runs Long-Term Indirect RSY	78
5.43	The First 0.05-0.5MByte Runs Long-Term Indirect RSY	79
5.44	The First 0.1-0.25MByte Runs Long-Term Indirect RSY	79
5.45	The First 0.1-0.5MByte Runs Long-Term Indirect RSY	80
5.46	The First 0.25-0.5MByte Runs Long-Term Indirect RSY	80
5.47	The First 0.05MByte Run Length Statistics	81
5.48	The Second 0.05MByte Run Length Statistics	82
5.49	The Third 0.05MByte Run Length Statistics	82
5.50	The Forth 0.05MByte Run Length Statistics	83
5.51	The First 0.1MByte Run Length Statistics	83
5.52	The Second 0.1MByte Run Length Statistics	84
5.53	The Third 0.1MByte Run Length Statistics	84
5.54	The Forth 0.1MByte Run Length Statistics	85
5.55	The First 0.25MByte Run Length Statistics	85

5.56	The Second 0.25MByte Run Length Statistics	86
5.57	The Third 0.25MByte Run Length Statistics	86
5.58	The Forth 0.25MByte Run Length Statistics	87
5.59	The First 0.5MByte Run Length Statistics	87
5.60	The Second 0.5MByte Run Length Statistics	88
5.61	The Third 0.5MByte Run Length Statistics	88
5.62	The Forth 0.5MByte Run Length Statistics	89
5.63	Delay and Distance Matching	90
5.64	The First 0.05MByte Run Direct Delay and Distance	91
5.65	The First 0.1MByte Run Direct Delay and Distance	92
5.66	The First 0.25MByte Run Direct Delay and Distance	92
5.67	The First 0.5MByte Run Direct Delay and Distance	93
5.68	The First 0.05MByte Run Indirect Delay and Distance	94
5.69	The First 0.1MByte Run Indirect Delay and Distance	94
5.70	The First 0.25MByte Run Indirect Delay and Distance	95
5.71	The First 0.5MByte Run Indirect Delay and Distance	95
5.72-A	The 0.05MByte Runs Routing Probabilities	97
5.72-В	The 0.05MByte Runs Routing Probabilities	98
5.72-C	The 0.05MByte Runs Routing Reachability	98
5.73-A	The 0.1MByte Runs Routing Probabilities	99
5.73-B	The 0.1MByte Runs Routing Probabilities	99
5.73-C	The 0.1MByte Runs Routing Reachability	100
5.74-A	The 0.25MByte Runs Routing Probabilities	100
5.74-B	The 0.25MByte Runs Routing Probabilities	101

5.74-C	The 0.25MByte Runs Routing Reachability	101
5.75-A	The 0.5MByte Runs Routing Probabilities	102
5.75-B	The 0.5MByte Runs Routing Probabilities	102
5.75-C	The 0.5MByte Runs Routing Reachability	103
5.76	The 0.05MByte Runs Total Routing Probabilities	104
5.77	The 0.1MByte Runs Total Routing Probabilities	104
5.78	The 0.25MByte Runs Total Routing Probabilities	105
5.79	The 0.5MByte Runs Total Routing Probabilities	105
6.1	The 0.05MByte Run Direct LSY Stability	111
6.2	The 0.1MByte Run Direct LSY Stability	111
6.3	The 0.25MByte Run Direct LSY Stability	112
6.4	The 0.5MByte Run Direct LSY Stability	112
6.5	The 0.05MByte Run Indirect LSY Stability	113
6.6	The 0.1MByte Run Indirect LSY Stability	114
6.7	The 0.25MByte Run Indirect <i>LSY</i> Stability	114
6.8	The 0.5MByte Run Indirect LSY Stability	115
6.9	The First 0.05MByte Run Short-Term Indirect RSY	116
6.10	The Second 0.05MByte Run Short-Term Indirect RSY	117
6.11	The First 0.1MByte Run Short-Term Indirect RSY	117
6.12	The Second 0.1MByte Run Short-Term Indirect RSY	118
6.13	The First 0.25MByte Run Short-Term Indirect RSY	118
6.14	The Second 0.25MByte Run Short-Term Indirect RSY	119
6.15	The First 0.5MByte Run Short-Term Indirect RSY	119

6.16	The Second 0.5MByte Run Short-Term Indirect RSY	120
6.17	The First 0.05-0.1MByte Runs Long-Term Indirect RSY	120
6.18	The First 0.05-0.25MByte Runs Long-Term Indirect RSY	121
6.19	The First 0.05-0.5MByte Runs Long-Term Indirect RSY	121
6.20	The First 0.1-0.25MByte Runs Long-Term Indirect RSY	122
6.21	The First 0.1-0.5MByte Runs Long-Term Indirect RSY	122
6.22	The First 0.25-0.5MByte Runs Long-Term Indirect RSY	123
6.23	The First 0.05MByte Run Length Statistics	124
6.24	The Second 0.05MByte Run Length Statistics	125
6.25	The First 0.1MByte Run Length Statistics	125
6.26	The Second 0.1MByte Run Length Statistics	126
6.27	The First 0.25MByte Run Length Statistics	126
6.28	The Second 0.25MByte Run Length Statistics	127
6.29	The First 0.5MByte Run Length Statistics	127
6.30	The Second 0.5MByte Run Length Statistics	128
6.31	The First 0.05MByte Run Direct and Indirect Loss	129
6.32	The First 0.1MByte Run Direct and Indirect Loss	130
6.33	The First 0.25MByte Run Direct and Indirect Loss	130
6.34	The First 0.5MByte Run Direct and Indirect Loss	131
6.35	The 0.5Mbps Run Short-Term Indirect RSY	132
6.36	The 4Mbps Run Short-Term Indirect RSY	133
6.37	The 80Mbps Run Short-Term Indirect RSY	133
6.38	The 400Mbps Run Short-Term Indirect RSY	134

6.39	The 0.5-4Mbps Runs Long-Term Indirect RSY	134
6.40	The 0.5-80Mbps Runs Long-Term Indirect RSY	135
6.41	The 0.5-400Mbps Runs Long-Term Indirect RSY	135
6.42	The 4-80Mbps Runs Long-Term Indirect RSY	136
6.43	The 4-400Mbps Runs Long-Term Indirect RSY	136
6.44	The 80-400Mbps Runs Long-Term Indirect RSY	137
6.45	The 0.5Mbps Run Direct Loss and Distance	138
6.46	The 4Mbps Run Direct Loss and Distance	138
6.47	The 6Mbps Run Direct Loss and Distance	139
6.48	The 8Mbps Run Direct Loss and Distance	139
6.49	The 10Mbps Run Direct Loss and Distance	140
6.50	The 20Mbps Run Direct Loss and Distance	140
6.51	The 40Mbps Run Direct Loss and Distance	141
6.52	The 80Mbps Run Direct Loss and Distance	141
6.53	The 100Mbps Run Direct Loss and Distance	142
6.54	The 200Mbps Run Direct Loss and Distance	142
6.55	The 400Mbps Run Direct Loss and Distance	143
6.56	The 800Mbps Run Direct Loss and Distance	143
6.57	The 0.5Mbps Run Indirect Loss and Distance	144
6.58	The 4Mbps Run Indirect Loss and Distance	145
6.59	The 6Mbps Run Indirect Loss and Distance	145
6.60	The 8Mbps Run Indirect Loss and Distance	146
6.61	The 10Mbps Run Indirect Loss and Distance	146

6.62	The 20Mbps Run Indirect Loss and Distance	147
6.63	The 40Mbps Run Indirect Loss and Distance	147
6.64	The 80Mbps Run Indirect Loss and Distance	148
6.65	The 100Mbps Run Indirect Loss and Distance	148
6.66	The 200Mbps Run Indirect Loss and Distance	149
6.67	The 400Mbps Run Indirect Loss and Distance	149
6.68	The 800Mbps Run Indirect Loss and Distance	150
6.69-A	The 0.5-8Mbps Runs Routing Probabilities	151
6.69-B	The 0.5-8Mbps Runs Routing Probabilities	152
6.70-A	The 10-80Mbps Runs Routing Probabilities	152
6.70-B	The 10-80Mbps Runs Routing Probabilities	153
6.71-A	The 100-800Mbps Run Routing Probabilities	153
6.71-B	The 100-800MbpsRuns Routing Probabilities	154
6.72	The 0.5-8Mbps Runs Total Routing Probabilities	155
6.73	The 10-80Mbps Runs Total Routing Probabilities	155
6.74	The 100-800Mbps Runs Total Routing Probabilities	156
7.1	The Bandwidth Terminology	158
7.2	The First Run Throughput	165
7.3	The Second Run Throughput	166
7.4	The Third Run Throughput	166
7.5	The Forth Run Throughput	167
7.6	The First Run Statistics	168
7.7	The Second Run Statistics	168
7.8	The Third Run Statistics	169

7.9	The Forth Run Statistics	169
7.10	The First Throughput and FTT	171
7.11	The Second Run Throughput and FTT	171
7.12	The Third Run Throughput and FTT	172
7.13	The Forth Run Throughput and FTT	172
7.14	The Number of Routes Suffer High Loss	175
7.15	The Estimated Horizontal Available Bandwidth	177
7.16	The 40Mbps Run $VAvB$ and $D_{VAvB}$	178
7.17	The 80Mbps Run $VAvB$ and $D_{VAvB}$	179
7.18	The 100Mbps Run $VAvB$ and $D_{VAvB}$	179
7.19	The 200Mbps Run $VAvB$ and $D_{VAvB}$	180
7.20	The 400Mbps Run $VAvB$ and $D_{VAvB}$	180
7.21	The 800Mbps Run $VAvB$ and $D_{VAvB}$	181
7.22	The <i>HAvB</i> Degree of Symmetry	182
7.23	The First 400Mbps Run Maximum AfTR and Packet Loss	183
7.24	The Second 400Mbps Run Maximum AfTR and Packet Loss	184
7.25	The First 800Mbps Run Maximum AfTR and Packet Loss	184
7.26	The Second 800Mbps Run Maximum AfTR and Packet Loss	185
7.27	The First 400Mbps Run AfTR Full View	185
7.28	The Second 400Mbps Run AfTR Full View	186
7.29	The First 800Mbps Run AfTR Full View	186
7.30	The Second 800Mbps Run AfTR Full View	187
7.31	The Direct and Indirect Jitter Minimum HAvBs	188

7.32	The Direct-Indirect Jitter and Loss Minimum HAvBs	189
7.33	The 800Mbps Run Direct-Indirect AfTR and Loss	190
7.34	The 400Mbps Run Direct-Indirect AfTR and Loss	190
7.35	The 200Mbps Run Direct-Indirect <i>AfTR</i> and Loss	191
7.36	The 100Mbps Run Direct-Indirect AfTR and Loss	191
7.37	The First 0.05MByte Run BTR and Loss	193
7.38	The Second 0.1MByte Run BTR and Loss	193
7.39	The Third 0.25MByte Run BTR and Loss	194
7.40	The Forth 0.5MByte Run BTR and Loss	194

### **CHAPTER 1**

#### **INTRODUCTION**

#### 1. Overview

Performing measurements on packet-switching systems is a fundamental challenge because of the shared infrastructure required by the complex design of scalable and cost-effective networks. The dynamic behavior of each element on a network is a result of sharing resources over short or long time-scales. Resulting congestion and reducing availability (i.e., a fraction of time, by which a service is reachable and functioning) and reliability are unfavorable implications of that sharing. The quality of routing depends on the pace in adjusting policies in response these implications. This chapter introduces some studied second order metrics and our method and motivation behind conducting the study.

Congestion oftentimes is considered as difficult to be eliminated perfectly because of the unpredictability of cross-traffics and network abilities. Delay and packet loss are common symptoms of congestion, and usually caused by bandwidth being maxed out at a router. The QoS in congested networks is a function of routing performance. The median path failure time is about 3 minutes and over 90% of failures last below 15 minutes to be resolved [27] such as with PGB while interior routing like OSPF spends about 10 seconds. There are strategies to avoid degrading Internet performance either by understanding first-order metrics such as how often and long in order to restrict causes or instead by analyzing second-order metrics to improve routing directly or indirectly.

The lack and inefficiency in precise diagnosing of second-order statistics causes researchers who study Internet routing to rely oftentimes on an assumption that considers network shortcomings as independent to address many QoS queries: the significance and speed in reacting with failures and stability of enhancements. The research presented in this thesis sheds closely the light on clarifying such queries, and relies heavily upon extensive measurements of Internet paths on long-haul and large-scale Planetlab networks. We investigated accuracy and efficiency of available tools outlined in chapter [3]. Traceroute, ping and iperf were utilized to measure three important metrics: delay, packet loss and available bandwidth, and assist common properties inherited among them. This chapter introduces utilized testbeds in section [2] and measurements classification in [3]. The studied metrics are outlined in section [5]. The terminology of overlay routing is explained in section [6]. Further we discuss our contribution in section [7] before summarizing in section [8].

#### 2. Testbeds

In networking, there are many available testbeds used for different purposes. Emulab has smooth interaction in between a user and pre designed modality type of emulated network. The Planetlab as an overlay testbed is more realistic than Emulab as a collection of universal distributed machines. The Emulated testbed provides more stability easiness in controlling utilized environments while Planetlab has less flexibility and guarantee in conforming identical experiments.

#### 2.1 Emulab

The Utah's Emulab provides sharing of resources among multiple concurrent experiments when enough nodes are available [23]. Emulab limits networks to 100 nodes [1] with arbitrary and artificial predetermined delays and bandwidths. Users via combination of Network Simulator (NS2) and web interface can specify structural experiment before swapping in. The Emulab process of swapping in to a new and real experiment consists of several steps: mapping the simulated network topology onto available nodes and switches, configuring VLANs on the switches to connect nodes into the network, installing an initial kernel and root file-system and lastly loading and running the chosen operating systems [23].

The user can access via Secure Shell (SHH) all nodes returned after swapping. The established VLANs allow multiple services such as wiping disks, changing operating systems and reshaping all simulated bandwidth-limited and loss-links. Despite the high degree of flexibility in control and security, Emulab was not suitable for our research due to the nature of emulated traffic and disappearance of full and real mapping of routes beneath VLANs. In addition, users cannot force involving particular nodes to be in their experiments.

#### 2.2 Planetlab

In contrary to Emulab, Planetlab affords an immediate practical approach to high-end clusters spread over globally distributed network, from which a user can select sites to construct experiments. Planetlab uses the following terminologies while setting an experiment:

- Site is a real location where Planetlab nodes are placed (e.g., MIT and MSU).
- Node is a dedicated machine that runs Planetlab services.
- Slice is an assigned space to contain a number of allocated nodes and resources distributed across Planetlab [24].
- Sliver is a slice runs on specific node, to which user can login via Secure Shell (SSH).

Figure [1.1] shows the above definitions more clearly. Planetlab approximately composed of more than 1000 nodes located in 40 countries [24]. Planetlab restricts only slices lifetime that must be constantly renewed. Every participating node is publicly at a minimum access to the Planetlab Central (PLC) management node through HTTPS and PLCAPI calls [25].



Figure [1.1]: The Planetlab Terminology (For interpretation of the references to color in this and all other figures, the reader is referred to the electronic version of this thesis).

The fundamental aim of Planetlab as a centrally managed collection overlay platform is to provide long-term service deployment and autonomous machines. Planetlab runs on Fedora Core 2 (FC2) operating system. The PLC supplies three functions: database for storing system state, web interface for management and RPM repository for bootstrapping and distributing software [25]. The web interface offers a readable and an XMLRPC interfaces called PLCAPI for internal usage [25]. Unfortunately, we did not directly gain access to Planetlab and visual web applications for monitoring and controlling tools like "MyPLC" that can be utilized to facilitate managing the most insurmountable problem "stability". Swapping in experiments indirectly is different than Emulab. For that, we used an application called "Omni" that allows users to access Planetlab users via Emulab accounts. Using Omni, a user can create a slice and sliver via simple and distinct Python scripts.

#### 3. Active-Passive Measurements

Networked systems can be monitored in either fashion active or passive. The active measuring indicates that a given utility is in active session of calculating metric while sending probes into the network. Unlikely, passive measuring is performed by copping portions of traffic without interfering via splitters, duplicating hubs, monitoring buffers or sniffers (e.g., tcpdump). Practically, active measurements are more reliable than passive measurements as the time gab between extracting and analyzing samples is quite small. The cost of passive monitoring oftentimes is much higher particularly as the network scales up to tens of links.

#### 4. Data Sampling

The term "measurement" throughout the study is defined by an operation of functioning traceroute, ping or iperf to assess used networks. In each experiment, hosts consecutively probe directories of domains "overlay neighbors" with some restrictions outlined in chapter [4]. These measurements are considered independently and exponentially spaced samples as suggested in [2]. Studies like [29] recommend the use of independent and exponential distributed time intervals to gain equally likely unbiased sampling of instantaneous signal values that have "Poisson" distribution in order to avoid self-synchronization. We followed the exact assumption in [30] that states an average time elapsed in waiting certain measurement is equal to the Internet time in that state.

#### 5. Second Order Metrics

Internet traffics carry associate dynamic and static parameters that describe their nature. Usually, transporting protocol's header values are momentarily constant for a period of time like an originator and receiver IP addresses and a type of flow. However, the challenge for an IP flow is to sustain acceptable demands of metrics such as delay, packet loss, bandwidth and jitter. These factors frequently have private and shared causes with others (i.e., acting beyond an

5

available bandwidth for instance produces higher delay and loss). These statistics appear in the study are defined as following:

- Delay is usually measured in milliseconds between the moments of sending packet and receiving its acknowledgment. The longer the delay the more difficult a conversation becomes. The International Telecommunication Union (ITU) recommends that one way delay should not exceed 400ms for acceptable speech quality [26].
- Packet loss is measured as a percentage of lost packets that occurs when packets get dropped or delayed beyond acceptable limit. A packet loss above 2% will go largely noticed, and reduces quality.
- Jitter is represented as a variation in milliseconds on packet's transit delay caused by queuing, contention and serialization effects on a route [26]. The higher bandwidth networks tend to have higher capabilities to eliminate jitter [26].
- Bandwidth is measured as the speed of pace, at which IP substrates can communicate in megabits transferred per second.

Many systems represent one-way packet loss as Poisson process with deterministic probability of capturing statistical unbiased samples. The lack of connectivity, sophisticated security and expanded delay are major motivators of packet loss. Characterizing bandwidth is a challenge because of prerequisite estimations about across traffic's interference. Generally, bandwidth is categorized into:

- Bottleneck bandwidth: natural maximum ability to forward data without interference.
- Available bandwidth: remaining capacity for traffic while utilizing a link.
- Throughput or achievable bandwidth: transmission speed that grantees successful delivery.
- Utilized bandwidth: difference between bandwidth capacity and available bandwidth.

#### 6. Overlay Routing

The current IP infrastructure manages conversation ability for hosts to communicate sensibly and reliably under an administration of either interior or exterior BGP protocol. Due to permanent or persistent failures or shortcomings, IP terminals may become partially or completely invisible from each other in terms of QoS metrics. This invisibility has been classified differently by applications. Bulk transfer applications for instance are sensitive to packet delay and loss [27]. Others, such as video streaming may seek to eliminate jitter as possible. The mentioned sort of problems and others brings up an essential need for an extra indirect routing "overlay routing" layer that can contribute in eliminating existed shortcomings.

The overlay routing sees a complete IP connection lies beneath as indirect hop. Figure [1.2] shows an indirect end-to-end abstraction between two IP substrates. There are two classifications of overlay networks: communication such as Resilient Overlay Routing (RON) and data dissemination like "Chord" [27]. The RON traffic is carried inefficiently via intermediate end links that connect overlay hosts to indirect layer as Figure [1.3] demonstrates [27]. The conversation between S and D instead of direct green route, it uses the indirect red route, on which hops 2 and 3 represents inefficiency.



Figure [1.2]: The Abstraction of Communication Overlay



Figure [1.3]: The RON Overlay Inefficiency

#### 7. Contribution

We believe that the work presented in this thesis provides considerable contribution to the field of networking, in particular measurements and routing analysis. This contribution is briefly divided into two sets. The first set is in terms of simple and extensive method in evaluating long and large-scale networks. The second set summarizes importance and stability of our findings.

#### 7.1 Measurements Perspective

The enormous collection of data samples "607,932" powerfully indicates real characteristics of Internet. The used utilities are all well-known and common benchmarks, and were combined of source-based and destination-based models. We argue that using repetitive runs for estimating specific metric profoundly supports and validates our measurements. Fair comparisons between the direct Internet routing and proactive indirect routing always appear throughout the study. Ping was used for the first time to assist packet loss on long and large-scale network, and monitor the BTR of four distinct data loads. A notion of conditional available bandwidth that was determined for 14,460 long-haul connections accurately beside other metrics using iperf. A defined degree of symmetry is used for the first time in this study. The terminology

of route symmetry was highly stable on both short and long-term scales. The stability of matching between geographical distance and both delay and packet loss is quite encouraging to be extended toward Internet distance analysis in future research.

#### 7.2 **Results Perspective**

We studied the most important three metrics in networking simultaneously oftentimes, and identified some shared properties among them. The majority of relations and analysis presented in this thesis were introduced for the first time as we aware of diverse combination of related studies. Our measurements were conducted using dissimilar experiments over wide time ranges. Some of our findings were perfectly powerful like the significance of performing indirect loss routing while increasing transmission speed. Other discoveries were relatively strong such as achieved enhancements via indirect packet delay routing. The disclosed stabilities of all studied behaviors were highly acceptable and encouraging. During each short-term analysis, we found that the utilized lengths<sup>1</sup> of indirect routes are always in a monotonic order regardless the network size; the maximum achieved length was 10 hops. Regarding traceroute, 15.5% of indirect routing has the ability to reduce delay by 100ms or more while 30% of routing decreased delay by less than 100ms. With iperf, however, eliminating packet loss rises up linearly up to 62% with high transmission rates.

#### 8. Summary

The QoS as an indicator of network performance is normally constrained via three different metrics: delay, packet loss and bandwidth. The effort introduced in this thesis is an attempt to discover our motivated belief about missing keys out of Internet measurements and their consequence findings such as the stability and significance of improvement on direct and indirect routing. The thesis will discuss related studies and motivation in chapter [2] while available tools and models in literature are presented in chapter [3]. The experimental

methodology of the study will appear in chapter [4]. The packet delay and related properties are in chapter [5]. Chapter [6] will address packet loss along with similar properties defined with packet delay. Chapter [7] will explore extensively an analysis of available bandwidth and related characteristics on both short and long time-scales.

### **CHAPTER 2**

#### **RELATED WORK AND MOTIVATION**

#### 1. Overview

The work presented in this thesis was built upon a broad of related research on Internet measurements. The scope of the majority of related studies was either on single metric behavior using the famous TCP and UDP protocols. Despite that, we have not come across studies that have clear and intensive connection between measurements and indirect routing. The following explanation in section [2] presents several studies conducted on Internet measurements and indirect routing. The study's motivation will be explained in section [3].

#### 2. Related Work

The study in [1], has an extensive analysis on the TCP behavior over Internet, and explored routing pathologies. The traceroute was used to extract routing properties such as symmetry and loops. They analyzed packet delay and loss and available bandwidth of TCP flows by monitoring different characteristics such as packet retransmission and reordering. The experimental results of the study were passively analyzed. In [2], researchers developed an overlay system called Resilient Overlay network (RON) in order to improve service availability by using delay, loss rate and available bandwidth data depository to guide RON. This system was able to discover Internet outages and provide alternative paths in 50% of failures. The design motivation is in exploiting redundancy of underlay routing. In [3], a proposed simple strategy for combining delay, jitter, packet loss rate and availability as QoS metrics into single cost function is used by Dijkstra's algorithm to determine indirect routing. The study calculated shortest routes over multi-domain routing network that has available information about domain reachability via service oriented architecture paradigm. The study shows improvement in QoS comparing to common methods.

K. Lai and M. Baker in [4] described a deterministic model of packet delay by deriving a technique referred as "packet tailgating" to assist link bandwidths. In [5] researchers developed tool called "ABwE" to monitor available bandwidth below 1Gbps based on a packet dispersion model. The traceroute developer V. Jacobson published "pathchar" in [6] as delay, loss and bandwidth estimator of every hop between source-destination pair. The black hole of this utility is time consumption. In [7], G. Jin and B. Tierney differentiated between bandwidth and throughput, and defined another new concept called Maximum Burst Size (MBS) in designing "netest" for estimating with an extinct of accuracy available bandwidth and throughput.

In [8], N. Hu, E. Li, Z. Mao, P. Steenkiste and J. Wang developed an interesting light-load tool identified as "pathneck". This tool is used to locate congested links over an end-to-end route. They used a Recursive Packet Train (RPT) methodology to measure the time-gap between every two responses per link via a pair of packets that has same Time to Live (TTL) values. The active and passive network measurement survey in [9] was conducted to understand differences between passive and active probing in detecting Internet characteristics like delay and loss. In [10], A. Johnson, B. Melander and M. Bjorkman designed a utility "DietTopp" to measure available bandwidth and bottleneck bandwidth on Internet. They compared the performance of estimation with pathload and pathrate that are more accurate but immoderate in consuming bandwidth and time.

