A Preliminary Analysis of TCP Performance in an Enterprise Network

Boris Nechaev[†], Mark Allman[‡], Vern Paxson[‡], Andrei Gurtov[†]
[†]*Helsinki Institute for Information Technology,* [‡]*International Computer Science Institute*

Abstract

Although TCP behavior is one of the most studied aspects of Internet traffic, little is known about TCP performance within modern enterprise networks. In this paper we analyze aspects of TCP performance observed in packet traces taken over four months from a medium-sized enterprise. We assess the prevalence of broken TCP transactions, applications used, throughput of TCP connections, and phenomena that influence performance, such as retransmissions, out-of-order delivery, and packet corruption. While much remains to explore, this work represents a first step towards understanding TCP performance in the under-studied environment.

1 Introduction

Researchers have studied TCP performance in many ways and in many different environments over the years [7, 4, 10, 9]. However, one area that has remained understudied is within complex enterprise networks. We believe two key reasons account for the lack of attention paid to enterprise networks. First, passively monitoring enterprise traffic is logistically difficult. Whereas monitoring wide-area traffic between a particular institution and the rest of the Internet can be readily accomplished by instrumenting at most a handful of vantage points, enterprise traffic often does not traverse central monitoring locations. Therefore, assessing enterprise networks involves taking measurements from a large number of vantage points—either from switches, as we do, or from end hosts themselves [2]—and then synthesizing a more complete picture from these. The second issue with enterprise networks is that they are often viewed as working "well enough". While complex, these networks do not have the same sort of cross-organizational issues that wide-area networks must contend with. In addition, low latencies and abundant network capacity can often mask problems with protocols and algorithms.

While enterprise networks may work "well enough," they are largely a black box at present: we have little idea whether the performance they deliver in fact matches our mental models of low-latency, loss-free, high-bandwidth pipes. Therefore, while logistically daunting, we argue that developing an understanding of performance across such networks holds the potential to drive innovation in terms of better-functioning network technologies.

This paper represents an initial step towards grappling with assessing performance issues within enterprise networks. We base our analysis on a dataset consisting of switch-level packet traces taken at the Lawrence Berkeley National Laboratory (LBNL) over the course of four months. Our study builds on our extensive previous efforts to calibrate the traces [5].

2 Data

The dataset used in this study comes from monitoring individual switch ports inside the Lawrence Berkeley National Laboratory (LBNL) from October 24 2005 through March 7 2006. Each of our 50 traces contains all traffic seen on each of ten switch ports mirrored simultaneously. Our goal was to monitor each set of switch ports for 24 hours and then move our monitoring apparatus to a different set of ports (a manual process). However, for logistical reasons we did not always achieve this ideal, and while our dataset includes 44 traces that cover more than 20 hours, we also have shorter traces. In total, the traces directly monitored 351 distinct hosts out of an estimated 8,000–10,000 wired hosts at the institute during that time. See [5] for a more detailed description of our data collection methodology, and for a discussion of the calibration techniques we applied to the data in this paper to remove measurement artifacts.

The overall dataset consists of 509 million TCP packets. As our interest in this work concerns intra-enterprise TCP traffic, we winnow the traces to the 292 million TCP packets that did not leave LBNL's network. We

analyze these packet traces using Bro [8] 1.5.1 to generate connection logs for each trace (including using Bro's `analy.bro` analysis script and the corresponding `TCPStats` module). The dataset contains 532K TCP connections.

Note, we disabled Bro's standard TCP inactivity timeout. We found that it led to multiple connections with the same addresses and ports being counted as different connections because of packets coming widely apart. These appear to reflect switch flooding, whereby the switch does not know to which port to direct a packet, and therefore transmits it on each port before (re-)learning which port it should be directed towards. (Note, these packets likely represent legitimate connections that are not within the general purview of our monitoring system.)

3 Connection Status

We first analyze each connection in our dataset for its "final state" as computed by Bro, which represents how each connection was instantiated and terminated. For instance, the "SF" state indicates a connection that was observed to have been established using TCP's standard three-way SYN handshake and terminated with a FIN handshake, whereas "REJ" indicates a SYN was used to start a connection, but the connection was immediately RST (thus rejected) by the responder. We observe 13 final states in our dataset in various frequencies across trace files. Figure 1 shows the relative frequencies of the states we observed for each trace in the dataset (sorted by percentage of SF connections). We include each state that comprises at least 1% of the overall connections in the dataset.

