LBNL-40319
UCB//CSD-97-945

Measurements and Analysis of End-to-End Internet Dynamics

Vern Paxson
Ph.D. Thesis

Computer Science Division
University of California, Berkeley

and

Information and Computing Sciences Division
Lawrence Berkeley National Laboratory
University of California
Berkeley, CA 94720

April, 1997

This work was supported by the Director, Office of Energy Research, Scientific Computing Staff, of the United States Department of Energy under Contract No. DE-AC03-76SF00098.

Measurements and Analysis of End-to-End Internet Dynamics

by

Vern Edward Paxson

B.S. (Stanford University) 1985
M.S. (University of California, Berkeley) 1991

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Computer Science

in the

GRADUATE DIVISION
of the
UNIVERSITY of CALIFORNIA at BERKELEY

Committee in charge:

Prof. Domenico Ferrari, Chair
Prof. Michael Luby
Prof. John Rice

1997

Measurements and Analysis of End-to-End Internet Dynamics

Copyright 1997
by
Vern Edward Paxson

The U.S. Department of Energy has the right to use this document
for any purpose whatsoever including the right to reproduce
all or any part thereof

Abstract

Measurements and Analysis of End-to-End Internet Dynamics

by

Vern Edward Paxson
Doctor of Philosophy in Computer Science

University of California at Berkeley

Prof. Domenico Ferrari, Chair

Accurately characterizing end-to-end Internet dynamics—the performance that a user actually obtains from the lengthy series of network links that comprise a path through the Internet—is exceptionally difficult, due to the network's immense heterogeneity. It can be impossible to gauge the generality of findings based on measurements of a handful of paths, yet logistically it has proven very difficult to obtain end-to-end measurements on larger scales.

At the heart of our work is a "measurement framework" we devised in which a number of sites around the Internet host a specialized measurement service. By coordinating "probes" between pairs of these sites we can measure end-to-end behavior along $O(N^2)$ paths for a framework consisting of N sites. Consequently, we obtain a superlinear scaling that allows us to measure a rich cross-section of Internet behavior without requiring huge numbers of observation points. 37 sites participated in our study, allowing us to measure more than 1,000 distinct Internet paths.

The first part of our work looks at the behavior of end-to-end routing: the series of routers over which a connection's packets travel. Based on 40,000 measurements made using our framework, we analyze: routing "pathologies" such as loops, outages, and flutter; the stability of routes over time; and the symmetry of routing along the two directions of an end-to-end path. We find that pathologies increased significantly over the course of 1995, indicating that, by one metric, routing *degraded* over the year; that Internet paths are heavily dominated by a single route, but that routing lifetimes range from seconds to many days, with most lasting for days; and that, at the end of 1995, about half of all Internet paths included a major routing asymmetry.

The second part of our work studies end-to-end Internet packet dynamics. We analyze 20,000 TCP transfers of 100 Kbyte each to investigate the performance of both the TCP endpoints and the Internet paths. The measurements used for this part of our study are much richer than those for the first part, but require a great degree of attention to issues of *calibration*, which we address by applying *self-consistency checks* to the measurements whenever possible. We find that packet filters are capable of a wide range of measurement errors, some of which, if undetected, can significantly taint subsequent analysis. We further find that network clocks exhibit adjustments and skews relative to other clocks frequently enough that a failure to detect and remove these effects will likewise pollute subsequent packet timing analysis.

Using TCP transfers for our network path "measurement probes" gains a number of advantages, the chief of which is the ability to probe fine time scales without unduly loading the network. However, using TCP also requires us to accurately distinguish between connection dy-

namics due to the behavior of the TCP endpoints, and dynamics due to the behavior of the network path between them. To address this problem, we develop an analysis program, `tcpanaly`, that has coded into it knowledge of how the different TCP implementations in our study function. In the process of developing `tcpanaly`, we thus in tandem develop detailed descriptions of the performance and congestion-avoidance behavior of the different implementations. We find that some of the implementations suffer from gross problems, the most serious of which would devastate overall Internet performance, were the implementations ubiquitously deployed.

With the measurements calibrated and the TCP behavior understood, we then can turn to analyzing the dynamics of Internet paths. We first need to determine a path's *bottleneck bandwidth*, meaning the fastest transfer rate the path can sustain. Knowing the bottleneck bandwidth then lets us determine which packets a sender transmits must necessarily *queue* behind their predecessors, due to the load the sender imposes on the path. This in turn allows us to determine which of our probes are perforce *correlated*. We identify several problems with the existing bottleneck estimation technique, "packet pair," and devise a robust estimation algorithm, PBM ("packet bunch modes"), that addresses these difficulties. We calibrate PBM by gauging the degree to which the bottleneck rates it estimates accord with known link speeds, and find good agreement. We then characterize the scope of Internet path bottleneck rates, finding wide variation, not infrequent asymmetries, but considerable stability over time.

We next turn to an analysis of packet loss along Internet paths. To do so, we distinguish between losses of "loaded" data packets, meaning those which necessarily queued behind a predecessor at the bottleneck; "unloaded" data packets, which did not do so; and the small "acknowledgement" packets returned to a TCP sender by the TCP receiver. We find that network paths are well characterized by two general states, "quiescent," in which no loss occurs, and "busy," in which one or more packets of a connection are lost. The prevalence of quiescent connections remained about 50% in both our datasets, but for busy connections, packet loss rates increased significantly over the course of 1995. We further find that loss rates vary dramatically between different regions of the network, with European and especially trans-Atlantic connections faring much worse than those confined to the United States.

