Addressing the Challenges of Web Data Transport

by

Venkata Narayana Padmanabhan

B.Tech. (Indian Institute of Technology, Delhi) 1993

M.S. (University of California at Berkeley) 1995

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:

Professor Randy H. Katz, Chair
Professor Steven R. McCanne
Professor J. George Shanthikumar

1998

Addressing the Challenges of Web Data Transport

Copyright © 1998

by

Venkata Narayana Padmanabhan

Abstract

Addressing the Challenges of Web Data Transport

by

Venkata Narayana Padmanabhan

Doctor of Philosophy in Computer Science

University of California at Berkeley

Professor Randy H. Katz, Chair

In just a few years since its inception, the World Wide Web has grown to be the most dominant application in the Internet. In large measure, this rapid growth is due to the Web's convenient point-and-click interface and its appealing graphical content. Since Web browsing is an interactive activity, minimizing user-perceived latency is an important goal. However, layering Web data transport on top of the TCP protocol poses several challenges to achieving this goal.

First, the transmission of a Web page from a server to a client involves the transfer of multiple distinct components, each in itself of some value to the user. To minimize user-perceived latency, it is desirable to transfer the components concurrently. TCP provides an ordered byte-stream abstraction with no mechanism to demarcate sub-streams. If a separate TCP connection is used for each component, as with HTTP/1.0, uncoordinated competition among the connections could exacerbate congestion, packet loss, unfairness, and latency.

Second, Web data transfers happen in relatively short bursts, with intervening idle periods. It is difficult to utilize bandwidth effectively during a short burst because discovering how much bandwidth is available requires time. Latency suffers as a consequence.

To address these problems, we first developed a new connection abstraction for HTTP, called *persistent-connection HTTP* (*P-HTTP*). The key ideas are to share a persistent TCP connection for multiple Web page components and to pipeline the transfers of these components to reduce latency. These ideas, developed by us in 1994, have been adopted by the HTTP/1.1 protocol. The main drawback of P-HTTP, though, is that the persistent TCP connection imposes a linear ordering on the Web page components, which are inherently independent.

This drawback of P-HTTP led us to develop a comprehensive solution, which has two components. The first component, *TCP session*, decouples TCP's ordered byte-stream service abstraction from its congestion control and loss recovery mechanisms. It *integrates* the latter mechanisms across the set of concurrent connections between a pair of hosts, thereby combining the flexibility of separate connections with the performance efficiency of a shared connection. This integration decreases download time by up to a factor of ten compared to HTTP/1.0 layered on standard TCP. TCP session does not alter TCP's messaging semantics, so deployment only involves local changes at the sender.

The second component of our solution, *TCP fast start*, improves bandwidth utilization for short transfers by reusing information about the network conditions cached in the recent past. To avoid adverse effects in case the cached information is stale, TCP fast start exploits priority dropping at routers, and augments TCP's loss recovery mechanisms to quickly detect and abort a failed fast start attempt.

In addition to the two challenges we have discussed, a third challenge arises from the increasing deployment of *asymmetric access networks*. Although Web browsing has asymmetric bandwidth requirements, bandwidth asymmetry could adversely impact Web download performance due to a disruption in the flow of acknowledgement feedback that is critical to sustaining good TCP throughput. To avoid performance degradation, we have developed end-host and router-based techniques, that both reduce disruption in the feedback and reduce TCP's dependence on such feedback. In certain situations, these techniques help decrease download time by a factor of fifteen.

This thesis includes mathematical and trace-based analysis, simulation, implementation and performance evaluation. In addition to the algorithms that we have designed and software that we have developed, our contributions include a set of paradigms for advancing the state-of-the-art in Internet transport protocols. These paradigms include the use of shared state and/or persistent state, and the exploitation of differentiated services mechanisms in routers.

Professor Randy H. Katz, Chair

To my parents and my sisters, Priya and Sumi.

Vidya dadaathi vinayam

vinayaath yaathi paathrathaam

paathrathwaath dhanam apnothi

dhanaath dharmam thatha sukham

(ancient Sanskrit saying)

English Translation

Education gives modesty;

through modesty one becomes worthy;

through worthiness one gets wealth;

through wealth one can be righteous and happy.

