Class-Based Thresholds: Lightweight Active Router-Queue Management for Multimedia Networking

by

Mark Anthony Parris

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Computer Science.

Chapel Hill

2001

Approved by:

Advisor: Kevin Jeffay

Reader: F. Donelson Smith

Reader: Bert Dempsey

Reader: David Stotts

Reader: Ketan Mayer-Patel

© 2001

Mark Anthony Parris

ALL RIGHTS RESERVED

the Intel Corporation, Sun Microsystems, Cabletron Systems, Cisco Systems, the IBM Corporation, and MCNC.

2.3.1. Streaming Control Protocol 29
2.3.2. Loss-Delay Based Adjustment Algorithm 30
2.3.3. Analysis of Responsive Unreliable Approaches 32
3. Application-level Approaches 32
4. Integrated End-System Congestion Management 36
5. Integrating Application Level Prioritization with Router Drop Policies 37
6. Router Based Quality of Service 38
6.1. Integrated Services Architecture 39
6.2. Differentiated Services 40
7. Summary 44
III. ROUTER QUEUE MANAGEMENT 47
1. Buffering in Routers 49
1.1. DropTail When Full (FIFO) 50
1.2. Active Queue Management Concept 52
2. The Evolution of Active Queue Management Policies 53
2.1. Solving the Problems of Lock-out and Full-queues 53
2.2. Providing Notification of Congestion 56
2.2.1. ICMP Source Quench 57
2.2.2. DECbit 58
2.2.3. ECN – Early Congestion Notification 59
2.2.4. RED – Random Early Detection 60
2.2.5. Summary 65
2.3. Non-TCP-Friendly flows 66
2.3.1. FRED 67
2.3.2. Floyd & Fall 69

2.3.3. RIO – RED with In and Out....70
2.3.4. Summary....72
2.4. Summary of Evolution of Active Queue Management....72
3. Key Algorithms in Greater Detail....74
3.1. Drop-Tail (FIFO)....74
3.1.1. Algorithm Description....75
3.1.2. Evaluation....76
3.2. RED....77
3.2.1. Algorithm Description....79
3.2.2. Evaluation....84
3.3. FRED....84
3.3.1. Algorithm....85
3.3.2. Evaluation....92
4. Packet Scheduling....94
4.1. Class-Based Queueing (CBQ)....95
4.1.1. Algorithm....96
4.1.2. Evaluation....99
5. Evaluation of Router Queue Management....101
6. Summary....104
IV. CLASS-BASED THRESHOLDS....106
1. Problem and Motivation....106
2. Goals....108
2.1. Traffic Types and Isolation....108
2.1.1. TCP....108
2.1.2. Multimedia....109

2.1.3. Other 109
2.2. Predictable Performance for Traffic Classes 110
2.2.1. Bandwidth Allocation 110
2.2.2. Manage latency 110
2.2.3. Resource Allocation via Queue Management 111
2.3. Minimize Complexity 111
3. The Algorithm 112
3.1. Design 112
3.1.1. Classification 112
3.1.2. Managing Queue Occupancy 113
3.1.3. Different Drop Policies for Different Types of Traffic 114
3.2. Algorithm Description 115
3.2.1. Declarations and Definitions 115
3.2.2. Drop Modes 118
3.2.3. Departing Packets 119
3.2.4. Early Drop Test 120
3.2.5. Computing the Average 121
4. Configuring CBT 122
4.1. Equations for Bandwidth Allocation 123
4.2. Setting the Other CBT Parameters 127
4.2.1. Weights for Computing Average Queue Occupancy 127
4.2.2. Maximum Drop Probability 127
4.2.3. Maximum Queue Size 128
5. Demonstrating the Effectiveness of CBT 128
5.1. Behavior When All Classes Consume Their Bandwidth Allocation 132

2.1.4. Summary of Blast Measurement Period 189
2.2. Multimedia Measurement Period 190
2.2.1. TCP Goodput 191
2.2.2. Multimedia 193
2.2.3. Summary of Results for Multimedia Measurement Period 201
3. Summary 202
VI. SUMMARY AND CONCLUSION 204
1. Class-Based Thresholds 206
2. Analyzing CBT 207
3. Empirical Analysis of Algorithms 208
4. Empirical Comparison of Algorithms 210
4.1. Blast Measurement Period 211
4.2. Multimedia Measurement Period 212
4.3. Summary 213
5. Future Work 213
5.1. Limitations and Suggested Refinements 213
5.1.1. Imprecision in CBT 213
5.1.2. Accurately Predicting Bandwidth Needs 217
5.2. Further Analysis 218
5.2.1. Complexity and Overhead 218
5.2.2. More Realistic Conditions 219
5.3. Deployment Issues 220
5.3.1. Packet Classification 220
5.3.2. Negotiating Allocations 222
6. Summary 223

