Performance Within A Fiber-To-The-Home Network

Matthew Sargent
Case Western Reserve University
Cleveland, OH
matthew.sargent@case.edu

Mark Allman
International Computer Science Institute
Berkeley, CA
mallman@icir.org

ABSTRACT

Fiber-To-The-Home (FTTH) networks are on the brink of bringing significantly higher capacity to residential users compared to today's commercial residential options. There are several burgeoning FTTH networks that provide capacities of up to 1 Gbps. We have been monitoring one such operational network—the Case Connection Zone—for 23 months. In this paper we seek to understand the extent to which the users in this network are in fact making use of the provided bi-directional 1 Gbps capacity. We find that even when given virtually unlimited capacity the majority of the time users do not retrieve information from the Internet in excess of commercially available data rates and transmit at only modestly higher rates than commodity networks support. Further, we find that end host issues—most prominently buffering at both end points—are often the cause of the lower-than-expected performance.

1. INTRODUCTION

Fiber-To-The-Home (FTTH) networks are starting to bring significantly higher capacity last mile networks to residential users when compared with traditional commercial offerings [1, 18, 3, 2]. Such networks provide upwards of 1 Gbps of capacity to residential users and this begs the obvious question of whether such users will be able to actually employ such capacity. This question breaks down into two components. The first component is whether the applications and services the users leverage will even try to utilize the capacity. We address this question in companion work [29] and in general find that FTTH users behave much like residential users that employ more traditional connection technologies. The second component is whether our protocols and implementations thereof are up to the task of utilizing the capacity these ultra-broadband networks provide.

We tackle this second question in the context of monitoring a neighborhood connected via FTTH known as the Case Connection Zone (CCZ). This network connects each of roughly 90 residences via a bi-directional 1 Gbps fiber link. After discussing the CCZ network, our data collection and the calibration of our data in the next section we then address the issues involved in TCP performance in the remainder of the paper. We focus on TCP performance in this initial paper as (*i*) TCP carries most of the bytes in the network and (*ii*) its reliability mechanism exposes enough information to assess performance using packet header traces, whereas UDP performance analysis requires a deep understanding of application payloads. In future work we intend to undertake an in-depth investigation of the performance of UDP traffic. Finally, we stress that while the monitored network is small, our goal is to gain an *initial empirical understanding* of the operation of protocols within FTTH networks *as experienced by users*. This leads us to a methodology that relies on observing normal user traffic rather than actively probing the network or parts thereof. While we believe we are the first to seek such an understanding, this study will certainly not be the last on this topic.

We find that even when given virtually unlimited capacity the majority of the time users do not retrieve information from the Internet in excess of commercially available data rates and transmit at only modestly higher rates than commodity networks support. Further, we find that end host issues—most prominently buffering at both end points—are often the cause of the lower-than-expected performance.

2. DATA

The Case Connection Zone (CCZ) provides a bi-directional 1 Gbps fiber connection to each home within the network. These fibers terminate in a switch, which connects to the rest of the ISP's network via a 1 Gbps link. Additionally, the traffic is mirrored to a port we monitor. Our monitoring setup is described in more detail in [29]. Our data gathering spans from January 25, 2011 to December 31, 2012. For the work we describe in this paper we collect two kinds of data, as follows.

Connection Logs: We use Bro 1.5.2 [27] to continuously monitor CCZ traffic and log each transport-level connection. The logs—denoted $\mathcal{L}_c$—contain information about each connection such as the duration, how much data is being sent, and which protocols and applications are being used. Before analyzing the logs we carefully filter out garbage connections—e.g., connection attempts from network scanners—to focus our attention only on connections we believe to accomplish useful work. Details about our filtering strategy are described in companion work [29]. In total our dataset contains 973M connections after the calibration process.

Packet Traces: We collect packet-level traces for a subset of the duration of our collection period. The torrent of traffic precludes the capture of all packets for the entire measurement period. We therefore collect packet traces from the 11^{th} through the 17^{th} of each month, as follows. We divide each day in the collection period into six hour blocks and collect a one-hour trace starting at a random time within each block. While we captured full packet payloads, saving all such traces quickly became logistically burdensome. We therefore randomly chose one trace to retain in full for

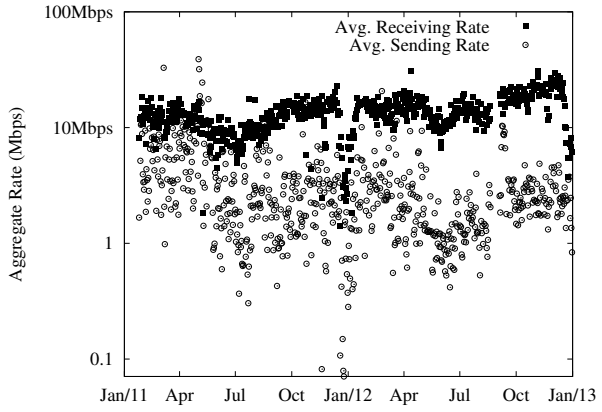

Figure 1: Aggregate transmission rates per day.

