On Dominant Characteristics of Residential Broadband Internet Traffic

Gregor Maier
TU-Berlin/T-Labs

Anja Feldmann
TU-Berlin/T-Labs

Vern Paxson
UC Berkeley/ICSI

Mark Allman
ICSI

ABSTRACT

While residential broadband Internet access is popular in many parts of the world, only a few studies have examined the characteristics of such traffic. In this paper we describe observations from monitoring the network activity for more than 20,000 residential DSL customers in an urban area. To ensure privacy, all data is immediately anonymized. We augment the anonymized packet traces with information about DSL-level sessions, IP (re-)assignments, and DSL link bandwidth.

Our analysis reveals a number of surprises in terms of the mental models we developed from the measurement literature. For example, we find that HTTP—not peer-to-peer—traffic dominates by a significant margin; that more often than not the home user's immediate ISP connectivity contributes more to the round-trip times the user experiences than the WAN portion of the path; and that the DSL lines are frequently not the bottleneck in bulk-transfer performance.

Categories and Subject Descriptors

C.2.2 [Computer-Communication Networks]: Network Protocols—Applications; C.2.3 [Computer-Communication Networks]: Network Operations—Network monitoring

General Terms

Measurement, Performance

Keywords

Network Measurement, Application Mix, HTTP usage, TCP performance, Residential Broadband Traffic, DSL

1. INTRODUCTION

Residential broadband Internet connectivity is a mature service in many countries. This foundation of rich access allows users to tightly integrate network use into their lives—from checking the weather or sports scores to shopping and banking to communicating with family and friends in myriad ways. However, the nature of the connectivity differs from previously studied environments such as campus networks and enterprises in salient ways.

First, users of residential broadband connections will often have different goals than those in other environments, and are not subject to the same sorts of strict acceptable use policies that may regulate their access at work or at school, such as prohibitions against accessing certain Web sites or employing certain applications. In addition, we expect that the users who set up hosts and ancillary equipment in residences often have no expertise in system administration, nor much desire to understand any more than is necessary to "make it work". Finally, unlike for campuses (and to a lesser extent, enterprises), researchers rarely have large-scale access to residential traffic, and thus its makeup, dynamics, and variations remain underexamined.

In this work we present observations developed from passive packet-level monitoring of more than 20,000 residential DSL lines from a major European ISP. This unique vantage point provides a broad view of residential traffic, enabling more comprehensive and detailed characterizations than was possible in previous work, such as Cho et al.'s studies based on backbone traces [19, 9, 10], other work that examined specific applications like P2P-assisted content distribution [27] and Skype [7], or studies using active measurements [12].

In this initial exploration we focus on studying a broad range of *dominant characteristics* of residential traffic across a number of dimensions, including DSL session characteristics, network and transport-level features, prominent applications, and network path dynamics. Our study discovered a number of results we found surprising in terms of the standard "mental models" one develops from the Internet measurement literature and by talking with operators and colleagues. For example:

- HTTP traffic, not peer-to-peer, dominates. Overall, HTTP makes up nearly 60% of traffic by bytes while peer-to-peer contributes roughly 14%. Even if we assume that all unclassified traffic is peer-to-peer, this latter figure only rises to one-quarter, confirming contemporaneous observations by Erman et al. [15] for a major US broadband provider.

- DSL sessions run quite short in duration, with a median length of only 20–30 min. The short lifetime affects the rate of IP address reassignments, and we find 50% of addresses are assigned at least twice in 24 h, and 1–5% of addresses more than 10 times, with significant implications for IP address aliasing.

IMC'09, November 4–6, 2009, Chicago, Illinois, USA.
Copyright 2009 ACM 978-1-60558-770-7/09/11 ...$10.00.

Name	Time	Duration	Size	Loss
WEEK	Aug 08	14x 90 min	100–600 GB ea.	none
SEP	Sep 08	24 h	>4 TB	several multi-second periods with no packets
APR	Apr 09	24 h	>4 TB	see above

Table 1: Summary of anonymized packet traces

- Delays experienced from a residence to the ISP's Internet gateway often exceed those over the wide-area path from the gateway to the remote peer. We find a median local component of 46 ms (due to DSL interleaving), versus a median remote component of 17 ms.
- Users rarely employ the full capacity of their lines, confirming observations by Siekkinen et al. [47]. 802.11 wireless networking in customers' homes, and TCP settings on the residential systems, appear to limit the achievable throughput.

We organize the paper as follows. After giving a short overview of our datasets and terminology in Section 2, we look at DSL session characteristics in Section 3. In Section 4 we explore which applications are popular among the user population, and take a closer look at the most predominant, HTTP, in Section 5. We briefly examine transport protocol features in Section 6, and examine path characteristics in Section 7. We summarize in Section 8.

2. DATA AND TERMINOLOGY

We base our study on passive, anonymized packet-level observations of residential DSL connections collected at aggregation points within a large European ISP. Overall, the ISP has roughly 10 million (4%) of the 251 million worldwide broadband subscribers [38]. They predominantly use DSL. The monitor operated at the broadband access router connecting customers to the ISP's backbone. The access bandwidth of the monitored lines varies between 1,200/200 Kbps (downstream/upstream) and 17,000/1,200 Kbps, with the exact rate depending on both the customer's contract and their distance from the DSLAM (the ISP's line-card). In the portion of the network we monitored most users had distances low enough to in principle support 17 Mbps.

For clarity of exposition, we define the following terms. A *line* denotes a physical DSL line as identified by a line-card identifier. We define a DSL-level *session* as the period when the DSL modem and the line-card are together in operation. We refer to the network between the monitoring point and the customer as the *local side*, as opposed to the *remote side* (remainder of the Internet). Similarly, the customer sends *upstream* traffic and receives *downstream* traffic. A *flow* refers to unidirectional data transmission at the usual 5-tuple granularity (IP addresses, transport protocol, transport ports). A *connection* is a bi-directional transport-level communication channel, demarked for TCP by the usual control packets (SYN, FIN/RST) and for UDP by the the arrival of the first packet and the absence of activity detected using an idle timeout of 20 s. Finally, the *originator* endpoint actively initiated the connection, as opposed to the *responder*, which passively awaited the connection request.

Our monitoring vantage point allowed us to observe more than 20,000 DSL lines from one urban area, connected to one access router, for which we employed Endace DAG network monitoring cards [14] for traffic capture. Immediately after capture we extract application classifications (using DPD [13]; see Section 4.1) and information such as HTTP headers from the traces using Bro [41], storing anonymized versions of the packet and application headers for later processing. Table 1 provides an overview of the data traces, including when gathered and overall size. WEEK reflects 14 intervals of 90 minutes each, gathered twice per day during the same hours over the course of one week. In addition, we gathered anonymized DSL session information, including the session start and end times, anonymized IP address, anonymized line-card identifier, and the configured access-bandwidth. Along with DSL session traces for each of our packet measurements, we obtained a 10-day DSL session-only trace from Jan 2009 (TEN), as well as six separate 24h session-only traces (see Table 2).

Name	Time	Duration	Loss
TEN	Feb 2009	10 days	none
EVERY4	Jan–Feb 2009	6x 24 h	none

Table 2: Summary of additional anonymized DSL session information

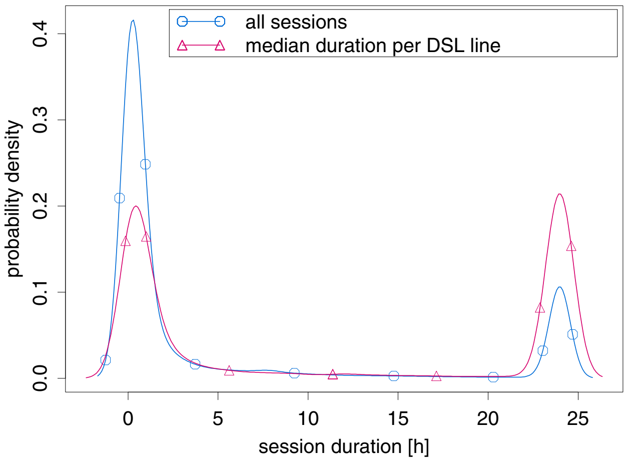

Figure 1: PDF of session durations for sessions with duration longer than 5 minutes for dataset TEN.

To simplify the presentation, we focus our discussion on SEP and TEN. However, we verified our results across all traces and explicitly point out differences. In particular, we use the 14 samples from WEEK to verify that there are no dominant day-of-week or other biases apparent in the 24 h traces (SEP, APR). In addition, we cross-checked our results with sampled NetFlow data exported by 10 of the ISP's routers. This further increases our confidence in the representativeness of our application mix results.