We also characterize: loss symmetry, finding that loss rates along the two directions of an Internet path are nearly uncorrelated; loss "outages," finding that outage durations exhibit clear Pareto distributions, indicating they span a large range of time scales; the degree to which a connection's loss patterns predict those of future connections, finding that observing quiescence is a good predictor of observing quiescence in the future, and likewise for observing a busy network path, but that the proportion of lost packets does not well predict the future proportion; and the efficacy of TCP implementations in dealing with loss efficiently, by retransmitting only when necessary. We find that most TCPs retransmit fairly efficiently, and that deploying the proposed "selective acknowledgement" option would eliminate almost all of their remaining unnecessary retransmissions. However, some TCPs incorrectly determine how long to wait before retransmitting, and these can suffer large numbers of unnecessary retransmissions.

We finish our study with a look at variations in packet transit delays. We find great "peak-to-peak" variation, meaning that maximum delays far exceed minimum delays. Delay variations along the two directions of an Internet path are only lightly correlated, but correlate well with loss rates observed in the same direction along the path. We identify three types of "timing compression," in which packets arrive at their receiver spaced more closely together than when originally

transmitted. The prevalence of none of the three is such as to significantly perturb network performance, but all three occur frequently enough to require judicious filtering by network measurement procedures to avoid deriving false timing conclusions.

We then look at the question of the time scales on which most of a path's queueing variations occur. We find that, overall, most variation occurs on time scales of 100–1000 msec, which means that transport connections might effectively adapt their transmission to the variations, but only if they act quickly. However, as with many Internet path properties, we find wide ranges of behavior, with not insignificant queueing variations occurring on time scales as small as 10 msec and as large as one minute.

The last aspect of packet delay variations we investigate is the degree to which it reflects an Internet path's *available bandwidth*. We show that the ratio between the delay variations packets incur due to their connection's own loading of the network, versus the total delay variations incurred, correlates well with the connection's overall throughput. We further find that Internet paths exhibit wide variation in available bandwidth, ranging from very little available to virtually all. The degree of available bandwidth diminished markedly over the course of 1995, but, as for packet loss rates, we also find sharp geographic differences, so the overall trend cannot be summarized in completely simple terms. Finally, we investigate the degree to which the available bandwidth observed by a connection accurately predicts that observed by future connections, finding that the predictive power is fairly good for time scales of minutes to hours, but diminishes significantly for longer time periods.

We argue that our work supports several general themes:

- The N^2 scaling property of our measurement framework serves to measure a sufficiently diverse set of Internet paths that we might plausibly interpret the resulting analysis as accurately reflecting general Internet behavior.
- To cope with such large-scaled measurements requires attention to calibration using self-consistency checks; robust statistics to avoid skewing by outliers; and automated "micro-analysis," such as that performed by `tcpanaly`, that we might see the forest as well as the trees.
- With due diligence to remove packet filter errors and TCP effects, TCP-based measurement provides a viable means for assessing end-to-end packet dynamics.
- We find wide ranges of behavior, so we must exercise great caution in regarding any aspect of packet dynamics as "typical."
- Some common assumptions such as in-order packet delivery, FIFO bottleneck queueing, independent loss events, single congestion time scales, and path symmetries are all sometimes violated.
- The combination of path asymmetries and reverse-path noise render sender-only measurement techniques markedly inferior to those that include receiver-cooperation.

Finally, we believe an important aspect of this work is how it might contribute towards developing a "measurement infrastructure" for the Internet: one that proves ubiquitous, informative, and sound.

To Lindsay —

For making it both possible
and worthwhile

— with all my love

Contents

List of Figures **xi**

List of Tables **xvi**

1 Introduction **1**

I End-to-End Routing Behavior in the Internet **4**

2 Overview of the Routing Study **5**

3 Related Research **8**
3.1 Studies of routing protocols 8
3.2 Studies of routing behavior 8
3.3 End-to-end routing dynamics 9
3.4 Routing in the Internet 10

4 Methodology **12**
4.1 Experimental apparatus 12
4.2 The `traceroute` Utility 13
4.2.1 The Time To Live field 13
4.2.2 How `traceroute` works 14
4.2.3 `Traceroute` limitations 15
4.3 Exponential sampling 18
4.4 Which observations are representative? 19
4.5 Testing for significant differences 20
4.6 A note on independence 22

5 The Raw Routing Data **23**
5.1 Participating sites 23
5.2 Measurement breakdown 27
5.3 Geography 30

6 Routing Pathologies **34**
6.1 Unresponsive routers . . . 34
6.2 Rate-limiting routers . . . 35
6.3 Routing loops . . . 35
6.3.1 Persistent routing loops . . . 36
6.3.2 Temporary routing loops . . . 41
6.3.3 Location of routing loops . . . 44
6.4 Erroneous routing . . . 44
6.5 Connectivity altered mid-stream . . . 45
6.6 Fluttering . . . 49
6.6.1 A simple example . . . 49
6.6.2 A more dramatic example . . . 50
6.6.3 Fluttering at another site . . . 55
6.6.4 Skipping . . . 56
6.6.5 Significance of fluttering . . . 57
6.7 Unreachability . . . 58
6.7.1 Host down . . . 58
6.7.2 Stub network outage . . . 58
6.7.3 Infrastructure failure . . . 60
6.7.4 Consistently unreachable hosts . . . 61
6.7.5 Unreachable due to too many hops . . . 61
6.8 Temporary outages . . . 62
6.9 Circuitous routing . . . 64
6.10 Summary . . . 69