Researchers in [11] presented "CapProbe" used in estimating bottleneck bandwidth between end hosts. The tool utilizes a packet delay dispersion strategy, and filters distorted information by interfered traffic. The accuracy of CapProbe was calibrated with pathrate and pathchar. An exponential flight pattern of probing packets named "chrips" is mixed up in designing "pathchirp" for calculating available bandwidth. Basically pathchirp increases probing rate per each chirp to obtain inter-arrival times before evaluating available bandwidth as presented in [12]. The "RPT" illustrated in [13] is almost similar in objective to pathneck in [8] form the point of detecting congested routers between end hosts. The methodology is simply by causing intermediate routers to drop an identical pair of packets out of the recursive train sequentially to calculate the time gap between their responses. Having cross-traffic will alter the train's length, and thus results wide time space from the router.

The study in [14] proposed flexible system "scriptrute" that allows third party remote terminal to conduct network measurements and debug on behalf on ordinary users. The package composed of many tools such as "sr-ally: alias resolver", "sr-sprobe: bandwidth estimator" and sr-rockettrace: robust traceroute". In [15], J. Strauss, D. Katabi and F. Kaashoek presented "spruce" as light utility for assisting available bandwidth using the packet-pair model presented in section [5.2] of chapter [3]. The "STAB" presented in [16] combines two probing concepts: the packet tailgating in [4] and chirps in [12] in a novel fashion to locate tight links using light traffic.

The study in [17] argued that existing mechanisms utilized in determining available bandwidth are more heuristic, unintuitive and inconsistent with any network model. Because of disturbance in network measurements, the researcher introduced a different calculation of available bandwidth using packet dispersion observation after improving byte-over-time calculation in "cprobe" studied in [20] to capture and deduct sampling errors introduced by the primitive design of cprobe. In [18], M. Jain and C. Dovrol presented "pathload" to evaluate available bandwidth based on monitoring increments on one-way delays of streams when transmission rate exceeds available bandwidth.

In [19], "pchar" as similar version of pathchar does link analysis while transmitting UDP packets and waiting for their ICMP responses. Controlling the TTL value allows pchar to
examine links respectively while alerting packet size contributes in determining delay and bandwidth. Packet duplication is used for estimating packet loss and queuing delay. The cprobe outlined in [20] uses unfair queuing for extracting available bandwidth measurements between end hosts. This calculation is performed simply using a train of ICMPs. When receiving the last echo, cprobe will calculate the ratio between the train size and arrival time difference between the initial and final echoes (i.e., an out-of-date utility called "pipechar" barely uses identical approach).

The study in [21] was aimed toward improving video perceived quality through a dynamic route selection (i.e., overlay routing). The distinguishing between routes was based on metrics like: available bandwidth, jitter and loss rate. The substantial result in this study was in showing that the most effective technique for overlay routing relies on an approximate estimation of the lower-bound of the available bandwidth variation range. In [22], the researchers present "sprobe" to measure bottleneck bandwidth of uncooperative networks by exploiting properties of TCP similarly to "sting". The core disadvantage of sprobe is a requirement of supper-root privileges in order to change firewall settings.

# 3. Motivation

We established our idea to investigate Internet routing over long and large-scale connections. The Internet graph topology partially offers numerous of partially disjoint paths between end terminals [1], and the field is still open for benefiting out of this available redundancy. However, several significant challenges such as the following can limit the usefulness of Internet redundancy.

- Since Internet exposes only single path to end-hosts, how can a failure masking system access additional paths? [1].
- How significant an alternative indirect route can be?

- How fast can the failure masking system respond to failures between *n* participants? For large *n*, the general solution scales up to  $O(n^2)$  [1].
- What metrics can benefit the route selection algorithm?
- What is degree of modification caused by an indirect routing on the existing Internet?

# 3.1 The Objective

The core aim of this study is to investigate whether or not a second effective indirect large-scale routing layer can be placed on top of the current Internet to enhance performance. We argue that both heterogeneity and scalability result an oscillatory in network utilization, delay in response and traffic characterizing among routers, and substitutional routes are always exist for each connection that carries traffic among source-destination pair. Implementing that layer can enhance performance in terms of the three metrics under concern in order to and minimize Internet growth's resultant side-effects. We believe that our study has extensively, accurately and carefully formed an extraordinary watch many interesting properties in Internet routing simultaneously and successively. Figure [2.1] shows a general abstraction of this study. We tried to shed the light on specific concerns: the usefulness of indirect routing, frequency of using direct and indirect routes, stability of virtual routing and shared relations among metrics.



Figure [2.1]: The General Abstraction of the Study

# 3.2 The Missing Stage

Although we have find that a considerable and sometimes high percentage of the current direct Internet routes have intention to operate virtually while maintaining common Internet characteristics and providing better performance, further and extensive investigation is required to discover some indirect drawbacks like the over-heading. Unfortunately, throughout this study we did not actively and effectively examine the influence of the extra overhead addressing on the performance of indirect routing due the shortness of the research time (i.e., designing comprehensive overlay routing protocol is a time consuming task).

## **3.3** Measurements Stability

One of the most essential properties of indirect routing is stability that relies on the steadiness of the under-laying Internet. For each metric of the studied, we performed the stability assessment on each relation or new-defined property simply by repeating probing over the period of measuring Planetlab four times in most cases (e.g., the traceroute average time interval was 6 hours and 2 hours for ping). Using ping, we reiterate each experiment run that was assigned for particular data size four times. In contrary, with iperf we only reiterated experiments of the two highest transmission rates. In fact, any data forwarder has transmission speed limitation, and because of that we consider the majority of experiments that were conducted by iperf using rates beyond maximum affordable rates as repetitive runs. The placed simplification was due to the lack of stability of Planetlab in maintaining the same physical structure of our experiment for quite long time.

# 4. Summary

The chapter summarized previous studies on packet measurements. The study was motivated by an amount of temporal alternatives offered by Internet in order to examine their values for overlay routing.

# **CHAPTER 3**

# MODELS AND TOOLS

#### 1. Overview

Over years, researchers have been in process of devolving different modes to assist or estimate different Internet characteristics. However, diagnosing cyber space as a fundamental procedure is still unresolved challenge due to different aspects such as [35]:

- Lack of standardized metrics.
- Lack of measurement and tools taxonomies.
- Measurement intervals (i.e., seconds, minutes, hours or days).
- The access to the core network and middle resources.
- Statistical analysis of measured data.
- Network performance prediction based on measurements.
- Proliferation of measurement non-scalable tools.
- Lack of experiment design.
- Security and privacy in network monitoring.

This chapter will briefly summarize common measurement models used in determining the metrics under study: delay, packet loss and available bandwidth. With each metric, we associated public utilized utilities that were under using attempts, but due to lack of scalability and other essential performance deficiencies oftentimes caused failures.

# 2. Probing Models

Designing an estimating utility for a certain metric requires profound understanding of a metric's dynamic nature. In networking, two models are used for managing flows: a source-based

and destination-based. Figure [3.1] shows the source-based model where a source "S" sends n probes, and should receive corresponding n responces. Figure [3.2] shows the destination-based model where S receives one response "R" that carries a required result. Each model has advantages and drawbacks as will be described in sections [2.1] and [2.2]. Performing precise active measurements with either model is depending on an affordable degree of control to network recourses, and a user should decide the following parameters carefully:

- The size of the probing packet.
- The size of data loaded into the network.
- The initial transmission rate.
- The initial time-gab among probes.



Figure [3.1]: The Source-Based Model



Figure [3.2]: The Destination-Based Model

Therefore, achieving accurate estimations is a function of the above four parameters although some applications are susceptible to one than another. Ping for instance is more controllable by the amount of load than iperf while iperf is more adjustable by the transmission rate (i.e., a small difference between transmission rate and receiving rate concludes an accurate measurement [32]). In addition, an initial gab between probes can indicate high correlation between its changes and a competing traffic utilized bandwidth on tight route [32].

# 2.1 Source-Based Model

In this design, tools use a single-direction control to send either TCP streams like sprobe or UDP packets such as traceroute. Using the TCP will force the receiver to send back TCP-FINs in response to incoming packets. However, with UDP, the sender will force its destination to replay upon transmitted ICMPs by UDP echoes. Table [3.1] shows advantages and disadvantages of the current model [39].

Positives	Negatives			
Flexible deployment.	ICMPs and Echoes may suffer filtering.			
No clock synchronization.	Requires time synchronization.			
	Requires reliability on the reverse direction.			
	Non-symmetric routes for responses.			
	Round-trip delays are more likely affected by a			
	cross-traffic than one-way delays.			
	More traffic on the reverse direction.			

Table [3.1]: The Source-Based Model Advantages and Disadvantages

# 2.2 Destination-Based Model

In contrast with the source-based model, both ends in this model are heavily involved in conducting measurements. Table [3.2] shows advantages and disadvantages of the destination-based model.

Positives	Negatives
More accurate than sender-based technique.	Difficult in deployment
Less traffic on the revere direction.	Requires time synchronization.
	Requires reliability on the reverse direction.

# Table [3.2]: The Destination-Based Model Advantages and Disadvantages

# **3.** Delay Estimation

Measuring a round-trip time of a packet while crossing a link or route is not a

complicated procedure using a tool operates as source-based model. Approximately, one can calculate the packet's one-way delay after assuming a forward and backward direction to be symmetric. In reality, considering Round Trip Time (RTT) as an indicator of the half-way delay can be a misleading fallacy if the two directions are completely distinct, and more over if the difference in size between the original packet and its replay is quite large.

However, the round-trip measure is commonly used to avoid complexity in controlling and synchronizing both ends for measuring the one direction delay. In networking, variety of tools successfully can estimate the round-trip delay as will be outlined in section [3.2].

## 3.1 Delay Models

Characterizing a total delay experienced by an orientated packet toward destination as a function of physical network properties and cross-traffic has been studied carefully throughout the past years. Normally, a packet can suffer of usual five delays: fabric, processing, queuing, transmission and propagation delay. For a train of packets of size "s", the fabric, processing and propagation delays are considered to be constant among all packets. The queuing "q(n)" delay of a packet n is the only nondeterministic that varies over time. The two simple models for analyzing a round-trip time of all packets of the train are:

- For a light load condition: since RTT(n) = d + q(n) for the n<sup>th</sup> packet where d is the summation of the other four delays, using FIFO queue results next RTTs to be simplified to RTT(n + 1) = RTT(n) + ε(n) where ε is an exponential random variable with small variance and zero mean [33]. Therefore all data points (RTT(n), RTT(n + 1)) will lie slightly above the diagonal RTT(n + 1) = RTT(n) in a phase plot.
- For a heavy load condition: the model uses neither plain autoregressive nor moving-average type of prober that can send probes every  $\delta$  seconds, and a FIFO queue that can receive two

distinct streams: a one with a fixed interval among probes and a second with a randomlydistributed Internet traffic [33]. Let the second stream receives *B* bits in between probe packets *n* and *n* + 1. Let's further assume that  $B/\mu \gg \delta$  so that q(n + 1) will be large enough to cause an accommodation of *k* probes behind the probe n + 1 with no intervening extra bits between their arrival times [33]. As a result, the accumulated probes will be queued for  $q(n + i) = q(n + i - 1) + s/\mu - \delta$  where  $2 \le i \le k+1$ . Therefore, and to be accurate, a phase plot of this scenario will demonstrate all *n* probes before *B* as in a light load model as described above, the probe n + 1 as special case with  $RTT(n + 1) = RTT(n) + B/\mu - \delta$ and *k* probes with  $RTT(n + 2) = RTT(n + 1) + (s/\mu - \delta)$ .

#### 3.2 Delay Tools

During the past years, an extensive research has been conducted on delay estimation. The vast majority of the available work on this topic focuses on round-trip delay. However, applications may have dissimilar sensitivity regarding packet delay, and necessitate periodic updates for delay statuses, and require careful selection of utilities. The next presents common tools for measuring round-trip delay.

# Ping

Ping runs the Internet Control Message (ICMP) mandatory echo request datagram to elicit echo responses from a receiver. For all request, ping will determine the average round-trip delay and accumulative packet loss. In this study, ping was functioned in administrative mode in order to utilize the flooding option (i.e., with 2ms time spacing, ping sends packets of size 1052Bytes = 1024Bytes payload + 20Bytes IP header + 8Bytes ICMP header).

#### - Traceroute

The traceroute is a tool that calculates route mapping between two end-hosts. It can monitor its packets in order to report a sequence of links until reaching destination, and identify miscreant intermediate router, which is discarding the flow. It uses the TTL flag to elicit ICMP responses from interposed hosts as packets transit. These ICMPs are received in response to 60Bytes UDP packets. The performance of traceroute may be limited for security purposes when the network firewall decides to discard all incoming traffic without sending back time exceeded packets [2]. The second challenge with traceroute happens when receiving replays from distinct IPs that belong to the same device.

#### - Tracepath

Tracepath can trace a route to given destination as datagrams proceed toward destination in order to detect the Maximum Transmission Unit (MTU) that can be assigned to a single datagram without being fragmented. The main dissimilarities comparing to traceroute are in using different UDP mechanism, having no fancy options and requiring no root privileges.

- Trout

Trout is combined of visual traceroute and the well-known "whois" utility (i.e., a domain lookup system). The trout has the ability to flexibly control pining procedure in contrast to the previous tools.

#### - Cing

This utility is similar to pathchar from the aspect of using ICMP timestamp probes in order to examine per link RTTs of a connection.

## 4. Loss Estimation

In packet switching systems data loss is possible due to use shared resources, link failures, outages, overloading and other causatives. Hence, assisting packet loss is not an easy task because of the difficulty in differentiating the loss direction and its originator when utilizing source-based models. As a result, a tradeoff between flexibility and accuracy appears while deciding a source-based or destination-based estimator. The operational mechanism in TCP

23

makes packet loss as unequivocal downside because of the splendid method applied [2], but in UDP, data loss can affect performance to unacceptable levels.

#### 4.1 Loss Tools

The challenge in prototyping any tool as mentioned earlier is in distinguishing precisely what caused packets to be lost. Many tools were designed to assist this metric, but yet predicting the real cause such as congestion, overloaded queues or large delay in response still requires further investigations. In this study, the attention is only toward using active loss determining on either TCP or UDP connections. The following utilities are common packet loss estimators.

- Sting

Sting is a TCP based packet loss measurement tool. According to [40], sting preserves a near universal applicability prototype to measure loss rates simultaneously on a forward and reverse direction in between a pair of nodes because of utilizing a destination-based model.

- Tulip

Tulip in [41] is used to capture packet reordering, estimate packet loss and queuing delay between two hosts. It leverages ICMP timestamps and sequential IP identifiers (i.e., two common features in current routers). The scriptroute package should be installed first as tulip one of its tools.

#### - Badabing

This tool is a destination-based type of design that can be parameterized to send a specific number of packets to determine packet loss. It uses a geometric model to demonstrate an explicit tradeoff between accuracy and consequence impacts when overloading network [42].

- **T-Rat** 

The T-RAT prototype is for analyzing TCP connections and determining limitations factors of transmission rate, namely packet loss and delay. The original design operated as a connectionless

24

UDP mode, and modified afterwards to be a connection oriented tool [43].

# 5. Bandwidth Estimation

Many probing techniques have been involved to measure bandwidth such as the onepacket model (e.g., pathchar) [4], packet-pair model (i.e., for calculating bottleneck bandwidth) [4] and multi-packet model. The third one is considered to be more accurate than the previous two because of relying on a variety of statistics out of a mixed flow [4].

# 5.1 One-Packet Model

Figure [3.3] shows the amount of time elapsed by a packet before terminating its journey over two links. In general, equation [3.1] calculates an overall delay while crossing l links. This formula includes transmission delay and others, but no queuing is involved [4]. Table [3.3] lists used parameters in the three models [4].

Parameter	Definition
n	The number of links
<i>d</i> ( <i>l</i> )	The delay of link <i>l</i>
D(l)	The summation of latencies up to <i>l</i>
b(l)	The bandwidth of link <i>l</i>
<i>s</i> ( <i>k</i> )	The size of packet k
t(k,l)	The fully arrival time of packet $k$ at link $l$
q(k, l)	The queuing delay of packet $k$ at link $l$
l(bn)	The index of bottleneck link

Table [3.3]: The Definitions of Bandwidth Parameters



Figure [3.3]: The One-Packet Delay Model Diagram

The one-way delay can be derived from figure [3.3] as represented in equation [3.1] that shows linear relationship between delay in seconds and packet size in bytes. This model is exactly used by pathchar to determine links' bandwidth using RTT. The assumption was that the linearity in relation also holds on the back-way (i.e., replays cross the same route over short time-scale equals to a time required in refreshing routing tables). For every link, after sending multiple packets of each utilized size sequentially, the slop of a resulting linear regression represents physical bandwidth "capacity" of a particular link.

$$t(0,l) = t(0,0) + \sum \left(\frac{s(0)}{b(i)} + d(i)\right), 0 \le i \le l - 1$$
[3.1]

- Used assumptions:
- Linear relation between transmission delay and packet size (i.e., not always true since a router may copy a 128Bytes faster than 129Bytes, but can be neglected) [4].

- Routers act in store-and-forward manner (i.e., receive the entire packet before sending the first bit, and this holds on the current Internet) [4].
- No cross traffic that causes a measurement packet to be queued, and alter queuing delay over time (i.e., almost not true) [4].
- Single forwarding channel is the only used by all routers. (e.g., the Basic Rate Interface (BRI) as standard Integrated Services Digital Network (ISDN) for small scale Internet connections, however, is composed of two 64Kbps channels appear as one at a receiving side) [4].
- Existed limitations:
- Non IP addressable invisible nodes that introduce an extra delay (i.e., do not decrement TTL) such as:
  - The node between a source's application and operating system [4] used for loading kernel space.
  - The node between a source's operating system and network interface card [4] used for unloading kernel space.
  - The node between a destination's network interface card and kernel's space [S].
- The queuing can possibly alter delay on either direction of a flow when applying estimating RTT.

# 5.2 Packet-Pair Model

This model is usually used for either locating or measuring bottleneck bandwidth. Figure [3.4] illustrates time differences after passing a bottleneck bandwidth link l. Equation [3.2] calculates the difference between arrival times of two identical packets of size s after traversing a route with bottleneck bandwidth b(l(bn)).

$$t(1,n) - t(0,n) = max(\frac{s(0)}{b(l(bn))}, t(1,0) - t(0,0))$$
[3.2]



Figure [3.4]: The Packet-Pair Model Diagram

More accurately, an estimated bottleneck bandwidth of a route via packet-pair model can be in three possible scenarios listed in table [3.4] and classified in figures [3.5], [3.6] and [3.7]. Figure [3.7] tracks only packets under concern P1 and P2 when passing two queues.

Route Status	Bottleneck Bandwidth
Unloaded route	b(l(bn)) = s(0)/(t(1,n)-t(0,n))
Cross traffic	b(l(bn)) > s(0)/(t(1,n)-t(0,n))
Multiple queues	b(l(bn)) < s(0)/(t(1,n)-t(0,n))

Table [3.4]: The Bottleneck Bandwidth Scenarios





 $\Delta t$  = Time to process *p* bytes of P2,  $BnB < p/\Delta t$ 

Figure [3.7]: Multiple Queues Scenario

- Used assumptions:
- FIFO queues are always implemented; if a system uses weighted fair queuing, then this model will measure the available bandwidth of the bottleneck segment instead [4].

- Linear characterization between transfer time and packet size.

Both packets will be queued at same location.

- Existed limitations:
- Does not count per link delays.
- Requires sending more than two packets to avoid single channel problem outlined in section [5.1].

# 5.3 Multi-Packet Model

This model is a general form of the previous two models. It was derived out of three different equations; a delay equation derived out of an arrival time formula and queuing delay formula. This model assumes that the first packet will never be queued [4], and concludes with a new equation [3.3] as a multi-packet delay model. Using this equation, the study in [4] concludes with estimation for a bandwidth of a link that causes queuing as expressed by equation [3.4].

$$t(k,n) = t(k,0)$$

$$+\sum_{i} \{\frac{s(k)}{b(i)} + d(i)$$
[3.3]

$$+ \max(0, t(k-1, i+1) - d(i) - t(k, i))\}$$
, where  $0 \le i \le n-1$ 

$$b(l(q) = \frac{s(k-1)}{t(k,n) + \frac{s(k) - s(k-1)}{b(l(q) - 1)} - \frac{s(k)}{b(n-1)} - t(k-1,0) - D(n-1)}$$
[3.4]

# 5.4 Bandwidth Tools

Existing bandwidth tools can be characterized in terms of four metrics, capacity, throughput, available bandwidth or Bulk Transfer Capacity (BTC: a measure of a network's capability to send significant quantities of load using a single congestion-aware transport connection like TCP. Intuitively, BTC is an expected long-term average transfer rate in bps of a single ideal TCP operation over a path in question [34]). In specific, determining any of the four

metrics can be either over a one-link route or an end-to-end connection. As bandwidth is a key factor in network systems, the following are common used tools in estimating bandwidth.

- Bing

Bing is used to compute an end-to-end throughput using two different sizes of ICMP echo request packet between the two terminals of the connection [44].

- Sprobe

Sprobe according to [22] can be considered as an accurate and quick utility to measure bottleneck bandwidth in uncooperative network by exploiting some TCP properties. Sprobe as similar to sting can work on asymmetric routes, flexible bandwidth changes and scalable networks. The dominant downside of this tool is the requirement to be functioned under specific firewall settings, which are not available usually for ordinary users.

- ABwE

This utility is a packet dispersion based type of design that can monitor distinct performance metrics such as congestion, route changes and available bandwidth.

- DietTopp

This tool is used to estimate an end-to-end link capacity as well as the end-to-end available bandwidth. However, in order to achieve accurate results the traced path should not contain more than one bottleneck [10].

- Spruce

This tool is used only to measure bandwidth between two terminals that are directly connected (i.e., link characterization utility). The main drawback of this tool is a prerequisite knowledge about the link capacity.

31

## - Clink

Clink as source-based utility was designed to calculate both delay and available bandwidth of a route. It uses UDP packets under the same mechanism implemented in trout, ping and traceroute in addition to high transmission rates for variety of sizes each time [45]. However, clink is one of the tools that generate heavy traffics on the route for long period of time. As a result, around 0.09 of the available bandwidth is consumed by probe packets if clink waits approximately ten times the round-trip delay [45]. In addition to that, all packets of different sizes are forced to follow every link on the connection instead of links that have smaller available bandwidths.

#### - Pathchar

Bellovin and Jacobson used the one-packet model in an inexpensive linear regression form to design pathchar; the high cost considers: time, scheduling acknowledgments, invisible hosts and reverse paths [4]. The researchers avoided receiving timely acknowledgments by using the ICMP echo replay protocol and UDP or ICMP time exceeding replay. They used equation [3.1] for the packet and corresponding acknowledgment to determine RTT. The issue of queuing is resolved with pathchar by using more extra traffic and consider the minimum delay of particular size to be zero in order to be consistence with the model [4].

## - Pathchirp

This tool uses UDP packet trains called chirps to estimate available bandwidth over Internet routes. Chirp's trains travel from a sender to receiver by an order from a third party master machine called pathchirp-run.

#### - Pchar

Pchar is an analogous to pathchar that was developed by a reimplementation of the pathchar

32

written by Van Jacobson. Similar to pathchar, pchar attempts to characterize delay, packet loss and bandwidth along end-to-end paths.

#### Pathload

Pathload is used to estimate available bandwidth of end-to-end connections. The available bandwidth as seen by pathload is the maximum IP-layer throughput, which can be achieved by a flow without reducing transmission speed over a period of time.

#### - Assolo

Assolo is a copy of pathchrip to estimate available bandwidth based on the concept of selfinduced congestion. The tool features with a new probing traffic profile called Reflected Exponential Chirp (REACH), which tests wide range of rates for accuracy purpose at the center of the probing interval. Moreover, the tool runs inside the real time operating system and uses some de noising mechanisms to further enhance accuracy.

#### - IGI-PTR

This paired utility is used to measure the residual bandwidth on a path using an active packet train probing. It implements two algorithms; the first called Initial Gap Increasing (IGI), and the second is the Packet Transmission Rate (PTR) [46]. Both IGI and PTR share the same probing procedure. IGI requires the bottleneck bandwidth while focusing on calculating background traffic load. In contrast, PTR directly calculates the packet transmission rate for estimating an end-to-end available bandwidth [46].

#### - Yaz

Usually, yaz is used to estimate an end-to-end available bandwidth. To some extent, yaz operates similarly to pathload, yet the contrary is in involving mean-spacing instead of normal spacing between packets to reduce errors resultant either by operating systems or hardware. Additionally, yaz uses a sort of expansion compression model (i.e., alerting trends in one-way

delays) to determine available bandwidth.

