WPI-CS-TR-04-04 February 2004

Inferring Queue Sizes in Access Networks by Active Measurement

by

Mark Claypool
Robert Kinicki
Mingzhe Li
James Nichols and Huahui Wu

Computer Science
Technical Report
Series

WORCESTER POLYTECHNIC INSTITUTE

Computer Science Department
100 Institute Road, Worcester, Massachusetts 01609-2280

Inferring Queue Sizes in Access Networks by Active Measurement

Mark Claypool, Robert Kinicki, Mingzhe Li, James Nichols, and Huahui Wu
{claypool,rek,lmz,jnick,flashine}@cs.wpi.edu
CS Department at Worcester Polytechnic Institute
Worcester, MA, 01609, USA

No Institute Given

Abstract. Router queues can impact both round-trip time and throughput. Yet little is publicly known about the queue provisioning employed by Internet services providers for the routers that control the access links to home computers. This paper proposes QFind, a black-box measurement technique, as a simple method to approximate the size of the access queue used at last mile router. We evaluate QFind through simulation, emulation, and measurement. Although precise access queue results are limited by receiver window sizes and other system events, we find there are distinct difference between DSL and cable access queue sizes.

1 Introduction

The current conventional wisdom is that over-provisioning in core network routers has moved Internet performance bottlenecks to network access points [ASS03]. Since typical broadband access link capacities (hundreds of kilobytes per second) are considerably lower than ISP core router capacities (millions of kilobytes per second), last-mile access links need queues to accommodate traffic bursts. Given the bursty nature of Internet traffic [JD03] that is partially due to flows with high round-trip times or large congestion windows, it is clear that the provider's choice for access link queue size may have a direct impact on a flow's achievable bitrate. A small queue can keep achieved bitrates significantly below the available capacity, while a large access queue can negatively impact a flow's end-to-end delay. Interactive applications, such as IP telephony and some network games, with strict delay bounds in the range of hundreds of milliseconds experience degraded Quality of Service when large access queues become saturated with other, concurrent flows.

Despite the importance of queue size to achievable throughput and added delay, there is little documentation on queue size settings in practice. Guidelines for determining the "best" queue sizes have often been debated on the e2e mailing list,[1] an active forum for network related discussion by researchers and practitioners alike. While general consensus has the access queue size ranging from

[1] In particular, see the e2e list archives at: ftp://ftp.isi.edu/end2end/end2end-interest-1998.mail and http://www.postel.org/pipermail/end2end-interest/2003-January/-002702.html.

one to four times the capacity-delay product of the link, measured round-trip times vary by at least two orders of magnitude (10 ms to 1 second) [JID+04]. Thus, this research consensus provides little help for network practitioners to select the best size for the access queue link. Moreover, a lack of proper queue size information has ramifications for network simulations, the most common form of evaluation in the network research community, where access queue sizes are often chosen with no confidence that these queue choices accurately reflect current practices.

A primary goal of this investigation is to experimentally estimate the queue size of numerous access links, for both cable modem and DSL connections managed by a variety of Internet Service Providers. Network researchers should find these results useful in designing simulations that more accurately depict current practices. A secondary goal of this investigation is to determine, using both emulation and simulation, to what extent access link queue sizes can impact the throughput of flows with high round-trip times and the delays of flows for delay-sensitive applications, such as IP telephony and network games. Network practitioners should find this information useful to better accommodate the QoS requirements of increasingly diverse traffic as well as network researchers who can use this information to plan for the next generation networks.

2 QFind

Based on related work and pilot studies, the following assumptions are made in this study: each access link has a relatively small queue size - between 10 and 100 packets; the maximum queue length is independent of the access link capacity or other specific link characteristics; and the queue size is constant and independent of the incoming traffic load with no attempt made by the router to increase the queue sizes under heavier loads or when flows with large round-trip times are detected. Below is the proposed *QFind* methodology for inferring the access network queue size from an end-host:

1. Locate an Internet host that is slightly upstream of the access link while still being "close" to the end-host. For the test results discussed in this paper, the DNS name server provided by the ISP is used since DNS servers are typically close in terms of round-trip time and easy to find by inexperienced end-users.
2. Start a `ping` from the end-host to the close Internet host and let it run for up to a minute. The minimum value returned during this time is typically the baseline latency without any queuing delays since there is no competing traffic causing congestion. This `ping` process continues to run until the end of the experiment.
3. Download a large file from a remote server to the end-host. For the test results in this paper, a 5 MByte file was used since it typically provided adequate time for TCP to reach congestion avoidance and saturate the access queue downlink capacity.