3. DSL SESSION CHARACTERISTICS

We begin our study with a look at the behavior of the users' DSL sessions (periods of connection to the ISP's network). A first basic question concerns the durations of such connections. Network analysis studies often make the assumption that one can use IP addresses as host identifiers (for example, for studies that count the number of systems exhibiting a particular phenomenon), and previous studies have found stability in these mappings on the order of several hours to days. Moore et al. analyzed the 2001 Code

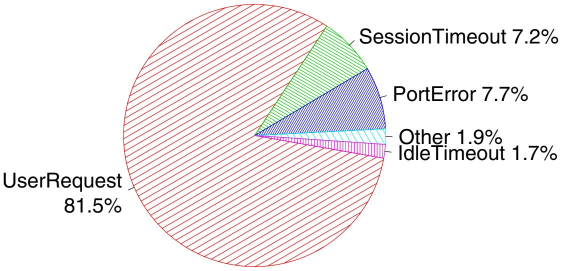

Figure 2: DSL (Radius) session termination causes distribution for sessions lasting longer than 5 minutes.

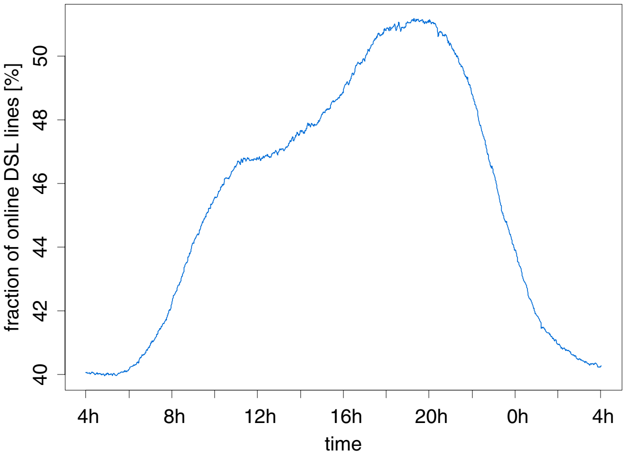

Figure 3: Relative number of concurrent DSL lines across time for one 24h weekday period of dataset TEN. Note the base-line.

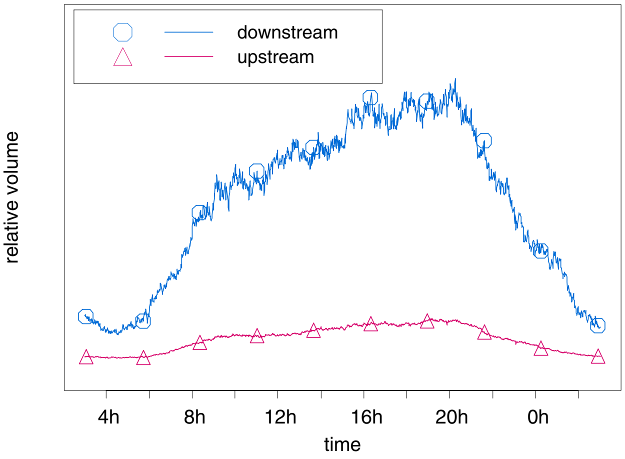

Figure 4: Bandwidth usage of all DSL lines across time (1 min bins).

Red outbreak and found that for larger timescales (days to weeks), IP addresses cannot be used as reliable host identifiers due to IP reassignment [35]; they did not examine timescales below several hours. Xie et al. observed some highly volatile dynamic IP address ranges, which they attributed mainly to dial-up hosts [54].

Thus, we expected to find typical session lengths of several hours. However, we find instead that many are quite short. We base our analysis on Radius [43] logs, which many European ISPs use for authentication and IP address leasing. Radius supports two timeouts, *SessionTimeout* and *IdleTimeout*, though the monitored ISP only makes use of the first. *SessionTimeout* performs a role similar to the DHCP lease time, limiting the maximum lifetime of a session. The ISP sets it to 24 hr (a popular choice among European ISPs [52, 37]). DSL home routers generally offer an option to reconnect immediately after a session expires. However, in contrast to DHCP, Radius does not provide an option to request a particular IP address (e.g., the previously used IP address), and the ISP allows addresses to change across sessions.

We analyzed the DSL session duration of the Radius logs, excluding sessions lasting under 5 minutes. Surprisingly, we find that sessions are quite short, with a median duration of only 20–30 minutes. Figure 1 shows the distribution of DSL session durations for those longer than 5 minutes, computed over all sessions, along with the distribution of the median session duration computed per DSL line. The data exhibits two strong modes around 20–30 minutes and 24 hr (the maximum duration given the Radius setup), partitioning the DSL lines in two large groups: always-connected lines, and lines that only connect on demand and disconnect shortly after. We do not find much in between (lines connected for several hours). While previous work found short sessions (70% lasting at most 1 hour) in the context of wireless university networks [30], we found it striking to discover such short DSL sessions in residential networks, in violation of our mental model that sessions would be significantly longer-lived.

To check if there is a significant difference in DSL session durations for P2P users vs. non-P2P users (see Section 4), we partitioned the DSL-lines into two groups. Overall, the characteristics of the distribution are similar, with two prevalent modes. However, we find that P2P users tend to have longer session durations and that a larger fraction of P2P users always remain connected.

To better understand the high prevalence of short sessions, we examined the Radius termination status in the logs. Radius differentiates between 18 termination causes. Figure 2 shows the distribution of causes for sessions longer than 5 minutes. We observe that more than 80% of sessions are terminated by user request (this rises to 95% for sessions under 5 minutes). Most likely these are caused by idle timeouts in the DSL modem on the client side. While most current broadband contracts are flat-rate, in the past time-based contracts were popular in Europe. Indeed, these latter are still offered by most European ISPs. Therefore, it is likely that consumer DSL routers come with a small idle timeout as a factory default in an effort to aid users in keeping down costs, and we verified this for several popular home routers. The second most common termination cause is PortError, which likely results when users power off their DSL modem as part of powering down their entire computing setup.

Since many DSL sessions are short and Radius does not preserve IP address assignments across sessions, we therefore expect (and find) IP addresses used for multiple DSL lines across each dataset. During a 24 hr period we find 50% of the IP addresses assigned to at least 2 distinct DSL lines, and 1–5% to more than 10 DSL lines. These results underscore *the peril involved in using an IP address as a long-term reliable host identifier*.

Previous work found that for consumers diurnal patterns start with activity in the morning, steadily increasing throughout the course of the day, with the height of activity starting in the early evening and lasting till midnight [19, 17]. We see this same overall pattern in terms of the number of active DSL sessions, as shown in Figure 3. However, we note that the variation is in fact modest, with 40% of the lines permanently connected. We also observe a slight day-of-week effect, with Sundays having larger numbers of concurrent sessions, and Friday/Saturday having lower daily maxima than other weekdays.

We also observe a diurnal pattern in bandwidth usage, per Figure 4, with the relative differences now being much more pronounced. After all, keeping a session alive does not imply any bandwidth usage per se.

Our data also offers us an opportunity to analyze the potential resource requirements of an ISP wide NAT deployment. In particular, we study how many public IP addresses are needed to support the traffic on the monitored lines. For this purpose we count the number of concurrently active TCP/UDP connections and add a 5-min or 10-min timeout to the duration of each 5-tuple. Doing so implies that we do not allow the immediate reuse of each 5-tuple. Under the assumption that a single public IP address can support 65,536 concurrent connections (due to available port space) we find that a single public IP address suffices to support 1,300–2,000 active lines with a 10-min timeout, and roughly twice that when using a 5-min timeout.

Given the maximum number of concurrently connected lines, 5–10 public addresses would in principle suffice to accommodate the monitored DSL-lines—a huge reduction of the required public IP address space.

So far we only considered outgoing connections, yet a NAT must also accommodate incoming connections. We find that very few lines utilize incoming connections for traditional services such as HTTP. Most successful incoming connections are to ports commonly used for VoIP (SIP and RTP), default P2P ports, IPSec key management, and traceroute destination ports. It is plausible that P2P applications can use Universal Plug-and-Play to dynamically negotiate ports with the NAT devices. SIP and RTP include NAT traversal solutions and proxy services. In addition, we find that almost all SIP connections are to/from the ISP's SIP server, since SIP is used as a transparent VoIP replacement for end-customers. Moreover, one does not have to support traceroute. As such it appears that one would not need too many additional public IP addresses for incoming connections.

While we acknowledge that more in-depth study is needed, it appears that such NAT deployment would indeed conserve a very large number of public IP addresses. Whether it proves manageable, and/or impedes innovation, remains a separate question.

4. APPLICATION USAGE

To understand the popular applications among our user population, we examine our application classifications (made at data-collection time) and anonymized application-layer header traces. We in addition assess how well purely port-based classification would perform for correctly identifying residential traffic patterns, and characterize traffic asymmetries.

Previous studies of Internet application mix found HTTP to predominate around the turn of the century. Fraleigh et al. [18] analyzed packet level traces recorded from the Sprint backbone in 2001, finding that in most traces HTTP contributed > 40% of all bytes, though several traces had P2P contributing 80%.