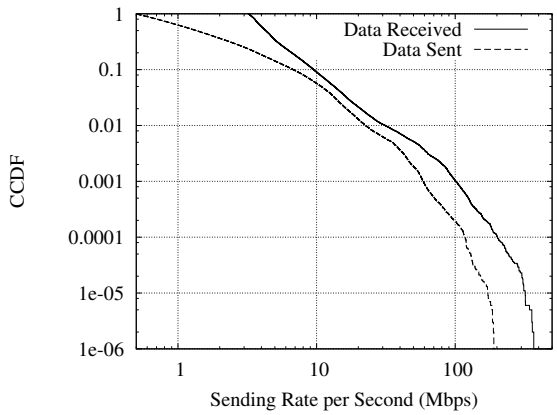

Figure 2: Throughput for top 1% bins.

each day and stripped the payload from the remaining three traces. For the analysis in this paper payloads are not used, except to calibrate the measurement apparatus.

Failing to record all packets that cross our monitoring point during our observation period can lead to biased or wrong conclusions. Therefore, before using our packet traces we must assess measurement-based packet loss. While *tcpdump* reports a number of losses this is often not telling as it is difficult for the *tcpdump* application to understand (and hence count) what was not observed. Rather, we analyze the traces themselves for signs of missing packets. In particular, we analyze TCP traffic for cases where we observe an acknowledgment for data that never appears in the trace. These "gaps" represent cases where the ultimate recipient clearly received the data, but the data was not recorded in our traces.

Bro's "gap analysis" only works for packet traces that contain full packet payloads as the analysis is part of the payload reassembly procedure. Therefore, we analyze the 162 payload traces in our corpus. Further, Bro's analysis only covers TCP traffic as it has sequence numbers that make it readily amenable to such analysis. Therefore, we stress that our analysis of measurement-based loss is not comprehensive. Rather, it is suggestive of the loss rates we expect from the collection apparatus. In 49 of the 162 traces we detect no measurement-based loss. In the remaining 113 traces we find loss rates of (*i*) less than 0.001% in 73 traces, (*ii*) $[0.001\%, 0.01\%)$ in 29 traces and (*iii*) $[0.01\%, 0.07\%)$ in 10 traces. In the remaining trace we find a measurement-based loss rate of 0.21%. Therefore, while individual measurement-based loss events may impact individual analyses we undertake, we conclude that in a general and statistical sense the measurement-based loss rate is low enough to not impact the insights we derive from our dataset.

Note, we do not assert that our dataset is "*typical*" or "*representative*". Networks are vastly heterogeneous and insights from various vantage points are necessary to draw general conclusions (as we discuss in more detail in [5]). This study is an initial investigation and we encourage the community to build on this work by studying additional FTTH networks.

3. OBSERVED TRANSMISSION SPEED

We now turn our attention to the salient feature of FTTH networks: speed. Figure 1 shows the aggregate sending and receiving rates for all TCP traffic across all hosts in the CCZ for each day in our dataset. The average daily aggregate incoming traffic rate is roughly 13.4 Mbps. We find that patterns in the data follow the academic calendar. As some students leave the CCZ during academic breaks, the overall population of CCZ is reduced.[1] This reduction in population naturally causes the overall sending and receiving rates to decline with the largest reductions taking place during the winter breaks. In terms of local hosts transmitting data we find that the average aggregate rate is 3.4 Mbps with similar modest dips during academic breaks.

3.1 Per-Host Speed

We will now focus on the capacity individual CCZ hosts consume.[2] For each connection in the $\mathcal{L}_c$ logs we evenly distribute the number of bytes transmitted over the duration of the connection. We then construct 1 second bins (86,400 bins per day) and assign the byte count to the appropriate bins. We track each direction independently. This even spreading of data across a connection does not reflect reality for two basic reasons: (*i*) applications do not send and receive data uniformly across the duration of a connection [28] and (*ii*) TCP's congestion control algorithms [6] constantly adjust the sending rate based on the perception of the network conditions. However, both of these dynamics happen outside our view and therefore for this initial analysis using uniform spreading suffices. However, future work will include a more fine-grained packet-level analysis.

As we sketch above, we break our data into 10.7 billion one-second bins—i.e., 86,400 bins for each direction, day and host in our dataset. To concentrate on periods when hosts are transmitting relatively rapidly we winnow our dataset to the top 1% (53.8M) bins in each direction. Figure 2 shows rate distributions for the top bins in each direction. The first point on each line shows the 99^{th} percentile of our entire distribution (since we focus this on the top 1%), which is a per-host sending rate of approximately 0.5 Mbps and a receiving rate of roughly 3.2 Mbps.

[1]The CCZ user population is roughly 60% students and 40% full-time residents of the neighborhood [17].

[2]We consider each CCZ IP address to be a "host". This is not necessarily correct due to NATs, but for our purposes only a rough approximation is necessary.

Service	Recvd (%)	Bins (%)
HTTP	82.5	96
Likely BitTorrent	5.4	19
BitTorrent	3.8	9.3
HTTPS	0.8	49.3
Unclassified	7.5	49.2

Table 1: Breakdown for top receiving applications.