4. Stop the `ping` process. Record the minimum and maximum round-trip times as reported by `ping` and the total time to download the large file. The maximum `ping` value recorded during the download typically represents the baseline latency plus the access link queuing delay.

The queue size of the access link can be inferred using the data obtained above. Let D_t be the total delay (the maximum delay seen by `ping`):

$$D_t = D_l + D_q \tag{1}$$

where D_l is the latency (the minimum delay seen by `ping`) and D_q is the queuing delay. Therefore:

$$D_q = D_t - D_l \tag{2}$$

Given throughput T (measured during the download), the access link queue size in bytes, q_b, can be computed by:

$$q_b = D_q \times T \tag{3}$$

For a packet size s (say 1500 bytes, a typical MTU), the queue size in packets, q_p, becomes:

$$q_p = \frac{(D_t - D_l) \times T}{s} \tag{4}$$

The strength of the QFind methodology lies in its simplicity. Unlike other approaches [ASS03,LP03], QFind does not require custom end-host software, making it easier to convince volunteers to participate in an Internet study. Moreover, the simple methodology makes the results reproducible from user to user and in both simulation and emulation environments.

2.1 Possible Sources of Error

The maximum `ping` time recorded may be due to congestion on a queue other than the access queue. However, this is unlikely since the typical path from the end-host to the DNS name server is short. Pilot tests [CKL+04a] suggest any congestion from the home node to the DNS name server typically causes less than 40 ms of added latency. Moreover, by having users repeat steps 2-4 of the QFind methodology multiple times (steps 2-4 take only a couple of minutes), apparent outliers can be discarded. This reduces the possibility of over-reporting queue sizes.

The queue size computed in Equation 4 may underestimate the actual queue size since it may happen that the `ping` packets always arrive to a nearly empty queue. However, if the file download is long enough, it is unlikely that every `ping` packet will be so lucky. Results in Section 3 suggest that the 5 MB file is of sufficient length to fill queues over a range of queue sizes.

If there is underutilization on the access link then the access queue will not build up and QFind may under-report the queue size. This can happen if there are sources of congestion at the home node network before ping packets even reach the ISP. Most notably, home users with wireless networks may have contention on the wireless medium between the `ping` and download packets. Pilot tests [CKL+04a] suggest that congestion on a wireless network during QFind tests adds at most 30 ms to any recorded `ping times`. As 30 ms may be significant in computing an access queue size, we ask QFind volunteers to indicate wireless/wired settings when reporting QFind results.

If the TCP download is limited by the receiver advertised window instead of by the network congestion window, then the queue sizes reported may be the limit imposed by TCP and not be the access link queue. However, recent versions of Microsoft Windows[2] as well as Linux[3] support TCP window scaling, allowing the receiver advertised window to grow up to 1 Gbyte [JBB92]. Even if window scaling is not used, it is still possible to detect when the receiver advertised window might limit the reported queue. The lack of `ping` packet losses during the download suggests that the access queue was not saturated and the queue size could actually be greater than reported.

For actual TCP receiver window settings, Windows 98 has a default of 8192 bytes[4], Windows 2000 has a default of 17520 bytes[5], Linux has a default of 65535 bytes [6], and Windows XP may have a window size of 17520, but it also has a mostly undocumented[7] ability to scale the receiver window size dynamically.

Additionally, some router interfaces may process `ping` packets differently than other data packets. However, in practice, hundreds of empirical measurements in [CCZ03] show `ping` packets usually provide round-trip time measurements that are effectively the same as those obtained by TCP.

3 Experiments

To determine whether the QFind methodology could effectively predict access link queue sizes in real last-mile Internet connections, we evaluated the QFind approach first with simulations using NS[8] (see Section 3.1) and then emulations using NIST Net[9] (see Section 3.2). After reviewing these proof-of-concept results, we enlisted many volunteers from the WPI community to run QFind experiments over a variety of DSL and cable modem configurations from home (see Section 3.3).

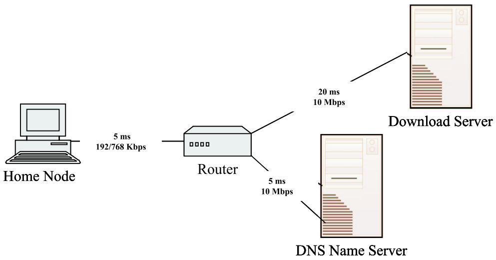

Fig. 1. Topology

3.1 Simulation

QFind was simulated with the configuration depicted in Figure 1 consisting of a home node, an ISP last-mile access router, a TCP download server and a DNS name server. The simulated link latencies used in the emulations were based on prototype QFind measurements.