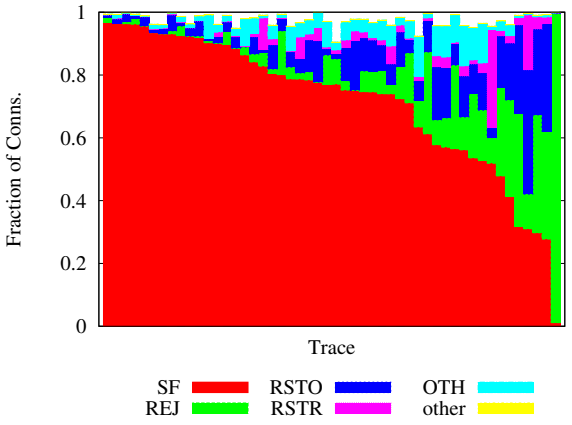

Figure 1: Distribution of final connection states.

In total, SF connections represent 58% of the connections in the dataset—however, this proportion varies greatly across different traces. Coupled with the RSTO and RSTR connections—which are properly established, but later closed with RSTs instead of FINs, sent by the originator or responder, respectively—these connections represent the 363K "good" connections we use in the analysis presented in the subsequent sections. In total these connections transferred about 50 GB of data.

Figure 1 shows that a large fraction of connections in each trace was rejected. Across the entire dataset we identify 158K (30%) rejected connections. However, we find that 129K of these originate from a single host, which scans nearly every port on two particular machines. This trace is on the far right of Figure 1, for which 98% of the connections are in the REJ state. The originator of these scans is a machine used by the network operators, and these scans reflect normal operations.

Additionally, Figure 1 shows that "OTH" connections also happen frequently. This state indicates that Bro did not observe the connection establishment nor teardown. As discussed above, these connections likely represent legitimate traffic that is mostly out of view of our monitoring point (i.e., we see a small bit of flooded traffic and nothing else). Across the dataset we observe 5.5K OTH connections. While 66% of these involve a single host (sending to port 80 in 99.5% of the cases), the remainder come from 275 distinct hosts, destined to 182 hosts. Over 90% of these connections contain only a single ACK or data packet.

We note that [3] uses the fraction of successfully established connections as its measure of "network health" and further finds the overall health of the network monitored to be roughly 66%. While our overall dataset yields a similar percentage of good connections (68%) when removing the port scan traffic—which we stress is a legitimate operational tool in this case—we find that just over 90% of the connections we observe are good. In addition, as shown in the figure individual traces (i.e., collections of hosts) contain a wide range of *variability* in the percentage of good connections.

4 Connection Characteristics

In this section we examine some basic characteristics of the connections in our dataset. We first note that we observe 202K connections (56%) involve a host outside the monitored subnet, while 161K (44%) involve a peer inside the subnet. (Again, this figure has high variability across traces, which is examined further in [5].) This shows that monitoring methodologies that focus on core routers in enterprise networks—such as used in a previous study of LBNL traffic [6]—miss a large fraction of the traffic. Therefore, switch measurements or end host measurements [2] are preferred to better understand the overall traffic characteristics.

Application	Bytes (%)	Conns. (%)	Scope
HTTP	9.10	18.5	I,O
NetBIOS-SSN	1.5	10.0	I,O
SMB	0.1	3.5	I,O
SIMAP	0.5	2.2	O
EPMapper	0.0	6.7	O
NFS	16.7	2.3	I
ssh	7.0	0.2	I,O
other-9100	1.2	0.1	I
printer	0.3	3.3	O
LDAP	0.0	1.1	O
Portmap	0.0	9.0	I,O
Dantz	26.7	0.3	I,O
other-9409	9.1	0.0	I
other-9406	8.2	0.0	I
other-9407	6.5	0.0	I
other-3365	2.3	0.0	O
MySQL	2.1	1.3	I,O
Warewulf Monitor	2.0	18.8	I,O
PostgreSQL	1.6	0.1	I
Braille prot.	1.4	0.0	O
other-1024	0.0	3.0	I
MacOS Serv. Admin	0.0	2.6	I
iSCSI	0.0	4.7	I,O

Table 1: Major applications observed.