Table of Contents

Chapter 1
Introduction **1**

1.1 Growth of the Internet ... 3
1.2 TCP/IP Protocol Architecture ... 5
1.3 World Wide Web ... 7
1.3.1 Web Data Characteristics ... 7
1.3.2 Web Traffic Characteristics ... 9
1.4 Transmission Control Protocol (TCP) ... 10
1.5 Mismatch Between the Web Application and the TCP Protocol ... 12
1.6 Impact of Heterogeneous Access Network Technologies ... 13
1.7 Addressing the Challenges ... 15
1.8 Design Principles ... 17
1.9 Thesis Outline ... 18

Chapter 2
Background and Related Work **20**

2.1 Transmission Control Protocol (TCP) ... 20
2.1.1 TCP Connection Setup ... 21
2.1.2 TCP Protocol Control Block (TCB) ... 22
2.1.3 TCP Acknowledgements ... 23
2.1.4 Loss Detection and Recovery ... 24
2.1.5 Congestion Avoidance and Control ... 25
2.1.6 TCP Options ... 28
2.1.7 Summary ... 29
2.2 World Wide Web ... 30
2.2.1 Hypertext Transfer Protocol version 1.0 (HTTP/1.0) ... 30
2.2.2 Performance Problems ... 32
2.2.3 Summary ... 33
2.3 TCP Enhancements ... 33
2.3.1 TCP NewReno ... 33
2.3.2 Explicit Congestion Notification (ECN) ... 34
2.3.3 TCP Vegas ... 35
2.3.4 Transaction TCP (T/TCP) ... 35
2.3.5 TCP Protocol Control Block Interdependence ... 36

2.3.6 Rate-Based Pacing (RBP) 37
2.3.6.1 TCP over Asymmetric Networks 38
2.4 Other Transport Protocols 38
2.4.1 NETBLT 39
2.4.2 Packet-Pair Bandwidth Probing 39
2.5 Summary 40

Chapter 3
Methodology **41**

3.1 Experimental Network Testbed 42
3.1.1 DirecPC Satellite Network 42
3.1.2 Hybrid Wireless Cable Modem Network 44
3.1.3 Ricochet Packet Radio Network 45
3.2 Web Server Traffic Trace Analysis 46
3.2.1 The IBM 1996 Olympics Web Server 46
3.2.2 Data Collection Methodology 47
3.2.3 Reconstructing the TCP Packet Trace 48
3.3 Simulation Study 49
3.4 Implementation 51
3.5 Evaluation 51
3.6 Summary 52

Chapter 4
Analysis of HTTP Performance **53**

4.1 HTTP/1.0 Basics 53
4.1.1 Sequential Retrieval 54
4.1.2 Concurrent Retrieval 55
4.2 Performance Analysis 55
4.2.1 Mathematical Analysis 55
4.2.2 Impact of Connection Length on Throughput 59
4.2.3 Impact of Limited Bandwidth 59
4.3 Analysis of Traces from the IBM Olympics Web Server 61
4.3.1 TCP Window Size Distribution 62
4.3.2 Effectiveness of TCP Loss Recovery 64
4.3.3 Effectiveness of TCP Congestion Control 65
4.3.3.1 Throughput Analysis 65
4.3.3.2 Congestion Avoidance Analysis 67
4.3.3.3 Congestion Control Analysis 68
4.4 Summary 71

Chapter 5
Persistent-Connection HTTP **72**

5.1 Goals of P-HTTP ... 73

5.2 Persistent Connections ... 73
5.2.1 Multiplexing Logical Data Streams ... 75
5.2.2 Negotiating the Use of Persistent Connections ... 77
5.2.3 Terminating Persistent Connections ... 78

5.3 Pipelining ... 79
5.3.1 The GETALL method ... 79
5.3.2 The GETLIST method ... 81

5.4 Implementation ... 81
5.4.1 Client-Server Implementation ... 81
5.4.2 Proxy-based Implementation ... 83

5.5 Experimental Results ... 86
5.5.1 Terrestrial Wide-Area Network ... 88
5.5.2 Satellite-based Wide-Area Network ... 88
5.5.3 Interaction of P-HTTP with TCP Congestion Control ... 91

5.6 Extensions of P-HTTP ... 92
5.6.1 Persistent Connections in HTTP/1.1 ... 93
5.6.2 Application-independent Multiplexing Protocols ... 93

5.7 Limitations of P-HTTP ... 93

5.8 Summary ... 95

Chapter 6
TCP Session **96**

6.1 Motivation ... 96

6.2 TCP Session Overview ... 98

6.3 Design and Operation of TCP Session ... 100
6.3.1 Session Control Block ... 100
6.3.2 Integrated Congestion Control ... 102
6.3.3 Connection Scheduling ... 105
6.3.4 Integrated Loss Recovery ... 106
6.3.4.1 Loss Recovery Rules ... 109

6.4 Implementation in BSD/OS ... 111

6.5 Performance Results ... 113
6.5.1 Competition versus Sharing between Concurrent Connections ... 113
6.5.2 Simulation Results ... 115
6.5.2.1 Impact of the Number of Simultaneous Web Downloads ... 117