7 End-to-End Routing Stability **71**
7.1 Importance of routing stability . . . 71
7.2 Why routes change . . . 73
7.3 Two definitions of stability . . . 74
7.4 Reducing the data . . . 75
7.5 Routing Prevalence . . . 77
7.6 Routing Persistence . . . 82
7.6.1 Rapid route alternation . . . 82
7.6.2 Medium-scale route alternation . . . 86
7.6.3 Large-scale route alternation . . . 86
7.6.4 Duration of long-lived routes . . . 87
7.6.5 Summary of routing persistence . . . 88
7.7 Detecting route changes . . . 89

8 Routing Symmetry **92**
8.1 Importance of routing symmetry . . . 92
8.2 Sources of routing asymmetries . . . 93
8.3 Definition of routing symmetry . . . 95
8.4 Analysis of routing symmetry . . . 97
8.5 Increasing prevalence of asymmetry . . . 98

8.6 Size of asymmetries . . . 98

II End-to-End Internet Packet Dynamics 101

9 Overview of the Packet Dynamics Study 102
9.1 Methodology . . . 103
9.1.1 Measurement considerations . . . 103
9.1.2 Using TCP . . . 104
9.1.3 Tracing at both sender and receiver . . . 106
9.1.4 Analysis strategies . . . 107
9.2 An overview of TCP . . . 109
9.2.1 Data delivery goals . . . 109
9.2.2 Achieving high performance . . . 110
9.2.3 Congestion control . . . 112
9.2.4 Slow start . . . 113
9.2.5 Self-clocking . . . 114
9.2.6 Responding to congestion . . . 117
9.2.7 Fast retransmit and recovery . . . 119
9.3 The Raw Measurements . . . 122

10 Calibrating Packet Filters 125
10.1 The notion of "wire time" . . . 125
10.2 How packet filters work . . . 126
10.3 Packet filter errors . . . 127
10.3.1 Drops . . . 128
10.3.2 Packet drop reports . . . 128
10.3.3 Inferring filter drops . . . 129
10.3.4 Trace truncation . . . 131
10.3.5 Additions . . . 131
10.3.6 Resequencing . . . 133
10.3.7 Timing . . . 135
10.3.8 Misfiltering . . . 137
10.4 Packet filter "vantage point" . . . 138
10.5 Pairing packet departures and arrivals . . . 139

11 Analyzing TCP Behavior 142
11.1 Analysis strategy . . . 142
11.2 Checking packet and measurement integrity . . . 145
11.3 Sender analysis . . . 146
11.3.1 Data liberations . . . 147
11.3.2 Inferring sender windows . . . 149
11.3.3 Inferring source quenches . . . 149
11.3.4 Inferring initial *ssthresh* . . . 151
11.4 Receiver analysis . . . 151

11.4.1 Ack obligations . . . 151
11.4.2 Inferring checksum errors . . . 153
11.5 Sender behavior of different TCP implementations . . . 155
11.5.1 Previous studies of TCP implementations . . . 156
11.5.2 Generic Tahoe behavior . . . 158
11.5.3 Generic Reno behavior . . . 158
11.5.4 BSDI TCP . . . 159
11.5.5 Digital OSF/1 TCP . . . 161
11.5.6 HP/UX TCP . . . 161
11.5.7 IRIX TCP . . . 162
11.5.8 Linux TCP . . . 162
11.5.9 NetBSD TCP . . . 164
11.5.10 Solaris TCP . . . 165
11.5.11 SunOS TCP . . . 168
11.5.12 VJ TCP . . . 168
11.6 Receiver behavior of different TCP implementations . . . 169
11.6.1 Acking in-sequence data . . . 169
11.6.2 Acking out-of-sequence data . . . 175
11.6.3 Gratuitous acks . . . 176
11.6.4 Response delays . . . 177
11.7 Behavior of additional TCP implementations . . . 179
11.7.1 Windows NT TCP . . . 180
11.7.2 Windows 95 TCP . . . 180
11.7.3 Trumpet/Winsock TCP . . . 181

12 Calibrating Pairs of Clocks **185**
12.1 Basic clock terminology . . . 185
12.1.1 Resolution . . . 186
12.1.2 Offset . . . 186
12.1.3 Accuracy . . . 186
12.1.4 Skew and drift . . . 186
12.2 Lack of synchronized clocks . . . 187
12.3 Terminology for comparing clocks . . . 187
12.4 Assessing clock resolution . . . 189
12.4.1 Method for assessing resolution . . . 189
12.4.2 Results of assessing resolution . . . 190
12.5 Assessing relative clock offset . . . 191
12.5.1 Method for assessing relative offset . . . 191
12.5.2 Relative offset for full-sized sender packets . . . 193
12.5.3 Results of assessing relative offset . . . 193
12.6 Detecting clock adjustments . . . 201
12.6.1 A graphical technique for detecting adjustments . . . 201
12.6.2 Removing noise from OTT measurements . . . 203
12.6.3 An algorithm for detecting adjustments . . . 204
12.6.4 Results of checking for adjustments . . . 207