Table 1 shows the prevalent applications in our dataset, each representing at least 1% of either the bytes or the connections across our dataset. The applications are broken into three groups. The top group contains applications seen in at least 80% of the traces in our dataset, while the applications in the bottom group are observed in at most 20% of the traces. The middle group of applications are represented in between 20% and 80% of the traces. The final column of the table indicates the scope of the traffic with "I" indicating the application meets the 1% byte or connection threshold when only considering intranet traffic and "O" indicating the threshold was met by traffic traversing subnet boundaries.

While the table shows a number of well-known and expected applications it also shows less well-known (and even unknown!) applications are not rare. In addition, we see that most applications are "unbalanced" in that they contribute a significant fraction of connections or bytes but not both. For instance, the Dantz backup system and NFS contribute a relatively large fraction of bytes (26.7% and 16.7%), but only a modest fraction of the connections (0.3% and 2.3%). On the other hand, NetBIOS-SSN and Warewulf Cluster monitoring show only small fractions of the bytes observed (1.5% and 2.0%), but larger fractions of connections (10.0% and 18.8%).

As a final piece of background we turn our attention to connection sizes. First, we note that we remove nearly

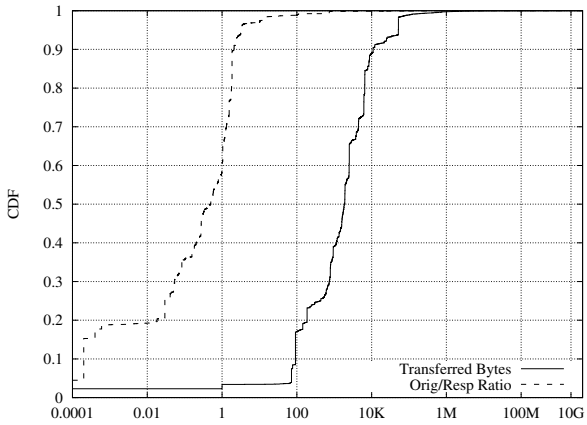

Figure 2: Distribution of connection sizes and ratio of originator data bytes to responder data bytes.

13K connections from this analysis due to a lack of a byte count in at least one of the directions.[1] Figure 2 shows the distribution of the total number of bytes (in both directions) transferred across each connection in our dataset. The median transfer size was just under 2 KB, while we observed a total of 4 connections that transferred a bit more than 5 GB. We find that in 78K (22.3%) connections, the connection originator sends no data bytes, while the responder does not send data in 10K (2.9%) connections. No data is exchanged in either direction in 8K (2.3%) connections. These are all "SF" connections (established and torn down correctly) and largely (92%) consist of "printer" connections. We find these printer connections to be between two pairs of IP addresses. The connections happen every 90 seconds in the first pair (36% of the cases). In the second pair we observe a similar 90 second periodicity, however in this case *two* connections are established—neither of which transfers any data—every 90 seconds.

Figure 2 also shows the distribution of the ratio of the data sent by the originator to the data sent by the responder to provide a sense of directionality.[2] The figure shows that nearly 60% of the connections consist of the responder sending more data than the originator (up to 1.2 million times as much data). Similarly in over 30% of the connections we find more bytes from the originator than the responder (up to 1.2 billion times as much data).

We also note that the top 15 flows (by volume) in our dataset (or 0.004% of the flows) account for 57% of the aggregate number of bytes transferred. This is on the low end of the findings in [9], which found that 50–60% of the bytes came from 0.006–0.09% of the connections (across several datasets, all but one of which had an order of magnitude larger percentages than what we ob-

serve). We further note that 90% of the traffic in our dataset comes from a mere 160 connections.

We conclude this section by considering the sizes of transactions across different application protocols. Table 2 provides various statistics for connections involving some of the more prevalent applications in our dataset. The data shows a range of application behavior. In general we see heavy tails with medians of only a small fraction of the 99th percentiles and maximums. However, while the heavy tails are present, the magnitude of the transfers varies across protocols. In addition, there are exceptions such as the port mapper and the Warewulf Cluster monitoring which do not show heavy tails.

App.	Med.	99th	Max
HTTP	1.9 KB	82 KB	835 MB
Dantz	6.4 KB	233 MB	4 GB
NFS	72 B	1.0 MB	1.1 GB
NetBIOS-SSN	2.0 KB	59 KB	137 MB
Warewulf	6.6 KB	52 KB	52 KB
Portmap	92 B	716 B	1.1 KB
ssh	5.5 KB	19 MB	2.6 GB

Table 2: Sizes of various prevalent applications.