6.5.2.2 Impact of Integrated Loss Recovery 119
6.5.2.3 Impact of the Number of Inline Images 120
6.5.2.4 Impact of the Inline Image Size 122
6.5.2.5 Impact of Interfering Bulk Transfer Traffic 123
6.5.2.6 Impact of Heterogeneous Traffic Mix 124
6.5.2.7 Impact of the Link Speed and Delay 125
6.5.3 Summary of Results 127
6.6 Limitations of TCP session 128
6.7 Summary 129

Chapter 7
TCP Session: Advanced Issues **130**

7.1 Connection Scheduling 130
7.1.1 Hierarchical Round-Robin Scheduler 131
7.1.2 Comparison With Alternatives 133
7.1.2.1 Scheduling Independent TCP Connections 133
7.1.2.2 Scheduling Logical Data Streams within a Single TCP Connection 135
7.1.3 Summary 136
7.2 Adapting Integrated Congestion Control for Proxy Hosts 136
7.3 Potential For False Retransmissions 138
7.4 Summary 140

Chapter 8
TCP Fast Start **141**

8.1 Motivation 142
8.2 Design of TCP Fast Start 143
8.2.1 Basic Idea 144
8.2.2 Sender Algorithm 145
8.2.2.1 Initiation and Termination of Fast Start 145
8.2.2.2 ..... Initialization of State Variables 145
8.2.2.3 Clocking Out Data During Fast Start 146
8.2.2.4 Quick Detection/Recovery From Failed Fast Start Attempt 146
8.2.3 Router Mechanism 148
8.3 Implementation in BSD/OS 149
8.4 Performance Results 151
8.4.1 BSD/OS Measurements 151
8.4.2 Simulation Results 153
8.4.2.1 Single Web Download with Constant Load 155
8.4.2.2 Single Web Download with Changed Load 156
8.4.2.3 Impact of RED Buffer Management 159
8.4.2.4 Competing Web Downloads with Constant Load 160

8.4.2.5 Competing Web Downloads with Changed Load 161
8.4.2.6 Competing, Heterogeneous Web Downloads with Changed Load 163
8.4.3 Summary of Results 164
8.5 Discussion 164
8.6 Summary 166

Chapter 9
Asymmetric Access Networks **167**

9.1 Motivation 168
9.2 Analysis of Performance Problems 169
9.2.1 Unidirectional Transfer 169
9.2.2 Bidirectional Transfers 172
9.2.3 Summary of Performance Problems 174
9.3 Solution to the Problems Caused by Bandwidth Asymmetry 174
9.3.1 Reducing Disruption in the Ack Stream 174
9.3.1.1 Ack Congestion Control (ACC) 175
9.3.1.2 Ack Filtering (AF) 176
9.3.1.3 Acks-first Scheduling 177
9.3.2 Adapting to Disruption in the Ack Stream 177
9.3.2.1 Sender Adaptation (SA) 177
9.3.2.2 Ack Reconstruction (AR) 178
9.4 Performance Evaluation 178
9.4.1 Unidirectional Traffic 180
9.4.2 Bidirectional Traffic 181
9.4.2.1 Bulk Transfers 181
9.4.2.2 Web Download 184
9.4.3 Summary of Results 186
9.5 Discussion 187
9.6 Summary 188

Chapter 10
Conclusions and Future Work **190**

10.1 Challenges and Solutions 190
10.2 Contributions 191
10.2.1 Separation of Protocol Service Model from Protocol Mechanisms 191
10.2.2 Shared State 192
10.2.3 Persistent State 192
10.2.4 Exploiting Support for Differentiated Services 193
10.2.5 Redefining the Coupling Between the Data and Control Paths 193
10.2.6 Summary 194

10.3 Design Principles Re-visited 194
10.3.1 Robustness 194
10.3.2 Wide Applicability 195
10.3.3 End-to-End Principle 195
10.3.4 Incremental Deployment 196
10.3.5 Summary 196
10.4 Directions for Future Work 196
10.4.1 Sharing State Across Hosts 196
10.4.2 Comprehensive Use of Persistent State 197
10.4.3 Guarding Against Malicious Users 197
10.4.4 Improved Metrics for Quantifying Perceived Performance 197
10.4.5 Multicast-Based Dissemination of Web Data 198
10.5 Availability 198