12.6.5 Problems with detection method . . . 207
12.6.6 Detecting adjustments via correlation . . . 212
12.7 Assessing relative clock skew . . . 213
12.7.1 Defining canonical sender/receiver skew . . . 215
12.7.2 Difficulties with noise . . . 216
12.7.3 Failure of line-fitting approaches . . . 218
12.7.4 A test based on cumulative minima . . . 218
12.7.5 Applying the test to a positive trend . . . 220
12.7.6 Identifying skew trends . . . 220
12.7.7 Results of checking for skew . . . 222
12.7.8 `oce`'s puzzling dynamics . . . 224
12.7.9 Removing relative skew . . . 227
12.8 Additional clock consistency checks . . . 228
12.8.1 Non-positive min-RTT$_{sr}$ . . . 228
12.8.2 Gap analysis . . . 229
12.9 Clock synchronization vs. stability . . . 230

13 Network Pathologies **232**
13.1 Out-of-order delivery . . . 232
13.1.1 Detecting out-of-order delivery . . . 233
13.1.2 Results of out-of-order analysis . . . 233
13.1.3 Impact of reordering . . . 237
13.2 Packet replication . . . 245
13.3 Packet corruption . . . 248

14 Bottleneck Bandwidth **252**
14.1 Bottleneck bandwidth as a fundamental quantity . . . 252
14.2 Packet pair . . . 254
14.3 Receiver-based packet pair . . . 256
14.4 Difficulties with packet pair . . . 257
14.4.1 Out-of-order delivery . . . 257
14.4.2 Limitations due to clock resolution . . . 258
14.4.3 Changes in bottleneck bandwidth . . . 260
14.4.4 Multi-channel bottleneck links . . . 261
14.5 Peak rate estimation . . . 263
14.6 Robust bottleneck estimation . . . 266
14.6.1 Forming estimates for each “extent” . . . 267
14.6.2 Searching for bottleneck bandwidth modes . . . 269
14.7 Analysis of bottleneck bandwidths in the Internet . . . 274
14.7.1 Single bottlenecks . . . 275
14.7.2 Bottleneck changes . . . 282
14.7.3 Multi-channel bottlenecks . . . 284
14.7.4 Estimation errors due to TCP behavior . . . 286
14.8 Efficacy of other estimation techniques . . . 287
14.8.1 Efficacy of PR . . . 287

14.8.2 Efficacy of RBPP . . . 288
14.8.3 Efficacy of SBPP . . . 288
14.8.4 Summary of different bottleneck estimators . . . 290

15 Packet Loss **291**
15.1 Loss rates . . . 291
15.2 Data packet loss vs. ack loss . . . 299
15.3 Loss bursts . . . 305
15.4 Loss location . . . 310
15.5 Evolution of packet loss rate . . . 313
15.6 Efficacy of TCP retransmission . . . 316

16 Packet Delay **323**
16.1 RTT variation . . . 324
16.1.1 The role of RTTs . . . 324
16.1.2 RTT measurement considerations . . . 324
16.1.3 RTT extremes . . . 325
16.1.4 RTT variation during a connection . . . 327
16.2 OTT variation . . . 332
16.2.1 Why we do not analyze OTT extremes . . . 332
16.2.2 Range of OTT variation . . . 332
16.2.3 Path symmetry of OTT variation . . . 333
16.2.4 Relationship between loss rate and OTT variation . . . 335
16.2.5 Evolution of OTT variation . . . 335
16.2.6 Removing load from OTTs . . . 338
16.2.7 Periodicity in OTTs . . . 342
16.3 Timing compression . . . 343
16.3.1 Ack compression . . . 344
16.3.2 Data packet timing compression . . . 345
16.3.3 Receiver compression . . . 348
16.4 Queueing analysis . . . 349
16.5 Available bandwidth . . . 354

17 Summary **365**
17.1 The routing study . . . 365
17.2 The packet dynamics study . . . 366
17.2.1 Measurement calibration and TCP behavior . . . 366
17.2.2 Timing calibration . . . 367
17.2.3 Network pathologies . . . 367
17.2.4 Estimating bottleneck bandwidth . . . 367
17.2.5 Packet loss . . . 368
17.2.6 Packet delay . . . 369
17.3 Future research . . . 370
17.4 Themes of the work . . . 371

Bibliography **372**

A The Network Probe Daemon **383**
A.1 Daemon operation 383
A.2 Security issues 385
A.2.1 Using `rtcpdump` instead of `tcpdump` 386
A.2.2 NPD authentication 386

List of Figures

5.1 Sites participating in routing study, North America and Asia 25
5.2 Sites participating in routing study, Europe 26
5.3 Number of measurements made for each Internet path, $\mathcal{R}_1$ dataset 28
5.4 Number of measurements made for each Internet path, $\mathcal{R}_2$ dataset 29
5.5 Links traversed during $\mathcal{R}_1$ and $\mathcal{R}_2$, North American perspective 33
5.6 Links traversed during $\mathcal{R}_1$ and $\mathcal{R}_2$, European perspective 33