5 Performance

We next examine different facets of performance observed within the enterprise. First, regarding basic network errors, we found that 583 TCP packets contained checksum errors. These were confined to eight traces, with three of the traces accounting for nearly 90% of the failures (likely indicating isolated hardware flakiness). Given their low rate, we do not believe these problems affect the insights obtained from the dataset.

We assessed out-of-order packet delivery by inspecting IP ID changes in TCP data packets. Bro tracks whether a given sender increments IP ID in a generally monotone fashion (either little-endian or big-endian), and flags a reordered packet if increments are generally monotone but a packet occurs in a trace with an IP ID slightly smaller than its predecessor in the trace. Using this definition, we observe reordering in 484 connections (0.1%), out of which 16 connections have reordered packets in both directions. Overall, only 0.0025% of data packets arrived out of order, a negligible level compared to the analogous figures reported in [7] and other wide-area studies. However, within a uni-directional flow that experienced reordering, the share of packets arriving out of order ranges from 0.003% to 100% (i.e., every packet that had an opportunity to arrive out of order indeed arrived out of order), with a median of 4.9% and 95th percentile of 40%. However, we note that a thorough assessment of packet reordering necessarily entails taking into account the *interval* between when the packets were originally sent; we will generally see lower reordering rates or larger intervals. Such an analysis is beyond the scope of our present initial study.

We also looked for *replicated* packets (sent once but arriving multiple times), as were (very rarely) noted in [7]. We did not observe any. However, we note that any replicated packets that arrived within 5 msec of each other would have been considered *phantoms* and removed from the traces during our calibration phase, as described in [5].

Another important aspect of TCP behavior is retransmissions. Assessing these robustly requires a measurement vantage point close to data sender, because the further away we monitor from the sender, the higher the risk we will miss retransmitted packets that get lost en route between the sender and the monitor's vantage point. For this reason, we consider only inter-subnet connections, for which given our knowledge of netmasks in each trace we can accurately identify which of the two hosts is in the monitored subnet (and thus for which we have a sender-side vantage point), and calculate the number of retransmitted packets sent by the local senders.

We find that 1,089 inter-subnet connections (0.5%) experienced retransmissions. A majority of these connections sent only one (724 conns.), two (159 conns.), or three (75 conns.) retransmissions. We also note that the maximum number of retransmissions we observed for a connection was 139, and the total number of retransmitted packets across all inter-subnet connections was 2,736.

Next we study the *maximum flight sizes* attained by uni-directional flows in the enterprise. We define this value as the maximum observed difference between a sender's highest outstanding sequence number and the cumulative acknowledgment point. (For this analysis we omit flows with sequence number "gaps" due to measurement loss.)

We find median and 99-th percentile maximum flight size for all flows equal to 214 and 5,296 bytes, respectively. 99.8% of flows had maximum flight sizes below 12.5KB. The remaining 0.2% flows have median and 99-th percentile maximum flight size of 19,280 and 65,334 bytes. (The highest value we observed is 878,860 bytes for a flow that lasted 6.5 hours and transmitted 2 GB in 16.7 M packets.)

Other than these final values (for just 0.2% of flows), these levels potentially indicate bandwidth underutilization; for 100Mb/s links such as those common within LBNL, a 1 msec propagation delay equates to a bandwidth-delay product of 12.5 KB. However, if actual latencies are significantly below 1 msec, then they might indeed completely utilize the available capacity. Thus,

the next steps in this analysis will be to undertake a robust analysis of connection round-trip latencies in order to confirm whether indeed most connections fail to utilize the available bandwidth, and, if so, why.

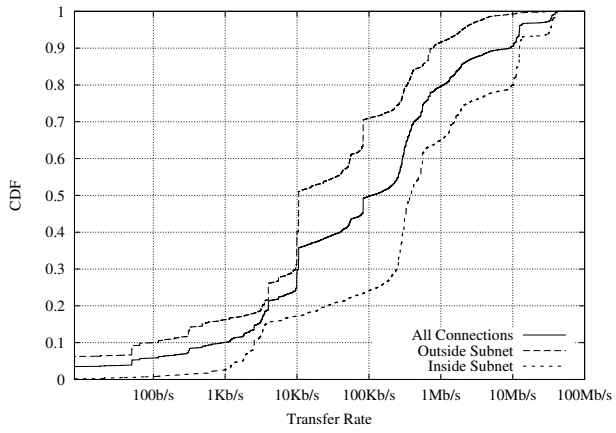

Figure 3: Observed transfer rates.