APPENDIX A. METHODOLOGY 224

1. Experimental Model 224

1.1. Alternatives for Experimentation 224

2. Network Configuration 228

2.1. Network Configuration 228

2.1.1. Physical Network Configuration 229

2.1.2. End-systems 231

2.1.3. Router Configuration 231

2.1.4. Induced Delay 231

3. Traffic 232

3.1. Understanding the Traffic Loads 234

3.2. Other Traffic 235

3.3. Multimedia 237

3.3.1. Proshare 237

3.3.2. MPEG 239

3.4. TCP 252

3.4.1. HTTP 254

3.4.2. BULK 260

4. Traffic Mixes 264

4.1. Timing of when each type of traffic is introduced 265

5. Data Collection 268

5.1. Network Monitoring 268

5.2. Application Level Monitoring 269

5.2.1. Measuring the Performance of Proshare 270

5.2.2. Measuring the Performance of MPEG 271

5.3. Router Queue Management Overhead 272
5.4. Summary 272
6. Statistical Issues 272
7. Performance metrics 274
7.1. Bandwidth Usage Measures 274
7.2. Latency 276
7.3. Frame-rate 277
8. Summary 278
APPENDIX B. CHOOSING OPTIMAL PARAMETERS 281
1. Methodology 282
2. FIFO 284
3. RED and FRED Analysis 291
3.1. Understanding the Charts and Figures 291
3.2. Fixed Parameters 294
4. RED 295
4.1. RED Threshold Settings 296
4.1.1. HTTP and Proshare 298
4.2. BULK and Proshare 311
5. FRED 318
5.1.1. BULK-Proshare 329
6. CBT 334
6.1. Constant Parameters 334
6.2. Determining Threshold Settings from Bandwidth Allocations 336
6.3. Evaluating Parameter Settings 339
6.4. HTTP-MPEG 340

6.5. BULK-MPEG 349
6.6. HTTP-Proshare 352
6.7. BULK-Proshare 354
6.8. Constraining Other 357
6.9. Summary 358
7. CBQ 359
7.1. HTTP-MPEG 360
7.2. BULK-MPEG 369
7.3. HTTP-Proshare 371
7.4. BULK-Proshare 372
7.5. Summary 374
8. Summary 375
REFERENCES 376

Figure 5.18 TCP Goodput Averaged Over 1 Second Intervals During the Multimedia Measurement Period for HTTP+Proshare 191

Figure 5.19 TCP Goodput Averaged Over 1 Second Intervals During the Multimedia Measurement Period for All Algorithms 192

Figure 5.20 Multimedia Throughput Averaged Over 1 Second Intervals During the Multimedia Measurement Period for HTTP+Proshare 193

Figure 5.21 Multimedia Throughput Averaged Over 1 Second Intervals During the Multimedia Measurement Period 194

Figure 5.22 End-to-End Multimedia Latency During the Multimedia Measurement Period for HTTP+Proshare 195

Figure 5.23 Multimedia Latency During the Multimedia Measurement Period 197

Figure 5.25 Actual Frame-Rate Averaged Over 1 Second Intervals During the Multimedia Measurement Period 199

Figure 5.26 Playable Frame-Rate for HTTP+MPEG 200

Figure 5.27 Playable Frame-Rate During the Multimedia Measurement Period 201

Figure A.1 Logical Network Configuration 229

Figure A.2 Physical Network Configuration 230

Figure A.3 Example of a Traffic Mix 234

Figure A.4 Load Generated by a UDP Blast 236

Figure A.5 Proshare Load 238

Figure A.6 MPEG Frame sizes by Type 241

Figure A.7 MPEG Packets per Frame (Ethernet Packets) 241

Figure A.8 MPEG Average Bytes per 66ms 242

Figure A.9 MPEG Average Packets per 66ms 243

Figure A.10 Average MPEG Loads from Different Traffic Mixes 244

Figure A.11 MPEG Load During Run #1 245

Figure A.12 MPEG Load During Run #2 245

Figure A.13 MPEG Frame Sizes over Time (Crocodile Dundee Movie) 247

Figure A.14 MPEG Load 248

Figure A.15 MPEG Packet Loss vs. Frame Loss 252

Figure A.18 Generated Load vs. Simulated Browsers 255

Figure A.19 HTTP Load Alone (but with bottleneck link of 10Mb/s) 257

Figure A.20 HTTP Load with and without other traffic types present. 258

Figure A.21 HTTP Average Load with and without Other Traffic Types 259

Figure A.22 BULK Load with no other traffic types (but with 10Mb/s bottleneck link) 262