Appendix A
Performance Analysis Tools 199

Appendix B
Implementing TCP Session 201

Bibliography **208**

List of Figures

Figure 1.1 Snapshot of a Web page. 2
Figure 1.2 The exponential growth of the Internet and the Web 4
Figure 1.3 The OSI network protocol stack. 6
Figure 1.4 Contrasting P-HTTP and TCP session. 16
Figure 2.1 TCP fast retransmission mechanism. 24
Figure 2.2 TCP congestion control dynamics. 27
Figure 2.3 A schematic depiction of Web clients, proxies, and servers. 30
Figure 2.4 HTTP/1.0 timeline. 32
Figure 3.1 DirecPC network topology. 43
Figure 3.2 Hybrid wireless cable modem network topology. 44
Figure 3.3 Ricochet packet radio network topology. 45
Figure 3.4 IBM Olympics server topology. 47
Figure 3.5 Example simulation topologies. 50
Figure 4.1 HTTP/1.0 timeline with sequential and concurrent connections. 54
Figure 4.2 Analytically computed Web page download time. 58
Figure 4.3 Throughput versus connection length. 60
Figure 4.4 Distribution of the receiver advertised window. 62
Figure 4.5 Correlation of throughput and number of concurrent connections. 66
Figure 4.6 Distribution of packet loss across concurrent connections. 69
Figure 4.7 Congestion backoff factor versus number of concurrent connections. 70
Figure 5.1 P-HTTP timeline. 74
Figure 5.2 GETALL and GETLIST timeline. 80
Figure 5.3 Proxy-based implementation of P-HTTP. 83
Figure 5.4 Proxy-based approach to emulating several protocol configurations. 85
Figure 5.5 Experimental network configuration. 87
Figure 5.6 P-HTTP: WAN results. 89
Figure 5.7 P-HTTP: DirecPC experimental results. 90
Figure 5.8 Illustration of persistent connection without pipelining. 91
Figure 5.9 Illustration of slow-start re-start. 92
Figure 6.1 TCP session congestion window dynamics. 104
Figure 6.2 Illustration of TCP fast retransmission. 107
Figure 6.3 Illustration of later acks in a TCP session. 108
Figure 6.4 Illustration of loss recovery using later ack. 110
Figure 6.5 Rules for integrated loss recovery. 111
Figure 6.6 Dynamics of concurrent connections with TCP and TCP session. 114
Figure 6.7 TCP session: Simulation topology and parameter settings. 116
Figure 6.8 Download time and packet loss versus simultaneous connections. 118

Figure 6.9 Impact of integrated loss recovery. 119
Figure 6.10 Download time and fairness versus number of inline images 120
Figure 6.11 Download time and fairness inline image size. 121
Figure 6.12 Download time versus interfering bulk transfers. 122
Figure 6.13 Impact of a heterogeneous traffic mix. 123
Figure 6.14 Impact of link bandwidth: 28.8 Kbps and 45 Mbps. 125
Figure 6.15 Impact of large RTT: 400 ms. 125
Figure 7.1 Hierarchical round-robin scheduling. 132
Figure 7.2 Sequence number trace with connection scheduling. 134
Figure 7.3 Sequence number trace with connection scheduling and stall. 135
Figure 7.4 Adapting integrated congestion control to proxy hosts. 138
Figure 7.5 Impact of integrated loss recovery on false retransmissions. 139
Figure 8.1 TCP fast start: download time versus Web page size. 152
Figure 8.2 TCP fast start: congestion window and sequence number plots. 153
Figure 8.3 TCP fast start: simulation topology and parameter settings. 154
Figure 8.4 Constant-load experiment. 155
Figure 8.5 Changed-load experiment. 157
Figure 8.6 Impact of fast start without priority dropping. 158
Figure 8.7 Impact of RED buffer management. 159
Figure 8.8 Constant-load experiment: Impact of bandwidth and delay. 160
Figure 8.9 Changed-load experiment: Impact of bandwidth and delay. 162
Figure 9.1 Cable network: downstream throughput versus upstream bandwidth. 170
Figure 9.2 Cable network: bidirectional transfers. 172
Figure 9.3 Bandwidth asymmetry: simulation topology. 179
Figure 9.4 Download time versus page size. 180
Figure 9.5 Ack filtering: bidirectional traffic. 182
Figure 9.6 Ack congestion control: bidirectional traffic. 184
Figure B.1 Ack congestion control: bidirectional traffic. 206

List of Tables

Table 1.1	Internet traffic distribution by application	5
Table 1.2	Web data types	8
Table 1.3	Access network technologies.	14
Table 4.1	Summary of IBM Olympics server trace analysis	63
Table 6.1	Alternative approaches to Web data transport	97
Table 8.1	TCP fast start: download time with heterogeneous hosts	163
Table 9.1	Cable network: throughput with bidirectional traffic	181
Table 9.2	Cable network: download time with bidirectional traffic	185

Acknowledgements

As I stand at the threshold of earning my doctorate, I am overwhelmed when I recall all the people who have helped me get this far.