Figure 3 shows the distribution of transfer rates for all connections (solid line), as well as by location of the peers. The rates are computed as the total number of bytes in each direction divided by the duration of the connection. We calculated the rates for each direction separately and as predicted by the transfer sizes discussed in the last section the server (responder) generally shows greater throughput than the originator. The median rates are roughly 2 KBps and 10 KBps for the originator and responder, respectively. In our traces 50% of the connections are faster than 100 Kbps which is about three times more than in [9], and 5-12 times more than in [10]. We believe the higher rates are likely explained by the low delays and high raw capacities provided in the enterprise environment. This underscores the point made in § 1 that enterprise networks are often found "good enough" in comparison to wide area networks. The figure also shows that *transfers within a subnet are generally an order of magnitude faster than those cross subnet boundaries*, even though the latter remain within the enterprise. We note that more than 20% of the transfers within a subnet obtain rates of more than 10 Mbps, while less than 1% of connections cross subnet boundaries achieve such rates.

Type.	Conns. (%)	Bytes (%)
Short-Small	57.2	0.6
Short-Large	2.6	0.8
Long-Small	31.8	0.8
Long-Large	8.4	97.8

Table 3: Breakdown of traffic classes.

We next separate flows into four classes by duration and size. We use a total transfer size threshold of 10 KB to classify each connection as "small" or "large" and a threshold of 1 second to classify each connection as "short" or "long". While the thresholds are arbitrarily chosen we note that using 50 KB and 5 seconds yields nearly the same classification as our chosen thresholds. Table 3 shows shares of connections and bytes in each category. As expected we observe that most connections (89%) are small, but most of the bytes are carried in the large connections (98.6% total).

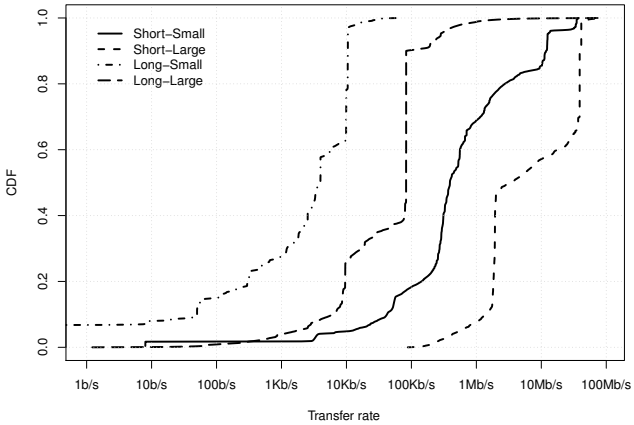

Figure 4: Connection rates.

Figure 4 shows the distribution of transfer rates for each of the four classes of connections. As expected, long-small show the worst performance. On the other hand, the short-small transfers show that it is possible to well utilize the capacity to send short transactions with nearly 20% of the connections sending at roughly 10 Mbps or more. The long-large flows somewhat surprisingly send at slow rates compared to the network capacity. While these connections carry 98% of the bytes, they do not transmit those bytes efficiently; these observations point to an obvious area for future study and performance improvement. The short-large flows have highest rates. Note, this class represents little of the network traffic in either connections or bytes (per Table 3), mostly consisting of HTTP traffic between two particular hosts and manifesting two modes in the rate distribution, at 1.7 Mbps and 37 Mbps.

Correlations between duration, size and rate have been studied in [10, 9] and in more detail in [1]. Table 4 suggests that correlations between ($\log D$,$\log S$) are similar to those in all the three other studies which also reported weak positive correlation. We observe strong negative correlation between ($\log D$,$\log R$), which is stronger than in [9] and much stronger than in [10, 1]. Our traces show negative ($\log S$,$\log R$) correlation for

Type.	Dur-Size	Dur-Rate	Size-Rate
All	0.43	-0.87	0.07
Short-Small	0.30	-0.71	0.46
Short-Large	0.63	-0.91	-0.26
Long-Small	-0.17	-0.79	0.74
Long-Large	0.39	-0.66	0.43

Table 4: Correlation coefficients of log(data).

the short-large subset. However, the three other classes have medium (as in [9]) to strong (as in [10, 1]) correlations. In [1] authors found that strong correlation between rate and size can be explained by TCP and application specifics.