Subsequent studies found that P2P became the dominant application. Ipoque and Cachelogic both used data from their deployed deep packet inspection and traffic management systems at selected customers sites to assess the application usage [45, 46, 40]. Cachelogic claimed that by 2006 P2P accounted for more than 70% of the traffic, with Ipoque supporting this claim for 2007. For 2008 Ipoque found that P2P in Europe accounted for more than 50% of traffic (with Web contributing another 25%).

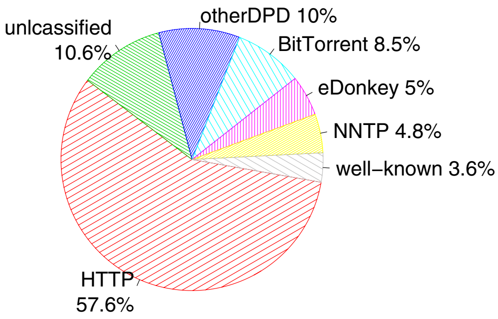

Figure 5: Application Mix for trace SEP.

On the other hand, Hyun-chul et al. reported that payload-based analysis conducted in 2004 from within the PAIX backbone found almost no P2P traffic, but more than 45% HTTP [23]. On the other hand, the same study developed how at various university networks the traffic differs; for example, at KAIST in 2006 they found under 10% HTTP, and 40–50% P2P.

Cho et al. [9, 10] also found in 2008 that TCP port 80 contributed only 14% of all bytes in Japanese ISP backbones (9% in 2005), with the bulk of traffic being on unassigned ports. None of the default P2P ports contributed more 1% of the traffic volume. (The authors point out that WINNY, the most prelevant P2P application in Japan, uses unassigned ports.) They found that residential traffic exhibited a shift to more streaming and video content, which agrees with recent blog and news reports that claim that P2P traffic has somewhat declined, with streaming media increasing [50, 3]. With an assumption that the unassigned ports indeed reflected P2P, their datasets indicated that P2P dominated the total traffic volume.

From a somewhat different perspective, Kotz and Essien [29, 30] reported that 50% of wireless traffic in 2001 on a university campus, which included residential buildings, used HTTP's well-known ports, with 40% of this traffic *incoming* to local servers. Henderson et al. [22] compared these results with newer traces from 2003/2004 of the same network, finding major shifts in the application mix (HTTP 63%→27%, File systems 5%→19%, P2P 5%→22%), and that more traffic stayed on-campus than in 2001 (70%, up from 34%). Of the P2P traffic, 73% remained internal. Therefore, we cannot easily compare these results to residential broadband use. Finally, Fraleigh et al. [18] also used a port-based approach on 2001 data, finding that on some links 60% of the bytes come from P2P and only 30% from HTTP, although most of their traces have more than 40% HTTP.

Given this context, we now turn to an analysis of application usage in our 2008/2009 residential traces.

4.1 Application usage analysis

To robustly identify application protocols, we employ the Bro system's Dynamic Protocol Detection (DPD) [13]. DPD essentially tries to parse each byte stream with parsers for numerous protocols, deferring determination of the corresponding application until only that application's parser recognizes the traffic. DPD also uses regular expression signatures to winnow down the initial set of candidate parsers. The Bro distribution includes full DPD parsers/recognizers for BitTorrent, FTP, HTTP, IRC, POP3, SMTP, SSH, and SSL. We extended the set of detectors with partial recognizers for eDonkey and Gnutella (both based on L7-filter signatures [32]), NNTP, RTP, RTSP, SHOUTcast, SOCKS, and Skype.

In the SEP trace we can classify more than 85% of all bytes, with another 3.6% using well-known ports, as reflected in Figure 5. We find that *HTTP, not P2P,* is the most significant protocol, accounting for 57% of residential bytes. We also find that NNTP contributes a significant amount of volume, nearly 5%. Almost all of the NNTP bytes arise due to transfers of binary files, with RAR-archives (application/rar) being among the most common file types, suggesting that the traffic reflects the equivalent of file-sharing.

We find that P2P applications—BitTorrent, Gnutella, and eDonkey—contribute < 14% of all bytes, with BitTorrent the most prevalent, and Gnutella almost non-existent. However, the L7-filter signatures for eDonkey may be incomplete. We observe a significant amount of traffic (1.2%) on well-known eDonkey ports that the classifier fails to detect as eDonkey. The distribution of connection sizes for this traffic closely matches that for traffic positively identified as eDonkey (and differs from other applications). If we presume that this indeed reflects eDonkey traffic, then the overall share of P2P traffic increases to 17–19%, with eDonkey's popularity roughly the same as BitTorrent's. But even if we assume that *all* unclassified traffic is P2P, the total P2P share still runs below 25%.

P2P applications could also in principle use HTTP for data download, thus "hiding" among the bulk of HTTP traffic and increasing the significance of P2P traffic volume. However, our in-depth analysis of HTTP traffic (Section 5) finds that this is not the case.

Streaming protocols[1] (RTSP, RTMP, SHOUTcast) account for 5% of the traffic in terms of bytes. We identify RTSP and SHOUTcast using partial DPD parsers, while we identify RTMP's based only on its well-known port. We also find noticeable Voice-over-IP traffic (Skype [7], RTP), about 1.3% of the total bytes.

In order to increase our confidence in the representativeness of our application mix results, we analyzed sampled NetFlow data exported by 10 of the ISP's routers. This data shows that 50% of the traffic comes from TCP port 80. We further compared our results with those from a commercial deep-packet-inspection system deployed at a different network location, finding a close match.

Our analysis of the other traces confirms the findings outlined above. In particular the other traces confirm that our results are not biased by the day-of-week we choose. However, while the HTTP traffic share in the APR trace is about the same, we find slightly more unclassified traffic. We note that the overall P2P traffic decreases somewhat, and shifts from eDonkey to BitTorrent (now 9.3%). Also the fraction of NNTP traffic decreases. On this day it only accounted for 2.2% of the traffic. Our hypothesis is that especially the latter observations reflect day-to-day variations rather than indications of trends, but we will require longer-time measurements to determine this definitively.

We might expect that application usage differs widely between users with different access speeds. Figure 6 shows the application mix seen for different downstream bandwidth rates. Although the mix does vary, the changes are modest, other than for more P2P traffic with higher bandwidths, and much higher NNTP prevalence for the 17000 Kbps class. However, only a small percentage of lines use NNTP, so its contribution to traffic mix can see more variation across different types of lines.

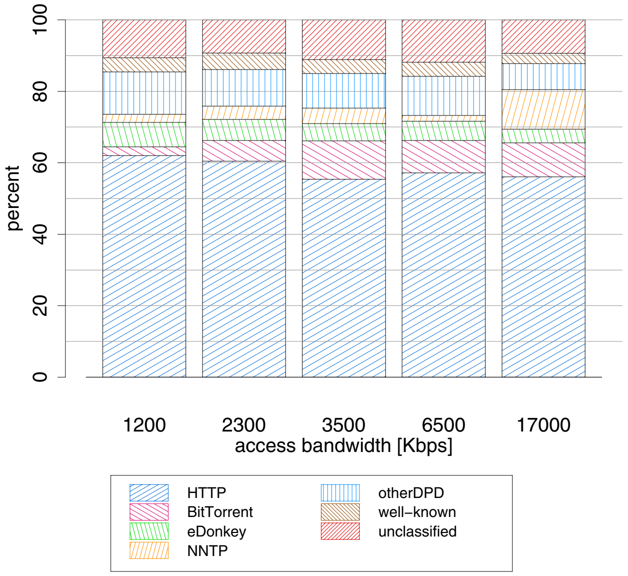

Figure 6: Relative application mix per access bandwidth. Bottom bar is HTTP, top bar unclassified.

However, we do find that lines with higher access bandwidth have a higher utilization in terms of average volume per line. Lines in the 3500 and 6500 Kbps categories contribute about twice as many bytes per line than lines in the 1200 Kbps class, and 17,000 Kbps lines three times more. We also find that general traffic per line is consistent with a heavy-tailed distribution, and the top 2.5% of lines account for 50% of the traffic.

To see if time-of-day effects influence the application mix, we examine the application mix per hour, see Figure 7. We would expect to to observe more bulk downloads and less interactive traffic during off-hour period, which our data confirms. Night-time traffic includes a larger fraction of P2P traffic, though HTTP remains dominant during every time slot. Also, we again note high variability in NNTP due to the small number of lines using it.

In contemporaneous work Erman et al. [15] studied the application mix and HTTP content type of a major US broadband provider in the context of understanding the potential for forward caching. They find that HTTP contributes 61% on average and 68% during the busy-hour to the traffic volume in the downstream direction while P2P only contributes 12%. As such, their results are strikingly similar to our results, strengthening the observation that HTTP is again on the rise and P2P on the decline.