6.1 Routes taken by alternating packets, `wustl` to `umann` 52
6.2 Distribution of long $\mathcal{R}_1$ outages 63
6.3 Distribution of long $\mathcal{R}_2$ outages 63
6.4 Circuitous route from `bsdi` to `usc` 64
6.5 Circuitous route from `lbli` to `ucol` 65
6.6 Circuitous route from `nrao` to `wustl` 65
6.7 Circuitous route from `lbl` to `wustl` 66
6.8 Individual routers comprising circuitous path from `lbl` to `wustl` 67
6.9 Circuitous route from `ncar` to `xor` 68
6.10 Circuitous route from `inria` to `oce` 68

7.1 Prevalence of the dominant route 78
7.2 Prevalence of the dominant route, for different source sites 80
7.3 Prevalence of the dominant route, for different destination sites 81
7.4 Site-to-site variation in $P^{10}_{\text{dst s}}$ 84
7.5 Estimated distribution of long-lived route durations 88

8.1 Route observed from `ucol` to `ucl` 96
8.2 Route observed from `ucl` to `ucol` 96
8.3 Second route observed from `ucl` to `ucol` 97
8.4 Distribution of asymmetry sizes 100

9.1 Sequence plot of a TCP connection during its "slow start" phase 113
9.2 Sequence plot of a "window-limited" TCP connection 115
9.3 TCP "self-clocking" 116
9.4 Sequence plot showing a TCP timeout retransmission 119
9.5 Sequence plot showing a TCP "fast retransmission" 120
9.6 Sequence plot showing TCP "fast recovery" 122

10.1 Packet filter replication . . . 132
10.2 Packet filter resequencing . . . 133
10.3 Enlargement of resequencing event in previous figure . . . 134
10.4 Example of "time travel" . . . 136
10.5 Same plot, with lines showing the ordering of the packets in the trace file . . . 136
10.6 Receiver sequence plot showing a forward clock adjustment, undetectable to the eye . . . 137
10.7 Example of an ambiguity caused by the packet filter's vantage point . . . 138

11.1 Sequence plot showing effects of unobserved source quench . . . 150
11.2 Receiver sequence plot showing two data checksum errors . . . 154
11.3 Sequence plot showing a burst of checksum errors . . . 154
11.4 Sequence plot showing the Net/3 uninitialized-*cwnd* bug . . . 160
11.5 Sequence plot showing the HP/UX congestion window advance with duplicate acks . . . 161
11.6 Sequence plot showing broken Linux 1.0 retransmission behavior . . . 163
11.7 Enlargement of righthand side of previous figure . . . 163
11.8 Sequence plot showing broken Solaris 2.3/2.4 retransmissions, RTT = 680 msec 165
11.9 Sequence plot showing broken Solaris 2.3/2.4 retransmissions, RTT = 2.6 sec . . 166
11.10 Solaris 2.4 retransmitting without cutting *cwnd* . . . 167
11.11 Sequence plot showing Solaris 2.4 acknowledgments during initial slow-start . . 171
11.12 Corresponding burstiness at sender . . . 172
11.13 Sequence plot showing retransmission timeout due to loss of single Solaris 2.4 ack 173
11.14 Receiver sequence plot showing lulls due to Solaris 2.3 acking policy . . . 174
11.15 Sequence plot showing more frequent acking leading to "filling the pipe" . . . 175
11.16 Sequence plot showing gratuitous acknowledgement . . . 177
11.17 Sequence plot showing false gratuitous acknowledgement . . . 178
11.18 Sequence plot showing Windows 95 TCP transmit problem . . . 180
11.19 Sequence plot showing Trumpet/Winsock TCP skipping initial slow start . . . 181
11.20 Sequence plot showing Trumpet/Winsock TCP skipping slow start after timeout 182
11.21 Sequence plot showing Trumpet/Winsock timer-driven acking . . . 183
11.22 Sequence plot showing Trumpet/Winsock failure to retain above-sequence data . 183

12.1 Median magnitude of clock offset, $\mathcal{N}_1$ tracing hosts . . . 194
12.2 Median magnitude of clock offset, $\mathcal{N}_2$ tracing hosts . . . 194
12.3 Evolution of `austr`'s relative clock offset over the course of $\mathcal{N}_1$ . . . 196
12.4 Evolution of `oce`'s relative clock offset over the course of $\mathcal{N}_1$ . . . 197
12.5 Evolution of `bnl`'s relative clock offset over the course of $\mathcal{N}_1$ . . . 197
12.6 Expanded view of the central line in the previous figure . . . 198
12.7 Evolution of `xor`'s relative clock offset over the course of $\mathcal{N}_1$ . . . 199
12.8 Evolution of `oce`'s relative clock offset over the course of $\mathcal{N}_2$ . . . 199
12.9 Evolution of `lbli`'s relative clock offset over the course of $\mathcal{N}_2$ . . . 200
12.10 Evolution of `sandia`'s relative clock offset over the course of $\mathcal{N}_2$ . . . 200
12.11 Evolution of `umont`'s relative clock offset over the course of $\mathcal{N}_2$ . . . 201
12.12 OTT-pair plot illustrating a clock adjustment . . . 202