The figure shows that more than 90% of the top 1% receiving bins represents a rate under 10 Mbps. In other words, over 99.9% of the overall bins do not exceed a rate available from common commodity residential networks.[3] Or, on average each user spends approximately 1.3 minutes per day employing higher-than-commodity network capacity. We also find that 0.1% of the top 1% receiving bins—or less than one second per day per host—shows an aggregate receiving rate of more than 100 Mbps.

Due to commercial networks often being asymmetric, the CCZ network provides a larger relative improvement in uplink capacity than in downlink capacity. While users only exceed a nominal commodity receiving rate (10 Mbps) 0.1% of the time, they exceed a nominal commodity uplink of 0.5 Mbps 1% of the time.[4] Further, we note that CCZ user transmission rates exceed 10 Mbps approximately 0.06% of the time and 100 Mbps roughly 0.0002% of the time. The data suggests that residential users' current usage patterns and applications are generally well-served by commodity downlinks, but when provided more outbound capacity users will take advantage of these resources to some degree.

Finally, we note that in 3 million instances (5.6%) whereby a given host's corresponding sending and receiving bins are both in the top 1% lists. This illustrates that in a non-trivial number of cases a particular host is engaged in high-speed data transfers in both directions, e.g., as part of a peer-to-peer network.

3.2 High-Rate Applications

We now briefly analyze which applications are active during periods of high capacity use. The following analysis takes into account only the top 1% of the bins as discussed in § 3.1. Table 1 shows the percentage of the incoming data volume for each application that receives at least approximately 1% of the total incoming data volume in the top bins. Additionally, the table shows the percentage of the top bins in which we find the service. The table shows that HTTP/HTTPS is responsible for over 82% of the data volume during high utilization periods. Further, 96% of the top bins contain HTTP/HTTPS traffic. Finally, we also find BitTorrent to be a mild contributor during periods of high rate data reception.[5]

Table 2 shows our findings for top applications in terms of data transmitted by CCZ users during the top 1% utilization periods. We find BitTorrent to be the largest contributor—both in volume and active bins. We additionally find web traffic, Minecraft and port 1111 traffic[6] to each modestly contribute to high rate data transmission.

Service	Sent (%)	Bins (%)
Likely BitTorrent	41	78
BitTorrent	35.4	55.3
Other-1111	8.9	1.4
Minecraft	6	10.6
HTTP	2.9	71
HTTPS	1.9	53
Unclassified	3.9	46.5

Table 2: Breakdown for top sending applications.

4. TRANSMISSION SPEED CAUSES

In § 3 we study transmission rates on a host-level basis. We find the hosts do not often use anywhere close to the available capacity. A natural question is: *why?* In this section we strive to analyze our packet traces to gain an initial understanding of what is limiting performance. TCP's performance is dictated by a set of congestion control algorithms [6] and has a number of dependencies, including (*i*) the TCP receiver's advertised window, (*ii*) the size of the TCP sender's retransmission buffer, (*iii*) the RTT of the network path, (*iv*) the loss rate along the network path and (*v*) the application's sending pattern. Of these, (*ii*) and (*v*) are not readily visible in packet traces, while the others are either exposed directly by the protocol or can be estimated from traces. In this section we use these pieces of information to study connection-level transmission speed.

4.1 Potential Speed

TCP's performance is ultimately constrained by the RTT of the network path and the receiver's advertised window. In particular, the upper-bound on performance is $\frac{advwin}{RTT}$. This upper-bound requires (*i*) the sender's retransmission buffer to be at least as big as the advertised window, (*ii*) the application to keep the TCP buffer full and (*iii*) no loss along the network path such that TCP's congestion window dynamically reduces the sending rate. For the purposes of assessing how fast FTTH-connected hosts *can* send and receive data we assume these requirements hold in this subsection.

Figure 3 shows two distributions for incoming (left) and outgoing (right) traffic. The solid line on each plot shows the performance if the connection were to use the maximum advertised window we observe over the course of the connection. We find that less than 0.2% of the connections can possibly utilize the full 1 Gbps at the disposal of these hosts based on their advertised window sizes (for both incoming and outgoing traffic). Meanwhile the median transmission rate is 13.6 Mbps for incoming traffic and 5.4 Mbps for outgoing traffic. Further, we find the entire distribution to show generally higher incoming rates than outgoing rates.

Some TCP implementations use "autotuned" socket buffers, whereby the size of the socket buffers—and therefore the advertised window—dynamically adjusts with the connection's rate [30]. In other words, if a host detects that the advertised window is hampering performance, additional buffer space is allocated to the connection and subsequently the size of the advertised window is increased. In this case, the analysis above does not correctly portray performance limits because advertised window is not necessarily hindering

[3]We are aware of faster commodity networks, but 10 Mbps is the right order.

[4]Again, our aim is not to quibble about commodity rates, but to illustrate the difference between the uplink use and downlink use by CCZ users.

[5]Note, "Likely BitTorrent" denotes otherwise unclassified traffic that involves a CCZ host that is simultaneously known to be using BitTorrent.