The delays built into the testbed emulations were 5 ms from home to router, 5 ms from router to DNS, and 20 ms from router to download server. Link capacities were set to reflect typical asymmetric broadband data rates [LP03], with the router-to-home downstream link capacity set at 768 Kbps, the home-to-router upstream link capacity set to 192 Kbps, and the link capacities in both directions between router and both upstream servers fixed at 10 Mbps. 1500 byte packets were used to model the typical Ethernet frame size found in home LANs and TCP receiver windows were set to 150 packets.

Figure 2 displays the cumulative density functions for 100 simulations of the QFind methodology (steps 2 to 4 in Section 2) with downstream access link queues of 10, 50 and 100 packets respectively. QFind predicts the access queue size remarkably well in this simulated environment. Of the 100 runs at each queue size, the *most* the predicted queue size was smaller than the actual queue size was 1 packet for the 10 packet queue, 1.5 packets for the 50 packet queue and 2.5 packets for the 100 packet queue. The median predicted queue size was less than the actual queue size by about 1 packet in all cases.

---

[2] The default in Windows 2000 and higher (see [Mic03]).
[3] The default in Linux kernel versions 2.2 and above.
[4] http://www.dslreports.com/tweaks/RWIN#howlarge
[5] http://rdweb.cns.vt.edu/public/notes/win2k-tcpip.htm
[6] See Documentation/networking/ip-sysctl.txt under a Linux source tree v2.4+.
[7] http://support.microsoft.com/default.aspx?scid=kb;en-us;Q314053
[8] http://www.isi.edu/nsnam/ns/
[9] http://snad.ncsl.nist.gov/itg/nistnet/

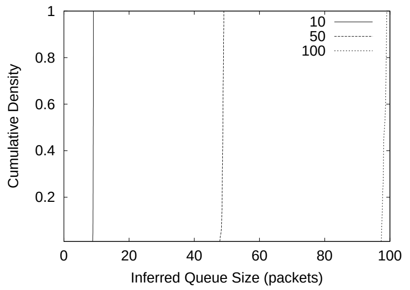

Fig. 2. Cumulative Density Functions of Inferred Queue Sizes for Actual Queue Sizes of 10, 50 and 100 Packets using NS Simulator

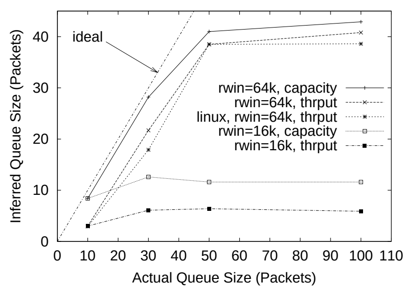

Fig. 3. Median of Inferred Queue Sizes versus Actual Queue Sizes using NIST Net Emulator

3.2 Emulation

To further investigate QFind feasibility, we setup a testbed to emulate a last-mile access router in a controlled LAN environment. Two computers were used as home nodes with one computer running Windows 2000 and the other running Linux in order to test the impact of the operating system type on QFind. The download server ran on Windows Server 2003 while the DNS name server ran on Linux. A NIST Net PC router emulated the ISP's Internet connection with link capacities set to reflect typical broadband asymmetry with the downstream router-to-home link capacity set to 768 Kbps, the upstream home-to-router link set to 192 Kbps, and the router link capacities to and from both servers using 10 Mbps LAN connections. The home-to-server round-trip delay was 20 ms for both the download server and the DNS server since the NIST Net implementation does not allow two host pairs to have different induced delays while sharing a router queue.

Using this testbed, the QFind methodology was emulated (steps 2 to 4 in Section 2) with home nodes running Windows 2000 with a TCP receiver window size of 16 Kbytes, Windows 2000 with a TCP receiver window sizes set to 64 Kbytes, and Linux with a TCP receiver window sizes set to 64 Kbytes. Three QFind emulations were run for each of the queue sizes of 10, 30, 50 and 100 packets, with a packet size of 1500 bytes.

Figure 3 presents the median of the inferred queue sizes. The inferred queue sizes labeled "thrput" are computed using the measured download capacity. The inferred queue sizes labeled "capacity" are computing using the capacity of the link. In those cases where the NIST Net queue size is smaller than the TCP receiver window size, QFind is able to infer the queue size closely, even for different operating systems. The queue sizes computed using link capacity are more accurate than those computed using download throughput. However, while the link capacity was, of course, known by us for our testbed, it is not, in general, known by an end-host operating systems nor by most of the home users who participated in our study.