12.13 Same measurements after de-noising pair-plot . . . 205
12.14 Clock adjustment via temporary skew . . . 208
12.15 Temporary skew leading to separate pivots . . . 208
12.16 Clock adjustment masked by excessive network delays . . . 209
12.17 Clock adjustment missed because too close to end of connection . . . 210
12.18 Double clock adjustment . . . 211
12.19 Clock adjustment "hiccup" . . . 211
12.20 An OTT pair plot showing strong negative correlation . . . 213
12.21 An OTT pair plot showing relative clock skew . . . 214
12.22 Clock skew obscured by network delays . . . 217
12.23 Enlargement of reverse path . . . 217
12.24 Distribution of $R(n, k)$ for $n = 15$ . . . 220
12.25 Example of extreme clock skew . . . 223
12.26 Strong relative clock skew of 6% . . . 224
12.27 Example of puzzling `oce` behavior . . . 225
12.28 Another example of puzzling `oce` behavior . . . 225
12.29 One more example of puzzling `oce` behavior . . . 226
12.30 Initial packet filter timing glitch . . . 229

13.1 Sequence plot showing a connection with 36% of data packets delivered out-of-order . . . 235
13.2 Sequence plot showing a connection with an out-of-order gap of 54 packets . . . 236
13.3 Out-of-order delivery with two distinct slopes . . . 236
13.4 Sequence plot of entire connection shown in previous figure . . . 237
13.5 Sequence plot of ack delivered out-of-order . . . 238
13.6 Sequence plot of two acks delivered out-of-order and very late . . . 238
13.7 Distribution of out-of-order delivery interval for $\mathcal{N}_1$ data packets . . . 240
13.8 Distribution of data packet out-of-order delivery interval for $\mathcal{N}_1$ and $\mathcal{N}_2$ . . . 241
13.9 Sequence plot showing retransmission event leading to **top** duplicate ack series . . . 244
13.10 Enlargement of **top** duplicate ack series . . . 245
13.11 Two acks replicated 8 times each . . . 246
13.12 Data packet replicated 22 times . . . 247
13.13 Data packet replicated at sender . . . 247

14.1 Paired sequence plot showing timing of data packets at sender and when received 256
14.2 Same plot with acks included . . . 257
14.3 Receiver sequence plot illustrating difficulties of packet-pair bottleneck bandwidth estimation in the presence of out-of-order arrivals . . . 258
14.4 Receiver sequence plot showing two distinct bottleneck bandwidths . . . 260
14.5 Enlargement of part of the previous figure . . . 261
14.6 Enlargement of part of the previous figure . . . 262
14.7 Multi-channel phasing effect . . . 263
14.8 Peak-rate optimistic and conservative bottleneck estimates, window-limited connection . . . 266
14.9 Erroneous optimistic estimate due to data packet compression . . . 267

14.10 Histogram of different single-bottleneck estimates for $\mathcal{N}_1$ . . . 276
14.11 Histogram of different single-bottleneck estimates for $\mathcal{N}_2$ . . . 277
14.12 Box plots of bottlenecks for different $\mathcal{N}_2$ receiving sites . . . 280
14.13 Time until a 20% shift in bottleneck bandwidth, if ever observed . . . 281
14.14 Symmetry of median bottleneck rate . . . 283
14.15 Sequence plot reflecting halving of bottleneck rate . . . 284
14.16 Excerpt from a trace exhibiting a false "multi-channel" bottleneck . . . 285
14.17 Self-clocking TCP "fast recovery" . . . 286

15.1 Connection durations for $\mathcal{N}_1$ (solid) and $\mathcal{N}_2$ (dotted) . . . 292
15.2 Connection durations for sites common to $\mathcal{N}_1$ (solid) and $\mathcal{N}_2$ (dotted) . . . 294
15.3 Hourly variation in ack loss rate for North American connections . . . 297
15.4 Hourly variation in ack loss rate for European connections . . . 298
15.5 Successful North American measurements, per hour . . . 298
15.6 Successful European measurements, per hour . . . 299
15.7 $\mathcal{N}_2$ loss rates for data packets and acks . . . 301
15.8 Complementary distribution plot of $\mathcal{N}_2$ unloaded data packet loss rate . . . 303
15.9 Complementary distribution plot of $\mathcal{N}_2$ loaded data packet loss rate . . . 304
15.10 Complementary distribution plot of $\mathcal{N}_2$ ack loss rate . . . 304
15.11 Distribution of packet loss outage durations . . . 307
15.12 Distribution of packet loss outage durations exceeding 200 msec . . . 308
15.13 Log-log complementary distribution plot of $\mathcal{N}_2$ ack outage durations . . . 308
15.14 Receiver sequence plot showing packet lost at or before bottleneck link . . . 311
15.15 Receiver sequence plot showing packet lost after bottleneck link . . . 311
15.16 Evolution of how well observing a zero-loss connection predicts that a future connection will also be zero-loss . . . 314
15.17 Evolution of how well observing a non-zero-loss connection predicts that a future connection will also be non-zero-loss . . . 315
15.18 Evolution of the mean difference in loss-rate between successive connections along the same path . . . 316
15.19 Receiver sequence plot showing large number of sequence holes . . . 317
15.20 Redundant retransmissions subsequent to previous figure . . . 318
15.21 Sender sequence plot showing failure of RTO adaption . . . 320