- Netest

This tool was designed based on a concept referred MBS for the benefit of calculating an approximate available bandwidth and throughput of a network.

- Pipechar

Pipechar in the application level is an available utility in the (NCS) package. However, Pipechar is considered to be out-of-date tool that was used to monitor routes by separately probing links toward destination in order to determine both original and utilized bandwidths (i.e., subtracting the two quantities results available bandwidth).

- Pathneck

This utility is not for measuring bandwidth rather than locating the influence of cross-traffic over the connection. Pathneck also uses RPT to locate routers that introduce wide delay gap (i.e., congested) between two packets of the same size.

- Nettimer

The nettimer was developed based on the multi-packet model to be operated in two distinct phases. The tailgating phase is to measure every link's bandwidth. The sigma phase is to determine the bottleneck bandwidth over the entire route. The design of this utility was implemented by assuming that there is no queuing experienced by the first packet while the second packet will face only single queuing at certain router behind the first packet.

#### - Stab

Similarly with pathneck, the main objective behind stab is to locate the thin link that has the smallest available bandwidth. This utility combines the concept of self-induced congestion "packet tailgating" and special probing trains called "chirps" to efficiently locate congested

segments.

#### - Nuttcp

The nuttcp is used to measure specific form of network performance namely packet loss and throughput. The raw-socket throughput of either TCP or UDP is measured by forwarding memory buffers between channel terminals. In addition to that, nuttcp estimates additional useful statistics related to data transferring such as system clock time, transmitter and receiver CPU utilization and loss percentage of UDP transfers [46]. The nuttcp over takes on nttcp and ttcp by several features such the used server mode, rate limiting, multiple parallel streams and timer based usage [46]. On the other side, the main downside is the request of gaining supper-user privileges to adjust the system service-file and the "xinetd" daemon configuration file.

#### - Thrulay-Thrulayd

The thrulay-thrulayd pair was implemented in destination-based prototype to calculate achievable bandwidth "throughput" and loss rate over an end-to-end connection either by broadcasting TCP bulks or UDP streams. This server-client model can be also used to estimate conditional available bandwidth after setting a suitable packet loss threshold.

#### - Iperf

The benchmark iperf is a common multi-function utility as a destination-based type of implementation. Iperf operates by using both TCP and UDP. In the TCP mode, iperf can estimate throughput, Maximum Segment Size (MSS) and the maximum transmission unit. Using UDP, iperf can measure packet loss, delay, jitter and conditional available bandwidth. Although iperf has the ability to perform simultaneous forward-reverse estimation on the same route, we instead preferred to separate each direction's measurement in order to avoid heavy loads over short time instances. The transferred load can be controlled by either time or a given size.

# 6. Summary

Measuring Internet dynamics has been under intensive research via credible tools. The chapter listed common models tools associated for tree significant dynamics: packet delay, loss rate and bandwidth. These utilities went into repetitive examinations throughout earlier stages of our study, but most of them failed due to different constrains.

# **CHAPTER 4**

## METHODOLOGY

#### 1. Overview

Implementing an efficient probing methodology is extremely difficult challenge in packet-switched networks. Throughout our research, designing a robust and accurate probing tactic was a secondary aim comparing to the direct and indirect analysis of Internet dynamics. The chapter will present utilized experimental classifications of probing schemes and probing algorithm in sections [3.1] and [3.3] respectively. Passively afterwards, and using reported measurements, we determined properties of the studied metrics in the remaining chapters beside the intensive bandwidth exploration in chapter [7].

## 2. **Probing Utilities**

In this study we utilized common benchmarks such as traceroute, ping and iperf. The traceroute was used to assist packet delay over long and large-scale network. We used the ping utility to evaluate three different factors: delay, packet loss and BTR. The iperf was operated in two different modes. The TCP mode was conducted to monitor the achievable bandwidth that can be useful in future window type of probing protocol design. In UDP mode, iperf can estimate both loss rate, and report the affordable transmission speed. The four distinct networks "experiments" that were assigned in the study are classified in appendixes from [I] to [IV].

## **3. Probing Accuracy**

Usually with interactive applications that run type of probing utility, performance and accuracy depend on the used prototype of the probing algorithm and its schedule. Therefore, before sending probes over broadband connection, carefully, the following questions should be addressed:

- What mechanism does match the application perfectly?
- What is the probing time-schedule?
- Until what degree of accuracy the probing should continue?
- What sort of challenges does face probes (e.g., outages, congestion, security and cost)?
- What is the proper scheme for collecting measurements (i.e., data collection and archiving)?

## 3.1 Utilized Probing Schemes

The high frequency in probing is required for some applications that are not delay and jitter tolerant like Voice over IP (VoIP). In the meantime, other applications are more sensitive toward available bandwidth such as video streaming. Therefore, choosing appropriate parameters to control a probing scheme in somehow depends on the metric under investigation. In our study we have used four different schemes as classified in table [4.1]. The first was for measuring delay by using light traceroute traffics. The second was utilized to calculate delay and packet loss using ping. Iperf used the third one to evaluate TCP throughput while the forth was for determining packet loss and available bandwidth. This table contained probing statistics of the quickest and slowest group of the first run of each experiment.

Experi	ment	Run	Metric	n	т	k	r	$\hat{g}t(l,r)$	$\hat{e}t(r)$
Tracer	oute	First	Delay	138	10	14	[1 → 4]	41.4	55.2
Ping		First	Delay, packet loss and BTR	140	2	70	70 $[1 \rightarrow 16]$ 11.2		14.0
Iperf	ТСР	First	Throughout	120	2	60	[1 → 4]	12.2	14.6
	UDP	Last	Loss rate and available bandwidth	122	2	61	[1 → 14]	16.8	21.6

Table [4.1]: The Parameters of the Probing Schemes

The parameter *n* is the number of machines per experiment. The parameter *m* represents the number of machines per group while *k* indicates the number of groups (i.e., we divided *n* to *k* groups that perform probing simultaneously). The value *r* refers to the range of repeating runs of an experiment, and *l* denotes groups' indices. The printed  $\hat{g}t(l,r)$  and  $\hat{e}t(r)$  are estimated probing delays in minutes of an actual gt(l,r) and et(r) respectively of the identified runs in using an averaged  $\hat{\zeta}$  in table [4.3] (i.e.,  $\hat{g}t(l,r) \leq \hat{e}t(r)$ ). The gt(l,r) is an actual required time by a group *l* of run *r* to accomplish probing while et(r) is the total run time. Since all groups are performed in parallel, an actual total running time of an experiment  $et(r) = \max\{gt(l,r): 1 \leq l \leq k\}$ .

# **3.2 Probability of Collision**

We defined a probability of collision to be the probability of an event when two hosts probe same host simultaneously. Clearly, this problem is almost similar to the Birth Day Problem<sup>1</sup>. The difference in here is that the number of individuals is equal to the number of groups k (i.e., only k nodes probe at any time instance). Meanwhile, the number of days in a year is corresponding to the total number n. In this study, k equals to n/2 with ping and iperf due to over limit time consumption, but  $\lfloor n/10 \rfloor$  was used in traceroute experiments. The subsequent procedure determines a correct upper bound of k.

We assume that the number of available nodes n is uniformly distributed random variable in a range  $[1 \rightarrow n]$ , and every host out of n is a probing subject "receives probes". Practically, kshould be equal to the number of active hosts in an overlay network at given time. The k probers need to be chosen whereby  $k \leq n$ . The target is to make the probability Ps(n, k) for at least two hosts of k probe a same point at particular time as minimum as possible.

The probability Pn(r1) = n/n is for a first prober  $r_1$  to not share a destination d1. Pn(r2) = 1 - 1/n and so on until Pn(rk) = 1 - (k - 1)/n. Intuitively, the total number of possibilities with sharing equals to  $n^k$ . Therefore, the simplified total probability with no single sharing is equal to the multiplication of all these probabilities as in equation [4.1].

$$Pn(n,k) = (1 - 1/n)(1 - 2/n) \dots (1 - (k - 1)/n)/n^{k}$$
[4.1]

In order to calculate an upper bound of k, we instead use a complement probability Ps(n,k) of Pn(n,k) as represented in equation [4.2].

$$Ps(n,k) = 1 - (1 - 1/n)(1 - 2/n) \dots (1 - (k - 1)/n)/n^{k}$$
[4.2]

For simplicity, we can substitute by a simplification from [31] in the later formula [4.2] to derive equation [4.3]. Due to this simplification, an attached error  $\varepsilon$  formed by the inequality [4.4] is associated to Ps(n, k) [31].

$$Ps(n,k) \approx 1 - \exp(-k(k-1)/2n) \approx 1 - (1 - k/2n)^{k-1}$$
 [4.3]

$$\varepsilon < n^3/6(n-k+1)^2$$
 [4.4]

1.

Therefore, In order to achieve Ps(n,k) = a, we can solve [4.3] for a minimum positive k that satisfies  $k^2 - k - 2n(\ln(1-a)) = 0$  as in [4.5].

$$k = \frac{1 \pm \sqrt{1 - 8n(\ln(1 - a))}}{2}$$
[4.5]

Regarding ping experiment for instance, a theoretical k should be 9 instead of 2. However, we did not abide to equation [4.5] because of a looping attribute of the probing algorithm in section [10.3]. This feature allows the algorithm meaningfully to resolve shared probing failures using a random ordering of the probing domains at every node (i.e., we postponed designing a pre agreement handshaking algorithm for future research). The direct correlation among k and gt(l) causes gt(l) to increase as k increases, and consequently rises up time consumption, which is not favorable in overlay routing.

In reality, nodes respond to probes with distinct rates, and thus the used assumption about n nodes being equally likely to be chosen is further diminished (i.e., a quick responder has small chance of being caught by more than one prober instantaneously, and likewise a quick prober has better chance to finish before another proper gains a role). Therefore, Ps(n,k) should be further improved to accommodate these statements.

#### **3.3 Probing Algorithm**

Throughout the measurements period, all conducted experiments were under the supervision of simple probing algorithm drawn in figure [4.1]. The algorithm has to two different time-loops. The first is for a given node *i* to probe the entire experiment based on a parameter  $\eta(i, j)$  defined in table [4.2], and the second one for re-probing hosts that failed to respond or their replies have been dropped during the previous time-loop.



Figure [4.1]: The State Diagram of the Probing Algorithm. The black box is a prober *i*. The short-dashed box represents success in probing while the long-dashed one refers to failure in the first loop.

Parameter	Definition
n	The total number of hosts.
fn(i)	The number of failures of prober <i>i</i> .
ft(i)	The failure loop time of prober <i>i</i> in minutes.
<i>pt(i)</i>	The probing loop time of prober <i>i</i> in minutes.
ζ	The average single probing time per host in minutes.
η(max)	The maximum re-probing threshold.
$\eta(i,j)$	The re-probing count of host <i>j</i> by prober <i>i</i> .
ht(i,j)	The total probing time of host <i>j</i> by prober <i>i</i> in minutes.

Table [4.2]: The Parameters of the Probing Algorithm

The first probing loop actual time - pt(i) can be broken down using equation [4.6] while the failure loop time is broken in equation [4.7]. The probing parameter  $\eta(i, j)$  whose value depends on status of a route that connects a prober *i* and a host under measuring is limited by  $\eta(\max)$  provided by a user. Equation [4.8] calculates the amount of time that prober *i* spends on *j*'s measurement. The  $\hat{\zeta}(l)$  is an  $\zeta$  approximated per group.

$$pt(i) = \sum ht(i,k), where \ 1 \le k \le n$$
[4.6]

$$ft(i) = \sum ht(i,k), where \ 1 \le k \le fn(i)$$

$$[4.7]$$

$$ht(i,j) = \eta(i,j) \times \zeta$$
[4.8]

$$\hat{\zeta}(l) = \sum \frac{\zeta(l)}{mn}$$
, where  $1 \le k \le mn$  [4.9]

The actual gt(l,r) defined in section [3.1] is equal to  $\sum (pt(i) + ft(i))$  where  $1 \le i \le m, l$ and m are as outlined in table [4.1]. The probing statistics of the quickest and slowest group of the first run of each experiment are shown in table [4.3]. The printed minimum and maximum  $\hat{\zeta}$ is an averaged  $\zeta$  of a group in minutes desired to be used in future study in order to limit heavy probing.

Experiment		Run	n	η(max)	fn(i)	$\eta(i,j)$	ζ
Traceroute		First	138	2	pprox 0	≈ 1	[0.03, 0.04]
Ping		First	140	2	pprox 0	$\approx 1$	[0.04, 0.05]
Iperf	ТСР	Last	120	2	$[0 \rightarrow n]$	$[1 \rightarrow 2]$	[0.07, 0.09]
	UDP	Last	122	2	$[0 \rightarrow n]$	[1 → 2]	[0.05, 0.09]
$\zeta$ depends on transmission speed, route status and load size.							

 $\eta(i, j)$  depends on route status.

Table [4.3]: The Statistics of the Probing Algorithm

## 3.4 Data Extraction and Analysis

The active probes sent into distinct networks assigned to Internet dynamics under analysis. Traceroute used 138 nodes as classified in appendix [I] to achieve 75,624 measurements on packet delay. Ping operated on 140 nodes in appendix [II] to resulted 311,360 delay and loss measurements. For TCP and UDP we used two semi-identical networks (i.e., the first of 120 hosts assigned to 109 sites and the second of 122 hosts on 110 sites) that generate 220,948 measurements using iperf. The TCP throughput was studied on 14,280 connections and 206,668 UDPs used for calculating packet loss and available bandwidth. The intention was not toward first-order metrics such as failures' frequency and period, or routing pathologies like outages, link symmetry and loops, as we considered the used networks to be fully connected graphs with infinite costs for routes suffer of either permanent or persistent outages. We did not target calibrating measurements or applying self-consistency reviews relying on using available benchmarks of common operating systems. The Dijkstra's algorithm was involved in paths that went into a delay and loss symmetries, stability and hop usage analysis.

Consequently, we analyzed 75,624 traces that were extracted via a Linux distribution traceroute, and collected over one day period. The first 18,906 measurements were performed over 40 minutes average run time. The remaining were spaced by a random interval in a range [4  $\rightarrow$  6] hours between consecutive runs for stability purpose. The worst scenario with traceroute occurs when a destination is reachable via 30 hops, and receives three 60Bytes packets on each link (i.e., approximately 5.27Kbytes in total). We argue that for long-haul connections, and with a probability equal to 0.27 for hosts to not probe a single node simultaneously, no heavy traffic type of concern will exist. We filtered responses from nonparticipating IP addresses although nodes may have multiple IP's assigned to handle traffics.

We switched next to evaluate packet delay and loss along distinct sets of Internet connections. Due to simplicity and time consumption comparing to available tools, we used ping as a benchmark with suitable efficiency and fastness. Then we analyzed 311,360 traces out of which 233,520 were extracted over the first day with random interval in range  $[1 \rightarrow 2]$  hours among iterating runs. The remaining four runs were slotted by the same interval in order to calibrate stability beyond the first day via extra overlapped runs. The 311,360 traces were performed using constant packet size (i.e., 1052Bytes) that forms different data streams shaped by the number of transmitted packets; the loads ranged in  $[0.05 \rightarrow 0.5Mbytes]$ . Beside delay and loss, we assisted another distinguished metric called Bulk Transfer Rate (BTR) and other interfered relations over direct and indirect routing. Yet again, Dijkstra's algorithm was used in characterizing indirect shortest and smallest delay and packet loss routes. We also compared

indirect cumulative packet loss on shortest delay paths with Internet loss. Similarly in manner but with different experiment, 206,668 UDPs were conducted via iperf to understand packet loss as a function transmission rate instead of packet size. This experiment consisted of 14,762 routes derived with distinct transmission rates for estimating packet loss and bandwidth.

Then we switched to estimate TCP throughput and UDP conditional-loss available bandwidth using iperf measurements. The difficulty in evaluating bandwidth is due to the absence of stable, accurate and quick utility that operates in either mode. Unfortunately, no acceptable success was achieved in estimating bandwidth with available tools due to either incompatibility with Planetlab or utilities' shortness. The cause is in not involving scalability although each tool has associated positives. In our study, however, we utilized a different mechanism in order to accurately measure available bandwidth via iperf (i.e., in UDP mode iperf does not provide an estimation of available bandwidth). Iperf generated UDP streams at different rates ranged in  $[0.5 \rightarrow 800Mbps]$ . The measurements were elicited over periods of four days on UDP and one day on TCP, and used in constructing an indirect routing with respect to packet loss, jitter or File Transfer Time (FTT). The available bandwidth was approximated to equal a maximum affordable transmission rate that causes no more than 2% packet loss.

Throughput and available bandwidth as crucial and challenging metrics cannot be used to perfectly define an indirect routing scheme in terms of best. Nevertheless, Dijkstra's algorithm as a link state algorithm can be modified to discover a maximum-minimum bandwidth route while vector state algorithms such as "Ford Fulkerson's algorithm" can find a feasible flow through a single source and sink network.

## 3.5 Validation

Regarding stability, we validate our findings simply by comparing repetitive delayed copies of a metric under study. For validating existence of an enhanced routing we used Dijkstra's algorithm to solve shortest route problem, and other developed algorithms for exploring minimum bandwidth route problem, accumulative packet loss, symmetry and distance matching. The use of Dijkstra's algorithm in our study is only to investigate a shortest route as an alternative to an existed one provided by Internet. We did not study existence of other routes that lies between the shortest and Internet route such as a second and third shortest routes.

# 4. Summary

The chapter introduced the utilized probing procedure along with parameters observed throughout the research period. The chapter also summarized the methodology used in extracting and analyzing measurements of four distinct experiments.

# **CHAPTER 5**

#### PACKET DELAY

#### 1. Overview

Technically, in a non-trivial network, a typical delay is composed of distinct sub-delays. Every fraction denotes an elapsed time between starting and finishing any logical or physical operation throughout the packet's journey to destination. Packet delay, on other hand, is a time difference between moments of sending packet and detecting arrival. Delay equals a summation of delays experienced on a forward and backward direction. However, in our study, we hold no restriction for using either term in expressing RTT as a total delay measure. In contrary to study in [2], our collected measurements can be considered as appropriate to explore an acceptable range of Internet RTTs due to utilizing variety of experiments and tools over wide-scaled network. We invested no time in applying a sophisticated data calibration in order to ensure precise assessments after verifying accuracies of reported results and discarding misleading samples.

Throughout our research we relied on a guarantee measure of utilized tools in providing subsequent information that reflects real characteristics of a network. The classified measurements were to assist metrics to an extent of accuracy over short and long-term periods (i.e., namely within a single run or among distinct runs). We did not perform any self-consistency techniques for calibrating accuracy and clock's resolution due to time consumption, which is a sensitive parameter in indirect routing. We analyzed packet delay in order to explore deterministic shared relations with packet loss and bandwidth. Likewise, packet delay is a wealthy source of information about routing quality as delay's dynamism starts to change when queues become packed. The estimation of packet delay was derived via two well-known tools: traceroute and ping. The contents of this chapter will be organized as follows. In section [2], we generally discuss delay components. Section [4] will illustrate traceroute analysis on elicited measurements while section [5] deliberates a profound direct and indirect analysis on ping measurements.

## 2. The Delay Components

In majority of deployed switched data or communication networks, any packet is most likely to suffer from an aggregated delay consists of five major sub-delays. These components are: fabric delay, processing delay, queuing delay, transmission "store-forward" delay and propagation delay "wire-line" as will be detailed next. Each delay ranges in magnitude and effect according to network design. Calculating a cumulative delay can be as a one-way or round-trip measure for a packet to traverse a route. The impact of delay can be visible in certain applications like VoIP or real-time transactions where performance has an opposite relation with increasing delay. The five delays except the queuing delay are deterministic, and yet queuing effects can be calculated by having prior knowledge about a total traffic in a network [52].

- Fabric delay *fd*: is an introductory delay because of executing logical functions by the interior design of a switch fabric (e.g., store and forward, MAC address lookup) [52].
- Processing delay pd: every packet's header is processed, and the payload passes an error checking mechanism. Consequently, pd can be a key component while performing a sophisticated error diagnosing demand in encryption algorithms.
- Queuing delay *qd*: a cross traffic can cause additional delay until delivering to a transmission buffer. Although priority schemes are implemented to mitigate *qd*, this portion is still nondeterministic due to unpredictability of cross traffic nature [52].
- Transmission delay rd: this delay as a ratio between a packet size and transmission rate is

proportional to packet size but not necessarily with systems that are not modeled as store then forward networks. Therefore, an RTT value can be further augmented by the transmission delay.

- The previous delays are either directly connected to hardware design like a queue size and transmission speed or software implications like error checking, fairness, priority algorithms and carried loads. However, a propagation delay - *gd* is directly attached to a utilized speed of transmission media and distance.

The total one-way delay - td on a given path consists of many routers is computed by aggregating all included partial delays (i.e.,  $td = \sum fd + pd + qd + rd + gd$ ).

## **3.** The Delay Experiments

The traceroute experiment consisted of 138 machines that are globally distributed as listed in appendix [I]. The 18,906 direct routes and their delayed pictures were extracted from 75,624 traces. The maximum pressure for tracing unreachable node within 30 links equals to 60Bytes in a worst case scenario. Traceroute was operated over four loops spaced by an approximate 6 hours interval. The used RTTs were elicited by averaging the last three samples. Because of simplicity carried on via traceroute, the experiment was used first to rapidly investigate existence of indirect layer that can reduce RTT as possible. Second, we determined the stability of ten randomly selected indirect routes of lengths 2, 3, 4, 5, 6, 7, 8, and 10 hops (i.e., 40 routes per length "320 samples" plus "8 samples" for few routes of 10 hops). Third, we analyzed the difference between direct and indirect routing in terms of utilizing physical links as covered in section [4.3]. After accomplishing this, we switched to use the power of ping in order to carefully study further relations.

The procedure with ping was slightly different. The used experiment consisted of 140 machines distributed as in appendix [II]. Similar to traceroute, ping also estimates the RTT
between two terminals of a channel. Ping, however, clearly reports RTT for each transmitted chunk of data. With all loads, we set  $\delta = 5$ ms as transmission interval between packets so that ping can probe and finish as quickly as the entire load is transferred. We assumed that the 5ms interval will satisfy the queuing inequality (i.e.,  $P/\mu - \delta < 0$  when sending a train of packets of size *P*, and spaced by  $\delta$  seconds to a queue with serving rate  $\mu$ ) although we might overwhelmed the network using such a mode with almost two heavy streams: 0.25 and 0.5MByte. The purpose was to investigate possible worst case enhancement and stability by rapidly recording results. We used ping under the control of the probing algorithm described in chapter [4] with no repetitions (i.e.,  $\eta(i) = 1$  for each host  $i \in [1 \rightarrow 140]$ ), and fortunately no failures were reported unless for unreachable destinations. Ping was utilized with distinguishable sizes of streams. The first stream has 50 packets of size 1052Bytes including the ICMP header in order to form 0.05MByte. The remaining streams form a 0.1MByte, 0.25MByte and 0.5MByte respectively. Section [5] will detail analyzing delay measurements that were derived on 16 different runs (i.e., each one represents 19,460 Internet paths).

# 4. Traceroute Analysis

The following primitive analysis was a first step to prove an existence of a redundancy of invisible alternatives for any path joins two endpoints. Usually, these alternatives are not noticeable by the Border Gateway Protocol (BGP). Although BGP is a core protocol that is backing main routing decisions on Internet [49] either domestically inside ASes (e.g., OSPF) or internationally between ASes (e.g., BGP), periodically BGP may not exchange any connection between two ASes when attached nodes constantly fail in sharing a direct route [27].

After strengthening the idea, we started to understand the possibility of using the extracted data to verify stability of corresponding indirect routes over time. We used Dijkstra's algorithm to find shortest delay routes among pairs. Regarding stability, we analyzed 10 samples

of all possible lengths across four runs over 24 hours period. In section [4.1], we calculated some length statistics, and demonstrated link utilization in section [4.3].

## 4.1 Indirect Length Statistics

#### 4.1.1 Definition

This section computes four diversity measures: mean, standard deviation, minimum and maximum lengths of 137 indirect routes connect every node to the rest of experiment. This analysis is relatively a measure of variability close to each machine.

#### 4.1.2 Result

The plots I's on each figure of [2.1] to [2.4] show that majority has mean lengths equal to 2 hops. The sorted mean curves are at one hop for same failed hosts and start raising as hosts change maximum lengths (i.e., three runs have almost identical sorted means). The sorted standard deviations indicate all length samples are almost away by one hop from a corresponding mean length except failed machines and other irregularities like in the last figure. Obviously, out of II's all hosts prefer using constant 1 hop direct routes while the maximum length is above 4 hops.



Figure [5.1]: The First Run Length Statistics



Figure [5.2]: The Second Run Length Statistics



Figure [5.3]: The Third Run Length Statistics



Figure [5.4]: The Forth Run Length Statistics

## 4.2 Indirect Length Stability

# 4.2.1 Definition

The referred length stability throughout our study indicates steadiness of utilizing hops via indirect paths. From the four repeated traceroute experiments, an analysis was performed on ten selected indirect routes of distinct lengths outlined earlier in section [3].