Intermediate results that can be drawn from these emulations even before evaluating actual QFind measurements include: the QFind emulation estimates of queue size are not as accurate as the simulation estimates; using the maximum link capacity provides a better estimate of the access queue size than using the measured download data rate; ping outliers in the testbed did not cause over prediction of the queue length; small TCP receiver windows result in significant underestimation of the access queue size since the ability of the download to fill the access queue is restricted by a small maximum TCP receiver window size setting.

3.3 Measurement

The final stage of this investigation involved putting together an easy-to-follow set of directions to be used by volunteers to execute three QFind experiments and record results such they could be easily emailed to a centralized repository. One of the key elements of the whole QFind concept was to develop a measurement procedure that could be run by a variety of volunteers using different cable and DSL providers on home computers with different speeds and operating systems. To maximize participation, the intent was to avoid having users download and run custom programs and avoid any changes to system configuration settings (such as packet size or receiver window). The final set of instructions arrived upon can be found at found at: http://www.cs.wpi.edu/~claypool/qfind/-instructions.html.

During January 2004, we received QFind experimental results[10] from 47 Qfind volunteers, primarily from within the WPI CS community of undergraduate students, graduate students and faculty. These users had 16 different ISPs: Charter (16 users), Verizon (11), Comcast (4), Speakeasy (4), Earthlink (2), AOL (1), Winternet (1), RR (1), RCN (1), NetAccess (1), MTS (1), Cyberonic (1), Cox (1), Covad (1) and Adelphia (1). The QFind home nodes had 5 different operating systems: WinXP (18 users), Win2k (11), Linux (6), Mac OS-X (3), and Win98 (1) and 12 unreported. Approximately one-third of the volunteers had a wireless LAN connecting their home node to their broadband ISP.

About 45% of the volunteers used DSL and 55% used cable modems. Figure 4 presents CDFs the throughput for all QFind tests, with separate CDFs for cable and DSL. The CDF for cable modems show a sharp increase corresponding to a standard 768 Kbps downlink capacity, which is also the median capacity. Above this median, however, the distributions separate with cable getting substantially higher throughput than DSL.

Figure 5 depicts CDFs of the maximum `ping` times reported for all QFind tests, with separate CDFs for cable and DSL. The median max `ping` time is about 200 ms, but is significantly higher for DSL (350 ms) than for cable (175

[10] The QFind data we collected can be downloaded from http://perform.wpi.edu/-downloads/#qfind

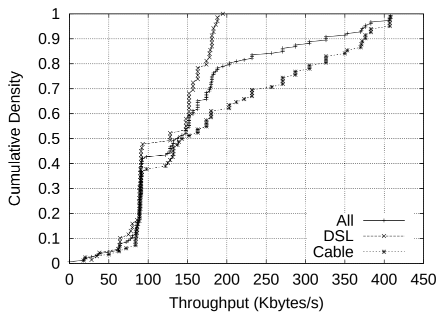

Fig. 4. Cumulative Density Functions of Throughput

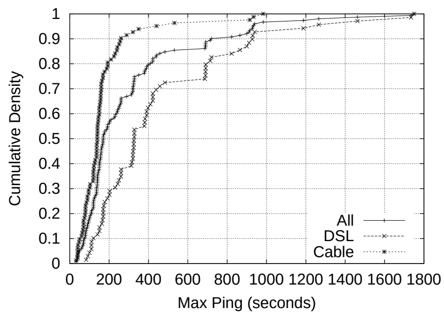

Fig. 5. Cumulative Density Functions of Max Ping Times

ms). Ping times of 350 ms are significant since this is enough delay to affect interactive applications [DCJ93,Hen01]. In fact, the entire body of the DSL CDF is to the right of the cable CDF, indicating a significant difference in the max ping times for DSL versus cable. Also, the maximum ping times for cable can be up to a second and can be well over a second for DSL, a detriment to any kind of real-time interaction.

In analyzing the full data set to infer queue sizes (see the analysis in [CKL+04a]), it appeared QFind may not clearly distinguish delays from the access queue from other system delays. We noted considerable variance in inferred access queue sizes even for volunteers within the same provider, an unlikely occurrence given that ISP providers tend to standardize their equipment at the edge by home users. This suggests that for some experiment runs, the `ping` delays that QFind uses to infer the queue size are a result of something other than delay at the access queue.

To remove data that does not accurately report access queue sizes, we winnow the full data set by taking advantage of the fact that the QFind volunteers produced three measurements. For each user's three measurements, if any pair have throughputs that differ by more than 10% or maximum `ping` times that differ by more than 10%, then all three measurements are removed. This winnowing removed the data from 17 users. All subsequent analysis is based on this winnowed data set.