[6]Port 1111 traffic seems to correspond to either RTMP (Flash) or Daodan malware [29].

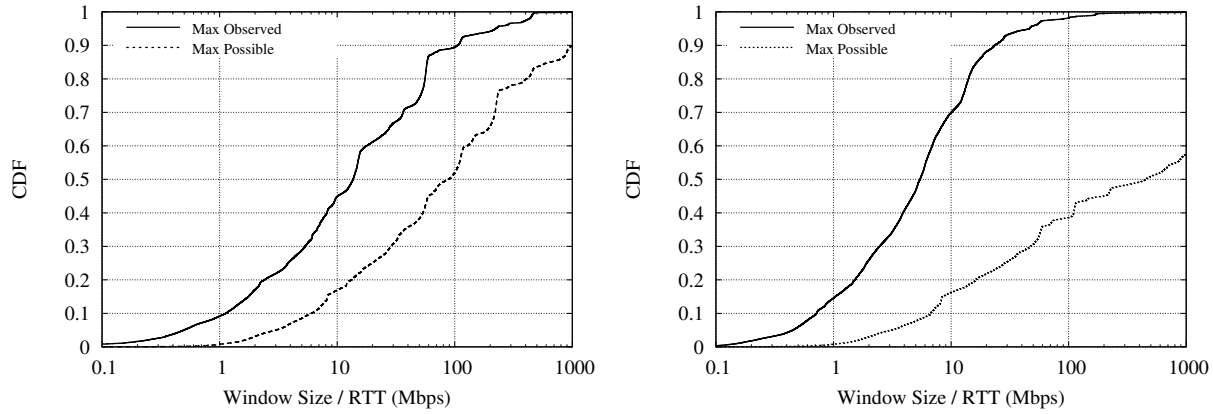

Figure 3: Distribution of maximum sending rates based on the advertised window for incoming (left) and outgoing (right) traffic.

a connection's performance, but rather is dictated by the connection's performance.

Unfortunately, there is no direct way to understand whether a host is autotuning its socket buffers. We therefore sketch a bound on how well hosts can perform, as shown by the second (dotted) line on the plots in figure 3. This line shows the distribution of the transmission rate based on the maximum possible advertised window size that can be expressed within the TCP headers. Nominally, the maximum advertised window is 64 KB, but the window scaling option can increase this to up to 1 GB. We find the theoretical window size is far greater than the maximum observed advertised window across directions, showing that hosts are not using the maximum socket buffers that can be encoded. We find that roughly 10% of the incoming connections and over 40% of the outgoing connections could encode windows that would yield a rate of at least 1 Gbps.

Above we establish that hosts *actually* advertise quite modest windows—which hamper performance—even though they *could* advertise larger windows. A final question is how often hosts increase their advertised window during the course of a connection. This speaks to the popularity of autotuned socket buffers. As a quick check we analyze each connection in our dataset for each hosts' initial and maximum advertised windows. We use the initial advertised window as a "base" of sorts since this is reflective of the buffering allotment at the connection's inception. We find that in roughly 80% of the connections the local host's maximum advertised window equals its initial advertised window. For remote hosts this equivalence holds in approximately 59% of the connections. This shows that autotuned socket buffers are in fact in use, but not on a majority of the connections.

4.2 Connections Without Loss

We now turn from the potential best-case rates that TCP can attain to examining the rates TCP does attain in our dataset and the reasons for those rates. As an initial investigation we examine the largest 10 connections in terms of data transmitted *from* a CCZ host in each of our 642 one-hour trace files. We winnow to only these large connections for several reasons. First, the performance of short connections is less dependent on capacity than on delay. Therefore, these are not useful to understanding whether TCP—or implementations thereof—can in fact use the capacity FTTH affords. Second, the process of analyzing connections for loss, RTT and flight size is burdensome in terms of both memory and computation—and especially so given that most connections that require memory and processing are short and irrelevant for our study.

Figure 4 shows the distribution of the connection sizes for our corpus of the top 10 connections per trace. We analyze the data flowing from CCZ hosts to remote hosts as our vantage point is then close to the sender and hence makes estimation of various sender properties straightforward (e.g., RTT, congestion window size, loss rate). A vantage point close to the receiver makes these sender properties difficult to estimate [26]. We will investigate traffic arriving to FTTH-connected hosts in more detail in future work. We find that 90% of the connections are at least 1 MB in size. The remaining 10% are legitimate, but come from low usage periods of our corpus (e.g., overnight on weekends or holidays). Our corpus of 6,420 connections involves 84 CCZ hosts (nearly the entire population) and 5,074 remote hosts. We first only consider connections that do not experience loss—or 926 of the 6,420 connections in our corpus. (We will consider the balance in the next subsection.)