Figure A.23 BULK Load in the Presence of Other Traffic Types 263

Figure A.24 Average Bulk Load with and without Traffic 264

Figure A.25 Traffic Mix for BULK+MPEG 265

Figure A.26 All Traffic mixes 267

Figure A.28 Location of Network Monitoring 269

Figure A.29 Location of Multimedia Traffic Generator 270

Figure B.1 Latency (ms) vs. Maximum Queue Size (packets) with FIFO during the Blast Measurement Period (HTTP-Proshare) 285

Figure B.2 Latency (ms) vs. Maximum Queue Size (packets) with FIFO (BULK-Proshare) 286

Figure B.3 Maximum Queue Size (packets) vs. Throughput (KB/s) with FIFO for All Traffic Mixes during Multimedia Measurement Period 288

Figure B.4 Maximum Queue Size (packets) vs. Packets Lost (packets/second) for FIFO with HTTP-Proshare during Blast Measurement Period 289

Figure B.5 Maximum Queue Size (packets) vs. Frame-rate (frames/second) for FIFO with HTTP-MPEG during the Blast Measurement Period 289

Figure B.6 Maximum Queue Size (packets) vs. Throughput (KB/s) with FIFO during the Blast Measurement Period 290

Figure B.7 Latency (ms) during Blast Measurement Period for RED and HTTP and Proshare 293

Figure B.8 TCP Goodput (KB/s) During Multimedia Measurement Period with RED for HTTP and Proshare. 299

Figure B.9 Aggregate Throughput (KB/s) during Multimedia Measurement Period with RED for HTTP and Proshare 301

Figure B.10 TCP Efficiency during the Multimedia Measurement Period with RED for HTTP and Proshare 303

Figure B.11 TCP Efficiency during Blast Measurement Period with RED for HTTP and Proshare 304

Figure B.12 TCP Load (KB/s) during Blast Measurement Period with RED for HTTP and Proshare 306

Figure B.13 TCP Throughput (KB/s) during Blast Measurement Period with RED for HTTP and Proshare 307

Figure B.14 Network Latency (ms) during Blast Measurement Period for RED with HTTP and Proshare 309

Figure B.15 Network Latency (ms) during Multimedia Measurement Period with RED for HTTP and Proshare 310

Figure B.16 TCP Goodput (KB/s) During Multimedia Measurement Period with RED for BULK and Proshare. 313

Figure B.17 Aggregate Throughput (KB/s) during Multimedia Measurement Period with RED for BULK and Proshare 314

Figure B.18 TCP Efficiency during Multimedia Measurement Period with RED for BULK and Proshare 315

Figure B.19 TCP Efficiency during Multimedia Measurement Period with RED for BULK and Proshare 316

Figure B.20 Network Latency (ms) during Multimedia Measurement Period for RED with BULK and Proshare 317

Figure B.21 TCP Goodput (KB/s) During the Multimedia Measurement Period with FRED for HTTP-Proshare 320

Figure B.22 Aggregate Throughput (KB/s) during Multimedia Measurement Period with FRED for HTTP-Proshare 321

Figure B.23 TCP Throughput (KB/s) during the Blast Measurement Period with FRED for HTTP-Proshare 322

Figure B.24 Aggregate UDP Throughput (KB/s) during the Blast Measurement Period with FRED for HTTP-Proshare 324

Figure B.25 Network Latency (ms) during the Multimedia Measurement Period for FRED with HTTP-Proshare..... 325

Figure B.26 Network Latency (ms) during the Blast Measurement Period with FRED for HTTP-Proshare 326

Figure B.27 TCP Efficiency during the Multimedia Measurement Period with FRED for HTTP-Proshare 328

Figure B.28 TCP Efficiency during the Blast Measurement Period for FRED with HTTP-Proshare 329

Figure B.31 TCP Goodput (KB/s) During Multimedia Measurement Period with FRED for BULK-Proshare 331

Figure B.32 Multimedia Bandwidth Allocation (KB/s) vs. Multimedia Throughput (KB/s) during Blast Measurement Period for HTTP-MPEG 342