16.1 Distribution of the ratio between a connection's maximum RTT to minimum RTT 328
16.2 Log-log complementary distribution plot of max-min RTT ratio . . . 328
16.3 Distribution of inverse ratio (minimum RTT to maximum RTT) . . . 329
16.4 Q-Q plot of ratio of minimum RTT to maximum RTT versus fitted normal distribution . . . 329
16.5 Distribution of RTT interquartile range . . . 330
16.6 Distribution of RTT interquartile range, normalized to minimum RTT . . . 331
16.7 Distribution of difference between maximum RTT and minimum RTT, normalized by interquartile range . . . 331
16.8 Distribution of interquartile and max-min OTT variation . . . 333

16.9 Scatter plot of interquartile ranges of unloaded data packet OTT variations versus acks . . . 334
16.10 Scatter plot of ack loss rate versus interquartile ack OTT variation, for $\mathcal{N}_2$ connections that lost at least one ack . . . 336
16.11 Evolution of how the interquartile range of normalized ack OTT variation differs with time . . . 337
16.12 Evolution of how the interquartile range of raw ack OTT variation differs with time 338
16.13 OTT plot revealing "broken" bottleneck estimate: one that is too low . . . 339
16.14 Another OTT plot revealing a "broken" bottleneck estimate: one that failed to detect a change in the bottleneck rate . . . 340
16.15 OTT plot showing virtually all OTT variation due to connection's own queueing load . . . 341
16.16 Enlargement of adjusted OTTs from previous figure . . . 341
16.17 Ack OTT plot showing 10-sec periodicities . . . 342
16.18 Paired sequence plot showing ack compression . . . 344
16.19 Data packet timing compression . . . 346
16.20 Rampant data packet timing compression . . . 347
16.21 Receiver sequence plot showing major receiver compression . . . 347
16.22 Ack OTT plot for a connection with $\widehat{\tau} = 4$ sec for ΔQ_τ . . . 350
16.23 Ack OTT plot for a connection with $\widehat{\tau} = 1$ sec for $Q_\tau^{\max}$ . . . 350
16.24 Proportion (normalized) of connections with given timescale of maximum sustained delay variation ($\widehat{\tau}$) . . . 352
16.25 Proportion (normalized) of connections with given timescale of maximum peak delay variation ($\widehat{\tau}$) . . . 353
16.26 Distribution of $\mathcal{N}_1$ inferred available bandwidth (β) . . . 357
16.27 Distribution of $\mathcal{N}_2$ inferred available bandwidth (β) . . . 357
16.28 Distribution of $\mathcal{N}_1$ inferred available bandwidth (β) for connections with bottleneck rates exceeding 100 Kbyte/sec . . . 359
16.29 Distribution of $\mathcal{N}_2$ inferred available bandwidth (β) for connections with bottleneck rates exceeding 100 Kbyte/sec . . . 359
16.30 Distribution of $\mathcal{N}_2$ inferred available bandwidth (β) for connections with bottleneck rates exceeding 250 Kbyte/sec . . . 360
16.31 Distribution of $\mathcal{N}_1$ minimum inferred available bandwidth (β) for connections with bottleneck rates exceeding 100 Kbyte/sec . . . 360
16.32 Distribution of $\mathcal{N}_1$ maximum inferred available bandwidth (β) for connections with bottleneck rates exceeding 100 Kbyte/sec . . . 361
16.33 Distribution of $\mathcal{N}_2$ inferred available bandwidth (β) for U.S. connections . . . 362
16.34 Distribution of $\mathcal{N}_2$ inferred available bandwidth (β) for European connections . 363
16.35 Evolution of difference between inferred available bandwidth (β) for successive connections . . . 363

List of Tables

I	Sites participating in first experiment ($\mathcal{R}_1$)	24
II	Additional sites participating in second experiment ($\mathcal{R}_2$)	25
III	Summary of routing experiment difficulties	27
IV	Uncertain router sites	30
V	Router cities	32
VI	Persistent routing loops in $\mathcal{R}_1$	37
VII	Persistent routing loops in $\mathcal{R}_2$	40
VIII	Failure modes for unreachable hosts in $\mathcal{R}_1$	58
IX	Failure modes for unreachable hosts in $\mathcal{R}_2$	58
X	Summary of representative routing pathologies	69
XI	Tightly-coupled routers	76
XII	Summary of persistence at different time scales	89
XIII	Summary of TTL method for detecting route changes	90
XIV	Sites participating in the packet dynamics study	123
XV	TCP Implementations known to `tcpanaly`	144
XVI	Relationship between relative clock accuracy and clock adjustments	230
XVII	Relationship between relative clock accuracy and clock skew	231
XVIII	Types of results of bottleneck estimation for $\mathcal{N}_1$ and $\mathcal{N}_2$	274
XIX	Types of results after eliminating trace pairs with `lbli`	274
XX	Raw and user-data rates of different common links	278
XXI	Ack loss rates for different connection geographies	295
XXII	Conditional ack loss rates for different connection geographies	296
XXIII	Unconditional and conditional loss rates for different packet types	306
XXIV	Proportion of redundant retransmissions (RRs) due to different causes	319

Acknowledgements

This work has its roots in the teaching, help, patience, and inspiration of a great number of people, to whom I wish to express my heartfelt gratitude.

Simply put, Van Jacobson is the reason I have studied networking; the reason I embarked on this study; and the reason I had faith that the work would, with sufficient diligence, yield a host of new insights. I am delighted that, having known him for nearly twenty years, I still find he has much to teach me.

Likewise, this work drew inspiration and invaluable support from Domenico Ferrari. The energy and respect that he affords to both his students' efforts, and to his students themselves, has made it a privilege to be advised by him.