4.2 Application mix of P2P VS. Non-P2P lines

Next we study if the application usage of those lines that frequently use P2P differs from those that do not. We find that roughly 3% of DSL-lines use P2P protocols and that their traffic contribution accounts for 30% of overall volume. If a line uses P2P protocols, they usually also account for most of the line's traffic: 29% BitTorrent and 17% eDonkey. However, HTTP is still popular and is responsible for 23% of transferred bytes. We also note that the fraction of unclassified traffic is higher at 23%, corresponding to roughly 64% of all unclassified traffic. There is hardly any NNTP usage, only 0.6% of bytes.

[1] We do not consider video delivery via HTTP as streaming. We refer to those as progressive HTTP downloads.

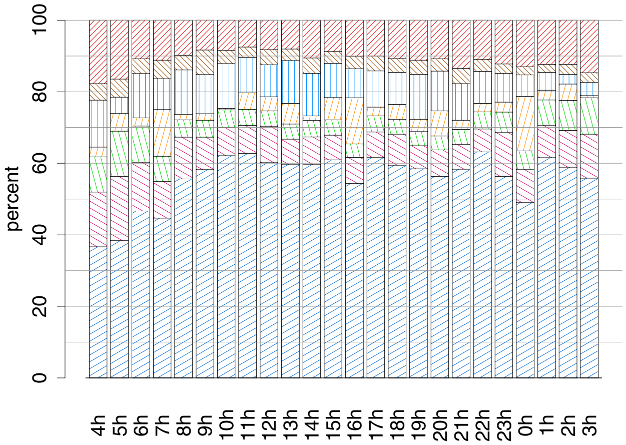

Figure 7: Relative application mix hour-by-hour. Same legend as in Figure 6.

Protocol	V_{PD}/V_D	V_{PD}/V_P
HTTP	97.5%	98.1%
BitTorrent	4.8%	66.1%
eDonkey	36.6%	55.9%
SSL	75.2%	86.1%
NNTP	66.7%	95.3%
RTSP	92.6%	99.1%

Table 3: DPD vs. destination port. V_D is the volume identified by DPD for a given protocol P, V_P is the volume observed on the P's default port(s), and V_{DP} is the intersection of the two (running on P's default port and detected as P).

Non-P2P lines predominantly use HTTP, for which it contributes 72% of their traffic volume, followed by NNTP with 6.5%, with only 5.2% of the traffic unclassified. Streaming services are also more dominant in this group (6.7%).

4.3 Does port-based classification work?

Very often in networking studies it is easier or more tenable to acquire TCP/IP transport information rather than relying on deep packet inspection systems. A significant question concerning the accuracy of such studies regards the degree to which one can soundly infer application protocols based solely on the TCP/UDP port numbers that connections use. Certainly, in adversarial settings, classification based on port numbers has quite limited power, due to the ease by which end systems can vary the ports they use. However, for non-adversarial situations, one might hope to leverage a predominant tendency for applications to indeed stick with the port assigned for their use.

Our DPD-based analysis—which is highly accurate for those applications where we have a full protocol parser, and still potentially quite accurate when we employ only a partial parser—presents an opportunity to assess the accuracy of port-based classification using fairly solid ground truth.

Numerous previous studies have indicated that the advent of P2P has rendered port-based approaches infeasible. Cho et al. [10] found that on Japanese Internet backbone links, 79% of traffic (by bytes) uses unknown ports, and that TCP port 80 contributes only 14% of bytes. In 2004 Karagiannis et al. [26] found P2P traffic increasingly moving away from well-known ports to dynamically negotiated ports. Kim et al. [23] found that port-based detection quality is inversely proportional to the fraction of P2P traffic.

We confirm that for current residential traffic a port-based approach works quite well. Table 3 shows how well a port-based approach would have performed for dominant application layer protocols. For each protocol P, column V_{PD}/V_D is the fraction of the traffic volume observed on P's default port(s) that DPD identifies as P. Column V_{PD}/V_P shows the proportion of the traffic on P's port that would be correctly identified by only inspecting the port number.

We interpret the table as follows. Most of the HTTP traffic (97.5% of bytes) does indeed appear on port 80 (middle column), and when looking at traffic on port 80 we find that 98.1% of those bytes come from HTTP (righthand column). The largest non-HTTP application on port 80 is SHOUTcast, a HTTP-like streaming protocol. We therefore conclude that for our traffic, classifying port 80 traffic as HTTP yields a good approximation for the total volume of HTTP traffic.

NNTP can only be partially identified by its default port (119). About two-thirds of NNTP traffic uses that port, and of the traffic appearing on that port, nearly all (95.3%) is indeed NNTP. From DPD, we know that the remainder uses the well-known HTTP proxy port, 3128. For SSL-based protocols (HTTPS, IMAPS, POP3S, SSMTP, NNTPS) we find roughly 75% using well-known ports. More than 90% of RTSP bytes appear on its default port (554).

The story is vastly different for P2P protocols, however. Since many institutions try to block P2P traffic with port-based filters, most P2P protocols have evolved to use non-standard, dynamically negotiated ports. Still, one third of the detected eDonkey traffic uses its well-known ports, and finding traffic on either those ports or on the BitTorrent ports generally means that the traffic is indeed caused by those protocols. (Interestingly, we find that 3% of BitTorrent traffic appears on *eDonkey* ports.)

4.4 Traffic symmetry

A common assumption regarding residential traffic is that the downstream dominates the upstream, i.e., most bytes are transfered to the local side. Indeed, this assumption has shaped—and is ingrained in—the bandwidth allocations of ADSL and cable broadband offerings. In addition, the prevalence of incoming connections affects the feasibility of carrier-grade network-address-translation (NAT).

In our datasets, we observe that most bytes appear in connections originated locally, with only 10% due to connections originated remotely. The largest fraction of incoming traffic is unclassified (33% of bytes), significantly higher than for outgoing connections, and with P2P the most significant contributor by volume (28% BitTorrent, 17% eDonkey). Voice-over-IP and streaming protocols also contribute significant volume to incoming connections (10%). Incoming FTP data connections for active FTP sessions account for just over 1% of bytes in incoming connections. Finally, we find that very few lines offer "classic" Internet services like SMTP or HTTP, nor did they appear significantly involved in DDoS or scanning activity (according to Bro's scan detector).

When looking at the number of bytes transfered upstream and downstream, i.e., the symmetry of traffic, we find that 85% of all bytes come downstream, i.e., the asymmetry assumption does hold (though likely bandwidth asymmetry helped shape this). This proportion is much higher than seen in the Japanese backbone studies [19, 9], which found only 55% of volume was downstream. However, they found P2P dominated their traffic mix, thus contributing to symmetry. For our traffic, we find that for P2P applications only 59% of bytes come downstream, yielding an upload/download "share-ratio" of $41/59 \approx 0.7$—still resulting in less symmetry than seen in the Japanese studies.

5. HTTP USAGE

As HTTP dominates the traffic in our datasets, we now examine it more closely to characterize its usage. A basic question concerns what has led to its resurgence in popularity versus P2P traffic, with two possible reasons being *(i)* HTTP offers popular high-volume content, e.g., [8, 42], and/or *(ii)* HTTP serves as a transport protocol for other application layer protocols, including possibly P2P [50, 3]. We find that 25% of all HTTP bytes carry Flash Video, and data exchanged via RAR archives contributes another 14%. Thus, clearly much of HTTP's predominance stems from its use in providing popular, high-volume content. We further find that in terms of volume, HTTP is *not* significantly used for tunneling or P2P downloads.

Many facets of HTTP usage have seen extensive study, as thoroughly surveyed by Krishnamurthy and Rexford [31]. Some studies have focused on understanding user behavior [4, 5, 11], while others have examined changes in content [53] and the performance of web caching [1, 5, 16]. Other work has looked at media server workloads regarding file popularity and temporal properties, such as in terms of live media streams collected from a large CDN [49], and file reference characteristics and user behavior of a production video-on-demand system in large-scale use [55].

More recently, various efforts have aimed at understanding from passive measurements how the rapid advent of "Web 2.0" applications has changed HTTP traffic patterns [44], as well as Web-based applications such as YouTube [20, 57] and online social networks [21, 36]. Others have employed active probing to study specific features of such applications [8].

Sites like alexa.com employ user-installed toolbars to track the popularity of various Web sites across demographic groups. They find that google.com, yahoo.com, youtube.com, and facebook.com currently rank among the most popular sites in terms of number of visits. In contrast, in this study we analyze popularity in terms of traffic volume.

5.1 Content Type Distribution

We use Bro's HTTP analyzer to parse the anonymized HTTP headers and compute the size of each HTTP request/response pair. To identify the content types of objects, we both examine the HTTP Content-Type header and analyze the initial part of the HTTP body using libmagic. We find more than 1,000 different content-types in HTTP headers. Surprisingly, the results of these two approaches often disagree: 43% of all HTTP bytes (28% of requests) exhibit a mismatch. Some disagreements are minor and easy to resolve. For example, in the absence of a standardized MIME type representation we can find several different strings used for the same type. We also often see generic use of application/octet-stream as Content-Type. In other cases, the sub-type differs: for example, the Content-Type header may specify "image/gif," while libmagic yields "image/jpeg".