6 Conclusions and Future Work

We have presented a preliminary analysis of TCP transport behavior seen inside a medium-sized enterprise. To do so we drew upon a collection of 50 traces that each record activity from 10 switch ports at the Lawrence Berkeley National Laboratory. In general, we confirm the common presumption that enterprise connections enjoy loss-less, high speed interconnection. But we also find that weighted by connections, the traces contain a wide range in the proportion of connections that are cleanly established and terminated vs. problematic; the proportion of TCP enterprise traffic that stays confined inside a single subnet likewise can vary greatly by vantage point and measurement time; and that applications differ sharply in whether their use remains within a subnet, includes both intra- and extra-subnet connections, or always leaves the subnet. We confirm that the distribution of the byte volume of most applications exhibits a heavy tail, but some applications quite prevalent within the enterprise lack this skew.

We find out-of-order packet delivery much more rare than observed for wide-area traffic, and likewise for packet corruption and replication. Additionally, we find that 0.5% of TCP senders send at least one retransmission. Our analysis of maximum flight sizes indicates that effective TCP windows are generally under 8 KB, though we have not yet determined whether these windows prevent connections from making full use of the available path capacity.

We find a wide range of transfer rates, with connections achieving throughputs of 3–12 times those seen for wide-area TCP. Far and away most bytes occur in long-lived, large transfers. When examining (log) correlations between connection sizes, rates, and durations, we often find agreement with previous work examining these relationships for wide-area traffic, other than the strong negative correlation we find between duration and rate, which may reflect the enterprise's "cleaner" paths, which have both little congestive loss due to cross-traffic, and high end-to-end capacity.

This analysis is only a beginning. Our immediate next steps are (1) an analysis of packet latency dynamics, (2) an assessment of retransmission timeout behavior, and, much more generally, (3) incorporation of a large set (1,000 switch ports) of traces that we have recently completed capturing.

Acknowledgments

This work was supported in part by TEKES as part of the Future Internet program of the ICT cluster of the Finnish Strategic Centre for Science, Technology and Innovation; and by the US National Science Foundation under grants CNS-0722035 and CNS-0831535.

References

[1] CHAN LAN, K., AND HEIDEMANN, J. A measurement study of correlation of Internet flow characteristics. *Computer Networks* (Jan 2006).

[2] GIROIRE, F., CHANDRASHEKAR, J., IANNACCONE, G., PAPAGIANNAKI, K., SCHOOLER, E., AND TAFT, N. The Cubicle vs. The Coffee Shop: Behavioral Modes in Enterprise End-Users. In *Proc. PAM* (2008).

[3] GUHA, S., CHANDRASHEKAR, J., TAFT, N., AND PAPAGIANNAKI, K. How healthy are today's enterprise networks? In *ACM SIGCOMM/USENIX Internet Measurement Conference* (2008).

[4] JAISWAL, S., IANNACCONE, G., DIOT, C., KUROSE, J., AND TOWSLEY, D. Inferring TCP connection characteristics through passive measurements. In *IEEE INFOCOM* (2004).

[5] NECHAEV, B., PAXSON, V., ALLMAN, M., AND GURTOV, A. On Calibrating Enterprise Switch Measurements. In *ACM SIGCOMM/USENIX Internet Measurement Conference* (Nov. 2009).

[6] PANG, R., ALLMAN, M., BENNETT, M., LEE, J., PAXSON, V., AND TIERNEY, B. A first look at modern enterprise traffic. In *ACM SIGCOMM/USENIX Internet Measurement Conference* (2005).

[7] PAXSON, V. End-to-End Internet Packet Dynamics. In *ACM SIGCOMM* (Sept. 1997).

[8] PAXSON, V. Bro: a system for detecting network intruders in real-time. *Comput. Netw. 31*, 23-24 (1999), 2435–2463.

[9] QIAN, F., GERBER, A., MAO, Z. M., SEN, S., SPATSCHECK, O., AND WILLINGER, W. TCP revisited: a fresh look at TCP in the wild. In *ACM SIGCOMM/USENIX Internet Measurement Conference* (2009).

[10] ZHANG, Y., BRESLAU, L., PAXSON, V., AND SHENKER, S. On the characteristics and origins of Internet flow rates. In *ACM SIGCOMM* (2002).

Notes

[1] By "lack of byte count" we do not mean zero bytes, but rather that we were unable to determine the number of bytes being transferred in a given direction—usually due to some puzzle whereby we do not observe a valid connection.

[2] Note, we added 1 byte to every byte count before computing the ratio to alleviate divide-by-zero issues.