I have also been delighted to have Sally Floyd as my mentor, colleague, and friend. She has listened to countless half-baked ideas of how to analyze and interpret various measurements, and has always patiently separated the promising from the harebrained. This calibration of ideas, and her suggestions on how to then pursue the more promising ones, has proved invaluable for fostering my sense of how to conduct sound research.

A number of others played major roles in shaping this work. I would particularly like to thank John Rice and Mike Luby for their industrious efforts in serving on my dissertation committee, which led to the work being much more solid than it would otherwise have been.[1]

My heartfelt thanks to Greg Minshall, for his detailed, insightful comments on nearly every page of the work (and for his willingness to burn an entire Friday evening discussing some of them); and to Amit Gupta, John Hawkinson, Kurt Lidl, Craig Partridge, and anonymous SIGCOMM and *IEEE/ACM Transactions on Networking* referees, all of whom contributed very helpful comments on earlier versions of the work.

I would like to also thank my colleagues at the Network Research Group: Kevin Fall, Craig Leres, and Steve McCanne, for their much appreciated ideas, support, and feedback.

Special thanks to Kathryn Crabtree, for her untiring help in surmounting innumerable administrative hurdles along the dissertation trail. She is an invaluable asset to UCB computer science.

This work would not have been possible without the efforts of the many volunteers who installed the Network Probe Daemon at their sites. In the process they endured debugging headaches, `inetd` crashes, software updates, and a seemingly endless stream of queries from me regarding their site's behavior. I am indebted to:

Guy Almes and Bob Camm (`adv`);
Jos Alsters (`unij`);
Jean-Chrysostome Bolot (`inria`);
Hans-Werner Braun, Kim Claffy, and Bilal Chinoy (`sdsc`);
Randy Bush (`rain`);
Jon Crowcroft and Atanu Ghosh (`ucl`);
Peter Danzig and Katia Obraczka (`usc`);
Mark Eliot (`sri`);
Robert Elz (`austr`);

[1] Particular thanks to Mike for throwing down the glove, and for knowing which glove to use.

Teus Hagen (oce);
Steinar Haug and Håvard Eidnes (sintef1, sintef2);
John Hawkinson (near and panix);
TR Hein (xor);
Tobias Helbig and Werner Sinze (ustutt);
Paul Hyder (ncar);
Alden Jackson (sandia);
Kate Lance (austr2);
Craig Leres (lbl);
Kurt Lidl (pubnix);
Peter Linington, Alan Ibbetson, Peter Collinson, and Ian Penny (ukc);
Steve McCanne (lbli);
John Milburn (korea);
Walter Mueller (umann);
Evi Nemeth, Mike Schwartz, Dirk Grunwald, Lynda McGinley (ucol, batman);
Francóis Pinard (umont);
Jeff Polk and Keith Bostic (bsdi);
Todd Satogata (bnl);
Doug Schmidt and Miranda Flory (wustl);
Sorell Slaymaker and Alan Hannan (mid);
Don Wells and Dave Brown (nrao);
Gary Wright (connix);
John Wroclawski (mit);
Cliff Young and Brad Karp (harv); and
Lixia Zhang, Mario Gerla, and Simon Walton (ucla).

I am likewise indebted to Keith Bostic, Evi Nemeth, Rich Stevens, George Varghese, Andres Albanese, Wieland Holfelder, and Bernd Lamparter for their invaluable help in recruiting NPD sites. Thanks, too, to Peter Danzig, Jeff Mogul, and Mike Schwartz for feedback on the design of NPD.

This work also benefited from discussions with Guy Almes, Tom Anderson, Robert Elz, Teus Hagen, John Krawczyk, Kate Lance, Dun Liu, Paul Love, Jamshid Mahdavi, Matt Mathis, Dave Mills, Pravin Varaiya, Curtis Villamizar, and Walter Willinger.

A preliminary analysis of the $\mathcal{R}_1$ routing dataset was done by Mark Stemm and Ketan Patel.

Often to understand the behavior of particular routers or to determine their location, I asked personnel from the organization responsible for the routers. I was delighted at how willing they were to help, and in this regard would like to acknowledge:

Vadim Antonov, Tony Bates, Michael Behringer, Per Gregers Bilse, Bjorn Carlsson, Peggy Cheng, Guy Davies, Sean Doran, Bjorn Eriksen, Amit Gupta, Tony Hain, John Hawkinson again!, Susan Harris, Ittai Hershman, Kevin Hoadley, Scott Huddle, James Jokl, Kristi Keith, Harald Koch, Craig Labovitz, Tony Li, Martijn Lindgreen, Ted Lindgreen, Dan Long, Bill Manning, Milo Medin, Keith Mitchell, Roderik Muit, Chris Myers, Torben Nielsen, Richard Nuttall, Mark Oros, Michael Ramsey, Juergen Rauschen-

bach, Douglas Ray, Brian Renaud, Jyrki Soini, Nigel Titley, Paul Vixie, and Rusty Zickefoose.

Finally, this work would never have been realized without the ongoing support provided by the Lawrence Berkeley National Laboratory. I am deeply grateful. In particular, I would like to thank Stu Loken and Ed Theil for their efforts and encouragement.

This work was supported by the Director, Office of Energy Research, Scientific Computing Staff, of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098.