#### 4.2.2 Result

Figure [5.5] shows that out of 40 routes of length 2, 29 routes maintained same length over one day period while 11 routes switch to either above or below length. The percentage of routes that appeared with higher lengths than 3 was 20% while 55% of routes have fixed three hops as in figure [5.6]. The percentage of maintaining same length starts to decay with routes of length 4 and above as indicated in figures [5.7] until [5.13]. Beyond five hops, routes always tend to change to small lengths.



Figure [5.5]: The Stability of Two Hops Length



Figure [5.6]: The Stability of Three Hops Length



Figure [5.7]: The Stability of Four Hops Length



Figure [5.8]: The Stability of Five Hops Length



Figure [5.9]: The Stability of Six Hops Length



Figure [5.10]: The Stability of Seven Hops Length



Figure [5.11]: The Stability of Eight Hops Length



Figure [5.12]: The Stability of Nine Hops Length



Figure [5.13]: The Stability of Ten Hops Length

The histograms [5.14] to [5.17] show that majority of routes uses one or two hops as utilization starts decaying exponentially afterwards. Obviously, these figures indicate a wide space of possible improvements on an enormous number of routes use lengths above 2 hops.



Figure [5.14]: The First Run Lengths Histogram



Figure [5.15]: The Second Run Lengths Histogram



Figure [5.16]: The Third Run Lengths Histogram



Figure [5.17]: The Forth Run Lengths Histogram

# 4.3 Indirect Link Utilization

This section explores the frequency of utilizing physical links via indirect routes combined with the stability of such utilization over time. The core objective is to check the extinct of using links before designing overlay routing. Indirect routes with little delay enhancements should not include more than 30 links to not introduce further delay due to addressing overhead. Interestingly, figures [5.18] to [5.21] show semi-Gaussian distributions with means slightly above 20 links. Routes that exploit more than 30 links are more likely indirect routes since traceroute limited to 30 links normally. The long column counts failed and possible indirect routes while beneath 30 links direct and indirect routes exist.



Figure [5.18]: The First Run Link Utilization







Figure [5.20]: The Third Run Link Utilization



Figure [5.21]: The Forth Run Link Utilization

## 5. Ping Analysis

Since the RTTs reported by traceroute depends only on three packets, which may not represent a real flow, with ping we analyzed different relations associated to more efficient packet delay samples. First, we defined a term called a degree of delay symmetry in order to assist 19,460 long-distance routes and their 291,900 delayed pictures more precisely. This degree is either attached to Internet routing "direct symmetry" or indirect routing extracted via Dijkstra's algorithm "indirect symmetry". We used both degrees to monitor delays between two nodes. The defined degree is not mentioned in literature, and differs from a symmetry term indicates structural path design. This degree is used to investigate RTT similarity between a source-destination pair. Further, different characteristics such as indirect length statistics, delay and distance and the significance of overlay routing were studied. The timesheet of the 16 runs is classified in figure [5.22].



Time [Hours]

Figure [5.22]: Ping Runs Timesheet

# 5.1 Direct Delay Symmetry

## 5.1.1 Definition

The defined degree of direct delay symmetry DSY(i) for a given machine *i* is a percentage of attached routes that cause an obsolete average delay difference less than 10ms between a forward and reverse direction. The DSY(i) is determined with respect to total

available routes "139" as simplified in figure [5.23]. Due to time constrains, we did not investigate if DSY(i) has any sort of relations with physical symmetry (i.e., in terms of involved links).

#### 5.1.2 Procedure

The extracted symmetry definition is to avoid complexity and difficulty in studying real structural symmetry over wide area networks. Hence, the study focused directly on the attitude of delay symmetry and its change as traffic last over distinct time periods. Although the used 10ms threshold for deciding DSY(i) has no real support, and needs further examination, we claim that if the delay gab between a pair's connections over long time is small and fixed with different streams, then the pair may use identical routes. The future object is to compare degree changes as a network becomes congested. From a performed primitive exploration, there is sort of correlation between packet loss and degree change although perfect decision requires further analysis. Real time video applications may prefer routes that offer high degrees in order to maintain strong live interaction between terminals.



Figure [5.23]: Direct Delay-Symmetry Example. Host *A* has a difference 50ms between the in-out directions to *B*. However, *B* has the 50ms with *A*, 3ms with *C* and 5ms with *D*. *C* has the 3ms with *B* and 0ms with *D* that has the 5ms with *B* and the 0ms with *C*. Therefore, DSY(A) = 0%,  $DSY(B) = \frac{2}{3} \times 100 = 66.6\%$ ,  $DSY(C) = \frac{2}{2} \times 100 = 100\%$  and DSY(D) = 100%.

#### 5.1.3 Result

Obviously, if the degree remains constant for multiple runs while having high packet loss, this will be a strong indicator about a failure in Internet in avoiding routing shortcomings. Figures [5.24] to [5.27] plot nodes' degrees over a 33 hours range. The four curves on each figure are separated by an average of 8 hours. The clear 17 hosts usually suffer from highly distinguishable delays on both directions are constant. The majority has degrees around 85%. As outline above, we argue that the clear change on the 0.05MByte first run degrees in figure [5.24] might be related to routing problems like packet delay but out of the study's scope.



Figure [5.24]: The 0.05MByte Runs Direct DSY Stability



Figure [5.25]: The 0.1MByte Runs Direct DSY Stability



Figure [5.26]: The 0.25MByte Runs Direct DSY Stability



Figure [5.27]: The 0.5MByteRuns Direct DSY Stability

# 5.2 Indirect Delay Symmetry

With similar terminology to section [5.1], we analyzed indirect delay symmetry in order to directly and indirectly compare delay symmetry. Figures from [5.28] to [5.31] show that indirect routes tend to be more close toward each other in terms of delay as the degree is increased to 90% for the majority of machines were at %80 in the previous section.



Figure [5.28]: The 0.05MByte Runs Indirect DSY Stability



Figure [5.29]: The 0.1MByte Runs Indirect DSY Stability



Figure [5.30]: The 0.25MByte Runs Indirect DSY Stability



Figure [5.31]: The 0.5MByte Runs Indirect DSY Stability

## 5.3 Short and Long-Term Route Symmetries

# 5.3.1 Definition

The route symmetry characterizes a percent of matching in indirect routing. This matching is classified among a forward-reverse pair -  $RSY(i \rightarrow j \mid t1, j \rightarrow i \mid t1 + \Delta t)$  elicited on the same run (i.e., a short-term symmetry, where  $0 < \Delta t \leq gt(l, r)$ ) or between a delayed picture and its original route -  $RSY(i \rightarrow j \mid t1, i \rightarrow j \mid t2)$  extracted on two distinct runs (i.e., a long-term symmetry).

#### 5.3.2 Procedure

The examination on this behavior part was based on comparing the percent of matching of intermediate hosts (i.e., node symmetry), and utilized direct hops (i.e., hop symmetry) via indirect connections. The referred node symmetry between two routes  $i \rightarrow j$  and  $j \rightarrow i$  represents the percentage of hosts on the short route, and also appear on the long route. The short-term hop symmetry between  $i \rightarrow j \mid t1$  and  $j \rightarrow i \mid t1 + \Delta t$  identifies a percentage of the short route's indirect hops (i.e., direct routes) whose reverses are also utilized by the long route. The long-term hop symmetry classifies the percentage of exact hops between  $i \rightarrow j \mid t1$  and  $i \rightarrow j \mid t2$ . Figure [5.32] identifies the above terminologies.



Figure [5.32]: The Terminologies of Route Symmetry

#### 5.3.3 Result

Figures from [5.33] to [5.40] show histograms of node and hop short-term symmetries of the first two runs of each load consequently. From I's, we can conclude that the hope symmetry is almost fixed no matter the size of the utilized load over time. The 100% matching has slightly above 5,000 indirect pairs of the total 9,730. The 3,000 pairs that have 0% indicate that at least one direction that connects a pair is longer than one hop. From II's, the defined node symmetry that represents the percentage intermediate matching among forward-reverse pair of connections, about 7,500 pairs have full matching regardless load size.

Figures from [5.41] to [5.46] show the full mapping of long-term symmetries of 19,460 routes that utilizes different loads. Each figure is a combination of two runs of the first four runs assigned to different loads. Both hope and node symmetries were almost duplicated comparing to the short-term symmetries as I's and II's of these figures show. This indicates that the routes  $i \rightarrow$ 

 $j \mid t1$  and  $j \rightarrow i \mid t1 + \Delta t$  are less identical than  $i \rightarrow j \mid t1$  and  $i \rightarrow j \mid t2$ , and strongly supports that the indirect delay routing preserves the same nature of Internet.



Figure [5.33]: The First 0.05MByte Run Short-Term Indirect RSY



Figure [5.34]: The Second 0.05MByte Run Short-Term Indirect RSY



Figure [5.35]: The First 0.1MByte Run Short-Term Indirect RSY



Figure [5.36]: The Second 0.1MByte Run Short-Term Indirect RSY



Figure [5.37]: The First 0.25MByte Run Short-Term Indirect RSY



Figure [5.38]: The Second 0.25MByte Run Short-Term Indirect RSY



Figure [5.39]: The First 0.5MByte Run Short-Term Indirect RSY



Figure [5.40]: The Second 0.5MByte Run Short-Term Indirect RSY



Figure [5.41]: The First 0.05-0.1MByte Runs Long-Term Indirect RSY



Figure [5.42]: The First 0.05-0.25MByte Runs Long-Term Indirect RSY



Figure [5.43]: The First 0.05-0.5MByte Runs Long-Term Indirect RSY



Figure [5.44]: The First 0.1-0.25MByte Runs Long-Term Indirect RSY



Figure [5.45]: The First 0.1-0.5MByte Runs Long-Term Indirect RSY



Figure [5.46]: The First 0.25-0.5MByte Runs Long-Term Indirect RSY

## 5.4 Indirect Length Statistics

This section characterizes same statistics in section [4.1] to watch whether or not these statists have similar behaviors while utilizing distinct host distribution, large and variable data sizes over different time scales. Figures starting with [5.47] and ending by [5.62] show the manners followed by the mean, standard deviation and maximum lengths of 16 runs. The mean length maintains stability at 2 hops and peaks at 4 hops in I's. The length dispersions are not that distinct from each other within the range  $[0.75 \rightarrow 1.5]$  across the entire network. From II's, a common maximum length utilized by indirect routes is 4 and above while the first nodes report one hop due to connectivity failures.



Figure [5.47]: The First 0.05MByte Run Length Statistics



Figure [5.48]: The Second 0.05MByte Run Length Statistics



Figure [5.49]: The Third 0.05MByte Run Length Statistics



Figure [5.50]: The Forth 0.05MByte Run Length Statistics



Figure [5.51]: The First 0.1MByte Run Length Statistics



Figure [5.52]: The Second 0.1MByte Run Length Statistics



Figure [5.53]: The Third 0.1MByte Run Length Statistics



Figure [5.54]: The Forth 0.1MByte Run Length Statistics



Figure [5.55]: The First 0.25MByte Run Length Statistics


Figure [5.56]: The Second 0.25MByte Run Length Statistics



Figure [5.57]: The Third 0.25MByte Run Length Statistics



Figure [5.58]: The Forth 0.25MByte Run Length Statistics



Figure [5.59]: The First 0.5MByte Run Length Statistics



Figure [5.60]: The Second 0.5MByte Run Length Statistics



Figure [5.61]: The Third 0.5MByte Run Length Statistics



Figure [5.62]: The Forth 0.5MByte Run Length Statistics

# 5.5 Direct Delay and Distance

# 5.5.1 Definition

This section investigates whether or not a geographical distance between a pair of nodes in appendix [II] is proportional to packet delay. The referred distance differs from Internet distance, which is an actual distance traversed on wires (i.e., the used geographical distance is the shortest possible Internet distance). The first goal behind understanding this relation is to check for a matching between delay and distance changes. The future objective is to compare such an easy mapping with respect to Internet distance. We calculated distance using the wellknown "Haversine Formula" [50] represented by the following three equations. The great-circle distance between two nodes a and b that have coordinates (lat(a), lon(a)) and(lat(b), lon(b)) respectively equals to the shortest distance D over the earth's surface after ignoring all topographies as follows:

$$R1 = \sin\left(\frac{\Delta \, lat}{2}\right) \sin\left(\frac{\Delta \, lat}{2}\right) + \cos(lat(a))\cos(lat(b))\sin\left(\frac{\Delta \, lon}{2}\right)\sin\left(\frac{\Delta \, lon}{2}\right)$$
[5.1]

$$R2 = 2atan2(\sqrt{R1}, \sqrt{1-R!})$$

$$[5.2]$$

$$D = RR_2$$
 [5.3]

Where  $R_1$  is the square of half the chord length between two ends,  $R_2$  is the angular distance in radians and *R* is earth's radius (i.e., 6,371Km).

#### 5.5.2 Procedure

Assuming all disabled connections have the largest delay value, for each machine, we simply monitored the motion of delay curve with respect to the corresponding distance curve. For a host *i*, we used the delay-distance matching between *i* and *j* as a base-line for the next route between *i* and *j* + 1 where  $1 \le j \le 139$  and  $i \ne j$ . If the next delay on the route to j + 1 has similar reaction as distance with respect to the previous route, we record +1 for *i*, otherwise *i* will have zero point on that route. The percentage of identical movements represents a total matching measure for *i* as shown in figure [5.63]. The only analyzed runs are the first of each load as illustrated in the following four figures.



Figure [5.63]: Delay and Distance Matching. The matching in movement between delay and distance equals  $4/5 \times 100 = 80\%$ 

# 5.5.3 Result

The figures from [5.64] to [5.67] obviously show that for the majority of hosts, delay follows the same behavior of distance with at most 70%.



Figure [5.64]: The First 0.05MByte Run Direct Delay and Distance



Figure [5.65]: The First 0.1MByte Run Direct Delay and Distance



Figure [5.66]: The First 0.25MByte Run Direct Delay and Distance



Figure [5.67]: The First 0.5MByte Run Direct Delay and Distance

# 5.6 Indirect Delay and Distance

The current section analyzes characteristics of indirect delay and distance in order to obtain aspects of convergence among them and compare their matching with delay-distance direct matching of section [5.5]. Figures from [5.68] to [5.71] show the relation between indirect RTT and distance of same runs in section [5.5]. The indirect delay's jumps track same attitudes of distance's jumps with 80% of matching.



Figure [5.68]: The First 0.05MByte Run Indirect Delay and Distance



Figure [5.69]: The First 0.1MByte Run Indirect Delay and Distance



Figure [5.70]: The First 0.25MByte Run Indirect Delay and Distance



Figure [5.71]: The First 0.5MByte Run Indirect Delay and Distance

# 5.7 The Significance of Indirect Delay Routing

# 5.7.1 Overview

In designing any network system, providing an estimate measure to the level of improvement is an essential task. This thesis evaluated the order of magnitude, by which delay can be decreased between any pair of the ping experiment in appendix [II]. We compared the difference in delay between the direct and indirect delays among all pairs as follows.

# 5.7.2 Procedure

We divided the level of improvement into four regions; region "A" contains the magnitude of differences equal or above 100ms, and "B" classifies enhancements in range [10  $\rightarrow$ 100) milliseconds. The third region "H" has cases when the direct route is exactly the shortest one. The forth "F" indicates failures in transforming from being unreachable to be reachable. While utilizing Dijkstra's algorithm with ping, we set delay to be 1E+05ms for any possible failure like unreachability or no responses on transmitted requests. Therefore, enhancements that are close to this value indicate moving from being unreachable via direct routing to be accessible indirectly and classified in region A.

# 5.7.3 Result

Figures [5.72-A], [5.73-A], [5.74-A] and [5.75-A] show the probability of being in region A in I's and B in II's for the performed 4 runs of each data load. The probabilities of being in regions H and F are illustrated in figures [5.72-B], [5.73-B], [5.74-B] and [5.75-B]. The number of successes of each machine (i.e., when recovering from no access via indirect route) and its zoom appear on V's and VI's respectively in figures [5.72-C], [5.73-C], [5.74-C] and [5.75-C]. The stability is obvious among the four runs, as the probability of being in region A is between 0.1 and 0.2 for majority. However, region B has bigger chance than region A with all hosts while H (i.e., also contains indirect routes with less than 10ms enhancements) has the wider gap as

expected. Only five machines always fail to access the entire network, and appear above 0.06 in IV's. Each VI (i.e., a zooming to analogous V) shows that the majority was able to reach at most 4 hosts that were directly unreachable while few others worked indirectly much better.



Figure [5.72-A]: The 0.05MByte Runs Routing Probabilities



Figure [5.72-B]: The 0.05MByte Runs Routing Probabilities



Figure [5.72-C]: The 0.05MByte Runs Routing Reachability



Figure [5.73-A]: The 0.1MByte Runs Routing Probabilities



Figure [5.73-B]: The 0.1MByte Runs Routing Probabilities



Figure [5.73-C]: The 0.1MByte Runs Routing Reachability



Figure [5.74-A]: The 0.25MByte Runs Routing Probabilities



Figure [5.74-B]: The 0.25MByte Runs Routing Probabilities



Figure [5.74-C]: The 0.25MByte Runs Routing Reachability



Figure [5.75-A]: The 0.5MByte Runs Routing Probabilities



Figure [5.75-B]: The 0.5MByte Runs Routing Probabilities



Figure [5.75-C]: The 0.5MByte Runs Routing Reachability

More clearly, in I's of the figures from [5.76] until [5.79], we can see the number of times of being in any of the four regions throughout the entire run (i.e., the summation of the corresponding counts of all machines). The indirect routing was able to achieve more than 100ms delay improvement with at most 3000 route out of the total 19,460 (i.e., 15.5% of the routing cases). Region B was almost the double (i.e., 30% of the total routing of each run) of region A, and the stability of this behavior is quite surprising. Including indirect routes with less than 10ms enhancements beside direct routes suggested by the existing routing, region H represents 46% of total routing in each of the 16 runs at most. The 8 failed hosts (i.e., either on both or one direction) caused being in region F with about 9% failures of the total routing. The second plot of each figure shows the probability of each region across the entire topology.



Figure [5.76]: The 0.05MByte Runs Total Routing Probabilities



Figure [5.77]: The 0.1MByte Runs Total Routing Probabilities



Figure [5.78]: The 0.25MByte Runs Total Routing Probabilities



Figure [5.79]: The 0.5MByte Runs Total Routing Probabilities

# 6. Summary

The chapter looked at some key issues in end-to-end packet delay. The analysis used measurements on two distinct experiments: traceroute and ping. The direct and indirect delay symmetries under the condition of 10ms delay difference were quite high and stable, and remain an area of further research. Performing indirect delay routing was able to enhance Internet performance (i.e., 15% at maximum), and avoid many failures.

# **CHAPTER 6**

# PACKET LOSS

#### 1. Overview

Packet loss is an inevitable consequence in cyberspace. Measuring packet loss over a broadband connection is not a straightforward task due to difficulties in anticipating a loss source and direction. The residual traffic may not sustain to be acceptable by voice or video applications for instance, and performance as a result can be severely degraded. Packet loss is a subsequent attitude of any path susceptible to, congestion, outage, overload, slowness or traffic prioritization. Reliable protocols like TCP usually mange packet loss automatically by retransmitting unacknowledged segments. The price of multiple retransmissions, however, can be even worse. External assistance can have a vantage with connection oriented and connectionless protocols that have no loss self-addressing mechanisms. In this chapter of our study we actively examined packet behavior more closely with advantage of the most readily available tools: ping and iperf.

The material of the chapter will be classified almost in a similar manner like packet delay since the two metrics are additive quantities. Section [4.1] will describe a concept of direct loss symmetry while section [4.2] will explain indirect loss symmetry. The physical indirect route symmetry will be analyzed in sections [4.3] and [5.1] with both ping and iperf. The relation between direct packet loss and a cumulative loss occurs on indirect shortest delay route is simply analyzed in [4.5]. We investigated a relation between distance and packet loss in section [5.2] and [5.3]. In section [5.4] we studied the significance of indirect loss routing.

#### 2. The Importance of Measuring Loss

The emergence of new data network applications caused estimating packet loss to

become increasingly necessitate for a network manager to accurately determine the impact of such applications. Recently, applications are able to be adaptable to the exploding nature of connections through timeout and retransmission functions of upper layer protocols in order maintain proper bandwidth allocation [47]. Unfortunately, this emergence impacts Internet scalability and heterogeneity, and enforces applications like online streaming to be more stochastic in manner. Consequently, an imperative threshold of performance requires careful understanding of a traffic dynamics before deploying such applications.

### **3.** The Loss Experiments

In this study we conducted our loss traces by using two distinct utilities: ping and iperf. Regarding ping, packet loss was estimated simultaneously with packet delay on a broadband network consists of 140 machines classified in appendix [II]. The usage of ping was operated in a flooding mode (i.e.,  $\delta = 5$ ms) while calculating the percentage of lost packets. The 16 ping runs with their detailed classifications were exactly as outlined in sections [3] and [5] of chapter [5].

The second utility, iperf was operated using five different loads attached to 12 distinct transmission rates as in table [6.1]. The first 12 runs of iperf were performed for a purpose of calculating conditional available bandwidth as will be discussed in chapter [7] while an extra two runs were conducted for monitoring measurements stability of bandwidths carried by highest transmission rates: 800Mbps and 400Mbps.The measured losses over a network of 122 hosts were calculated for distinct rates and sizes of UDP streams.

		Transmission Rate [Mbps]											
		0.5	4	6	8	10	20	40	80	100	200	400	800
Data Load [MByte]	0.5	*	-	-	-	-	-	-	-	-	-	-	-
	2	-	*	-	-	-	-	-	-	-	-	-	-
	3	-	-	*	-	-	-	-	-	-	-	-	-
	5	-	-	-	*	-	*	*	-	-	-	-	-
	10	-	-	-	-	*	-	-	*	*	*	*	*

Table [6.1]: The Ordered Probing Rates

Although it is not an easy effort to accurately characterize packet loss over wide networks due to hardware and software heterogeneities (e.g., a node may have a slow network interface cards, out-of-date processor or slow Internet connection), we attempted to achieve a profound accuracy by sending unified sizes of data ranged in  $[0.5 \rightarrow 10MByte]$ , and study 14,762 routes assigned per individual rate. For about 5 hosts fail in sending a required bulk within a specific time period, we forced their transmissions for smaller loads instead. The reported packet loss is based on actual amount of data received by the second terminal. Each server sends back to client five statistics: packet loss, total receiving time, actual received load and jitter.

The iperf was utilized under a supervision of the probing algorithm mentioned in chapter [4] over an approximate four days period. After that, we started analyzing 44,286 of a total of 206,668 traces per day (i.e., 3 runs associated to distinct transmission speeds). We used a random intervals in the range  $[5 \rightarrow 7]$  among performed runs while ascending transmission speed. The only two repetitive runs via 400Mbps and 800Mbps were performed first.

#### 4. Ping Analysis

In this section, we analyzed almost similar applicable relations to these studied in chapter [5]. First we defined a measure called a degree of loss symmetry to evaluate packet loss over 19,460 long-distance routes. The defined degree is also to compare packet loss over a forward

route  $[s \rightarrow d]$  and parallel reverse direction  $[d \rightarrow s]$ . This chapter further discusses an indirect relation between packet delay and loss, and characterizes indirect length statistics and physical route symmetries.

# 4.1 Direct Loss Symmetry

# 4.1.1 Definition

The defined direct loss symmetry degree - LSY(i) for a given machine *i* is the percentage of the out direct connections whose loss percentage differences with their reverses are smaller or equal to 1%.

#### 4.1.2 Procedure

Using the same concept outlined in chapter [5] section [5.1] without a concrete background on choosing the utilized 1% threshold as distinct applications may require distinguishable demands. The defined degree can be not related to direct route mapping, on which we hold further examination to future. The main objective is in understanding routing functionality among paired nodes in terms of packet loss.

#### 4.1.3 Result

Figures from [6.1] to [6.4] show that a major portion of the network has degrees around 80%. There are 17 machines with degrees below 40% irrespective of load size. Few hosts start to decrease their degrees as the transmitted load increased like the first 5 hosts. The motivating achievement with direct packet loss symmetry is in its similarity to the degree of direct delay symmetry presented in chapter [5] for a considerable number of hosts. This may support the fact that as a packet delay varies between parallel connections, packet loss also follows similar attitude (i.e., we postpone this claim for future investigation).