First and foremost, I would like to thank my Ph.D. advisor, Professor Randy Katz, for his constant support, guidance, and inspiration. Randy is a truly remarkable advisor who grants students a lot of freedom to explore new ideas, but at the same time interacts closely with them. I have greatly benefited both from his excellent technical advice and from his role-model as a successful team leader and researcher. I look forward to continuing my association with him in the future.

I am indebted to Jeff Mogul for introducing me to the problem of improving HTTP performance, which eventually led to my dissertation. Jeff has been a great mentor and collaborator. I would like to thank him and the Compaq (formerly Digital) Western Research Laboratory for providing me generous support, beginning with a summer internship, and continuing with access to their computing facilities as an honorary researcher.

I would like to thank Professor Steve McCanne for being on my qualifying exam and thesis committees. Steve's insightful comments and constructive criticisms have been extremely useful to me. I am also grateful to Professors George Shanthikumar and Bob Brodersen for serving on my committees and providing valuable feedback on the ideas presented in this thesis.

I have had very productive collaborations with Hari Balakrishnan, Jeff Mogul, Srini Seshan and Mark Stemm, which have contributed significantly to this thesis. I have also had valuable discussions on various aspects of the thesis with B. R. Badrinath, Sally Floyd, Tom Henderson, T. V. Lakshman, Giao Nguyen, Vern Paxson, and K. K. Ramakrishnan. My graduate student career has also been enriched by interactions with several researchers including Mary Baker, Eric Brewer, Ramón Cáceres, Domenico Ferrari, Akthar Jameel, Barry Leiner, Reiner Ludwig, Kevin Mills, Rob Ruth, Subir Varma, and Yongguang Zhang. I would like to thank them all.

This work was supported primarily by DARPA under contract DAAB07-95-C-D154. It was also supported by grants from Hughes Aircraft Corporation, Hybrid Networks Inc., Metricom Inc. and the California MICRO program. Son Dao and Yongguang Zhang of Hughes Research Labs, Ed Moura of Hybrid Inc., and Mike Ritter of Metricom Inc. helped set up the network testbed for my

experimental work. Brian Shiratsuki, Keith Sklower and Ken Lutz were always willing to help straighten out any kinks that developed in the testbed. Keith Sklower was largely responsible for getting our connection to the Hughes DirecPC network off the ground. I am grateful to them all.

Kathryn Crabtree was ever helpful in guiding me through the administrative labyrinth in Berkeley. I am convinced that every research organization should have someone as adept as her in keeping bureaucratic red tape at bay! Terry Lessard-Smith and Bob Miller were great project administrators. I doubt if large systems projects in Berkeley would run half as smoothly as they do without their tireless efforts, or be half as much fun as they are without the excellent project retreats they organize.

My graduate student colleagues have made my stay at Berkeley a truly enjoyable one. Their collective intellectual energy has been a great motivator for me. I would like to thank Elan Amir, Hari Balakrishnan, Yatin Chawathe, Armando Fox, Jeff Gilbert, Steve Gribble, Tom Henderson, Todd Hodes, Dan Jiang, Bruce Mah, Giao Nguyen, Mark Stemm, Helen Wang, and Tao Ye.

The "443 Soda gang" of Elan, Hari, Todd, Mark, and members-at-large, Tom, Giao, and Srini was always an enthusiastic (and leak-proof!) audience for almost any topic under the sun, including many politically-incorrect ones. I certainly hope the gang will continue to flourish in cyberspace (with the protection of PGP encryption, of course!) as the ranks of the members-at-large swells.

I would like to thank all my teachers, from elementary school through graduate school, for providing me with an excellent education. I am particularly indebted to my professors at the Indian Institute of Technology for giving me a solid foundation in Computer Science and Engineering.

Most important of all, I would like to express my gratitude to my family for being an unstinting source of support and encouragement. My grandparents have inspired me through their courage in overcoming the challenges of life. I am saddened that they did not live to see my graduation. My parents have taught me the value of education and have worked hard to provide me the very best of it. They have always been there when I have needed them. My sisters, Priya and Sumi, have been wonderful sources of support. The tinge of jealousy I feel in them both having beaten me to earning the title "Doctor" is effectively neutralised by their abiding affection! Last but not the least, my brothers-in-law, Sekhar and Umesh, have been great friends and have given me excellent advice drawing on their own careers as doctoral students, and subsequently, researchers.