Raw Performance: Figure 5 shows the throughput for each connection without loss as a function of the data volume. The clear trend is that performance increases with data volume. This is expected for two reasons. First, connections start transmission using slow start [21, 6], which by definition aims to begin transmission at a low rate and ramp up the speed based on the capabilities of both the network path and the remote peer. Slow start ends when congestion is detected (usually via lost data) or the sender fills the receiver's advertised window. During the process TCP will generally be underutilizing the available network capacity while searching for an appropriate sending rate at the beginning of each connection. The impact of slow start on overall transmission speed is greater for shorter transfers than for longer transfers whereby the impact is ameliorated over more time. Second, we conjecture that as data volume increases the applications better manage the transmission process (adjust buffer sizes, etc.) in an attempt to well utilize the available capacity. Such efforts are less useful for small transfers that are more dependent on the raw delay

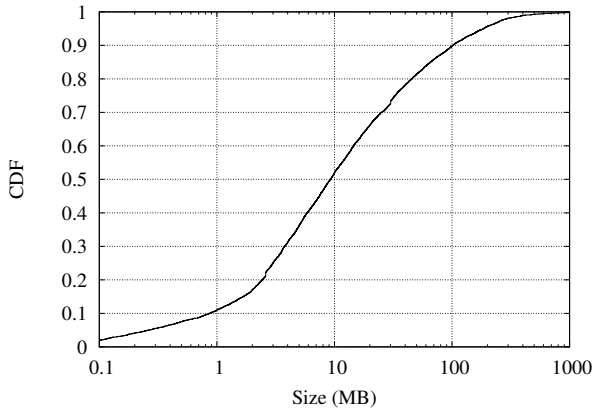

Figure 4: Sizes of the top 10 connections per trace.

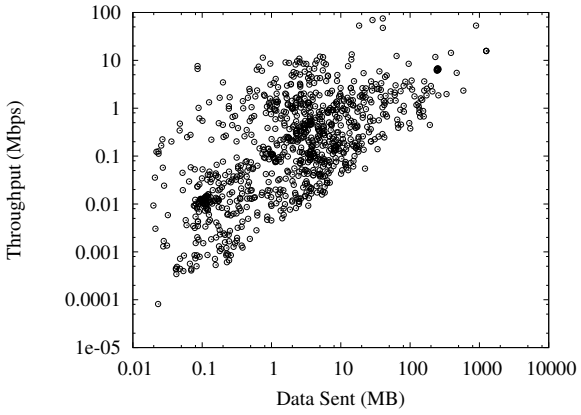

Figure 5: Connection sizes vs. throughput for connections without loss.

between endpoints than the actual capacity. We find that sending rates generally top out at around 10 Mbps, but on occasion do reach nearly 100 Mbps. We now turn to investigating why the TCP performance is much lower than the available capacity.

Advertised Window Limits: We use Bro to determine the advertised window and the maximum flight size for each connection. The flight size is the amount of data transmitted but not acknowledged at any given time and approximates TCP's congestion window. We then compare the maximum flight size with the maximum advertised window to assess whether the TCP sender is limited by the receiver's advertised window. We find that in 11.6% of the connections without loss the sending TCP is in fact constrained by receiver's advertised window. In the remainder of the connections there is some other phenomenon that is constraining the sending rate.

Sender Buffer Limits: In Figure 6 we plot the distribution of maximum flight size over all connections with no loss and for which we do not find to be advertised window limited. We find modes of varying size in this plot at 16KB, 32KB, 64KB, 96KB and 128KB. These are suggestive of some sender-side buffering issue that is limiting the flight size. The natural candidate would be the sender's TCP retransmission buffer—which limits the amount of data that

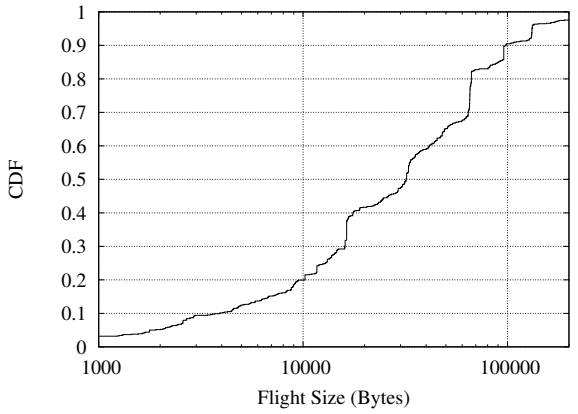

Figure 6: Distribution of flight size in connections without loss.

can be transmitted before receiving an ACK in case the data is lost and needs resent. The limit could also come from an application—e.g., in an attempt to limit the overall sending rate. While the 1 Gbps fiber link is unlikely to become overloaded it is possible that some applications are trying to protect infrastructure within a house (e.g., a wireless network). Alternatively, some applications may have a "natural rate" and exceeding this rate is pointless (e.g., vastly exceeding a video playout rate just means the end host will have to buffer content until its appointed playout time). When summing the various modes we find they account for roughly 45% of the connections with no loss.

Remote Peer Limits: We next compare the remote peers' IP address to the SpamHaus PBL [32] to determine whether the remote is a likely residential host. While this heuristic is not perfect, our results here and previous work in the literature [4] show it is a reasonable approximation. Using this definition we find that 26% of the remote peers are in residential settings, indicating that there is likely a fairly low capacity limit imposed on the remote side of the connection that could explain some of the low performance we observe.[7] The fastest connection involving a remote peer in a residential setting is 4.5 Mbps. Additionally, we observe 30% of the connections to non-residential remote peers attain throughput in excess of 4.5 Mbps—and topping out at 75 Mbps.