When Content-Type and libmagic disagree, we try to identify the most likely "true" content type by using heuristics. We start by normalizing the content types and giving priority to libmagic for those content types with well-known formats, e.g., most image and video types. For other formats, we manually examine the mismatches and pick the most likely resolution. We report mismatches we could not resolve as "x/x" in our results, and generic or unidentified content types, such as application/octet-stream, as "n/n". All in all, our analysis illustrates the need for considerable caution when basing an assessment of content types solely on the Content-Type header.

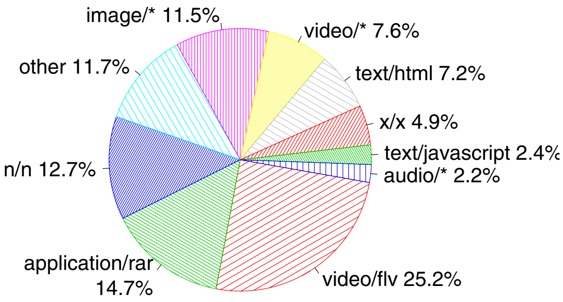

Figure 8: Top content-types for HTTP by bytes for trace SEP.

Figure 8 shows a pie chart of the distribution of bytes per content type from the SEP trace. The most common content-type by volume is Flash Video (video/flv)—the format used by sites such as youtube.com and many news sites—which contributes 25% of the bytes. This is followed by the archive format RAR (application/rar), which accounts for 15% of HTTP traffic.

The unknown or unidentifiable content-types together account for 18% of the HTTP traffic. We find that a significant portion of this traffic reflects automated software updates, as 14% of the unidentifiable bytes come from a single software update site. Image types (GIF, PNG, and JPEG) contribute 11.4% of bytes, while video types other than Flash account for only 7.6%.

During the night we observe a higher fraction of RAR objects and unknown objects, while the relative popularity of HTML and image types decreases. This indicates that the former arise due to bulk transfers rather than interactive browsing.

The general content-type distribution is essentially unchanged when considering the APR trace. However, the fraction of non-Flash Video (video/flv) video content increases (to 9%), while audio content decreases. Moreover, the fraction of unknown content types from the automated software site falls to 7.5% in APR. We also confirmed that the presented results are not subject to day-of-week effects by comparing them with results from WEEK trace.

Drawing upon recent data from a major US broadband provider, Erman et al. [15] also report similar content type distributions. They find that video content corresponds to 32% of HTTP traffic, and compressed file downloads, e.g., RAR, for 16% of traffic.

When separating lines with and without P2P protocol usage, we find that the content-type distribution for non-P2P lines closely matches the overall one. However, lines that use P2P have a smaller fraction of Flash Video (20%) and RAR archives (11%), and a larger fraction of unidentified content-types (25%) We note that 28% of this unidentified traffic is served from CDNs and 8% from a Direct Download Provider.

5.2 Distribution Across Domains

Next we examine the distribution across domains, presenting the results for the SEP trace in Table 4. We base our analysis on extracting the second-level domain from the HTTP Host header. We find that the byte distribution per domain fairly closely matches a Zipf distribution, per Figure 9. The top 15 domains account for 43% of all HTTP bytes. Since Flash Video is the most voluminous

Rank	Domain	Fraction of Traffic
1	Direct Download Provider	15.3%
2	Video portal	6.1%
3	Video portal	3.3%
4	Video portal	3.2%
5	Software updates	3.0%
6	CDN	2.1%
7	Search engine	1.8%
8	Software company	1.7%
9	Web portal	1.3%
10	Video Portal	1.2%

Table 4: Top domains (anonymized) for trace SEP

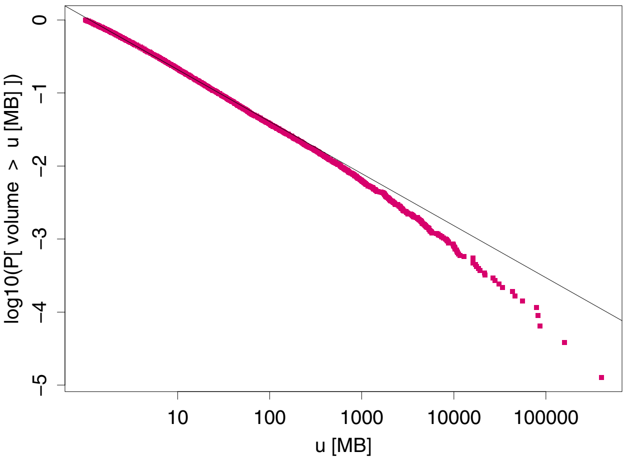

Figure 9: CCDF of HTTP volume per domain, for domains with >1 MB of total traffic for trace SEP.

content-type, it is not surprising to find sites offering videos among the top domains, and indeed most of the traffic to/from these video portals has type video/flv. A Direct Download (DDL) provider also accounts for a significant fraction of HTTP traffic. These DDL providers (also called "One-click providers") host large files for their customers. When a user uploads a file, they receive a (encoded) URL that provides subsequent access for downloading the file. Users can then distribute the URLs to friends or share them in online forums. About 16% of the HTTP traffic involves Direct Download providers, with one provider in particular heavily dominating this traffic (93% of DDL traffic volume). Nighttime traffic exhibits a strong shift towards DDL sites; they account for 24% of HTTP bytes during the 4 AM hour. DDL providers also originate almost 90% of all application/rar bytes.

Similar results hold for the APR trace, with only some changes in the lower ranks. Given the small difference in volume for these domains, we attribute such changes to normal day-to-day differences rather than long-term trends.

5.3 User-Agent Popularity

To assess the popularity of different types of web clients, we extract the User-Agent headers from the HTTP requests, group them into broader categories, and then rank these categories by transfered volume. We group user-agents that we cannot classify, and requests lacking a User-Agent header, as "Unclassified". Table 5 shows the results. We can attribute more than 82% of HTTP traffic to traditional Web browsers, with Firefox and Internet Explorer each having a share of approximately 35% each, while Safari and Opera only contribute 6% and 3% of HTTP traffic. We also crosschecked with the results described above to verify that a large fraction of the traffic due to these traditional web clients involves well-known domains. We do not see a significant volume contribution by advertised P2P clients. Further, even if such P2P traffic falls into the "Unclassified" bin, it represents little in terms of overall volume. Therefore, in our dataset we do not observe a large proportion of P2P systems running on top of HTTP, unless they employ mimicry of well-known browsers, and also manipulate content types and domains.

Rank	User-agent	Fraction of Traffic
1	Firefox 3	24.6%
2	MSIE 7	20.4%
3	MSIE 6	13.6%
4	Firefox 2	11.9%
5	Unclassified	5.5%
6	Safari	4.3%
7	Network libraries	4.0%
8	Opera	2.8%
9	Streaming clients	2.5%
10	Download managers	1.6%

Table 5: Top user-agents by volume

6. TRANSPORT PROTOCOL FEATURES

We next delve into exploring which of the various TCP options and configurations we see in actual use. Doing so allows us to calibrate our expectations with regard to TCP throughput performance, which we then explore in Section 7. We limit our analysis to connections that transfer some actual TCP payload, which excludes a large number of unproductive connections caused by backscatter, scanning, or other establishment failures. The excluded connections contribute about 0.1% of all bytes, but amount to 35% of all connections.

To compare our results to previous studies, we need to determine the usage of options on a *per-host* basis. However, unlike previous studies we expect to find our dataset rife with NATs (within the DSL customers' home networks). Therefore, isolating individual hosts presents a challenge, since multiple hosts may share a single DSL line. To address this difficulty, we assess option usage in two ways. The first technique considers each DSL line identifier as a single host, and attributes any options observed in packets associated with the line to that host. Doing so obviously undercounts the number of hosts. For the second approach, we assume that each distinct TCP option set represents a distinct host. This likely overcounts the number of hosts, so by employing both strategies we can bracket the ranges for host-based use of various TCP options.

Window Scaling

Window Scaling enables efficient data transfer when the bandwidth-delay product exceeds 64 KB. We find window scaling advertisements in 32–35% of the SYNs in our dataset, with 4% of the connections failing to successfully negotiate the use of window scaling. When focusing on only connections transferring more than 50 KB, we find only a small change, with 34–38% successfully negotiated window scaling. Finally, we observe that 45–62% of the hosts in our datasets advertise window scaling (across traces and across our under- and over-estimates for host count). In contrast, Medina et al. reported that 27% of the observed client hosts advertised window scaling in early 2004 [34]. Of those advertisements, 97% were found to be zero (i.e., the client advertises the *ability* to scale windows, but not the desire to do so). In our dataset, we do not find a predominance of scale factors of zero; most scale factors are in fact non-zero, and cover a wide range. Even with our rough counting of hosts, we can see that use of larger windows has become more routine over the past 5 years.