Figure B.33 Multimedia Bandwidth Allocation (KB/s) vs. Multimedia Throughput (KB/s) during the Blast Period with HTTP-MPEG on a One-to-One Scale..... 343

Figure B.34 Multimedia Bandwidth Allocation (KB/s) vs. Packet Loss (packets/s) during the Blast Measurement Period for HTTP-MPEG 344

Figure B.35 Bandwidth Allocation for Multimedia (KB/s) vs. Frame Rate (Frames/s) during the Blast Measurement Period for HTTP-MPEG..... 345

Figure B.36 Multimedia Bandwidth Allocation (KB/s) vs. Latency (ms) during the Blast Measurement Period for HTTP-MPEG..... 346

Figure B.37 Multimedia Bandwidth Allocation (KB/s) vs. TCP Throughput (KB/s) during the Blast Measurement Period for HTTP-MPEG 347

Figure B.38 TCP Bandwidth Allocation (KB/s) vs. TCP Throughput (KB/s) during the Blast Measurement Period for HTTP-MPEG 348

Figure B.39 Multimedia Bandwidth Allocation (KB/s) vs. Selected Metrics during the Blast Measurement Period with BULK-MPEG 350

Figure B.40 TCP Bandwidth Allocation (KB/s) vs. TCP Throughput during the Blast Measurement Period with BULK-MPEG 352

Figure B.41 Multimedia Bandwidth Allocation (KB/s) vs. Selected Metrics during the Blast Measurement Period with HTTP-Proshare 354

Figure B.42 Multimedia Bandwidth Allocation (KB/s) vs. Selected Metrics during the Blast Measurement Period with BULK-Proshare 356

Figure B.43 Multimedia and TCP Throughput vs. Allocations on a 1:1 scale during the blast measurement period with BULK-Proshare.....357

Figure B.44 B_{mm} (KB/s) vs. Other Throughput during the Blast Measurement Period Across All Traffic Mixes.....358

Figure B.45 Multimedia Bandwidth Allocation (KB/s) vs. Multimedia Throughput (KB/s) during the Blast Measurement Period for HTTP-MPEG.....362

Figure B.46 Multimedia Bandwidth Allocation (KB/s) vs. Multimedia Throughput (KB/s) during the Blast Period with HTTP-MPEG on a One-to-One Scale.....363

Figure B.47 Multimedia Bandwidth Allocation (KB/s) vs. Packet Loss (packets/s) during the Blast Measurement Period for HTTP-MPEG 364

Figure B.48 Bandwidth Allocation for Multimedia (KB/s) vs. Frame Rate (Frames/sec) during the Blast Measurement Period for HTTP-MPEG.....365

Figure B.50 Multimedia Bandwidth Allocation (KB/s) vs. Latency (ms) during the Blast Measurement Period with HTTP-MPEG.....366

Figure B.51 Multimedia Bandwidth Allocation (KB/s) vs. TCP Throughput (KB/s) during the Blast Measurement Period with HTTP-MPEG 368

Figure B.52 Multimedia Bandwidth Allocation (KB/s) vs. Selected Metrics during the Blast Measurement Period with BULK-MPEG 370

Figure B.53 Multimedia Bandwidth Allocations (KB/s) vs. Selected Metrics during the Blast Measurement Period forHTTP-Proshare 372

Figure B.54 Multimedia Bandwidth Allocations vs. Selected Metrics during the Blast Measurement Period with BULK-Proshare 374

portant performance factors. We explain these factors below and then explain why UDP is a better choice than TCP for interactive multimedia.

While reliable data transfers have long dominated the Internet, multimedia, such as audio and video, is one of the newest forms of traffic in the Internet. It can be widely used interactively, either to allow two or more individuals to converse or to allow a user at one end-system to interact with a remote environment. This remote interaction may be real, such as controlling some remote robotic device, or virtual, such as interacting with a virtual environment being generated at a remote location.

Interactive multimedia's primary characteristic is that the quality of the content is time-dependent. Data may arrive at the viewing station slightly later than the ideal time it should be played and still be useful. However, data that is delayed significantly may be useless. If the throughput is lower than intended the quality of the play-out may suffer but still be useful. However if the throughput falls below some minimum level the media stream may be useless.

The fundamental generic data units for multimedia are frames that are periodically generated. Depending on the specific media type, the receiver can tolerate some lost frames. In particular, loss may be preferable to the increased latency required to insure reliable delivery. Increased reliability usually comes at the cost of the increased latency to sense the loss and then retransmit the lost data. Because of concerns over smooth play-back as well as interactivity, significant latency may degrade the properties of an interactive multimedia interaction. We discuss each of these properties, throughput, loss, latency, and jitter, in greater detail below.