Figure [6.1]: The 0.05MByte Run Direct *LSY* Stability



Figure [6.2]: The 0.1MByte Run Direct LSY Stability



Figure [6.3]: The 0.25MByte Run Direct *LSY* Stability



Figure [6.4]: The 0.5MByte Run Direct LSY Stability

# 4.2 Indirect Loss Symmetry

Similar to direct loss symmetry studied in section [4.1], the analysis of indirect routes is to understand same aspects of packet loss occurred on forward and backward indirect routes. Figure [6.5], [6.6], [6.7] and [6.8] show a solid stability of indirect loss symmetry on 16 runs. The common peak degree value is perfectly identical to the corresponding indirect delay degree in section [5.2] of chapter [5], and the remaining behaviors of the four loads are almost indistinguishable. For majority of nodes, indirect loss routing adds an extra 10% to direct degrees, and contributed heavily in bringing loss values of additional paired connections (i.e., approximately 14 pairs) to be further close of each other.



Figure [6.5]: The 0.05MByte Run Indirect LSY Stability



Figure [6.6]: The 0.1MByte Run Indirect LSY Stability



Figure [6.7]: The 0.25MByte Run Indirect LSY Stability



Figure [6.8]: The 0.5MByte Run Indirect LSY Stability

# 4.3 Short and Long-Term Route Symmetries

## 4.3.1 Detention

This concept is exactly as described in section [5.3] of chapter [5]. The only difference is in examining smallest packet loss connections instead. The performed analysis on intermediate machines and hops shaped in indirect routes is to understand traffic directing among pairs.

## 4.3.2 Result

Figures from [6.9] to [6.16] display the short-term comparisons of the first and second run of each utilized load. The zero percent of hop symmetry indicates that at least one route of the pair is indirect. The 100% counts a number of routes in a range [6,000  $\rightarrow$  8,000], and indicates that the reversed short route between a pair is perfectly used by the long one as displayed in I's. The 100% node matching peaks at 9,000 pairs out of a total 9,730 pairs. Figures from [6.17] to [6.22] show the long-term route symmetry of the same combinations of runs used



in chapter [5] section [5.3]. A similar duplication as with delay appears on hop and node symmetries, which again preserves Internet characteristics.

Figure [6.9]: The First 0.05MByte Run Short-Term Indirect RSY



Figure [6.10]: The Second 0.05MByte Run Short-Term Indirect RSY



Figure [6.11]: The First 0.1MByte Run Short-Term Indirect RSY



Figure [6.12]: The Second 0.1MByte Run Short-Term Indirect RSY



Figure [6.13]: The First 0.25MByte Run Short-Term Indirect RSY



Figure [6.14]: The Second 0.25MByte Run Short-Term Indirect RSY



Figure [6.15]: The First 0.5MByte Run Short-Term Indirect RSY



Figure [6.16]: The Second 0.5MByte Run Short-Term Indirect RSY



Figure [6.17]: The First 0.05-0.1MByte Runs Long-Term Indirect RSY



Figure [6.18]: The First 0.05-0.25MByte Runs Long-Term Indirect RSY



Figure [6.19]: The First 0.05-0.5MByte Runs Long-Term Indirect RSY


Figure [6.20]: The First 0.1-0.25MByte Runs Long-Term Indirect RSY



Figure [6.21]: The First 0.1-0.5MByte Runs Long-Term Indirect RSY



Figure [6.22]: The First 0.25-0.5MByte Runs Long-Term Indirect RSY

## 4.4 Indirect Route Length Statistics

## 4.4.1 Definition

This section uses the same collection of measures: mean, standard deviation, minimum and maximum length of 139 indirect loss routes. The analysis is only for the first two runs of each utilized data load. The main objective is to examine if there is any similarity between the statistics of packet loss and delay as described in section [5.4] of chapter [5].

#### 4.4.2 Result

Disjointed by an average interval of 8 hours between distinct loads and 2 hours between identical loads, figures from [6.23] to [6.30] demonstrate manners of the above statistics. The mean length oftentimes stable at 1 hop as I's display. The standard deviation is ranged at 0.5 for all except 5 hosts that report zeros due to reachability failures similar to packet delay. The maximum length is ranged in  $[2 \rightarrow 6]$  hops after excluding the 5 hosts suffer from unreachability.

The first important finding is that indirect loss routing is very conservative in hop utilization comparing to indirect delay routing. The second observation is that as the load amplified above 0.05MB, hosts start to use longer routes than 2 hops.



Figure [6.23]: The First 0.05MByte Run Length Statistics



Figure [6.24]: The Second 0.05MByte Run Length Statistics



Figure [6.25]: The First 0.1MByte Run Length Statistics



Figure [6.26]: The Second 0.1MByte Run Length Statistics



Figure [6.27]: The First 0.25MByte Run Length Statistics



Figure [6.28]: The Second 0.25MByte Run Length Statistics



Figure [6.29]: The First 0.5MByte Run Length Statistics



Figure [6.30]: The Second 0.5MByte Run Length Statistics

## 4.5 Indirect Delay and Loss

#### 4.5.1 Definition

In this part, we combined an analysis on packet delay and loss in a modified indirect routing scheme. The main aim is to compare a cumulative packet loss on indirect shortest delay routes with a corresponding direct loss. Consequently, the amount of degradation of the indirect cumulative loss can be a correlation indicator between delay and packet loss.

#### 4.5.2 Result

Using the first run of each load, figures from [6.31] to [6.34] show the direct packet loss in I's and the analogous cumulative packet loss on shortest delay route in II's. Although accurate judgment on this regard requires further research, the preliminary view shows a weak relation among the direct and indirect packet loss. The plot I of figure [6.31] shows that out of the 19,460 direct Internet connections, only 14,745 routes maintained packet loss below 2% (i.e., valid routes). For a unified direction pair of routes, we considered the routes to be different if their losses are distinct (i.e., a direct packet loss is not equal to a corresponding indirect cumulative packet loss). The two routes otherwise are categorized as exactly equal. As clear from the following figures, the number of valid routes extracted by Internet is always higher than the number of indirect cumulative packet loss valid routes. Moreover, the number of direct and indirect valid routes starts to decay when transmitting heavy data loads. Similarly, the number of equal routes was in a descending range [14,147  $\rightarrow$  13,367], and correspondingly the number of different routes follows an ascending range [5,313  $\rightarrow$  6,093] when sending heavy streams.



Figure [6.31]: The First 0.05MByte Run Direct and Indirect Loss



Figure [6.32]: The First 0.1MByte Run Direct and Indirect Loss



Figure [6.33]: The First 0.25MByte Run Direct and Indirect Loss



Figure [6.34]: The First 0.5MByte Run Direct and Indirect Loss

## 5. Iperf Analysis

This part will examine different characteristics on UDP measurements carried out via iperf. The following assessments of packet loss depend on transmission rate between serverclient pairs instead of data weight. We evaluated the relation between direct packet loss and distance as a function of sending speed before analyzing the magnitude of indirect loss routing. We concluded by characterizing the short and long-term route symmetries.

## 5.1 Short and Long-Term Route Symmetries

### 5.1.1 Definition

Finally on packet loss, we analyzed short and long-term physical indirect symmetries as in previous sections. We studied runs associated to rates: 0.5Mbps, 4Mbps, 80Mbps and 400Mbps as a selection of distinct speeds to determine hop and node symmetry.

#### 5.1.2 Result

Figures from [6.35] to [6.38] display the short-term hop symmetry. From I's, we noticed a decrement on short-term symmetries as the transmission rate increased. The node symmetry started to descend as sending rates exceeded 80Mbps as displayed in II's while a dramatic decay from about 5,500 matched pairs to 1,000 pairs with high rates. Figures from [6.39] to [6.44] cover hop and node long-term symmetries. The second remark is that the long-term hop symmetry is much affected than the node symmetry as transmission speed increased. The small gaps between compared rates tend to result high matching among unidirectional indirect pair of routes (i.e., indirect loss routing start to act asymmetrically as altering transmission rate).



Figure [6.35]: The 0.5Mbps Run Short-Term Indirect RSY



Figure [6.36]: The 4Mbps Run Short-Term Indirect RSY



Figure [6.37]: The 80Mbps Run Short-Term Indirect RSY



Figure [6.38]: The 400Mbps Run Short-Term Indirect RSY



Figure [6.39]: The 0.5-4Mbps Runs Long-Term Indirect RSY



Figure [6.40]: The 0.5-80Mbps Runs Long-Term Indirect RSY



Figure [6.41]: The 0.5-400Mbps Runs Long-Term Indirect RSY



Figure [6.42]: The 4-80Mbps Runs Long-Term Indirect RSY



Figure [6.43]: The 4-400Mbps Runs Long-Term Indirect RSY



Figure [6.44]: The 80-400Mbps Runs Long-Term Indirect RSY

#### 5.2 Direct Packet Loss and Distance

This section focuses on understanding the relation between direct packet loss and topographical distance as a function of transmission rate. The main objective behind this analysis is to explore the amount of correlation between direct packet loss and distance as performed with packet delay. The utilized distance again is derived via equation [5.3]. The following figures from [6.45] to [6.56] illustrate results of sensitive altering criteria mentioned in chapter [5] section [5.5], and applied on the delay-distance relation for 12 ordered rates. The matching percent is about 20% on low transmission rates. High rates elevate matching to maximum 50%. Holding the pervious assumption (i.e., as real distances increase, Internet distances also increase) indicates an obvious weakness in correlation comparing with the corresponding delay-distance relation in chapter [5].



Figure [6.45]: The 0.5Mbps Run Direct Loss and Distance



Figure [6.46]: The 4Mbps Run Direct Loss and Distance



Figure [6.47]: The 6Mbps Run Direct Loss and Distance



Figure [6.48]: The 8Mbps Run Direct Loss and Distance



Figure [6.49]: The 10Mbps Run Direct Loss and Distance



Figure [6.50]: The 20Mbps Run Direct Loss and Distance



Figure [6.51]: The 40Mbps Run Direct Loss and Distance



Figure [6.52]: The 80Mbps Run Direct Loss and Distance



Figure [6.53]: The 100Mbps Run Direct Loss and Distance



Figure [6.54]: The 200Mbps Run Direct Loss and Distance



Figure [6.55]: The 400Mbps Run Direct Loss and Distance



Figure [6.56]: The 800Mbps Run Direct Loss and Distance

#### 5.3 Indirect Packet Loss and Distance

The following is an analysis of the relation between packet loss over indirect smallest loss routes and indirect total distance in order to obtain a clear comparison with the previous criteria in section [5.2]. The next 12 figures from [6.57] to [6.68] summarize a stable but weaker reaction between the change in packet loss and distance in virtual routing using 12 different rates as matching is maximum at 15% with small rates, and on average less than 10% with high rates.



Figure [6.57]: The 0.5Mbps Run Indirect Loss and Distance



Figure [6.58]: The 4Mbps Run Indirect Loss and Distance



Figure [6.59]: The 6Mbps Run Indirect Loss and Distance



Figure [6.60]: The 8Mbps Run Indirect Loss and Distance



Figure [6.61]: The 10Mbps Run Indirect Loss and Distance



Figure [6.62]: The 20Mbps Run Indirect Loss and Distance



Figure [6.63]: The 40Mbps Run Indirect Loss and Distance



Figure [6.64]: The 80Mbps Run Indirect Loss and Distance



Figure [6.65]: The 100Mbps Run Indirect Loss and Distance



Figure [6.66]: The 200Mbps Run Indirect Loss and Distance



Figure [6.67]: The 400Mbps Run Indirect Loss and Distance



Figure [6.68]: The 800Mbps Run Indirect Loss and Distance

## 5.4 The Significance of Indirect Loss Routing

## 5.4.1 Definition

Importantly, we characterized provided enhancements via indirect packet loss routes. Following a similar manner to delay's significance analysis in chapter [5] section [5.7] we considered four different regions A, B, H and F to represent indirect loss routes' impact. The region A counts successes in reducing packet loss below the 2% threshold (i.e., also accommodates further decrementing on loss below threshold originally). The second identifies cases when a minimized loss percentage is not below 2%. The third represents holding acceptable existing Internet routines. The last region identifies failures in recovering routes that have high packet losses.

#### 5.4.1 Result

For combination of four distinct transmission rates, figures [6.69-A], [6.70-A] and [6.71-A] display A's and B's probabilities. The probabilities of H and F are identified in figures [6.69-B], [6.70-B] and [6.71-B]. The superiority of indirect routing (i.e., represented via A) over Internet (i.e., represented via H) at high transmission rates is obvious. The A's probabilities start to increase from the range  $[0.1 \rightarrow 0.2]$  to  $[0.4 \rightarrow 1]$  while B's degradation is from  $[0.8 \rightarrow 0.9]$  down to  $[0 \rightarrow 0.6]$ . Both B's and F's probabilities remains close to zero for the majority of hosts except for a few that experience persistent failures when incrementing the transmission rate.



Figure [6.69-A]: The 0.5-8Mbps Runs Routing Probabilities



Figure [6.69-B]: The 0.5-8Mbps Runs Routing Probabilities



Figure [6.70-A]: The 10-80Mbps Runs Routing Probabilities



Figure [6.70-B]: The 10-80Mbps Runs Routing Probabilities



Figure [6.71-A]: The 100-800Mbps Run Routing Probabilities



Figure [6.71-B]: The 100-800MbpsRuns Routing Probabilities

More obvious, figures from [6.72] to [6.74] display regional counters (i.e., a total number of routes in a region) across complete runs. As more connections start to fall in A when increasing transmission rate, the linearity in A's amelioration and B's degradation is clear in I's. The confusion on linearity in figure [6.73] is due to utilizing different loads as outlined in table [6.1]. With 8Mbps for instance, by 29% in total, indirect routing was able to shift direct packet loss below 2%, or even further if originally below. Ordering 80Mbps if possible, increased total enhancements to 46% while probing at 800Mbps results 61% success. The II's show associated total probabilities over the entire network.



Figure [6.72]: The 0.5-8Mbps Runs Total Routing Probabilities



Figure [6.73]: The 10-80Mbps Runs Total Routing Probabilities



Figure [6.74]: The 100-800Mbps Runs Total Routing Probabilities

#### 6. Summary

Packet loss is a central dynamic to be analyzed as function of forwarding rate. Loss routes seem to be more identical with Internet and indirect packet loss routing. This offers a space to investigate a possibility of facing unavoidable single-source of loss that cannot avoid loss when sensing with high rates on both directions. The linear manner between overlay and underlay performances indicates an open field of magnificent improvements on Internet.

# **CHAPTER 7**

#### BANDWIDTH

#### 1. Overview

Estimating bandwidth (i.e., bottleneck, available, achievable or utilized) is an essential function especially for adaptive flows as a step toward Bandwidth Aware Applications (BAA). Researchers look differently toward this metric based on their application's demand either control or data communication. The bottleneck bandwidth is a static property in packet-switched networks, but the available, achievable and utilized bandwidths are dynamic quantities as they change over short or long time-scales, and adjust position oftentimes. Applications, therefore, have divergent concerns as their flows start to traverse into a network. Active applications for instance: online video or audio streams and interactive telnet transactions require adequate broadcasting speeds meanwhile indirect data transfers such as electronic mails do not necessitate elevated forwarding rates, and consider bandwidth as surmountable factor oftentimes. Before deploying active applications, a developer might be required to characterize different network concerns related to bandwidth such as locating the quick path and calculating the speed of a particular path. Figure [7.1] shows the clear picture of the following bandwidth terms:

- Link capacity: the physical capacity C(i) is associated with a link *i*.
- Bottleneck bandwidth: for a route k, the associated bottleneck bandwidth BnB(k) = min(C(i)), where  $i \in [1 \rightarrow n]$  and n is the number of links involved in the route k.
- Link utilization: for a link i, U(i) ranges in  $[0 \rightarrow 1]$ .
- Available bandwidth: for a route k, the associated available bandwidth is AvB(k) = min((1 U(i))C(i)) where  $i \in [1 \rightarrow n]$ .
- Throughput: the achievable speed *ThrB*, by which a sender grantees successful devilry.
- Bulk Transfer Rate (BTR): the amount of transferred data per unit time (i.e., the quotient of division between a load size and its total transfer time).



Figure [7.1]: The Bandwidth Terminology

Discussion and analysis on TCP throughput is outlined in section [5]. We deeply analyzed our approach to measure UDP conditional available bandwidth (i.e., controlled by packet loss threshold), and discussed different relations between bandwidth and other metrics in section [6]. Finally, a brief analysis on ping BTR and associate data loss appears in section [7].

## 2. Bottleneck and Available Bandwidths

Practically, there is a clear distinction between bottleneck bandwidth and available bandwidth. For any end-to-end route, the upper bound of the available bandwidth equals to bottleneck bandwidth provided by the slowest element on the route. The lower bound, however, can decay with orders of magnitude down to zero when having a particular link under full utilization or persistent failure. In addition, the available bandwidth can be relatively small when having high packet loss or large delay in case of handshaking protocols. The inequality [7.1] expresses the later explanation mathematically.

$$0 \le AvB(t) \le BnB \tag{7.1}$$

Where AvB(i) is the available bandwidth at a time t. The later terminology bounded by the

expression in [7.1] shows how fast a connection can possibly transmit data; in other words, how fast a connection should be while preserving network stability [2] so that a high throughput can be achieved. However, as the main difference between the available bandwidth and bottleneck bandwidth is time scale, timing the bottleneck bandwidth (i.e., becomes a dynamic behavior) will result the available bandwidth at specific time.

In networking, guaranteeing enough available bandwidth does not ensure successful data delivery. The other interesting quantity though, achievable bandwidth can notify a sender about a devilry status in case of utilizing connection oriented protocols such as TCP. The throughput can range within the limits of the following inequality:

$$0 \le ThrB(t) \le AvB(t) \tag{7.2}$$

Where ThrB(t) is the achievable flow speed at time t. In the study, in order to determine available bandwidth, we placed a packet loss threshold equals to 2% of a transmitted UDP stream. This threshold can involve dropping, reordering, duplicating and corrupting, and slightly higher can be considered as acceptable [1] for many applications.

Unfortunately, the majority of tools measure peak rates as described in chapter [3] do not provide robust estimations on large-scale networks. A number of them failed to work on Planetlab aggregates such as sprobe due to a requirement of supper-root privileges. Others have insufficient accuracies such as clink and pipechar while utilities like pathload consumes long time in reporting a route statistics. Spruce, however, is used only for partial estimation beside requiring prior knowledge about link capacity.

### 3. What's Missing?

Yet, the future has not come for networks that can preserve flows on behalf of their sources by handling loss, errors and duplications. Undoubtedly, any transferred packet has a certain probability to be delayed, duplicated, corrupted or even dropped somewhere while continuing a journey. That location, therefore, or the ones before in case of corruption can have a copy before applying any decision on the packet. Missing such a design caused every forwarding element in between a source-destination pair to be blind with respect to all remaining resources. This blindness forced any estimator to operate on the concept of Poisson modularity on both short and long time-scales rather than the concept of self-similarity as recent studies claimed that the behavior of a switched traffic seems to be a self-similar process "looks the same" across four or five orders of magnitude (e.g., beyond several hours of time-scale, there is a clear evidence about daily periodicity caused by human traffic patterns [53]). Within a small time scale than the previous, analysis shows burstiness across aggregates while the Poisson representation has shown fast smoothness over time.

Therefore, having information from intermediate routers can allow estimating bandwidths more accurately. In large-scaled networks, characterizing the entire structure, and calculating available bandwidth with a normal manner (i.e., subtracting link utilization from capacity) can be inefficient. The simple instead, is by filling in the residual capacity carefully until an imposed condition exists.

## 4. Bandwidth Impediments

Resources resides on an end-to-end chain have distinguishable abilities to push packets into their upstream ports or withdraw packets out of downstream ports. This nature can be affected directly by a physical limitations (e.g., slow processor, the frequency bandwidth of a wire) or more complex shortages (e.g., time required for looking addresses up and forwarding packets or slowness in reading data out of a transport stack) [1]. In addition to the previous, data compression, asymmetric routing and measurement time can significantly affect bandwidth [36].

### 5. Achievable Bandwidth

Technically, the term throughput is connected to handshaking protocols rather than

connectionless protocols (i.e., because of the existence of other limitations such as the TCP window protocol that affects the flow wheatear or not there is an available bandwidth). The term throughput is another metric that is highly correlated to both physical and software limitations, and accordingly can be controlled via available bandwidth and bottleneck bandwidth as outlined in the inequality in [4.2]. Although available bandwidth and throughput might appear similar, in this study we consider the TCP throughput as the amount of packets that can be correctly delivered to its receiver over time unit while the available bandwidth is the smallest residual capacity over a path.

### 5.1 TCP Throughput

With a pure TCP as a reliable connection oriented protocol, the throughput is adjustable on behalf of the user to grantee better performance by not overloading destination buffers and path resources. That is automatically implemented by prohibiting any manned interference in order to prevent bottlenecks from functioning at 100% utilization [28]. Attempts to reach a high throughput quickly may accomplish long packet queues, and resultant upper bound end-to-end delays and an increase in packet losses. The later undesirable appurtenances will certainly lead to a large number of retransmissions. The TCP as window-based protocol runs two meticulous but not perfect conversations between parties; the first called end-to-end flow control (i.e., an end-toend protocol to avoid overwhelming a receiver's application), and the second referred as end-toend congestion control (i.e., an end-to-end protocol uses a combination of techniques to avoid overloading a network) [28].

By implementing the two protocols, estimating throughput is not a difficult task with the existing tools such as iperf since two ends can agree on a flow control window, and a sender can manipulate MSS. Therefore, end users can monitor their throughputs according to formula [7.3].

$$ThrB(t) = \frac{cwnd(t)}{RTT(t)}$$
[7.3]

Where cwnd(t) is the control widow size at time instance *t*. Therefore, *cwnd* should be equivalent at least to an estimated RTT multiplied by the bottleneck bandwidth thereby ensuring less time consumption in achieving maximum throughput on the well-known schematic relation between the later and the offered load. In the lack of knowledge, the bottleneck bandwidth can be assumed to be the slowest Network Interface Card (NIC) of the two ends [48]. We applied a simple TCP mechanism to estimate throughout by using iperf to fire TCP flows over a real time experiment consists of 120 machines. This network is almost identical to the used in determining UDP packet loss in chapter [6] section [5] and UDP bandwidth in section [6]. This experiment has a total of 14,280 direct routes (i.e., operative and inoperative). The most fundamental issue for TCP is in deciding the control window size that controls the amount of transmitted data into network at particular time [50]. If a choice was small, the sender will be idle at times in waiting for less number of acknowledgments, and provide a degraded performance. The theoretical guess, however, is the bottleneck bandwidth delay product as mentioned earlier in equation [7.3] by approximating initial throughput -  $ThrB(\hat{t})$  to equal the BnB of a particular route.

With k equals to 61 as in table [4.1], each machine probed the 120 daemons randomly until fulfilling the entire network. A 2MByte load was used to derive throughput samples every 2 seconds. The first objective behind functioning iperf in TCP mode is to understand the functionality of throughput out of four delayed network pictures. The second is to evaluate the minimum transmission rate for iperf in UDP mode (i.e., we did not input these bandwidth starters to UDP mode rather than monitoring the difference between them and the effectiveness of the suggested transmission rates in table [6.1]). The reason is to avoid any estimation errors possibly introduced by the used 0.5MByte window size in the TCP scenario. Unfortunately, iperf as any existing tool has associated errors and failures that affect measurements usually, and introduce a type of insufficiency; the following are the common failures were encountered in order of significance:

- In TCP mode, hosts such as "planetlab1.cs.stevens-tech.edu" have intention to down active background daemons, and result "write-read failed" message.
- Hosts like "ricepl-1.cs.rice.edu" do not accept connections, and respond by a "time-out" error when attempting to establish TCP sessions.
- Hosts such as "planetlab2.eecs.jacobs-university.de" always fail in UDP mode to receive reports from servers.

According to [37], measuring the pace, at which iperf sends packets correctly depends on both the buffer size at the TCP-IP stack and the TCP acknowledgment notion itself. The TCP-IP stack will only accept segments when not full. That is obvious when choosing large window (i.e., the initial reported throughput is usually high), and after a while this initial becomes less in later reports. Therefore, the theoretical throughput can be not achievable due to the TCP nature, delay, asymmetry, CPU power and other constraints [37].

The following is an example of instructing iperf on "pl1.pku.edu.cn" to estimate throughput toward "planetlab2.umassd.edu". Every 2 seconds iperf reports a throughput sample for each transferred piece of 2Mbyte load, and the final 1.06Mbps is the achievable throughput. The file transfer time is 15.9 seconds. The remaining unreported portion of data "0.47MByte" (i.e., a subtraction between 2MByte and previous cumulative delivered sub-loads) required an extra 3.9 seconds, which is higher than previous 2 seconds for 0.54MByte. In some measurements, an extra time is required although a summation of earlier sub-loads is 2MByte. This deficiency is due the claim in [37] outlined previously about TCP acknowledging when a control window already received 2MByte, but the sender has not been notified yet about a

portion that requires a retransmission.