TCP Timestamp

Timestamps help TCP to compute more accurate round-trip time estimates, and serve to disambiguate old packets from new ones in very high-speed transfers. We observe timestamps advertised in 11–12% of the connections in our dataset, with 8% of the connections ultimately negotiating their use. We further observe that 21–39% of the hosts (across traces and host-counting methods) advertise timestamps, versus 22% as observed by Medina et al. [34]. Further, Veal [51] probed a variety of web servers and concluded that 76% of the servers will use timestamps when requested by the client.

Selective Acknowledgment (SACK)

SACK facilitates more effective recovery from lost data segments. We find that 97% of connections in our dataset advertise support for SACK, with 82% of the connections successfully negotiating its use. In addition, we observe that roughly 9% of the connections that negotiate SACK have at least one instance whereby a receiver uses SACK to report a discontinuous arrival (either due to loss or reordering). Finally, we observe 82–94% of the hosts in our dataset advertising SACK (across traces and host-counting strategies). Medina et al. reported that in 2004 88% of the clients attempted to use SACK [34], and that active probing found roughly 69% of successfully contacted servers supported SACK.

Maximum Segment Size (MSS)

The MSS governs the largest data segment a TCP sender will transmit. Across all TCP traffic, we find advertised values in the 1300–1460 byte range in 98% of the connections. These values arise from the very common 1500 byte Ethernet MTU, minus space required for TCP/IP headers, as well as space for additional tunneling headers.

Explicit Congestion Notification (ECN)

ECN enables routers to signal conditions of congestion without necessarily employing packet drops. We find virtually no support for ECN, observing only a handful of hosts (no matter how they are counted) advertising support for it in their SYN packets.

Summary

We find that usage of performance improving TCP options varies considerably. SACK enjoys widespread deployment and use; window scaling is quite common in terms of both support and effective (non-zero) employment; ECN sees almost no use.

7. PERFORMANCE/PATH CHARACTERISTICS

We now turn our attention to factors that affect the performance that users experience—spanning network effects, transport protocol settings, application behavior, and home networking equipment.

In a previous study, Dischinger et al. [12] recently used active measurements to probe 1,900 broadband host from 11 major providers in Europe and North America. They found that the last-mile predominates as the performance bottleneck and induces high jitter in the achievable throughput. They also found that broadband links have large queuing buffers of several hundred to several thousand ms, and that 15% of last-mile RTTs exceed 20 ms. However, they do not compare access versus remote contributions to RTT. While their study covers a more diverse set of hosts, our approach leverages capturing all activity of residential hosts.

Jiang and Dovrolis [25] estimated TCP RTTs from passive measurements of unidirectional packet data using SYN-SYN/ACK-ACK handshakes and a slow-start based approach. They found that 90–95% of connections have RTTs < 500 ms at various academic links. Aikat et al. [2] examined the variability of RTTs *within* a connection using data from the University of North Carolina. They report that a striking 15% of TCP connections have median RTTs >1 s. However, their analysis does not take delayed ACKs into account. Fraleigh et al. [18] analyzed packet level traces from the Sprint backbone from 2001, finding that the median RTT never exceeded 450 ms across their 9 traces. Only 3 traces had median RTTs >300 ms, while 6 traces had median RTTs of <50 ms.

Siekkinen et al. [47, 48] analyzed performance limitations experienced by ADSL users using passive measurements of approximately 1,300 DSL clients. They found that most users do not utilize the available bandwidth, and that most traffic is application-limited—particularly for P2P applications, which often actively limit the transfer rate. Network limitations like congestion or TCP windows only affected a small number of transfered bytes.

Zhang et al. [56] analyzed Internet flow traces from various access, peering, and regional links within a Tier-1 provider in 2002 to understand from where performance bottlenecks arose. They found that the most frequent performance limitations were network congestion and advertised receiver window sizes.

Given this context, we now turn to an analysis of performance limitations in our 2008/2009 residential traces.

7.1 TCP performance limitations

TCP's advertised window can have a significant impact on performance, as the window must equal or exceed the bandwidth-delay product for a connection to fully utilize the network path's capacity. If too small, the data sender must pause and wait for ACKs before sending additional data, whereas with a large enough window data can steadily stream. We use the access bandwidth to compute bandwidth-delay products for all connections and find that in the downstream direction, 44% of all connections that transferred at least 50 KB have a bandwidth-delay product that exceeds the maximum advertised window, but this proportion drops to 15% for the upstream direction (which due to bandwidth asymmetry does not require as large of a window).

We find that the maximum advertised window observed per connection tends to be fairly small, with a median across all connections in our dataset of 64 KB. Interestingly, the use of window scaling does not significantly affect advertised window size; the median for such connections increases only slightly, to 65–67 KB. However, the 75^{th} percentile for connections with window scaling is roughly 190 KB, as opposed to the limit of 64 KB imposed by a lack of window scaling.

We note, however, that connections with small advertised windows might in fact have their performance more significantly limited by TCP's response to congestion. We assess loss/reordering events by checking whether a sender ever fails to send monotonically increasing sequence numbers. Loss plays a key role in achiev-

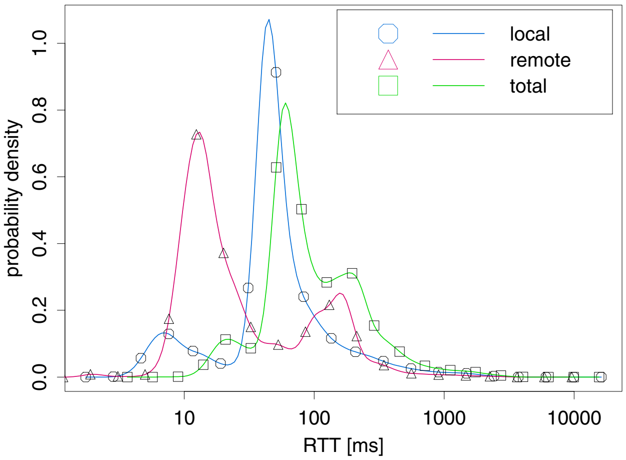

Figure 10: TCP round trip times for trace SEP.

able TCP performance [33, 39], and TCP can confuse reordering for loss [6], causing it to perform congestion control actions that hinder performance. We find that roughly 10% of TCP connections experience such events. Furthermore, 33% of connections that transfer >50 KB experience loss or reordering. These rates are consistent with the observation that 8% of connections that negotiated SACK actually exchanged a SACK block, as did 30% of connections that transfered at least 50 KB. In addition, we find that about 1% of connections required SYN retransmissions in order to successfully establish.

Finally, we find that at some points the receiver's advertised window "closes" (drops to zero). Generally, this behavior indicates that the receiving application has failed to drain the operating system's TCP buffer quickly enough, and therefore TCP must gradually advertise less available buffer. As the advertised buffer space decreases, the sender's ability to keep enough data in flight to fully fill the network path diminishes. We find that for 4% of the downstream connections the advertised window drops to zero, while this phenomenon occurs for 3% of the upstream connections.

7.2 Round-trip-times (RTT)

We gathered our measurements at the ISP's broadband access router, which is the first IP router that traffic from the local hosts encounters. We can therefore divide the end-to-end RTT that the residential connections experience into a local component, measured from our monitor to the end system and back, and a remote component, from our monitor over the wide-area Internet path to the host at the other end of the connection.

We estimate TCP RTTs using the connection setup handshake (SYN, SYN/ACK, ACK) [25], ignoring connections with SYN or SYN/ACK retransmissions, and connections in which the final ACK carries data (which can indicate that an "empty" ACK has been lost). Figure 10 shows the smoothed probability distribution of the RTTs. We found it quite surprising to observe that in many cases the local RTT exceeds the remote RTT, i.e., *the time to simply get to the Internet dominates over the time spent traveling the Internet*.

The difference manifests itself throughout most of the distribution. For example, the median, 75^{th}, 90^{th}, and 99^{th} percentiles of the local RTTs are all substantially larger than their remote counterparts, and we find that 1% of local RTTs exceed 946 ms, while for remote RTTs the corresponding delay quantile is only 528 ms. The 99^{th} percentile of total RTT is 1328 ms, with a 90^{th} percentile of 278 ms and a median of 74 ms. While RTTs are often fairly low, we also observe several cases for which the local RTT reaches values in the 2–6 sec range and beyond.

Local RTTs follow a bi-modal distribution, with one peak at 7 ms and another, larger one at 45 ms. This is consistent with the fact that most DSL lines use *interleaving* [28, 24], which increases delay, while a smaller number of the DSL lines use the "fast path" feature, which does not contribute any significant delay.