Summary: We find 11.6% of the connections without loss are constrained by the advertised window and the data suggests that another 45% are hampered by some sender-side buffer. Further, approximately one-quarter of the connections involve residential remote peers that likely have an anemic capacity limit (relative to the 1 Gbps available to CCZ users). Without additional insight from the end hosts themselves it is hard to reason about the performance of the remaining connections. While we cannot pinpoint the cause, we can say that buffering issues (on both hosts) do not appear to be the first order constraints.

4.3 Connections With Loss

Finally, we turn to the 5,494 connections in our large con-

[7]We are noting that residential networks likely place a capacity limit on their hosts, not that non-residential networks do not.

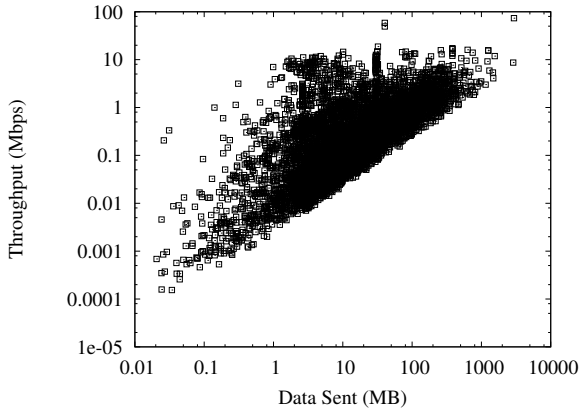

Figure 7: Connection sizes and throughput for connections with loss.

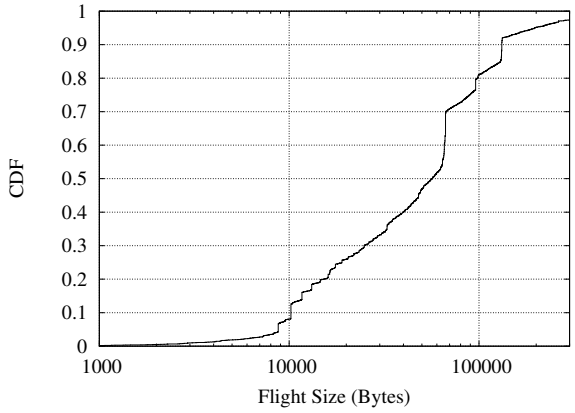

Figure 8: Distribution of flight size in connections with loss.

nection corpus that experience loss. In these cases theory suggests the loss rate and RTT combine to dictate performance [24, 25]. However, the advertised window, retransmission buffers and application behavior can still limit performance. As such, we repeat the step-wise analysis we conduct for the no loss case above.

Raw Performance: As with the no loss case we describe above we first seek to understand the relationship between transfer size and performance. Figure 7 shows the results for the connections in our corpus with loss. We again find that the minimum performance increases with the transfer size. Further, we find that 77% of the connections in the corpus do not attain even 1 Mbps. Finally, 1% of connections exceed 10 Mbps.

Advertised Window Limits: Next, we find that the maximum flight size reaches the maximum advertised window size in 15% of the connections with loss, which is a slightly higher proportion than in the no loss connections.

Sender Buffer Limits: As above we next plot the maximum flight size distribution in Figure 8. As with the no loss case we find several modes that suggest a sender-side buffer limit that ultimately constrains performance. In this case we find this happens in over 35% of the connections that experience loss.

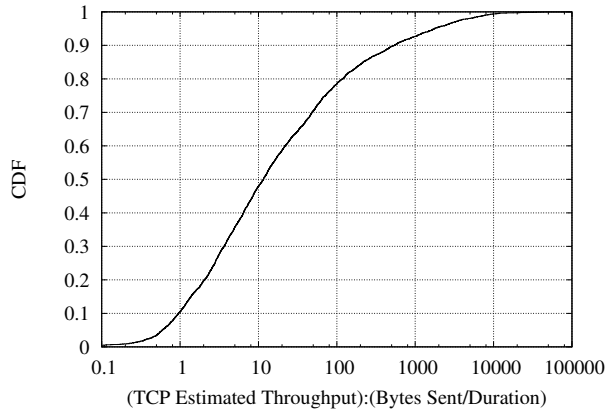

Figure 9: Ratio of theoretical throughput to actual throughput for each connection.

Remote Peer Limits: As in the no loss case above, we classify the remote peers as residential or non-residential using the SpamHaus PBL. In the set of connections with loss we find roughly half involve a residential peer. This is roughly twice as many as in the no loss case, which shows that these connections are prone to congestion—as one would expect. We find that all of the connections with loss and a remote residential peer in our dataset have throughput under 14 Mbps. Meanwhile the performance exceeds 14 Mbps to non-residential peers in just 1% of the connections. This shows the speeds are more homogeneous across type of remote endpoint than in the case of no loss. We do find a maximum throughput of 73 Mbps for non-residential peers.