\_\_\_\_\_

[pl1.pku.edu.cn]\$ iperf -c planetlab2.umassd.edu -w 0.5M -i 2 -m -n 2M

-----

[3] local 162.105.205.21 port 41721 connected with 134.88.5.253 port 5001

[ id] Interval	Transfer	Bandwidth
[ 3] 0.0- 2.0 sec	0.26 MByte	1.08 Mbps
[ 3] 2.0- 4.0 sec	0.23 MByte	0.95 Mbps
[ 3] 4.0- 6.0 sec	0.30 MByte	1.25 Mbps
[ 3] 6.0- 8.0 sec	0.54 MByte	2.26 Mbps
[ 3] 8.0-10.0 sec	0.01 MByte	0.03 Mbps
[ 3] 10.0-12.0 sec	0.19 MByte	0.79 Mbps
[ 3] 0.0-15.9 sec	2.00 MByte	1.06 Mbps

[ 3] MSS size 1448 bytes (MTU 1500 bytes, Ethernet)

\_\_\_\_\_

The final throughput is computed as an average of the first six throughputs. Although each sample appears to be calculated similarly as BTR, the final throughput did not follow this rule to be 1.006Mbps (i.e., no clear identification on this issue has been achieved).

We studied the measured throughput stability over one day period in section [5.1.1]. In section [5.1.2], we will demonstrate the relation between the reported throughput and FTT, which is the time required by TCP-IP stack to fulfill withdrawing 2MByte load from a sender in section [5.1.3].

## 5.1.1 Throughput Stability

This section analyzes stability of TCP throughputs distributed over a 24 hors period. The

utilized window size for every connection was 0.5MByte. Throughputs were reported every 2 seconds time interval. The entire experiment was forced to send 2MByte, and report the maximum segment size on each flow. Figure [7.2] demonstrates the first run reported throughputs for 120 machines. Figures from [7.3] until [7.5] show delayed runs spaced by an interval in  $[4 \rightarrow 6]$  hours.

In the following four figures, we found that approximately 28 direct routes can provide throughputs between 200Mbps and 800Mbps while majority fall below 100Mbps. In general, the four pictures of TCP throughputs are indistinguishable in terms of magnitude except few irregularities, and heavy concentration is below 10Mbps.



Figure [7.2]: The First Run Throughput



Figure [7.3]: The Second Run Throughput



Figure [7.4]: The Third Run Throughput



Figure [7.5]: The Forth Run Throughput

## 5.1.2 Direct Throughput Statistics

For each machine we calculated the mean, standard deviation, minimum and maximum direct throughputs of 119 out routes to remaining hosts. Figures from [7.6] to [7.9] show the above statistics per machine and ascending mean and standard deviation across network in I's while the minimum and maximum per host throughputs are in II's. The sorted mean is almost identical in the four pictures, and peaks at 29Mbps. The standard deviation on the other side is below 20Mbps for about 100 machines, and roses up in  $[20 \rightarrow 80Mbps]$  for the remaining hosts. Hosts that afford throughputs above 400Mbps were usually constant. The minimum throughputs were always on zero except for one host (i.e., had been successfully connected to all nodes), and zero minimum occurred because of having hosts mentioned earlier that always fail in establishing TCP sessions.



Figure [7.6]: The First Run Statistics



Figure [7.7]: The Second Run Statistics



Figure [7.8]: The Third Run Statistics



Figure [7.9]: The Forth Run Statistics

### 5.1.3 Throughput and FTT

The aim of analyzing FTT on multi-sessions TCP connection is to check dissimilarity between direct mean throughput and indirect minimum achievable mean throughput over shortest FTT route while sending 2Mbytes over all direct connections. The required FTT in achieving final throughput can be more important than throughput. Since indirect single-session TCP flow is a function of new RTT and *BnB*, analyzing indirect throughput over such a flow requires careful and real implementation of overlay routing. The study in [51] envisioned a method called TCP pipelining by splitting a high delay connection to independent end-joined TCP flows in order to partially minimize RTT and increase cwnd and ThrB consequently so that all hops' RTTs are less than the direct connection's RTT. Due time limit, we were not able to analyze RTTs of the TCP experiment in appendix [III]. Instead, we used FTT as an amplified delay measure to evoke indirect TCP pipelining. The analysis is under assumption of having persistent and synchronized connections (i.e., small time spaces among joined flows).

Figure [7.10] shows a relation between direct mean throughput of 119 routes per host, indirect minimum mean throughput over the shortest FTT route and required length. The remaining figures from [7.11] to [7.13] display that indirect minimum mean throughput is slightly improved for majority of hosts. The maximum indirect length was widely held on four hops while between six and seven with few machines.



Figure [7.10]: The First Throughput and FTT



Figure [7.11]: The Second Run Throughput and FTT



Figure [7.12]: The Third Run Throughput and FTT



Figure [7.13]: The Forth Run Throughput and FTT

### 6. UDP Available Bandwidth

On contrary to TCP, there is no in advance agreement between end users who use a UDPconnectionless channel. Therefore, measuring bandwidth should be done carefully by a suitable tool on behalf of this unreliable media. The significant parameter associate with bandwidth an application can watch is loss rate, while in TCP-IP stack many parameters can indicate throughput behavior such as a change in congestion window (i.e., decides the number of unacknowledged packets in network) [28] and the frequency of retransmissions in addition to loss rate.

Iperf was used in UDP mode for measuring available bandwidth despite overwhelming routes via an iperf client when using high transmission rates. To partially overcome the later downside, we conducted our measurements only over short durations as possible; practically, we utilized loads between 0.5MByte and 10MByte on server-client tics as in table [6.1] so to not disturb other traffics. The objective of the current analysis is predict a maximum UDP transmission rate that caused an acceptable packet loss over an end-to-end segment (i.e., Horizontal Available Bandwidth - HAvB). Observing UDP maximum sending rates was under the condition of resulting at most 2% packet loss on every route of 14,762 direct broadband connections. The UDP probing for all used rates was organized via the probing algorithm classified in chapter [4].

#### 6.1 Practical Approaches

Beside two used approaches described in section [5.1] for TCP and earlier in section [6] for UDP, there are slightly different methodologies that can be used for estimating available bandwidth at a given time instance as summarized below.

In [48], a practicable procedure to estimate available bandwidth by starting with an initial UDP transfer rate slightly below a reported TCP throughput, probing for 10 seconds and

reporting interim results every one second. If a measured packet loss is below a stated threshold, a new measurement with slightly increased rate will start (i.e., may not be always correct since a loss rate at time t can exceed the threshold comparing to a loss at  $(t + \Delta t)$ ). Using such an incremental procedure can waste time as supposed to test a desired bandwidth first and reduce the rate if loss exceeds threshold. Moreover, examining bandwidth with such a prototype will cause heavy loads over small intervals. In this chapter, however, our objective was not only measuring conditional available bandwidth but also functioning in a time sensitive operation to find possible alternative routing at  $(t + \Delta t)$ .

## 6.2 The Affordable Transmission Rate and Loss

The Affordable Transmission Rate - AfTR represents a maximum transmission speed that can be offered via a host, and close to an ordered rate. This section analyzes 206,668 traces resultant via distinct ordered rates (i.e., can be limited by CPU, and not be achievable as not used in any validation process rather than classifying measurements). The details about extracting traces were outlined in chapter [6] section [3]. Figure [7.14] shows a number of direct routes that suffer from packet loss greater than 2% for each affordable transmission rate.



Figure [7.14]: The Number of Routes Suffer High Loss

# 6.3 The Horizontal Available Bandwidth

## 6.3.1 Definition

This section illustrates our approach in estimating a long-term Horizontal Available Bandwidth - HAvB for 14,762 direct routes after extracting required measurements via iperf and applying equation [7.4] once on the first 12 runs in order to calculate HAvB per each direct route (i.e., HAvB(i, j) for every pair (i, j) directly attached).  $HAvB(i,j) = \arg \max(AfTR(i,j) \mid tr(k)) \mid \min(l(i,j))$ 

$$\begin{bmatrix} (AfTR(1,1)|tr(1),l(1,1)|tr(1)) & \cdots & (AfTR(1,1)|tr(k),l(1,1)|tr(k)) \\ \vdots & \ddots & \vdots \\ (AfTR(1,n)|tr(1),l(1,n)|tr(1)) & \cdots & (AfTR(1,n)|tr(k),l(1,n)|tr(k)) \end{bmatrix}$$

$$\begin{bmatrix} (AfTR(n,1)|tr(1),l(n,1)|tr(1)) & \cdots & (AfTR(n,1)|tr(k),l(n,1)|tr(k)) \\ \vdots & \ddots & \vdots \\ (AfTR(n,n)|tr(1),l(n,n)|tr(1)) & \cdots & (AfTR(n,n)|tr(k),l(n,n)|tr(k)) \end{bmatrix} \end{bmatrix}$$

$$\begin{bmatrix} (AfTR(n,n)|tr(1),l(n,n)|tr(1)) & \cdots & (AfTR(n,n)|tr(k),l(n,n)|tr(k)) \\ \vdots & \ddots & \vdots \\ (AfTR(n,n)|tr(1),l(n,n)|tr(1)) & \cdots & (AfTR(n,n)|tr(k),l(n,n)|tr(k)) \end{bmatrix} \end{bmatrix}$$

$$\begin{bmatrix} (AfTR(n,n)|tr(1),l(n,n)|tr(1)) & \cdots & (AfTR(n,n)|tr(k),l(n,n)|tr(k)) \\ \vdots & \ddots & \vdots \\ (AfTR(n,n)|tr(1),l(n,n)|tr(1)) & \cdots & (AfTR(n,n)|tr(k),l(n,n)|tr(k)) \end{bmatrix} \end{bmatrix}$$

Both *i* and *j* are in range  $[1 \rightarrow n]$  where *n* is the number of hosts in experiment. The term AfTR(i,j)|tr(k), is an affordable transmission rate at the k<sup>th</sup> demanding speed (i.e., not every host is able to probe at a given transmission rate - tr(k)) on the direct route  $i \rightarrow j$ . The l(i,j)|tr(k) represents the packet loss between a source *i* and destination *j*. The main objective behind the analysis is to understand full view about possible bandwidths in order to perform quick and close decision on demanding rates' limitation an application can request.

## 6.3.2 Result

Figure [7.15] shows estimated HAvBs of the studied iperf experiment after applying equation [7.4]. From the figure, HAvB is highly concentrated below the 100Mbps and then between 100Mbps and 400Mbps. This indicates that a sufficient number of routes can provide high bandwidths when utilized as parts of indirect paths. There is also an adequate number that have bandwidths in range [400  $\rightarrow$  800Mbps] while maintaining 2% loss threshold.



Figure [7.15]: The Estimated Horizontal Available Bandwidth

## 6.4 The Vertical Available Bandwidth

## 6.4.1 Definition

Following the full characterization in section [6.3], a second important analysis while studying UDP bandwidth is to identify per machine *i* a maximum direct available bandwidth route that causes no more than 2% packet loss (i.e., referred as short-term direct Vertical Available Bandwidth - VAvB(i,k), where *k* is an index of down machine on a route). For each host, if VAvB is relatively high comparing to remaining 120 routes, then its route should have better chance to be involved via indirect routes as traffic exits. The second defined related measure to VAvB called Degree of Vertical Available Bandwidth  $D_{VAvB}(i)$  for a host *i*. The degree represents for machine *i* a number of direct routes that have short-term direct available bandwidths close in value to VAvB(i,k) (i.e.,  $0 \le VAvB(i,k) - AfTR(i,j) \le 30$  and  $l(i,j) \le$ 2%, where  $1 \le j \le 120$  and  $j \ne k$ ).

## 6.4.2 Result

Figures from [7.16] to [7.21] display the *VAvB* in I's, associate packet loss in II's and  $D_{VAvB}$  in III's for transmission rates from 40Mbps to 800Mbps. For 40Mbps run, majority of hosts have at least one *VAvB* equals to demanding rate (i.e., 40Mbps) while few others can only provide 10Mbps at maximum. Regarding  $D_{VAvB}$ , except four all hosts have 100 out of 120 routes that afford short-term direct bandwidths within the above condition. The situation was almost identical with 80Mbps many of previous hosts have ability to probe at 80Mbps instead of 40Mbps while providing below threshold packet loss. The degree started degrading as demanding rates rose above 80Mbps, and *VAvB* is less than the demanding rate with some hosts as clear from the last three runs. The plot II indicates any machine that has zero *HAvB*.



Figure [7.16]: The 40Mbps Run VAvB and D<sub>VAvB</sub>



Figure [7.17]: The 80Mbps Run VAvB and D<sub>VAvB</sub>



Figure [7.18]: The 100Mbps Run VAvB and D<sub>VAvB</sub>



Figure [7.19]: The 200Mbps Run VAvB and D<sub>VAvB</sub>



Figure [7.20]: The 400Mbps Run VAvB and D<sub>VAvB</sub>



Figure [7.21]: The 800Mbps Run VAvB and D<sub>VAvB</sub>

## 6.5 The Horizontal Available Bandwidth Symmetry

Generally, this analysis is an attempt to understand whether Internet works symmetrically in terms of bandwidth or not (i.e., available bandwidths on forward-reverse pair are close to each other). We defined the degree of symmetry of the Horizontal Available Bandwidth - HAvBSY(i)for a machine *i*. The degree refers to percentage of paired direct connections that have available bandwidth difference less than 0.2Mbps. Figure [7.22] shows degrees of utilized 122 machines composed in UDP experiment. We conclude that most machines have different HAvBs on both directions except three that act symmetrically.



Figure [7.22]: The HAvB Degree of Symmetry

## 6.6 The Maximum *AfTR* and Loss

A limited stability measure was performed on two runs of highest demanding rates: 400Mbps and 800Mbps. We simply compared similarity of the maximum affordable transmission speeds "AfTR" between the two runs and their four days delayed pictures. The similarity was on basis of checking the maximum affordable rate of each machine and the attached packet loss. Figures [7.23] to [7.26] show this similarity among the four runs. The conclusion is that oftentimes the maximum AfTR is attached to high loss except few hosts that can probe with maximum AfTR with an acceptable packet loss.

Figures from [7.27] to [7.30] have the full behavior of the AfTR on all direct paths along with associated packet loss, and how it was concentrated below 100Mbps. The comparison of 400Mbps runs indicates that the relation between AfTR and packet loss was almost identical after 4 days and with 800Mbps runs as well. This manner provides additional support to the



stability of the estimated *HAvB* in section [6.3].

Figure [7.23]: The First 400Mbps Run Maximum AfTR and Loss



Figure [7.24]: The Second 400Mbps Run Maximum AfTR and Loss



Figure [7.25]: The First 800Mbps Run Maximum AfTR and Loss



Figure [7.26]: The Second 800Mbps Run Maximum AfTR and Loss



Figure [7.27]: The First 400Mbps Run AfTR Full View



Figure [7.28]: The Second 400Mbps Run AfTR Full View



Figure [7.29]: The First 800Mbps Run AfTR Full View



Figure [7.30]: The Second 800Mbps Run AfTR Full View

## 6.7 The Horizontal Available Bandwidth and Jitter

Unfortunately, due to Planetlab instability, we were not able to study the relation between the HAvB and packet delay (i.e., comparing a direct HAvB between a source-destination pair with the minimum HAvB on the indirect short delay route of the pair). We instead, included another important metric, which is jitter in order to build a long-term indirect jitter routing and therefore, calculate the associate indirect horizontal available bandwidth. Jitter is mentioned in the Service-Level Agreement (SLA) should be within specific constrains is applicable to Dijkstra's algorithm [3].

Jitter is the amount of variation in delay, and usually associated with unreliable connections because they consistently report back same delay over and over again [38]. Figure [7.31] shows the relation between the estimated direct HAvB and indirect minimum HAvB on the smallest indirect jitter path. Except on few regions, the indirect HAvB was almost similar to

direct HAvB derived in section [6.3], and that possibly can indicate the tight relation between packet loss and jitter while attempting to bring jitter further down. Figure [7.32] illustrates the triple relation with the previous two and the minimum HAvB on the long-term smallest loss route. This was implemented to understand the change on HAvB if an application has interest to further diminish long-term packet loss (i.e., smaller than the 2% if possible). The minimum HAvB on both jitter and loss indirect routes was highly identical, which further support that jitter and loss are strongly correlated in their occurrence.



Figure [7.31]: The Direct and Indirect Jitter Minimum HAvBs



Figure [7.32]: The Direct-Indirect Jitter and Loss Minimum HAvBs

## 6.8 The Short-Term *AfTR* and Packet Loss

In this section we are interested to investigate the amount of short term improvements per single run on AfTR and packet loss. The reason behind performing the following analysis is to investigate the possibility to forward traffic with indirect short-term bandwidth while preserving packets from suffering high loss. Figures from [7.33] to [7.36] demonstrate this relation for the highest four demanding rates and number of hops required for any change. From these plots we can see the significant improvement in terms of packet loss for each run. The indirect packet loss is almost decreased to zero for large number of routes over each transmission rate while utilizing seven indirect hops at maximum as III's show. The minimum indirect AfTR however, was shifted to either 100Mbps or between zero and 30Mbps.We noticed also some indirect bandwidths above 100Mbps as well. The finding can benefit short-term indirect loss routing with rates up to 10Mbps.



Figure [7.33]: The 800Mbps Run Direct-Indirect AfTR and Loss



Figure [7.34]: The 400Mbps Run Direct-Indirect AfTR and Loss



Figure [7.35]: The 200Mbps Run Direct-Indirect AfTR and Loss



Figure [7.36]: The 100Mbps Run Direct-Indirect AfTR and Loss

### 7. The Ping Bulk Transfer Rate

Finally on bandwidth, we simply analyzed the first, second, third and fourth run respectively of each utilized load's BTR, by which ping is forwarding packets. The BTR was calculated as ratio between the total transferred load and time. Figures [7.37], [7.38], [7.39] and [7.40] show stability of BTR over an approximate period of 30 hours. These figures were spaced by a random interval within the Range [ $8 \rightarrow 10$ ]. The BTR was almost averaged at 0.5Mbps over the majority of 19,460 direct routes except 12 clear clusters that have BTRs a little below. The minority of routes forwarded the traffic with rates above the 0.5Mbps. The loss behavior was almost following the same shape despite the size of the used load with high concentration slightly above the zero level. The CDF's in III's show the cumulative distribution of the number of direct paths that suffer from loss above 2% per machine. For example, if this event equals to 20, then the corresponding CDF is showing the cumulative probability of the number of machines that have the at most 20 routes experience more than 2% of packet loss. From the plot III in figure [8.38], we can see that some of 140 machines have a maximum number of 19 routes that cause loss above the given threshold.



Figure [7.37]: The First 0.05MByte Run BTR and Loss



Figure [7.38]: The Second 0.1MByte Run BTR and Loss


Figure [7.39]: The Third 0.25MByte Run BTR and Loss



Figure [7.40]: The Forth 0.5MByte Run BTR and Loss

### 8. Summary

The dynamism of bandwidth caused estimating available bandwidth to be a continuous challenge in networking. The analysis shows an enormous amount of distinguished available bandwidths that can be managed indirectly for Internet advancement. Improving bandwidth requires applications to set time constrains for overlay routing to benefit from each single measurement.

## **CHAPTER 8**

### **CONCLUSION AND FUTURE WORK**

### 1. Conclusion

#### **1.1** Packet Delay Analysis

Traceroute: The hop utilization appears to be stable over a period of 24 hours with identical light flows. The stability analysis indicates that short indirect routes are more stable, and have higher intention to accommodate additional hops. However, long routes start to shorten themselves usually. The full mapping indicates that 60% of routes are indirect, and can provide distinct levels of improvements. The overall link utilization is semi-Gaussian with mean of 20 links. Surprisingly, all indirect length sequences found a cross entire experiments follow; with traceroute for instance,  $2 \leq indirect \, length \, in \, hops \leq 9$  and  $\min(length) = 2 \, hops$ .

Ping: we found that indirect delay routing acts more heavily when sending heavy loads than light loads (i.e., 12.8% with 0.05MByte and 15.4% with 0.25Mbyte). The hop utilization is quite stable and refers to an open space of indirect delay routing with an average mean length of 2 hops. We found that the maximum indirect length clearly diminish on average with traffics above 0.05MByte. Performing indirect routing results an approximate 90% symmetry degree, which caused additional 14 pairs of forward-back routes per host to act symmetrically in terms of delay. Moreover, indirect routing caused delay-distance relation to be more overlapped by extra 10%. The overall structural short-term hop and node symmetries tend to be more stable and independent of load size.

### 1.2 Packet Loss Analysis

Ping: the defined loss symmetry degrees are similarly concentrated as delay's degrees, but further examination is required with lower thresholds to establish a solid argument. The general performed route symmetry shows that indirect loss routing consumes fewer hops than delay routing (e.g., 64% difference with the first 0.05Mbyte and 43% with the second 0.05Mbyte), and that elevated the percentage of hop and node symmetries as direct routes become more involved.

Iperf: indirect loss routing as a function of transmission rate can offer significant enhancements in routing. From section [5.4] of chapter [6], we can notice a perfect linear attitude between region A and B. We found that indirect packet loss avoidance is a function of the transmission rate (e.g., in case of 800Mbps, packet loss is reduced indirectly below 2% with probability 0.62 while Internet routing has a success probability of 0.26). The overall short-term route symmetry has a light opposite relation with transmission rate while the long-term symmetry more is adjustable (i.e., highly asymmetric).

### **1.3 Bandwidth Analysis**

Iperf: in TCP Mode, throughput samples indicate about 35 hosts suffer high dispersion in range  $[10 \rightarrow 80Mbps]$ , and the majority is clustered below 10 Mbps. The achieved mean throughput is approximately 5Mbps and slightly higher with fewer hosts. In UDP mode, the packet loss is almost linear with transmission rate as indicated in chapter [6] section [5.4]. The majority of estimated 14,762 long-term available bandwidths are below 100Mbps while a second concentration is in  $[10 \rightarrow 400Mbps]$ . The short-term per host available bandwidth and its degree degrade when ordering above 200Mbps. The long-term available bandwidth acts asymmetrically with symmetry degrees below 2%. We found that high transmission rates (i.e., 400 and 800Mbps) are attached to high packet loss except 47 hosts. Moreover, simultaneously minimizing packet loss in the process of determining conditional bandwidths also minimizes jitter. In close, short- term loss elimination can be perfectly implemented with indirect minimum available bandwidths smaller than desired rates.

### 2. Future Work

The utilized probing algorithm and tool are core stones in any estimation system. Based

on our investigation of available tools in literature, we intend to develop a new utility that combines a robust pre agreement handshaking and window-based probing algorithm as an estimation procedure for Internet dynamics. The tool will operate in TCP and UDP light modes, and will evaluate performance metrics: packet delay and loss, conditional available bandwidth, throughput and possibly others. The utility will be used afterward to actively feed a metric-based data base.

The main aim behind our study is to explore the way to design a WAN application prototype as a performance enhanced overlay routing that can analyze the previous data base carefully and quickly in order to update routing tables of participating nodes. The virtual layer will be on top of the core protocol "BGB" as we have noticed during the period of our research major deficiencies and failures occur on long-haul routes more often. In UDP mode, the extra routing will be directed to minimize delay within regions A and B that were classified in chapter [5] section [5.7]. The second desired task is to perform linear rate-based loss avoidance according to the outcomes in chapter [6] section [5.4]. Lastly, the indirect forwarder will classify routes on short and long-term bandwidth bases to join pairs via indirect maximum-minimum available bandwidth connections or pre stated delay or loss routes. In TCP mode, we are interested to design a single-session indirect TCP routing after investigating negatives of the over-head addressing. This is because of the severe lack of such studies on dominant protocols like TCP (i.e., barely non-existent). We expect TCP indirect routing to necessitate perfect characterizing of indirect routes' cwnds and RTTs as key components in improving achievable bandwidth.

The ON-OFF Control Routing (OFCR) will be operated in two phases: active and idle routing. The active mode will guide OFCR via an updated primary data repository composed of packet delay, loss rate and bandwidth. The inactive mode will function when temporal failing in measuring a route (e.g., indirect routing experience failures in updating a desired minimum delay route). The off mode can utilize pre-determined metrics such as link utilization. The conducted results in chapter [5] section [5.6] are quite engorging to be extended further on exploring a timely assistant delay-distance model for OFCR. Based on the achieved stability of the performed offline indirect routing, this off-control will function overlay over short-time scales (i.e., among failures). The second off-control metric will be the defined degree delay and loss symmetries in chapter [5] and [6].

The next will be in studying per rout a possible number of better alternatives that reside between the shortest and Internet routes. The objective of analyzing such a problem is to build reliability measure of OFCR beside the studied stability in chapters [5], [6] and [7]. From available measurements, we look to invent a dependent and theoretical probabilistic failure characterization as a second reliability measure. This model will assign a failure probability (e.g., a packet loss failure when exceeding a stated threshold) to every indirect hop, and study correlation among hop failures.