Remote RTTs exhibit three modes, at 13 ms, 100 ms, and 160 ms, with the latter two somewhat blurred in the plot. Likely these modes reflect the geographic distribution of remote hosts (e.g., Europe, US East coast, US West coast).

7.3 Impact of Access Technology

The not infrequent appearance of large local RTTs led us to investigate their possible cause. Typically, large RTTs reflect large queuing delays. Indeed, Dischinger et al. [12] found that residential broadband links can exhibit queuing delays of several seconds when a DSL line is fully utilized.

Manual inspection of sequence number plots of some connections with large RTTs (>1000 ms) indeed shows such queues building up. We therefore checked whether those lines utilized their access bandwidth during these events. We found, however, that this is *not* always the case: while we often see significant traffic on these DSL lines, they do not necessarily utilize their upstream or downstream bandwidth fully. A more detailed manual analysis reveals other effects, too, such as RTTs within a connection suddenly jumping by an order of magnitude.

One possible cause could be wireless links in users' homes, given the plausibility of a large fraction of broadband users employing 802.11 wireless to connect their computers to the Internet. In densely populated, urban areas, users often "see" numerous wireless networks, and therefore can experience non-negligible contention for the medium.

To assess this hypothesis, we used several DSL links (1x 8000 Kbps and 3x 2000 Kbps downstream) to estimate upstream and downstream throughput and queuing delays using active measurements done with the nettest tool.

Using wired connections, we are able to fully utilize the DSL link's bandwidth. When using wireless connections, the achieved throughput often drops to 400–1000 Kbps. In both cases, we experience queuing delays of several seconds. However, the reduced throughput when using wireless access causes the queue to start building up at lower rates. In addition, while we were unable to saturate the 8000 Kbps link[2] with a wired connection, and therefore had low or imperceptible queuing delay, using wireless the queuing delay still rose to several seconds.

These results show that wireless networks can have a significant impact on the achievable throughput. In particular, 11 Mbps wireless cards and wireless connections in areas with many other wireless senders, and/or with poor link quality, face significant performance degradation. We verified that wireless connections, in uncontested environments and with current 54 Mbps wireless devices, offer the same throughput and queuing delay as wired connections.

7.4 Achieved Throughput

Next, we examine how many lines actually utilize their available access bandwidth across a substantial period of time. We count the number of transfered bytes per DSL line across 1 sec bins and then calculate the throughput per bin. We call a line *active* if it sent at

[2] Due to a bottleneck in the Internet between the DSL line and the measurement server

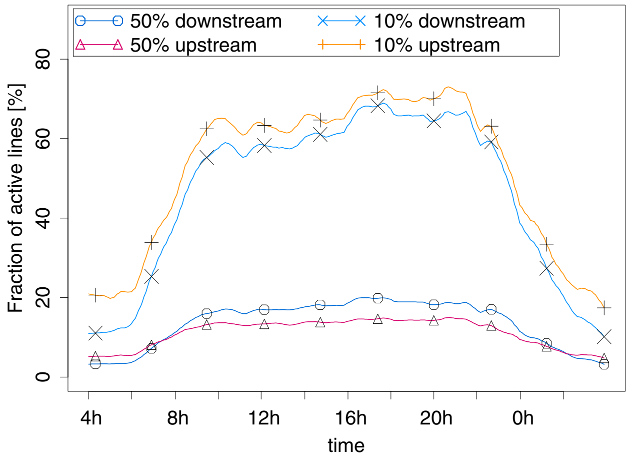

Figure 11: Fraction of active lines using 50%/10% of their available upstream/downstream bandwidth at least once per 5 minute bin (smoothed).

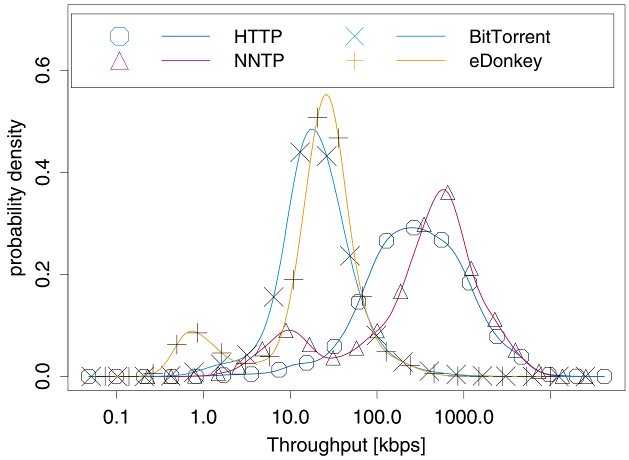

Figure 12: Achieved throughput of flows with size >50 KB by application protocol.

least one packet, or received at least 5 KB, in each bin. We then compare these results to the available access bandwidth for each DSL line, determining how many lines exceeded 10% or 50% of their bandwidth for at least one second during a given 5 min period.

Figure 11 shows that most lines use only a small fraction of their bandwidth. Less than a quarter of the active lines exceed 50% of their bandwidth for even *one second over a 5 minute* time period. However, during the day we observe 50–60% of active lines achieving at least a 10% bandwidth utilization. These results are consistent with findings from Siekkinen et al. [47].

To gauge whether there is a principle network limitation on obtainable performance, we analyzed the achieved throughput per unidirectional flow, distinguishing flows by their application-layer protocol. To do so, we constructed the equivalent of NetFlow data from our packet traces, using an inactivity timeout of 5 sec. Figure 12 shows the distribution of the achieved throughput for these flows, given they transfered at least 50 KB. We observe that HTTP and NNTP achieve throughputs an order of magnitude larger than those for P2P and unclassified traffic (note the logarithmic scale). We also find that other DPD-classified traffic, as well as traffic on well-known ports, achieves throughput similar to that for HTTP and NNTP. These findings suggest that a portion of unclassified traffic is likely P2P. For flows with more data (> 500 KB), the difference in throughput actually increases slightly. Furthermore, we see that the throughput for all of these larger flows increases as well.

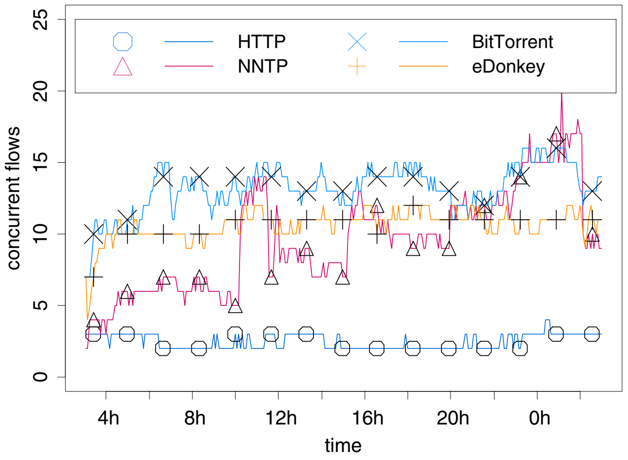

Figure 13: Number of mean parallel flows with size >50 KB per application protocol and line (in 5 min bins).

Some P2P applications open multiple parallel connections in order to download content from several peers at the same time. To analyze this behavior, we investigated the mean number of parallel flows per application; see Figure 13. The plot confirms that P2P protocols use more parallel flows than HTTP. However, the difference is substantially smaller than the difference in achieved throughput. As such, the upstream capacity of other peers combined with application restrictions effectively throttles P2P transfers. Interestingly, we find that NNTP behaves similar to the P2P protocols, using a larger number of parallel flows. This is most likely a result of users using a customized NNTP client for bulk download, rather than a traditional newsgroup reader.

8. SUMMARY

In this paper we have studied residential broadband Internet traffic using anonymized packet-level traces augmented with DSL session information. Our data covers more than 20,000 customers from a major European ISP. Our initial exploration of the datasets unearthed a number of surprises that alter some of our mental models of such traffic.

We started with DSL level characteristics, examining session durations, their termination causes, and the number of concurrent sessions. Session durations are surprisingly short, with a median duration of only 20–30 minutes, while we would have expected several hours to days. Our termination cause analysis turned up that most sessions end due to termination from the user end, which we attribute to default router configurations based on former timed contracts. As a consequence, IP addresses are reassigned frequently, with up to 4% of addresses assigned more than 10 times a day. This indicates that the use of IP addresses as host identifiers can prove quite misleading over fairly short time scales.

Next we examined usage of different applications and their impact on overall traffic. We observed that P2P no longer dominates in terms of bytes. Rather, HTTP once more carries most of the traffic, by a significant margin (>50%). While we used Bro's DPD [13] to identify applications, we also examined the efficacy we would obtain from a simple, purely port-based approach for application classification, finding it works quite well for our datasets, due to the prelevance of HTTP, NNTP, and streaming applications. It does not work as well for P2P, however.