Figure 6 depicts a CDF of the access queue sizes measured by QFind, with separate CDFs for DSL and cable. There is a marked difference between the DSL and cable inferred queues, with cable having queue sizes under 20 Kbytes while DSL queues are generally larger. The steep increase in the DSL queue sizes around 60 Kbytes is near the limit of the receiver window size of most OSes (64 Kbytes), so the actual queue limits may be higher.

Figure 7 depicts CDFs for the access queue sizes for Charter,[11] the primary cable provider in our data set, and non-Charter cable customers. There are some

[11] http://www.charter.com/

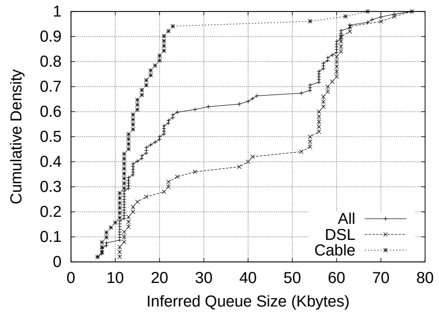

Fig. 6. Cumulative Density Functions of Access Queue Size Inferred by QFind

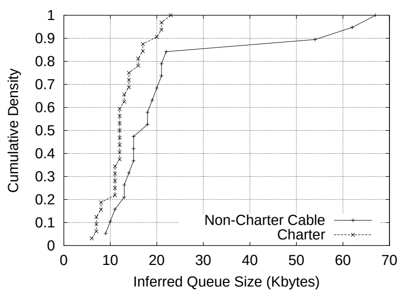

Fig. 7. Charter-Non-Charter: Cumulative Density Functions of Access Queue Size Inferred by QFind

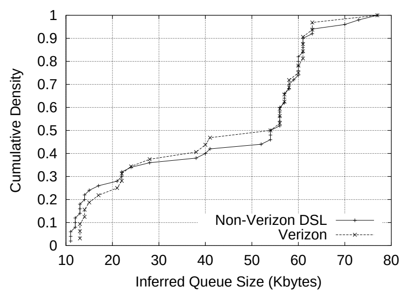

Fig. 8. Verizon-Non-Verizon: Cumulative Density Functions of Access Queue Size Inferred by QFind

marked differences in the distributions, with Charter cable queues appearing to be slightly smaller than non-Charter cable queues. The extremely large non-Charter cable queue reported above 0.8 is from one cable provider.

Figure 8 depicts similar CDFs for the access queue sizes for Verizon,[12] the primary DSL provider in our data set, and non-Verizon DSL customers. Here, there are very few differences between the different DSL provider distributions, suggesting there may be common queue size settings across providers.

4 The Impact of Access Queue Size on Performance

The apparent differences between access queue sizes for DSL and for cable and even for different cable providers brings forth the question what size *should* access queues be? Instead of debating the merits of particular queue sizes as has been done in discussion forums (see Section 1), this Section briefly explores the impact of access queue size on throughput and round-trip times through simulation.

The simulations used the topology depicted in Figure 1, with the exception that the router-server link delays were varied from 50-800 ms, a typical range of round-trip times on the Internet [JID+04]. We ran simulations for five different queue sizes: 5, 10, 20, 50 and 100 with the varying link latencies. The receiver window size was set to 200. We computed throughput as the size of the simulated file divided by the download time and round-trip time as the average of the `ping` times during the download period.

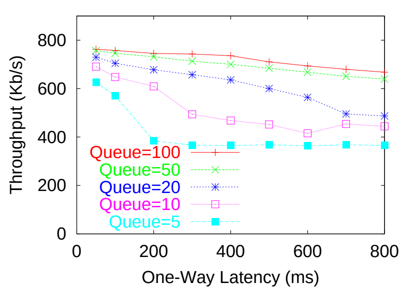

Fig. 9. Throughput versus Queue Size

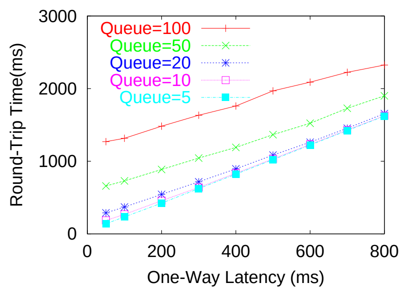

Fig. 10. Round-trip time versus Queue Size

Figure 9 depicts the throughput for each queue size. The x-axis is the one-way link latency from router to download server, and the y-axis is the throughput of the router-to-home downstream link. The five curves represent results with queue sizes of 100, 50, 20, 10 and 5 from top to bottom. The curves clearly depicts that larger queue sizes have higher throughput, even for very low latencies, but especially when link latency are high. Notice, however, that the

[12] http://www.verizon.com/

curves for Queue=100 and Queue=50 very close, suggesting decreasing returns on throughput for larger queues.