The similarity between packet delay and loss symmetry degrees requires further investigation. Measuring correlation among symmetry degrees can be used to establish an optimal performance routing that overcomes major side-effects of packet switching. Measuring packet delay using light loads carried via traceroute causes a dense hop utilization comparing with heavy loads that ping transmitted. We are interested to explore the load's extent that causes a shift of overlay routing toward Internet more quickly in order to avoid such shortcomings in the futuristic OFCR. We further intend to understand whether a relation between direct and indirect delay symmetries and the structural design of the corresponding direct and indirect or not. The validating of this relation can assist in determining link symmetries among routes more efficiently as no proficient and fast mapping tool is available. The planned OFCR will raise bandwidth decisions either on short-term and long-term bases as illustrated in chapter [7]. The short-term routing will utilize determine routes that have VAvB to send traffics (e.g., online video and voice are possible applications) while implementations like file transfer will be operated in a long-term phase.

# APPENDICES

# **APPENDIX I**

## TRACEROUTE EXPERIMENT

IP Domain	Site	Country
108.58.13.206	Optimum Online, Mineola (NY)	USA
124.124.247.3	Reliance Communications, New Delhi (Delhi)	India
128.10.19.53	Purdue University, West Lafayette (IN)	USA
128.111.52.58	University of California - Santa Barbara, Santa Barbara (CA)	USA
128.111.52.63	University of California - Santa Barbara, Santa Barbara (CA)	USA
128.119.41.210	University of Massachusetts, Amherst (MA)	USA
128.119.41.211	University of Massachusetts, Amherst (MA)	USA
128.135.164.193	University of Chicago, Chicago (IL)	USA
128.143.6.130	University of Virginia, Charlottesville (VA)	USA
128.151.65.101	University of Rochester, Rochester (NY)	USA
128.151.65.102	University of Rochester, Rochester (NY)	USA
128.163.142.20	University of Kentucky, Lexington (KY)	USA
128.163.142.21	University of Kentucky, Lexington (KY)	USA
128.187.223.211	Brigham Young University, Provo (UT)	USA
128.208.3.180	University of Washington, Seattle (WA)	USA
128.220.251.50	Johns Hopkins University, Baltimore (MD)	USA
128.223.8.114	University of Oregon, Eugene (OR)	USA
128.227.56.82	University of Florida, Gainesville (FL)	USA

128.233.252.12	University of Saskatchewan, Saskatoon (SK)	Canada
128.42.142.45	Rice University, Houston (TX)	USA
128.59.20.226	Columbia University, New York (NY)	USA
128.8.126.79	University of Maryland, Hyattsville (MD)	USA
129.10.120.194	Northeastern University, Boston (MA)	USA
129.105.15.38	Northwestern University, Evanston (IL)	USA
129.110.125.52	University of Texas at Dallas, Richardson (TX)	USA
129.130.252.141	Kansas State University, Manhattan (KS)	USA
129.15.78.30	University of Oklahoma, Norman (OK)	USA
129.15.78.31	University of Oklahoma, Norman (OK)	USA
129.186.205.79	Iowa State University, Ames (IA)	USA
129.21.30.116	Rochester Institute of Technology, Rochester (NY)	USA
129.237.161.193	University of Kansas, Lawrence (KS)	USA
129.237.161.194	University of Kansas, Lawrence (KS)	USA
129.82.12.187	Colorado State University, Fort Collins (CO)	USA
129.82.12.188	Colorado State University, Fort Collins (CO)	USA
130.127.39.152	Clemson University, Anderson (SC)	USA
130.127.39.153	Clemson University, Anderson (SC)	USA
130.161.40.154	Delft University of Technology Network, Delft (Zuid-	Netherlands
	Holland)	
130.195.4.69	Victoria University of Wellington, Wellington	New Zealand

130.216.1.23	Auckla, Auckland	New Zealand
130.75.87.84	Gottfried Wilhelm Leibniz Universitaet Hannover, Hanover (Niedersachsen)	Germany
130.92.70.254	University of Berne, Berne (Bern)	Switzerland
131.179.150.70	University of California - Los Angeles, Los Angeles (CA)	USA
131.179.150.72	University of California - Los Angeles, Los Angeles (CA)	USA
131.247.2.248	University of South Florida, Tampa (FL)	USA
132.239.17.226	University of California - San Diego, La Jolla (CA)	USA
132.72.23.10	Ben Gurion University Network, Yafo (Tel Aviv)	Israel
133.1.74.162	Osaka University, Toyonaka (Osaka)	Japan
133.11.240.56	University of Tokyo, Tokyo (Tokyo)	Japan
133.15.59.2	Toyohashi University of Technology, Toyohashi (Aichi)	Japan
134.121.64.4	Washington State University, Pullman (WA)	USA
134.226.52.34	Trinity College, Dublin (Dublin)	Ireland
134.226.52.35	Trinity College, Dublin (Dublin)	Ireland
134.76.81.91	GWD Goettingen, Göttingen (Niedersachsen)	Germany
136.159.220.42	University of Calgary, Calgary (AB)	Canada
137.132.80.110	National University of Singapore, Singapore	Singapore
137.148.16.11	Cleveland State University Computer Services, Cleveland (OH)	USA
137.165.1.113	Williams College Campus, Williamstown (MA)	USA

137.99.11.86	University of Connecticut, Storrs Mansfield (CT)	USA
137.99.11.87	University of Connecticut, Storrs Mansfield (CT)	USA
138.15.10.56	NEC Laboratories America, Sunnyvale (CA)	USA
138.238.250.157	Howard University, Washington (DC)	USA
139.19.142.6	Max-planck-institut FuerInformatik, Saarbruecken, Saarbrücken (Saarland)	Germany
140.112.107.82	Taiwan Academic Network, Taipei (T'ai-pei)	Taiwan
140.112.42.158	Taiwan Academic Network, Taipei (T'ai-pei)	Taiwan
140.247.60.123	Harvard University, Cambridge (MA)	USA
141.161.20.33	Georgetown University, Washington (DC)	USA
141.20.103.211	Humboldt-Universitaet Zu Berlin, Berlin	Germany
141.213.4.201	University of Michigan - College of Engineering, Ann Arbor (MI)	USA
141.219.252.133	Michigan Technological University, Houghton (MI)	USA
142.104.21.245	University of Victoria, Victoria (BC)	Canada
143.107.111.235	Universidade De Sao Paulo, São Paulo (Sao Paulo)	Brazil
143.215.131.198	Georgia Institute of Technology, Atlanta (GA)	USA
143.89.49.73	Hong Kong University of Science and Technology, Central District	Hong Kong
143.89.49.74	Hong Kong University of Science and Technology, Central District	Hong Kong

Table [I]: Traceroute Experime	ent (cont'd).
--------------------------------	---------------

145.99.179.147	SURFNET, Utrecht	Netherlands
146.57.249.99	University of Minnesota, Minneapolis (MN)	USA
147.102.224.227	National Technical University of Athens, Athens (Attiki)	Greece
149.169.227.131	Arizona State University, Tempe (AZ)	USA
149.43.80.22	Colgate University, Hamilton (NY)	USA
150.65.32.68	Japan Advanced Institute of Science and Technology, Nomi (Kyoto)	Japan
152.3.138.7	Duke University, Durham (NC)	USA
155.225.2.72	The Citadel, North Charleston (SC)	USA
155.246.12.163	Stevens Institute of Technology, Hoboken (NJ)	USA
155.246.12.164	Stevens Institute of Technology, Hoboken (NJ)	USA
155.98.35.5	University of Utah, Salt Lake City (UT)	USA
156.56.250.227	Indiana University, Bloomington (IN)	USA
158.130.6.253	University of Pennsylvania, Philadelphia (PA)	USA
158.130.6.254	University of Pennsylvania, Philadelphia (PA)	USA
160.193.163.106	Osaka City University, Osaka (Gifu)	Japan
160.36.57.172	The University of Tennessee Health Science Center, Memphis (TN)	USA
160.36.57.173	The University of Tennessee Health Science Center, Memphis (TN)	USA
164.107.127.12	Ohio State University, Columbus (OH)	USA

Table [I]: Traceroute Experiment	(cont'd).
----------------------------------	-----------

165.230.49.114	Rutgers University, Piscataway (NJ)	USA
169.229.50.10	University of California at Berkeley, Oakland (CA)	USA
169.229.50.4	University of California at Berkeley, Oakland (CA)	USA
170.140.119.70	Emory University, Atlanta (GA)	USA
192.1.249.138	BBN Communications, Cambridge (MA)	USA
192.138.213.236	Suffolk University, Boston (MA)	USA
192.138.213.238	Suffolk University, Boston (MA)	USA
192.16.125.11	Swedish Institute of Microelectronics, Kista (Stockholms Lan)	Sweden
192.16.125.12	Swedish Institute of Microelectronics, Kista (Stockholms Lan)	Sweden
192.33.90.67	ETHZ, Swiss Federal Institute of Technology, Zürich	Switzerland
192.42.83.250	Illinois Institute Of Technology, Chicago (IL)	USA
192.42.83.253	Illinois Institute Of Technology, Chicago (IL)	USA
193.167.187.186	Helsinki Institute For Information Technology, Helsinki (Southern Finland)	Finland
193.63.58.70	University College of London - Computer Science Department, London	UK
194.29.178.13	Warsaw University of Technology, Warsaw	Poland
195.113.161.83	CESNET, Prague (Hlavni Mesto Praha)	Czech Republic

195.116.53.14	Telekomunikacja Polska S.A., Polska (Kujawsko-Pomorskie)	Poland
198.163.152.229	Telecommunications Research Labs, Winnipeg (MB)	Canada
198.82.160.220	Virginia Polytechnic Institute and State University, Blacksburg (VA)	USA
198.82.160.221	Virginia Polytechnic Institute and State University, Blacksburg (VA)	USA
198.82.160.238	Virginia Polytechnic Institute and State University, Blacksburg (VA)	USA
198.82.160.239	Virginia Polytechnic Institute and State University, Blacksburg (VA)	USA
199.26.254.69	George Mason University, Fairfax (VA)	USA
200.0.206.169	Cooperación Latino Americana de Redes Avanzadas, Montevideo	Uruguay
200.0.206.203	Cooperación Latino Americana de Redes Avanzadas, Montevideo	Uruguay
200.10.150.253	Escuela Politecnica Del Litoral, Guayaquil (Guayas)	Ecuador
200.19.159.35	Fundação de Desenvolvimento da Pesquisa	Brazil
202.23.159.52	Doshisha University, Nishinotoindori (Kyoto)	Japan
203.178.133.10	High Speed Network Lab - WIDE Project, Communication Research Lab	Japan
203.178.133.11	High Speed Network Lab - WIDE Project, Com. Lab	Japan

204.8.155.226	Boston University, Boston (MA)	USA
206.117.37.7	Los Nettos, Los Angeles (CA)	USA
206.12.16.155	Bcnet, Vancouver (BC)	Canada
206.207.248.38	University Of Arizona, Tucson (AZ)	USA
207.197.40.251	Nevada System Of Higher Education, Reno (NV)	USA
210.123.39.102	Korea Telecom, Seoul (Seoul-T'ukpyolsi)	South Korea
210.125.84.44	Kwangju Institute of Science and Technology, Kwangju	South Korea
211.68.70.36	Beijing Telecommunication & Research Academy, Dian	China
	(Beijing)	
213.131.1.101	Department of Computational Methods, Moscow (Moscow	Russia
	City)	
216.48.80.14	University Of Ottawa, Ottawa (ON)	Canada
219.243.208.62	China Education and Research Network, Beijing (Beijing)	China
220.245.140.196	TPG Internet Pty Ltd, Frankston (Victoria)	Australia
220.245.140.197	TPG Internet Pty Ltd, Frankston (Victoria)	Australia
72.36.112.78	University of Illinois, Urbana (IL)	USA
82.179.176.44	Moscow Institute of Electronic Engineering, Moscow	Russia
	(Moscow City)	
87.236.232.153	Jordanian Universities Network L.L.C., Irbid (Al Balqa)	Jordan

Table [I]: Traceroute Experiment

# **APPENDIX II**

## PING EXPERIMENT

IP Domain	Site	Country
108.58.13.206	Optimum Online, Mineola (NY)	USA
128.10.19.52	Purdue University, West Lafayette (IN)	USA
128.10.19.53	Purdue University, West Lafayette (IN)	USA
128.111.52.63	University of California - Santa Barbara, Santa Barbara (CA)	USA
128.138.207.54	University of Colorado, Boulder (CO)	USA
128.143.6.130	University of Virginia, Charlottesville (VA)	USA
128.151.65.101	University of Rochester, Rochester (NY)	USA
128.187.223.212	Brigham Young University, Provo (UT)	USA
128.208.3.184	University of Washington,- Seattle (WA)	USA
128.208.4.199	University of Washington, Seattle (WA)	USA
128.220.251.50	Johns Hopkins University, Baltimore (MD)	USA
128.220.251.52	Johns Hopkins University, Baltimore (MD)	USA
128.223.8.113	University of Oregon, Eugene (OR)	USA
128.227.150.12	University of Florida, Gainesville (FL)	USA
128.42.142.41	Rice University, Houston (TX)	USA
128.59.20.227	Columbia University, New York (NY)	USA
128.59.20.228	Columbia University, New York (NY)	USA
128.6.192.156	Rutgers University, Hillsborough (NJ)	USA

128.8.126.111	University of Maryland, College Park (MD)	USA
128.8.126.78	University of Maryland, College Park (MD)	USA
128.83.122.143	University of Texas at Austin, Austin (TX)	USA
129.10.120.193	Northeastern University, Boston (MA)	USA
129.10.120.194	Northeastern University, Boston (MA)	USA
129.105.15.38	Northwestern University, Evanston (IL)	USA
129.107.35.131	University of Texas at Arlington, Arlington (TX)	USA
129.108.202.10	University of Texas at El Paso, El Paso (TX)	USA
129.110.125.51	University of Texas at Dallas, Richardson (TX)	USA
129.130.252.141	Kansas State University, Manhattan (KS)	USA
129.186.205.77	Iowa State University, Ames (IA)	USA
129.22.150.78	Case Western Reserve University, Cleveland (OH)	USA
129.237.161.194	University of Kansas, Lawrence (KS)	USA
129.74.74.19	University of Notre Dame, Notre Dame (IN)	USA
129.82.12.187	Colorado State University, Fort Collins (CO)	USA
129.93.229.139	University of Nebraska - Lincoln, Lincoln (NE)	USA
130.127.39.153	Clemson University, Anderson (SC)	USA
130.194.252.9	Monash University, Richmond (Victoria)	Australia
130.195.4.68	Victoria University of Wellington, Wellington	New Zealand
130.216.1.23	Auckla, Auckland	New Zealand
130.253.21.123	University of Denver, Denver (CO)	USA

130.49.221.40	University of Pittsburgh, Pittsburgh (PA)	USA
131.123.34.36	Kent State University, Kent (OH)	USA
131.193.34.21	University of Illinois, Chicago (IL)	USA
131.193.34.38	University of Illinois, Chicago (IL)	USA
131.247.2.245	University of South Florida, Tampa (FL)	USA
131.247.2.248	University of South Florida, Tampa (FL)	USA
132.239.17.226	University of California - San Diego, La Jolla (CA)	USA
132.68.237.36	Technion Network, Haifa	Israel
132.72.23.10	Ben Gurion University Network, Yafo (Tel Aviv)	Israel
133.68.253.243	Nagoya Institute of Technology, Nagoya (Aichi)	Japan
133.9.81.164	Waseda University	Japan
134.88.5.253	University of Massachusetts Dartmouth, North Dartmouth (MA)	USA
136.145.115.194	University of Puerto Rico, San Juan	Puerto Rico
136.159.220.40	University of Calgary, Calgary (AB)	Canada
137.132.80.105	National University of Singapore, Singapore	Singapore
137.148.16.11	Cleveland State University Computer Services, Cleveland (OH)	USA
137.165.1.115	Williams College Campus, Williamstown (MA)	USA
137.99.11.86	University of Connecticut, Storrs Mansfield (CT)	USA
138.15.10.55	NEC Laboratories America, Sunnyvale (CA)	USA

138.238.250.157	Howard University, Washington (DC)	USA
139.80.206.133	University of Otago, Dunedin (Otago)	New Zealand
140.109.17.181	Academia Sinica, Taipei (T'ai-pei)	Taiwan
140.112.107.80	Taiwan Academic Network, Taipei (T'ai-pei)	Taiwan
140.112.42.158	Taiwan Academic Network, Taipei (T'ai-pei)	Taiwan
140.114.79.231	Taiwan Academic Network, Hsinchu (T'ai-wan)	Taiwan
140.123.230.249	National Chung Cheng University, Taipei (T'ai-pei)	Taiwan
140.192.249.204	Depaul University, Chicago (IL)	USA
141.161.20.32	Georgetown University, Washington (DC)	USA
141.20.103.210	Humboldt-Universitaet Zu Berlin, Berlin	Germany
141.212.113.179	University of Michigan - College of Engineering, Ann Arbor (MI)	USA
141.213.4.202	University of Michigan - College of Engineering, Ann Arbor (MI)	USA
141.219.252.133	Michigan Technological University, Houghton (MI)	USA
142.104.21.241	University of Victoria, Victoria (BC)	Canada
143.107.111.235	Universidade De Sao Paulo, São Paulo (Sao Paulo)	Brazil
143.215.131.197	Georgia Institute of Technology, Atlanta (GA)	USA
143.215.131.206	Georgia Institute of Technology, Atlanta (GA)	USA
143.89.49.74	Hong Kong University of Science and Technology, Central District	Hong Kong

145.99.179.147	SURFNET, Utrecht	Netherlands
147.46.240.165	Seoul National University, Seoul (Seoul-T'ukpyolsi)	South Korea
149.43.80.22	Colgate University, Hamilton (NY)	USA
150.140.140.93	University of Patras Network, Patras (Akhaia)	Greece
150.65.32.66	Japan Advanced Institute of Science and Technology, Nomi (Kyoto)	Japan
152.14.93.140	North Carolina Research and Education Network, Raleigh (NC)	USA
155.223.52.3	EGE University, Izmir	Turkey
155.225.2.72	The Citadel, North Charleston (SC)	USA
155.246.12.163	Stevens Institute of Technology, Hoboken (NJ)	USA
156.56.250.227	Indiana University, Bloomington (IN)	USA
157.92.44.102	Universidad Nacional De Buenos Aires, Buenos Aires (Distrito Federal)	Argentina
158.130.6.253	University of Pennsylvania, Philadelphia (PA)	USA
160.193.163.106	Osaka City University, Osaka (Gifu)	Japan
160.36.57.173	The University of Tennessee Health Science Center, Memphis (TN)	USA
162.105.205.21	Peking, Beijing (Beijing)	China
164.73.47.244	Servicio Central De Informatica, Montevideo (Montevideo)	Uruguay
165.230.49.119	Rutgers University, Piscataway (NJ)	USA

169.226.40.4	State University of New York, Albany (NY)	USA
169.229.50.18	University of California at Berkeley, Oakland (CA)	USA
170.140.119.69	Emory University, Atlanta (GA)	USA
170.140.119.70	Emory University, Atlanta (GA)	USA
192.16.125.12	Swedish Institute of Microelectronics, Kista (Stockholms Lan)	Sweden
192.33.90.68	ETHZ, Swiss Federal Institute of Technology, Zürich	Switzerland
192.42.83.253	Illinois Institute of Technology, Chicago (IL)	USA
193.63.58.70	University College of London - Computer Science Department, London	UK
194.29.178.14	Warsaw University of Technology, Warsaw	Poland
195.113.161.83	CESNET, Prague (Hlavni Mesto Praha)	Czech Republic
195.116.53.14	Telekomunikacja Polska S.A., Polska (Kujawsko-Pomorskie)	Poland
198.128.56.12	Lawrence Berkeley National Laboratory, Berkeley (CA)	USA
198.133.224.149	Computer Sciences Department University of Wiscons, Madison (WI)	USA
198.175.112.105	Intel Corporation - Pleasant Hill, Fair Oaks (CA)	USA
198.7.242.42	Ocean State Higher Education and Administration Network, Providence (RI)	USA

198.82.160.221	Virginia Polytechnic Institute and State University, Blacksburg (VA)	USA
198.82.160.239	Virginia Polytechnic Institute and State University, Blacksburg (VA)	USA
200.0.206.137	Cooperación Latino Americana de Redes Avanzadas, Montevideo	Uruguay
200.0.206.203	Cooperación Latino Americana de Redes Avanzadas, Montevideo	Uruguay
200.129.132.19	Associação Rede Nacional De Ensino E Pesquisa, Belém (Para)	Brazil
200.132.1.4	Associação Rede Nacional De Ensino E Pesquisa, Porto Alegre	Brazil
200.17.202.194	Universidade Federal Do Paraná, Curitiba (Parana)	Brazil
200.19.159.35	Fundação de Desenvolvimento da Pesquisa	Brazil
201.155.87.63	Uninet S.A. De C.V., Mexico (Distrito Federal)	Mexico
202.112.128.11	Beijing University of Aeronautics and Astronautics, Beijing (Beijing)	China
202.112.28.100	China Education and Research Network Backbone, Beijing (Beijing)	China
202.116.81.195	Zhongshan University, Guangzhou (Guangdong)	China
202.141.161.44	Scitech Group of University of Science, Hefei (Anhui)	China

202.189.126.85	The University of Hong Kong, Central District	Hong Kong
202.23.159.52	Doshisha University, Nishinotoindori (Kyoto)	Japan
202.244.160.252	Kurashiki University of Science and Arts, Kurashiki (Okayama)	Japan
202.249.37.67	High Speed Network Lab - WIDE Project, Communication Research Lab	Japan
203.178.133.11	High Speed Network Lab - WIDE Project, Communication Research Lab	Japan
203.178.133.2	High Speed Network Lab - WIDE Project, Communication Research Lab	Japan
203.30.39.242	Singapore Advanced Research and Education Network	Singapore
204.8.155.226	Boston University, Boston (MA)	USA
204.85.191.11	North Carolina Research and Education Network, Charlotte (NC)	USA
210.125.84.42	Kwangju Institute of Science and Technology, Kwangju	South Korea
211.68.70.36	Beijing Telecommunication and Research Academy, Dian (Beijing)	China
212.201.44.82	Jacobs University Bremen gGmbH, Bremen	Germany
213.131.1.101	Department of Computational Methods, Moscow (Moscow City)	Russia
216.165.109.79	New York University, New York (NY)	USA

72.36.112.74	University of Illinois, Urbana (IL)	USA
72.36.112.78	University of Illinois, Urbana (IL)	USA
82.179.176.42	Moscow Institute of Electronic Engineering, Moscow (Moscow City)	Russia
87.236.232.174	Jordanian Universities Network L.L.C., Amman	Jordan
88.255.65.220	Turk Telekom, Istanbul	Turkey

Table [II]: Ping Experiment

# **APPENDIX III**

## **IPERF-TCP EXPERIMENT**

IP Domain	Site	Country
128.10.19.52	Purdue University, West Lafayette (IN)	USA
128.10.19.53	Purdue University, West Lafayette (IN)	USA
128.138.207.54	University of Colorado, Boulder (CO)	USA
128.151.65.101	University of Rochester, Rochester (NY)	USA
128.208.3.184	University of Washington, Seattle (WA)	USA
128.208.4.199	University of Washington, Seattle (WA)	USA
128.220.251.50	Johns Hopkins University, Baltimore (MD)	USA
128.220.251.52	Johns Hopkins University, Baltimore (MD)	USA
128.223.8.113	University of Oregon, Eugene (OR)	USA
128.227.150.12	University of Florida, Gainesville (FL)	USA
128.42.142.41	Rice University, Houston (TX)	USA
128.59.20.227	Columbia University, New York (NY)	USA
128.6.192.156	Rutgers University, Hillsborough (NJ)	USA
128.8.126.111	University of Maryland, Hyattsville (MD)	USA
128.8.126.78	University of Maryland, Hyattsville (MD)	USA
128.83.122.143	University of Texas at Austin, Austin (TX)	USA
129.10.120.193	Northeastern University, Boston (MA)	USA
129.10.120.194	Northeastern University, Boston (MA)	USA

129.105.15.38	Northwestern University, Evanston (IL)	USA
129.107.35.131	University of Texas at Arlington, Arlington (TX)	USA
129.110.125.51	University of Texas at Dallas, Richardson (TX)	USA
129.130.252.141	Kansas State University, Manhattan (KS)	USA
129.22.150.78	Case Western Reserve University, Cleveland (OH)	USA
129.237.161.194	University of Kansas, Lawrence (KS)	USA
129.74.74.19	University of Notre Dame, Notre Dame (IN)	USA
129.82.12.187	Colorado State University, Fort Collins, CO	USA
129.93.229.139	University of Nebraska-Lincoln, Lincoln, NE	USA
130.127.39.153	Clemson University, Anderson (SC)	USA
130.194.252.9	Monash University, Richmond (Victoria)	Australia
130.195.4.68	Victoria University of Wellington, Wellington	New Zealand
130.216.1.23	Auckla, Auckland	New Zealand
130.253.21.123	University of Denver, Denver (CO)	USA
130.49.221.40	University of Pittsburgh, Pittsburgh (PA)	USA
131.179.150.72	University of California - Los Angeles, Los Angeles (CA)	USA
131.193.34.38	University of Illinois at Chicago, Chicago (IL)	USA
131.247.2.245	University of South Florida, Tampa (FL)	USA
131.247.2.248	University of South Florida, Tampa (FL)	USA
132.239.17.226	University of California - San Diego, La Jolla (CA)	USA
132.72.23.10	Ben Gurion University Network, Yafo (Tel Aviv)	Israel