Path Characteristics: As a final test we assess how the observed performance relative to theoretical throughput predictions derived from the given network path characteristics. As there are 50% of connections with loss and 43% of connections without loss being constrained by an unknown phenomenon, we attempt to determine if TCP itself is a limiting factor given network characteristics. We used the TCP model that was developed in [24, 25] and applied in [20] to calculate the rate each connection could potentially use given the connection's RTT and loss rate.[8] Figure 9 shows the distribution of the rate suggested by the TCP model versus the rate we observe for each connection with loss in our dataset. We find that in less than 10.4% of the cases the observed throughput actually outperforms the model. In the remaining roughly 89% of the cases the observed throughput is less—often by orders of magnitude—than the performance predicted by the model. This plot re-enforces our finding that the network path and TCP's congestion control algorithms should allow connections to transmit more rapidly than they in fact do. This means that host limits and application behavior—such as playing data out at a constant rate rather than as fast as possible—are causing lower performance than TCP could theoretically attain across the given path. As a final check, we examine the performance relative to the model when the data is partitioned by remote endpoint type (residential vs. non-residential). We find the results based on this split—elided due to space constraints—are similar to the distribution for all of the connections. This

[8]We used the median RTT for the connection and $b = 1$ for the model given in [20].

suggests that the type of remote host we connect to does not have an effect on this metric.

5. RELATED WORK

We are not aware of any study that directly aims to understand performance of residential FTTH-connected hosts. However, much previous work attempts to understand TCP performance in the face of various network dynamics—e.g., loss [15], reordering [11, 33], wireless bit-errors [8, 13], path asymmetry [7], excessively low bandwidth paths [12]. We stress that these are illustrative and the literature is filled with additional examples of TCP modifications to mitigate specific issues. An area of direct relation to our work is the observation that TCP's congestion control algorithms do not work well within high-speed network environments [16]. The community has addressed this short-coming with a variety of proposals for changing TCP's fundamental congestion controller (e.g., [22, 19]). Another class of work is trying to understand broadband performance (or its components) via active measurements (e.g., [14, 23]) Finally, a number of researchers leverage residential gateways to gain an understanding of home networks (e.g., [31, 10, 9]).

Our work is distinct from this previous work in that we observe performance obtained by users during their regular Internet use. We do not alter protocols or implementations thereof to try to coax better performance. Rather, we focus on determining how the protocols perform in the wild and making initial observations on what may cause the given performance.

6. SUMMARY

This paper aims to present an initial investigation of in situ performance in FTTH networks. Based on packet-level traces from an operational FTTH network we make several contributions, as follows.

- We are the first (to our knowledge) to study capacity utilization issues within an operational FTTH network over a 23 month time period.
- We find that even when given virtually unlimited capacity the majority of the time users do not retrieve information from the Internet in excess of commercially available data rates. However, in terms of transmitting data we find the FTTH users in our study use modestly more capacity than (nominally) available via commodity broadband.
- Similar to the last point we find that since 1 Gbps links to a single household are not well utilized, FTTH presents an opportunity for new applications and services to capitalize on such resources.
- We find that TCP theory suggests hosts should be attaining (much) higher performance than they are in the CCZ network. This suggests that TCP implementations are artificially limiting performance.
- We find that one of the likely reasons—but not the only—for TCP's low performance is end host buffering issues. In some cases this manifests in TCP's advertised window, but in others we find evidence of a sender-side buffer limitation.

Finally, we stress that our goal in this paper is an initial study with only passive observations. Our future work will also involve active engagement to determine whether the low performance we observe is deep-seated or requires simple tweaks to address. Additionally, we encourage others with access to FTTH networks to begin investigating performance in these emerging networks.

Acknowledgments

Lev Gonick, Mike Chalkwater, Lou Changeri, Tu Ouyang and Michael Rabinovich facilitated our monitoring of the CCZ network. Discussions with Michael Rabinovich aided this work. Katrina LaCurts developed the Bro TCP analyzer we leverage in this work. The work was funded in part by NSF grants CNS-1213157, CNS-0831535 and CNS-1161799. Our thanks to all!

7. REFERENCES

[1] Bringing ultra high-speed broadband to Stanford homes. `http://googleblog.blogspot.com/2010/10/bringing-ultra-high-speed-broadband-to.html`.

[2] Case Connection Zone. `http://caseconnectionzone.org/`.

[3] Chattanooga, Tenn Announces only 1 Gigabit Broadband Service in U.S. for Both Residential and Business Customer. `http://chattanoogagig.com/pdf/Chattanooga_GPON_EPB.pdf`, 2010.

[4] M. Allman. Comments on Bufferbloat. *ACM SIGCOMM Computer Communication Review*, 43(1), Jan. 2013.

[5] M. Allman. On Changing the Culture of Empirical Internet Assessment. *ACM Computer Communication Review*, 43(3), July 2013. Editorial Contribution.

[6] M. Allman, V. Paxson, and E. Blanton. TCP Congestion Control, Sept. 2009. RFC 5681.