Figure 10 depicts the round-trip time for each queue size. The x-axis is the one-way link latency from router to download server, and the y-axis is the round-trip time between home node and download server. The five curves represent results with queue sizes 100, 50, 20, 10 and 5 from top to bottom. The curves clearly depict higher round-trip times for higher latencies but also higher round-trip times for larger queue sizes. Even when latencies are low (100 ms or under), a large access queue size can cause latencies that seriously degrade real-time interactive applications. Although it is worth noting that the curves for queue sizes of 5, 10 and 20 are all very close, suggesting decreasing returns on round-trip time for smaller queues.

Overall, based on these brief simulations, it appears an access queue size around 20 packets provides reasonable throughput without severely impacting round-trip times when downloading. Alternatively, the simulations also reinforce the need for Quality of Service which could allow real-time interactive applications low delays while providing high throughput for other applications.

5 Summary

The QFind methodology for inferring queue sizes is attractive in several ways: 1) by using a standard `ping` and a download through a Web browser, QFind does not require any custom software or special end-host configuration; 2) by using a single TCP flow, QFind so does not cause excessive congestion. This provides the potential for QFind to be used to measure access queues from a wide-range of volunteers.

Simulation and emulation results show that QFind can be effective at inferring queue sizes, even across multiple operating systems, as long as receiver window sizes are large enough and access queues are not so small as to limit throughput.

Unfortunately, measurement results suggest QFind is substantially less accurate than in simulation for determining access queue sizes. By doing multiple QFind experiments, it is possible to ensure analysis on only consistent results, but this results in the discarding of many data samples, thus somewhat defeating the purpose of having a readily available, non-intrusive methodology.

Based on the winnowed data set from our 47 QFind volunteers, DSL appears to have significantly smaller access queues than does cable, and the corresponding `ping` delays when such a queue is full can significantly degrade interactive applications with real-time constraints.

Future work could include exploring technologies that have been used for bandwidth estimation (a survey of such technologies is in [PMDC03]). In particular, techniques such as [RRB$^+$03] that detect congestion by a filling router queue may be used to determine maximum queue sizes. The drawback of such techniques is they require custom software and may be intrusive, so these drawbacks would need to be weighed against the benefit of possibly more accurate results.

References

[ASS03] Aditya Akella, Srinivasan Seshan, and Anees Shaikh. An Empirical Evaluation of Wide-Area Internet Bottlenecks. In *Proceedings of the ACM Internet Measurement Conference (IMC)*, October 2003.

[CCZ03] Jae Chung, Mark Claypool, and Yali Zhu. Measurement of the Congestion Responsiveness of RealPlayer Streaming Video Over UDP. In *Proceedings of the Packet Video Workshop (PV)*, April 2003.

[CKL+04a] Mark Claypool, Robert Kinicki, Mingzhe Li, James Nichols, and Huahui Wu. Inferring Queue Sizes in Access Networks by Active Measurement. Technical Report WPI-CS-TR-04-04, CS Department, Worcester Polytechnic Institute, February 2004.

[CKL+04b] Mark Claypool, Robert Kinicki, Mingzhe Li, James Nichols, and Huahui Wu. Inferring Queue Sizes in Access Networks by Active Measurement. In *Proceedings of the 5th Passive and Active Measurement Workshop (PAM)*, April 2004.

[DCJ93] Spiros Dimolitsas, Franklin L. Corcoran, and John G. Phipps Jr. Impact of Transmission Delay on ISDN Videotelephony. In *Proceedings of Globecom '93 – IEEE Telecommunications Conference*, pages 376 – 379, Houston, TX, November 1993.

[Hen01] Tristan Henderson. Latency and User Behaviour on a Multiplayer Game Server. In *Proceedings of the Third International COST Workshop (NGC 2001)*, number 2233 in LNCS, pages 1–13, London, UK, November 2001. Springer-Verlag.

[JBB92] V. Jacobson, R. Braden, and D. Borman. TCP Extensions for High Performance. *IETF Request for Comments (RFC) 1323*, May 1992.

[JD03] H. Jiang and C. Dovrolis. Source-Level IP Packet Bursts: Causes and Effects. In *Proceedings of the ACM Internet Measurement Conference (IMC)*, October 2003.