133.68.253.243	Nagoya Institute of Technology, Nagoya (Aichi)	Japan
133.9.81.164	Waseda University	Japan
134.88.5.253	University of Massachusetts Dartmouth, North Dartmouth	USA
	(MA)	
136.145.115.194	University of Puerto Rico, San Juan	Puerto Rico
136.159.220.40	University of Calgary, Calgary (AB)	Canada
137.132.80.105	National University of Singapore, Singapore	Singapore
137.148.16.11	Cleveland State University Computer Services, Cleveland	USA
	(OH)	
137.165.1.115	Williams College Campus, Williamstown (MA)	USA
137.99.11.86	University of Connecticut, Storrs Mansfield (CT)	USA
138.15.10.55	NEC Laboratories America, Sunnyvale (CA)	USA
138.238.250.157	Howard University, Washington (DC)	USA
139.80.206.133	University of Otago, Dunedin (Otago)	New Zealand
140.109.17.181	Academia Sinica, Taipei (T'ai-pei)	Taiwan
140.114.79.231	Taiwan Academic Network, Hsinchu (T'ai-wan),	Taiwan
140.123.230.249	National Chung Cheng University, Taipei (T'ai-pei)	Taiwan
140.192.249.204	Depaul University, Chicago (IL)	USA
141.161.20.32	Georgetown University, Washington (DC)	USA
141.212.113.179	University of Michigan - College of Engineering, Ann Arbor	USA
	(MI)	

141.213.4.202	University of Michigan - College of Engineering, Ann Arbor (MI)	USA
141.219.252.133	Michigan Technological University, Houghton (MI)	USA
142.104.21.241	University of Victoria, Victoria (BC)	Canada
143.107.111.234	Universidade De Sao Paulo, São Paulo (Sao Paulo)	Brazil
143.107.111.235	Universidade De Sao Paulo, São Paulo (Sao Paulo)	Brazil
143.89.49.74	Hong Kong University of Science and Technology, Central District	Hong Kong
147.46.240.165	Seoul National University, Seoul (Seoul-T'ukpyolsi)	South Korea
149.43.80.22	Colgate University, Hamilton (NY)	USA
150.140.140.93	University of Patras Network, Patras (Akhaia)	Greece
150.65.32.66	Japan Advanced Institute of Science and Technology, Nomi (Kyoto)	Japan
152.14.93.140	North Carolina Research and Education Network, Raleigh (NC)	USA
155.225.2.72	The Citadel, North Charleston (SC)	USA
156.56.250.227	Indiana University, Bloomington (IN)	USA
157.92.44.102	Universidad Nacional De Buenos Aires, Buenos Aires (Distrito Federal)	Argentina
160.193.163.106	Osaka City University, Osaka (Gifu)	Japan
160.36.57.173	The University of Tennessee, Memphis (TN)	USA

162.105.205.21	Peking, Beijing (Beijing)	China
164.73.47.244	Servicio Central De Informatica, Montevideo (Montevideo)	Uruguay
165.230.49.119	Rutgers University, Piscataway (NJ)	USA
165.91.55.11	Texas A&M University, College Station (TX)	USA
165.91.55.9	Texas A&M University, College Station (TX)	USA
169.226.40.4	State University of New York, Albany (NY)	USA
169.229.50.18	University of California at Berkeley, Oakland (CA)	USA
169.235.24.232	University of California - Riverside, Riverside (CA)	USA
170.140.119.70	Emory University, Atlanta (GA)	USA
178.22.88.44	Limited Liability Company Data Center M	Russia
192.16.125.12	Swedish Institute of Microelectronics, Kista (Stockholms	Sweden
	Lan)	
192.33.90.68	ETHZ, Swiss Federal Institute of Technology, Zürich	Switzerland
192.42.83.253	Illinois Institute of Technology, Chicago (IL)	USA
193.63.58.70	University College of London - Computer Science	UK
	Department, London	
194.29.178.14	Warsaw University of Technology, Warsaw	Poland
195.113.161.83	CESNET, Prague (Hlavni Mesto Praha)	Czech
		Republic
195.116.53.14	Telekomunikacja Polska S.A, Polska (Kujawsko-Pomorskie)	Poland
198.128.56.12	Lawrence Berkeley National Laboratory, Berkeley (CA)	USA

198.133.224.149	Computer Sciences Department University of Wiscons, Madison (WI)	USA
198.175.112.105	Intel Corporation - Pleasant Hill, Fair Oaks (CA)	USA
198.7.242.42	Ocean State Higher Education and Administration Network,	USA
	Providence (RI)	
198.82.160.221	Virginia Polytechnic Institute and State University,	USA
	Blacksburg (VA)	
198.82.160.239	Virginia Polytechnic Institute and State University,	USA
	Blacksburg (VA)	
200.0.206.137	Cooperación Latino Americana de Redes Avanzadas,	Uruguay
	Montevideo	
200.0.206.202	Cooperación Latino Americana de Redes Avanzadas,	Uruguay
	Montevideo	
200.129.132.19	Associação Rede Nacional De Ensino E Pesquisa, Belém	Brazil
	(Para)	
200.132.1.4	Associação Rede Nacional De Ensino E Pesquisa, Porto	Brazil
	Alegre	
200.17.202.194	Universidade Federal Do Paraná, Curitiba (Parana)	Brazil
200.19.159.35	Fundação de Desenvolvimento da Pesquisa	Brazil
202.112.128.11	Beijing University of Aeronautics and Astronautics, Beijing	China
	(Beijing)	

202.112.28.100	China Education and Research Network Backbone, Beijing (Beijing)	China
202.116.81.195	Zhongshan University, Guangzhou (Guangdong)	China
202.125.215.10	The Hong Kong Polytechnic University	Hong Kong
202.141.161.44	Scitech Group of University of Science, Hefei (Anhui)	China
202.189.126.85	The University of Hong Kong, Central District	Hong Kong
202.23.159.52	Doshisha University, Nishinotoindori (Kyoto)	Japan
202.249.37.67	High Speed Network Lab - WIDE Project, Communication Research Lab	Japan
203.178.133.11	High Speed Network Lab - WIDE Project, Communication Research Lab	Japan
203.178.133.2	High Speed Network Lab - WIDE Project, Communication Research Lab	Japan
203.30.39.242	Singapore Advanced Research and Education Network	Singapore
204.8.155.226	Boston University, Boston (MA)	USA
204.85.191.11	North Carolina Research and Education Network, Charlotte (NC)	USA
210.125.84.42	Kwangju Institute of Science and Technology, Kwangju	South Korea
211.68.70.36	Beijing Telecommunication & Research Academy, Dian (Beijing)	China
213.131.1.101	Department of Computational, Moscow (Moscow City)	Russia

216.165.109.79	New York University, New York (NY)	USA
72.36.112.74	University of Illinois, Urbana (IL)	USA
72.36.112.78	University of Illinois, Urbana (IL)	USA
87.236.232.174	Jordanian Universities Network L.L.C., Amman	Jordan

Table [III]: Iperf-TCP Experiment

# **APPENDIX IV**

## **IPERF-UDP EXPERIMENT**

IP Domain	Site	Country
108.58.13.206	Optimum Online, Mineola (NY)	USA
128.10.19.53	Purdue University, West Lafayette (IN)	USA
128.143.6.130	University of Virginia, Charlottesville (VA)	USA
128.151.65.101	University of Rochester, Rochester (NY)	USA
128.187.223.212	Brigham Young University, Provo (UT)	USA
128.208.3.184	University of Washington, Seattle (WA)	USA
128.208.4.199	University of Washington, Seattle (WA)	USA
128.220.251.50	Johns Hopkins University, Baltimore (MD)	USA
128.220.251.52	Johns Hopkins University, Baltimore (MD)	USA
128.223.8.113	University of Oregon, Eugene (OR)	USA
128.227.150.12	University of Florida, Gainesville (FL)	USA
128.42.142.41	Rice University, Houston (TX)	USA
128.59.20.227	Columbia University, New York (NY)	USA
128.6.192.156	Rutgers University, Hillsborough (NJ)	USA
128.8.126.111	University of Maryland, Hyattsville (MD)	USA
128.8.126.78	University of Maryland, Hyattsville (MD)	USA
128.83.122.143	University of Texas at Austin, Austin (TX)	USA
129.10.120.193	Northeastern University, Boston (MA)	USA

129.10.120.194	Northeastern University, Boston (MA)	USA
129.105.15.38	Northwestern University, Evanston (IL)	USA
129.107.35.131	University of Texas at Arlington, Arlington (TX)	USA
129.108.202.10	University of Texas at El Paso, El Paso (TX)	USA
129.110.125.51	University of Texas at Dallas, Richardson (TX)	USA
129.130.252.141	Kansas State University, Manhattan (KS)	USA
129.22.150.78	Case Western Reserve University, Cleveland (OH)	USA
129.237.161.194	University of Kansas, Lawrence (KS)	USA
129.74.74.19	University of Notre Dame, Notre Dame (IN)	USA
129.82.12.187	Colorado State University, Fort Collins, CO	USA
129.93.229.139	University of Nebraska-Lincoln, Lincoln, NE	USA
130.127.39.153	Clemson University, Anderson (SC)	USA
130.194.252.9	Monash University, Richmond (Victoria)	Australia
130.195.4.68	Victoria University of Wellington, Wellington	New Zealand
130.216.1.23	Auckla, Auckland	New Zealand
130.49.221.40	University of Pittsburgh, Pittsburgh (PA)	USA
131.193.34.38	University of Illinois at Chicago, Chicago (IL)	USA
131.247.2.245	University of South Florida, Tampa (FL)	USA
131.247.2.248	University of South Florida, Tampa (FL)	USA
132.239.17.226	University of California - San Diego, La Jolla (CA)	USA

132.72.23.10	Ben Gurion University Network, Yafo (Tel Aviv)	Israel
133.68.253.243	Nagoya Institute of Technology, Nagoya (Aichi)	Japan
133.9.81.164	Waseda University	Japan
134.88.5.253	University of Massachusetts Dartmouth, North Dartmouth (MA)	USA
136.145.115.194	University of Puerto Rico, San Juan	Puerto Rico
136.159.220.40	University of Calgary, Calgary (AB)	Canada
137.132.80.105	National University of Singapore, Singapore	Singapore
137.148.16.11	Cleveland State University Computer Services, Cleveland (OH)	USA
137.165.1.115	Williams College Campus, Williamstown (MA)	USA
137.99.11.86	University of Connecticut, Storrs Mansfield (CT)	USA
138.15.10.55	NEC Laboratories America, Sunnyvale (CA)	USA
138.238.250.157	Howard University, Washington (DC)	USA
139.80.206.133	University of Otago, Dunedin (Otago)	New Zealand
140.109.17.181	Academia Sinica, Taipei (T'ai-pei)	Taiwan
140.112.42.158	Taiwan Academic Network, Taipei (T'ai-pei)	Taiwan
140.123.230.248	National Chung Cheng University, Taipei (T'ai-pei)	Taiwan
140.192.249.204	Depaul University, Chicago (IL)	USA
141.161.20.32	Georgetown University, Washington (DC)	USA
141.212.113.179	University of Michigan - Engineering, Ann Arbor (MI)	USA
141.213.4.202	University of Michigan - College of Engineering, Ann Arbor (MI)	USA
-----------------	--	-------------
141.219.252.133	Michigan Technological University, Houghton (MI)	USA
142.104.21.241	University of Victoria, Victoria (BC)	Canada
143.107.111.234	Universidade De Sao Paulo, São Paulo (Sao Paulo)	Brazil
143.107.111.235	Universidade De Sao Paulo, São Paulo (Sao Paulo)	Brazil
147.46.240.165	Seoul National University, Seoul (Seoul-T'ukpyolsi)	South Korea
149.43.80.22	Colgate University, Hamilton (NY)	USA
150.140.140.93	University of Patras Network, Patras (Akhaia)	Greece
150.65.32.66	Japan Advanced Institute of Science and Technology, Nomi (Kyoto)	Japan
152.14.93.140	North Carolina Research and Education Network, Raleigh (NC)	USA
155.223.52.3	The Citadel, North Charleston (SC)	USA
155.246.12.163	Stevens Institute of Technology, Hoboken (NJ)	
156.56.250.227	Indiana University, Bloomington (IN)	USA
157.92.44.102	Universidad Nacional De Buenos Aires, Buenos Aires (Distrito Federal)	Argentina
158.130.6.253	University of Pennsylvania, Philadelphia (PA)	
160.193.163.106	Osaka City University, Osaka (Gifu)	Japan
162.105.205.21	Peking, Beijing (Beijing)	China

165.230.49.119	Rutgers University, Piscataway (NJ)	USA
165.91.55.11	Texas A&M University, College Station (TX)	USA
165.91.55.9	Texas A&M University, College Station (TX)	USA
169.226.40.4	State University of New York, Albany (NY)	USA
169.229.50.18	University of California at Berkeley, Oakland (CA)	USA
169.235.24.232	University of California - Riverside, Riverside (CA)	USA
170.140.119.69	Emory University, Atlanta (GA)	USA
170.140.119.70	Emory University, Atlanta (GA)	USA
192.16.125.12	Swedish Institute of Microelectronics, Kista (Stockholms	Sweden
	Lan)	
192.33.90.68	ETHZ, Swiss Federal Institute of Technology, Zürich	Switzerland
192.42.83.253	Illinois Institute of Technology, Chicago (IL)	USA
193.63.58.70	University College of London - Computer Science	UK
	Department, London	
194.29.178.14	Warsaw University of Technology, Warsaw	Poland
195.113.161.83	CESNET, Prague (Hlavni Mesto Praha)	Czech
		Republic
195.116.53.14	Telekomunikacja Polska S.A, Polska (Kujawsko-Pomorskie)	Poland
198.133.224.149	Computer Sciences Department University of Wiscons,	USA
	Madison (WI)	
198.175.112.105	Intel Corporation - Pleasant Hill, Fair Oaks (CA)	USA

198.7.242.42	Ocean State Higher Education and Administration Network, Providence (RI)	USA
198.82.160.221	Virginia Polytechnic Institute and State University, Blacksburg (VA)	USA
198.82.160.239	Virginia Polytechnic Institute and State University, Blacksburg (VA)	USA
200.0.206.137	Cooperación Latino Americana de Redes Avanzadas, Montevideo	Uruguay
200.0.206.202	Cooperación Latino Americana de Redes Avanzadas, Montevideo	Uruguay
200.129.132.19	Associação Rede Nacional De Ensino E Pesquisa, Belém (Para)	Brazil
200.132.1.4	Associação Rede Nacional De Ensino E Pesquisa, Porto Alegre	Brazil
200.17.202.194	Universidade Federal Do Paraná, Curitiba (Parana)	Brazil
200.19.159.35	Fundação de Desenvolvimento da Pesquisa	Brazil
202.112.128.11	Beijing University of Aeronautics and Astronautics, Beijing (Beijing)	China
202.112.28.100	China Education and Research Network Backbone, Beijing (Beijing)	China
202.116.81.195	Zhongshan University, Guangzhou (Guangdong)	China

202.125.215.10	The Hong Kong Polytechnic University	Hong Kong
202.189.126.85	The University of Hong Kong, Central District	Hong Kong
202.23.159.52	Doshisha University, Nishinotoindori (Kyoto)	Japan
202.249.37.67	High Speed Network Lab - WIDE Project, Communication Research Lab	Japan
203.178.133.11	High Speed Network Lab - WIDE Project, Communication Research Lab	Japan
203.178.133.2	High Speed Network Lab - WIDE Project, Communication Research Lab	Japan
203.30.39.242	Singapore Advanced Research and Education Network	Singapore
204.8.155.226	Boston University, Boston (MA)	USA
204.85.191.11	North Carolina Research and Education Network, Charlotte (NC)	USA
210.125.84.42	Kwangju Institute of Science and Technology, Kwangju	South Korea
211.68.70.36	Beijing Telecommunication & Research Academy, Dian (Beijing)	China
212.201.44.82	Jacobs University Bremen gGmbH, Bremen	Germany
213.131.1.101	Department of Computational Methods, Moscow (Moscow City)	Russia
216.165.109.79	New York University, New York (NY)	USA
72.36.112.74	University of Illinois, Urbana (IL)	USA

72.36.112.78	University of Illinois, Urbana (IL)	USA
87.236.232.174	Jordanian Universities Network L.L.C., Amman	Jordan

Table [IV]: Iperf-UDP Experiment

## **BIBLIOGRAPHY**

## **BIBLIOGRAPHY**

- [1] D. Andersen, PhD Thesis: Improving End-to-End Availability Using Overlay Networks, MIT, 2004.
- [2] V. Paxson, PhD Thesis: Measurements and Analysis of End-to-End Internet Dynamics, University of California at Berkeley, 1997.
- [3] D. Yiltas and H. Perros, QoS Based Multi-Domain Routing under Multiple QoS Metrics, NC State University.
- [4] K. Lai and M. Baker, Measuring Link Bandwidths Using a Deterministic Model of Packet Delay, Stanford University.
- [5] J. Navratil and R. Cottrell, ABwE: A Practical Approach to Available Bandwidth Estimation, Stanford Linear Accelerator Center (SLAC).
- [6] A. Downey, Using Pathchar to Determine Internet Link Characteristics, Colby College, ACM SIGCOMM, 1999.
- [7] G. Jin and B. Tierney, Netest: A Tool to Measure the Maximum Burst Size, Available Bandwidth and Achievable Throughput, Lawrence Berkeley National Laboratory.
- [8] N. Hu, E. Li, Z. Mao, P. Steenkiste and J. Wang, Locating Internet Bottlenecks: Algorithms, Measurements, and Implications, Carnegie Mellon University, Bell Laboratories, University of Michigan, AT&T Labs Research, SIGCOMM 2004.
- [9] V. Mohan, Y. Reddy and K. Kalpana, Survey: Active and Passive Network Measurements, G. Pulla Reddy Engineering College, International Journal of Computer Science and Information Technologies (IJCSIT) 2011.
- [10] A. Johnsson, B. Melander and M. Bjorkman, DietTopp: A First Implementation and Evaluation of a Simplified Bandwidth Measurement Method, Malardalen University.
- [11] R. Kapoor, L. Chen, L. Lao, M. Gerla and M. Sanadidi, CapProbe: A Simple and Accurate Capacity Estimation Technique, Qualcomm and UCLA, SIGCOMM 2004.
- [12] V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil and L. Cottrell, PathChirp: Efficient Availble Bandwidth Estimation for Network Paths, Rice University and Stanford Linear Accelerator Center (SLAC), 2003.
- [13] N. Hu and P. Steenkiste, RPT: A Low Overhead Single-End Probing Toolfor Detecting Network Congestion Positions, Carnegie Mellon University, 2003.
- [14] N. Spring, D. Wetherall and T. Anderson, Scriptroute: A Public Internet Measurement Facility, University of Washington, USENIX 2003.

- [15] J. Strauss, D. Katabi and F. Kaashoek, A Measurement Study of Available Bandwidth Estimation Tools, MIT, IMC 2003.
- [16] V. Ribeiro, R. Riedi and R. Baraniuk, Spatio-Temporal Available Bandwidth Estimation with STAB, Rice University, SIGMETRICS-Performance 2004.
- [17] M. Kazantzidis, Technical Report: How to Measure Available Bandwidth on the Internet, UCLA.
- [18] M. Jain and C. Dovrolis, Pathload: A Measurement Tool for End-to-End Available Bandwidth, University of Delaware, In Proceedings of PAM Workshop, March 2002.
- [19] B. Mah, Estimating Bandwidth and Other Network Properties, Cisco Systems, 1999.
- [20] R. Prasad, M. Murray, C. Dovrolis and K. Claffy, Bandwidth Estimation: Metrics Measurement Techniques and Tools, Georgia Institute of Technology and Cooperative Association for Internet Data Analysis.
- [21] M. Jain and C. Dovrolis, Path Selection Using Available Bandwidth Estimation in Overlay-Based Video Streaming, Georgia Institute of Technology.
- [22] S. Sariou, P. Gummadi and S. Gribble, SProbe: A Fast Technique for Measuring Bottleneck Bandwidth in Uncooperative Environments, In Proceedings of IEEE INFOCOM, 2002.
- [23] K. Sklower and A. Joseph, Very Large Scale Cooperative Experiments in Emulab-Derived Systems, University of California at Berkeley, 2007.
- [24] T. Friedman, PlanetLab Europe (PLE), PlanetLab Europe Consortium. http://www.ictfire.eu/fileadmin/events/2011-09-2OpenCall/OpenLab/01-Friedman-PLE\_presentation.pdf
- [25] E. Jaffe, D. Bickson, and S. Kirkpatrick, Everlab A Production Platform for Research in Network Experimentation and Computation, Hebrew University of Jerusalem, TheIn Proceedings of the 21<sup>st</sup> Large Installation System Administration Conference, 2007.
- [26] Experts in VoIP Service Provider solutions, VoIP Speed Test Broadband Bandwidth Test - Broadband Jitter Test. http://www.whichvoip.com/voip/speed\_test/ppspeed.html
- [27] N. Feamster, Lecture: Overlay Routing in the Internet, MIT, 2002.
- [28] H. Balakrishnan, Lecture: End-to-End Congestion Control, MIT, 2002.
- [29] I. Bilinskis and A. Mikelsons, Randomized Signal Processing, Prentice Hall International, 1992.
- [30] R. Wolff, Poisson Arrivals See Time Averages, Operations Research, 1982.
- [31] E. Weisstein, Birthday Problem, MathWorld Wolfram Web Resource. http://mathworld.wolfram.com/BirthdayProblem.html

- [32] N. Hu and P. Steenkiste, Evaluation and Characterization of Available Bandwidth Probing Techniques, In the IEEE JSAC, 2003.
- [33] J. Bolot and, Characterizing End-to-End Packet delay and Loss in the Internet, In Journal of High-Speed Networks, ACM Sigcomm, 1993.
- [34] M. Mathis and M. Allman, A Framework for Defining Empirical Bulk Transfer Capacity Metrics, Network Working Group, Pittsburgh Supercomputing Center and BBN/NASA Glenn, In Request for Comments: 3148, 2001.
- [35] T. Ndousse, Network Measurement and Analysis for E2E Performance, Department of Energy at Advanced Scientific Computing Research Office, 2002.
- [36] Using Iperf to Measure Available Network Bandwidth When Troubleshooting Veritas Volume Replicator (VVR) performance issues, 2010. http://www.symantec.com/docs/TECH76447
- [37] M. Kozelj, Iperf Users How iperf works? 2009. http://www.mail-archive.com/iperfusers@lists.sourceforge.net/msg00147.html
- [38] What is Jitter? http://www.nessoft.com/kb/article/what-is-jitter-57.html
- [39] X. Lu, "Bandwidth Metrics and Measurement Tools, High-Performance Computing Group - Computer Science University of Windsor.
- [40] S. Savage, Sting: A TCP-based Network Measurement Tool, University of Washington, USENIX Symposium on Internet Technologies and Systems, 1999.
- [41] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson, Tulip: A Tool for Performance Problems along Internet Paths, University of Washington. http://www.icir.org/models/tools.html
- [42] J. Sommers, P. Barford, N. Duffield, and A. Ron, Improving Accuracy in End-to-end Packet Loss Measurement, SIGCOMM, 2005.
- [43] Matti, U. Guillaume and B. Ernst, Research Report: Towards a Quantitative T-RAT, Institut Eurecom - Department of Corporate Communications, 2004.
- [44] Bing Manual, 1995. http://fgouget.free.fr/bing/bing\_src-readme.shtml
- [45] A. Downey, Clink: a Tool for Estimating Internet Link Characteristics, Wellesley College. http://rocky.wellesley.edu/downey
- [46] http://nuttcp.org/nuttcp/Welcome%20Page.html
- [47] D. Moskaluk, VoIP using Wireless Mesh Infrastructure, 2007.

http://www.moskaluk.com/voip\_using\_wireless\_mesh\_infrast.htm

- [48] Iperf Usage Examples. http://kb.pert.geant.net/PERTKB/IperfTool
- [49] Border Gateway Protocol. http://en.wikipedia.org/wiki/Border\_Gateway\_Protocol
- [50] Calculate Distance between Latitude-Longitude Points. http://www.movabletype.co.uk/scripts/latlong.html
- [51] H. Pucha and Y. Hu, Overlay TCP: Ending End-to-End Transport for Higher Throughput, Purdue University, ACM SIGCOMM, 2005.
- [52] Application Note: Latency on a Switched Ethernet Network, RuggedCom Inc, 2008.
- [53] H. Balakrishnan, Lecture: Traffic Self-Similarity, MIT, 2002.