[7] H. Balakrishnan, V. N. Padmanabhan, G. Fairhurst, and M. Sooriyabandara. TCP Performance Implications of Network Path Asymmetry, Dec. 2002. RFC 3449.

[8] H. Balakrishnan, S. Seshan, E. Amir, and R. Katz. Improving TCP/IP performance over wireless networks. *MobiCom*, 95:2–11, Nov. 1995.

[9] Z. S. Bischof, J. S. Otto, , and F. E. Bustamante. Up, Down and Around the Stack: ISP Characterization from Network Intensive Applications. In *ACM SIGCOMM Workshop on Measurements Up the STack (W-MUST)*, Aug. 2012.

[10] Z. S. Bischof, J. S. Otto, M. A. Sánchez, J. P. Rula, D. R. Choffnes, , and F. E. Bustamante. Crowdsourcing ISP Characterization to the Network Edge. In *ACM SIGCOMM Workshop on Measurements Up the STack (W-MUST)*, Aug. 2011.

[11] E. Blanton and M. Allman. On Making TCP More Robust to Packet Reordering. *ACM Computer Communication Review*, 32(1):20–30, Jan. 2002.

[12] S. Dawkins, G. Montenegro, M. Kojo, and V. Magret. End-to-end Performance Implications of Slow Links, July 2001. RFC 3150.

[13] S. Dawkins, G. Montenegro, M. Kojo, V. Magret, and N. Vaidya. End-to-end Performance Implications of Links with Errors, Aug. 2001. RFC 3155.

[14] M. Dischinger, M. Marcon, S. Guha, K. Gummadi, R. Mahajan, and S. Saroiu. Glasnost: Enabling end

users to detect traffic differentiation. In *Proceedings of the 7th USENIX conference on Networked systems design and implementation*, Apr. 2010.

[15] K. Fall and S. Floyd. Simulation-based Comparisons of Tahoe, Reno, and SACK TCP. *Computer Communications Review*, 26(3), July 1996.

[16] S. Floyd. HighSpeed TCP for Large Congestion Windows, Dec. 2003. RFC 3649.

[17] L. Gonick. Personal Communication, Apr. 2021.

[18] Ultra high-speed broadband is coming to Kansas City, Kansas. `http://googleblog.blogspot.com/2011/03/ultra-high-speed-broadband-is-coming-to.html`.

[19] S. Ha, I. Rhee, and L. Xu. CUBIC: A New TCP-Friendly High-Speed TCP Variant. *ACM SIGOPS Operating System Review*, 42(5), July 2008.

[20] M. Handley, S. Floyd, J. Padhye, and J. Widmer. TCP Friendly Rate Control (TFRC): Protocol Specification, Jan. 2003. RFC 3448.

[21] V. Jacobson. Congestion Avoidance and Control. In *ACM SIGCOMM*, 1988.

[22] C. Jin, D. Wei, S. Low, J. Bunn, H. Choe, J. Doyle, H. Newman, S. Ravot, S. Singh, F. Paganini, G. Buhmaster, L. Cottrell, O. Martin, and W. chun Feng. FAST TCP: From Theory to Experiments. *IEEE Network*, 19(1), 2005.

[23] C. Kreibich, N. Weaver, B. Nechaev, and V. Paxson. Netalyzr: illuminating the edge network. In *ACM Internet Measurement Conference*, Nov. 2010.

[24] M. Mathis, J. Semke, J. Mahdavi, and T. Ott. The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm. *Computer Communication Review*, 27(3), July 1997.

[25] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP Throughput: A Simple Model and its Empirical Validation. In *ACM SIGCOMM*, Sept. 1998.

[26] V. Paxson. Automated Packet Trace Analysis of TCP Implementations. In *ACM SIGCOMM*, Sept. 1997.

[27] V. Paxson. Bro: A System for Detecting Network Intruders in Real-Time. *Computer Networks*, Dec. 1999.

[28] M. Sargent, E. Blanton, and M. Allman. Modern Application Layer Transmission Patterns from a Transport Perspective. In *Passive and Active Measurement Conference*, Mar. 2014.

[29] M. Sargent, B. Stack, T. Dooner, and M. Allman. A First Look at 1 Gbps Fiber-To-The-Home Traffic. Technical Report 12-009, International Computer Science Institute, Aug. 2012.

[30] Semke, J. and Mahdavi, J. and Mathis, M. Automatic TCP buffer tuning. In *ACM SIGCOMM Computer Communication Review*, volume 28, pages 315–323. ACM, 1998.

[31] S. Sundaresan, W. de Donato, N. Feamster, R. Teixeira, S. Crawford, and A. Pescapè. Broadband Internet Performance: A View From The Gateway. *SIGCOMM-Computer Communication Review*, 41(4):134, 2011.

[32] The Spamhaus Project - PBL. `http://www.spamhaus.org/pbl/`.

[33] M. Zhang, B. Karp, S. Floyd, and L. Peterson. RR-TCP: A Reordering-Robust TCP with DSACK. In *Proceedings of the Eleventh IEEE International Conference on Networking Protocols (ICNP)*, Nov. 2003.