[JID+04] Sharad Jaiswal, Gianluca Iannaccone, Christophe Diot, Jim Kurose, and Don Towsley. Inferring TCP Connection Characteristics Through Passive Measurements. In *Proceedings of IEEE Infocom*, March 2004.

[LP03] Karthik Lakshminarayanan and Venkata Padmanabhan. Some Findings on the Network Performance of Broadband Hosts. In *Proceedings of the ACM Internet Measurement Conference (IMC)*, October 2003.

[Mic03] Microsoft. Description of Windows 2000 TCP Features, September 2003. Online at: http://support.microsoft.com/default.aspx?scid=kb;EN-US;224829.

[PMDC03] R.S. Prasad, M. Murray, C. Dovrolis, and K.C. Claffy. Bandwidth Estimation: Metrics, Measurement Techniques, and Tools. *IEEE Network*, November-December 2003.

[RRB+03] Vinay Ribeiro, Rudolf Riedi, Richard Baraniuk, Jiri Navratil, and Les Cottrell. pathChirp: Efficient Available Bandwidth Estimation for Network Paths. In *Proceedings of the 4th Passive and Active Measurement Workshop (PAM)*, 2003.

Appendix

Supplementary figures that did not fit in the PAM'03 paper [CKL+04b].

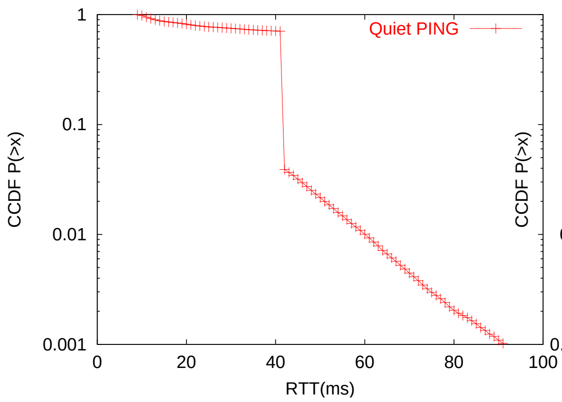

Fig. 11. Complementary Cumulative Density Function of Ping Times. Home node and home network was quiet. Ping times were two days straight, Friday morning through Saturday morning.

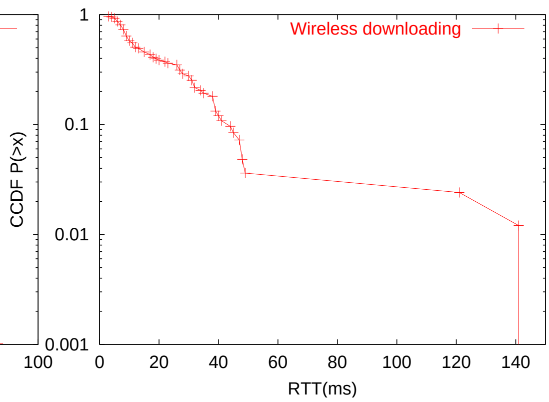

Fig. 12. Complementary Cumulative Density Function of Ping Times. Home node was wireless. Ping was during download but ping was only to gateway (did not travel access queue).

NISTNet emulation (see Section 3.2) setup:

```
Setup:
1. Nistnet:
  csta04.WPI.EDU 0.0.0.0 --delay 10 (ms) --bandwidth 24000 (Bps)
  0.0.0.0 csta04.WPI.EDU --delay 10 (ms) --bandwidth 96000 (Bps) --drd 0 (10-100)P
  packet size = 1500 bytes

2. Clients:
  OS = Windows 2000 service pack 4
  CPU = Intel Celeron 1.2GHz
  RAM = 256MB
  Receive Win1 = 17520 Bytes
  Receive Win2 = 65535 Bytes

  OS = Linux merlot 2.4.20-8
  CPU = Pentium MMX 233
  RAM = 128MB
  Receive Win = 65535 Bytes

3. Software Tools:
  Dr. TCP - to modify the RWIN for Windows 2000.
```

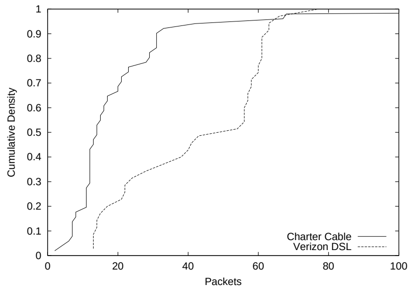

Fig. 13. Cumulative Density Functions of Access Queue Sizes Inferred by QFind. All data collected from Verizon and Comcast users is shown.