Consider how interactive multimedia works, in a perfect network. In the simplest case, a continuous stream of independent and discrete samples (frames) of analog audio or video are periodically recorded at the recording station, transmitted across the network to the viewing station, and then played back with the same periodicity. The periodicity at the recording station is the recording rate and the periodicity at the viewing station is the display rate. With a sufficiently high recording and display rate these consecutive frames give the illusion of continuity. However, sometimes the display rate may be insufficient to

Users expect to see the results of their input adjustments with no apparent delay, on the order of 100 ms [Schneiderman98]. In some cases, delays of 50ms are noticeable and delays of 80ms are irritating [Bhola98]. Interactive multimedia demands low latency for the interaction to be useful.

Even when dealing with non-interactive media, latency, particularly variance in latency, is a serious issue. We refer to the variability in latency as "jitter". For example, jitter results in irregular or random movement that appears in a video playback with variable latency. If the latency is variable, the inter-arrival time between frames will be variable. If frames are played back as soon as they arrive, the display of the frames will lose the illusion of continuity. In this context, jitter applies to all types of media, referring simply to a non-periodic playback rate of the frames. Jitter can be addressed by artificially adding delay to those packets that arrive quickly at the receiver by buffering those packets [Stone95]. This technique causes all packets to have the same, potentially large, delay but avoids variation in latency. Typically the buffer is configured to be sure all packets incur a delay on the order of the maximum end-to-end delay. Thus, it is useful to have an upper bound on the end-to-end latency in order to limit the amount of buffering and latency that must be allocated at the receiver. The variability in latency needs to have a reasonable bound in order for end-system jitter management techniques to be effective.

The protocol commonly used for interactive multimedia is UDP. UDP offers a datagram service that makes a best-effort attempt to deliver data to the destination. UDP is basically a transport-level wrapper for IP.

UDP appeals to designers of multimedia applications because is an extremely lightweight protocol. Because it simply guarantees the transmission and not the delivery of the packet, it offers a more stable transmission rate and less transport-level buffering. This lack of transport-level buffering eliminates one potential source of latency. Datagrams are never retransmitted. Transmission is never delayed because a previous packet is unacknowledged. Datagrams are only buffered if the network interface lacks capacity to transmit them, or, at the receiver, if the receiver is not ready to receive them. Since inter-

Congestion on a link occurs when the traffic arriving at a given link (i.e. load) exceeds the maximum capacity of that link. This congestion may occur because the outbound link's capacity is less than that of a single inbound link and all of the traffic arriving on that inbound link is destined for the lower capacity outbound link. It may also, more commonly, occur because packets arriving from multiple inbound links are all destined for the same outbound link and, in aggregate, those packets exceed the capacity of the outbound link. The congestion may be transient, resulting from the chance simultaneous arrival of packets on multiple inbound links or it may be persistent as the aggregate load consistently exceeds the outbound link's capacity.

Traditionally, there have been no widely used explicit indicators of congestion in the Internet. There was no widely used mechanism for providing feedback to end-systems to indicate that a particular link was congested. (Although the source quench option of ICMP was designed to allow routers to signal congestion, it has not been widely used in routers or end-systems. For more discussion of source quench, see Chapter III.) Instead, end-systems had to infer the presence of congestion by noting packet loss and assuming that loss was due to overflow in a congested router's queue. In order for loss to be an effective indication of congestion, either the transport or application-level protocol must detect loss by the absence of expected packets at the receiver, and report the loss to the sender. TCP must detect loss in order to implement reliability through retransmissions. It can use this same mechanism to infer congestion and respond accordingly.

There are proposals for explicit congestion indicators as well. Recent proposed extensions to TCP recommend taking advantage of a an explicit congestion notification (ECN) bit in the IP header which would indicate congestion may be imminent [Floyd94]. Those proposals recommend that routers set this ECN bit in the IP header of packets when persistently occupied queues indicate congestion. The receivers of messages with the ECN bit set would then be responsible for informing the senders that congestion may be imminent. Since this mechanism would indicate congestion before queues overflow, the senders would be able to adjust their transmission rate and actually avoid loss, rather than simply respond to it. This would reduce the need for retransmissions since the notification occurs before packet loss actually occurs. Unfortunately, ECN, like many Internet pro-