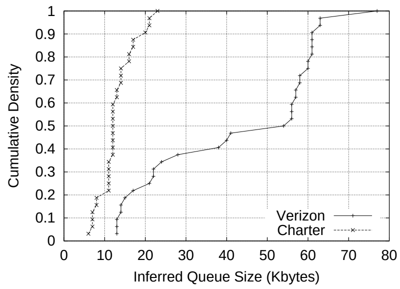

Fig. 14. Cumulative Density Functions of Access Queue Sizes Measured with QFind. The above data has removed 17 user sets, where max `ping` differences or throughput differences were 10+%.

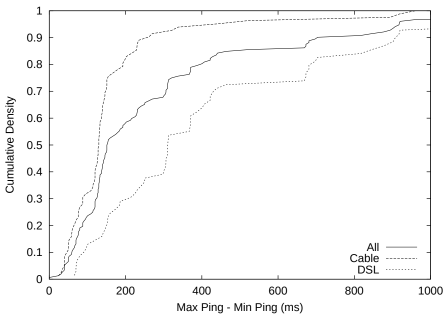

Fig. 15. Cumulative Density Functions for Difference in Maximum Ping and Mininimum Ping Values. The difference for all DSL and cable modems is shown.

```
http://www.dslreports.com/drtcp
```

NISTNet emulation (see Section 3.2) data:

```
Q_bytes(bw) and Q_pack(bw) are computed using download speed.
Q_bytes(cp) Q_pack(cp) are computed using capacity (768kbps).
```

Linux

Qset	Max_rtt	Min_rtt	dl_bw	Q_bytes(bw)	Q_pack(bw)	Q_bytes(cp)	Q_pack(cp)
100	694	21	89460	60206.58	40.13772	64608	43.072
100	695	21	84140	56710.36	37.80690667	64704	43.136
100	673	21	88780	57884.56	38.58970667	62592	41.728
50	688	21	86460	57668.82	38.44588	64032	42.688
50	650	21	89460	56270.34	37.51356	60384	40.256
50	684	21	89460	59311.98	39.54132	63648	42.432
30	448	21	62460	26670.42	17.78028	40992	27.328
30	442	21	63310	26653.51	17.76900667	40416	26.944
30	431	21	66250	27162.5	18.10833333	39360	26.24
10	151	21	33230	4319.9	2.879933333	12480	8.32
10	146	21	32050	4006.25	2.670833333	12000	8
10	147	21	35160	4430.16	2.95344	12096	8.064

Win2K RWIN=16K (17520 Bytes, default)

Qset	Max_rtt	Min_rtt	dl_bw	Q_bytes(bw)	Q_pack(bw)	Q_bytes(cp)	Q_pack(cp)
100	201	20	49127	8891.987	5.927991333	17376	11.584
100	201	20	50423	9126.563	6.084375333	17376	11.584
100	201	20	48760	8825.56	5.883706667	17376	11.584
50	205	20	51701	9564.685	6.376456667	17760	11.84
50	201	20	52708	9540.148	6.360098667	17376	11.584
50	200	20	51438	9258.84	6.17256	17280	11.52
30	231	20	44989	9492.679	6.328452667	20256	13.504
30	210	20	46712	8875.28	5.916853333	18240	12.16
30	201	20	50354	9114.074	6.076049333	17376	11.584
10	160	20	34320	4804.8	3.2032	13440	8.96
10	151	20	34737	4550.547	3.033698	12576	8.384
10	151	20	34730	4549.63	3.033086667	12576	8.384

Win2K RWIN=64K

Qset	Max_rtt	Min_rtt	dl_bw	Q_bytes(bw)	Q_pack(bw)	Q_bytes(cp)	Q_pack(cp)
100	701	20	89685	61075.485	40.71699	65376	43.584
100	691	20	91671	61511.241	41.007494	64416	42.944
100	681	20	92641	61235.701	40.82380067	63456	42.304
50	661	20	90084	57743.844	38.495896	61536	41.024
50	661	20	89924	57641.284	38.42752267	61536	41.024
50	671	20	90957	59213.007	39.475338	62496	41.664
30	460	20	73884	32508.96	21.67264	42240	28.16
30	460	20	74167	32633.48	21.75565333	42240	28.16
30	450	20	74579	32068.97	21.37931333	41280	27.52
10	160	20	35832	5016.48	3.34432	13440	8.96
10	150	20	32346	4204.98	2.80332	12480	8.32

10	151	20	34480	4516.88	3.011253333	12576	8.384