To understand why HTTP is again the dominant application, we looked at a number of facets of its usage. We found that Flash Video, the format used by video portals such as `youtube.com` and news sites, contributes 25% of all HTTP traffic, followed by RAR archives. The latter are mostly downloaded from Direct Download providers associated with file-sharing. We did not find a significant share of HTTP traffic attributable to P2P protocols or application protocols using HTTP as a transport protocol.

We note that a number of these results agree with those of Erman et al.'s contemporaneous study [15], suggesting that the trends are representative for a significant fraction of the Internet.

We analyzed transport protocol characteristics in terms of TCP options. We found that window scaling and SACK have become more popular since Medina et al.'s previous study [34], with SACK employed by more than 90% of clients. Window scaling is also often used, but does not in fact result in larger advertised receiver windows.

We assessed performance and path characteristics of TCP connections, noting that most DSL lines fail to utilize their available bandwidth. Examining TCP round-trip-times, we found that for many TCP connections the access bandwidth-delay product exceeds the advertised window, thus making it impossible for the connection to saturate the access link. Our RTT analysis also revealed that, surprisingly, the latency from the DSL-connected host to its first Internet hop dominates the WAN path delay. This discrepancy can however be explained by ADSL's interleaving mechanism. We found that WAN delays are often as little as 13 ms, but local RTTs not infrequently exceed 1000 ms, a phenomenon that is likely caused by the use of wireless equipment in the customers home and ensuing contention on the wireless hop. We also observed that connections from client-server applications, like HTTP and NNTP, achieve an order of magnitude higher throughput per flow than P2P connections.

In future work we plan to explore application characteristics and network capacity issues in more depth, as well as to obtain longitudinal data to perform trend analysis. Furthermore, we plan to investigate interactive and real-time sensitive traffic such as VoIP and gaming. Although these do not yet contribute a significant number of bytes, these protocols are important for perceived Quality-of-Service by customers.

9. ACKNOWLEDGEMENTS

This work was supported in part by US National Science Foundation grants CNS-0722035, CNS-0831535, and a grant from Deutsche Telekom Laboratories, Berlin. We would like to thank the anonymous reviewers for numerous valuable comments.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors or originators and do not necessarily reflect the views of the National Science Foundation.

10. REFERENCES

[1] Abrams, M., Standridge, C. R., Abdulla, G., Williams, S., and Fox, E. A. Caching Proxies: Limitations and Potentials. In *Proc. World Wide Web Conference* (1995).

[2] Aikat, J., Kaur, J., Smith, F. D., and Jeffay, K. Variability in TCP Round-trip Times. In *Proc. ACM Internet Measurement Conference* (2003).

[3] Anderson, N. P2P Traffic Drops as Streaming Video Grows in Popularity. `http://arstechnica.com/old/content/2008/09/p2p-traffic-drops-as-streaming-video-grows-in-popularity.ars`.

[4] Arlitt, M., and Williamson, C. Internet Web Servers: Workload Characterization and Implications. *IEEE/ACM Trans. Networking* (1997).

[5] Barford, P., Bestavros, A., Bradley, A., and Crovella, M. Changes in Web Client Access Patterns: Characteristics and Caching Implications. *World Wide Web 2* (1999).

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

[7] Bonfiglio, D., Mellia, M., Meo, M., Rossi, D., and Tofanelli, P. Revealing Skype Traffic: When Randomness Plays With You. In *Proc. ACM SIGCOMM* (2007).

[8] Cha, M., Kwak, H., Rodriguez, P., Ahn, Y.-Y., and Moon, S. I Tube, You Tube, Everybody Tubes. In *Proc. ACM Internet Measurement Conference* (2007).

[9] Cho, K., Fukuda, K., Esaki, H., and Kato, A. The Impact and Implications of the Growth in Residential User-to-User Traffic. In *Proc. ACM SIGCOMM* (2006).

[10] Cho, K., Fukuda, K., Esaki, H., and Kato, A. Observing Slow Crustal Movement in Residential User Traffic. In *Proc. ACM CoNEXT* (2008).

[11] Crovella, M., and Bestavros, A. Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes. *IEEE/ACM Trans. Networking* (1997).

[12] Dischinger, M., Haeberlen, A., Gummadi, K. P., and Saroiu, S. Characterizing Residential Broadband Networks. In *Proc. ACM Internet Measurement Conference* (2007).

[13] Dreger, H., Feldmann, A., Mai, M., Paxson, V., and Sommer, R. Dynamic Application-Layer Protocol Analysis for Network Intrusion Detection. In *Proc. Usenix Security Symp.* (2006).

[14] Endace Measurement Systems. `http://www.endace.com/`, 2008.

[15] Erman, J., Gerber, A., Hajiaghayi, M. T., Pei, D., and Spatscheck, O. Network-aware Forward Caching. In *World Wide Web* (2009), pp. 291–300.

[16] Feldmann, A., Caceres, R., Douglis, F., Glass, G., and Rabinovich, M. Performance of Web Proxy Caching in Heterogeneous Bandwidth Environments. In *Proc. IEEE INFOCOM* (1999).

[17] Feldmann, A., Greenberg, A., Lund, C., Reingold, N., Rexford, J., and True, F. Deriving Traffic Demands for Operational IP Networks: Methodology and Experience. *IEEE/ACM Trans. Networking 9* (2001).

[18] Fraleigh, C., Moon, S., Lyles, B., Cotton, C., Khan, M., Moll, D., Rockell, R., Seely, T., and Diot, S. C. Packet-Level Traffic Measurements from the Sprint IP backbone. *IEEE Network Magazine 17*, 6 (2003).

[19] Fukuda, K., Cho, K., and Esaki, H. The Impact of Residential Broadband Traffic on Japanese ISP backbones. *ACM Comp. Comm. Review 35*, 1 (2005).

[20] Gill, P., Arlitt, M., Li, Z., and Mahanti, A. YouTube Traffic Characterization: A View From the Edge. In *Proc. ACM Internet Measurement Conference* (2007).

[21] Golder, S., Wilkinson, D., and Huberman, B. A. Rhythms of Social Interaction: Messaging within a Massive Online Network. In *International Conference on Communities and Technologies* (2007).

[22] Henderson, T., Kotz, D., and Abyzov, I. The Changing Usage of a Mature Campus-wide Wireless Network. In *Proc. ACM MOBICOM* (2004).

[23] Hyun-chul, K., Claffy, K., Fomenkov, M., Barman, D., Faloutsos, M., and Lee, K. Internet Traffic Classification Demystified: Myths, Caveats, and the Best Practices. In *Proc. ACM CoNEXT* (2008).

[24] ITU. ADSL standards ITU G.992.1-5.

[25] Jiang, H., and Dovrolis, C. Passive Estimation of TCP Round-Trip Times. *ACM Comp. Comm. Review 32*, 3 (2002).

[26] Karagiannis, T., Broido, A., Brownlee, N., Claffy, K. C., and Faloutsos, M. Is P2P dying or just hiding? In *Proc. IEEE GLOBECOM* (Nov 2004).

[27] Karagiannis, T., Rodriguez, P., and Papagiannaki, K. Should Internet Service Providers Fear Peer-Assisted Content Distribution? In *Proc. ACM Internet Measurement Conference* (2005).

[28] Kitz.co.uk. Interlaving Explained. `http://www.kitz.co.uk/adsl/interleaving.htm`.

[29] Kotz, D., and Essien, K. Characterizing Usage of a Campus-wide Wireless Network. Tech. rep., Dept. of Comp.Sci, Dartmouth College, Mar 2002.

[30] Kotz, D., and Essien, K. Analysis of a campus-wide Wireless Network. *Wireless Networks 11*, 1-2 (2005), 115–133.

[31] Krishnamurthy, B., and Rexford, J. *Web Protocols and Practice*. Addison-Wesley, 2001.

[32] L7filter. Application Layer Packet Classifier for Linux. `http://l7-filter.sourceforge.net/`.

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

[34] Medina, A., Allman, M., and Floyd, S. Measuring the Evolution of Transport Protocols in the Internet. *ACM Comp. Comm. Review 35*, 2 (2004).

[35] Moore, D., Shannon, C., and claffy, k. Code-Red: A Case Study on the Spread and Victims of an Internet Worm. In *Proc. ACM Measurement Workshop* (2002), pp. 273–284.

[36] Nazir, A., Raza, S., and Chuah, C.-N. Unveiling Facebook: A Measurement Study of Social Network Based Applications. In *Proc. ACM Internet Measurement Conference* (2008).

[37] netzikon. Zwangstrennung (in German). `http://netzikon.net/lexikon/z/dsl-zwangstrennung.html`, 2009.

[38] OECD. OECD Broadband Portal. `http://www.oecd.org/sti/ict/broadband`, 2008.

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

[40] Parker, A. CacheLogic: P2P Media Summit.

[41] Paxson, V. Bro: A System for Detecting Network Intruders in Real-Time. *Comuter Networks 31*, 23–24 (1999).

[42] Rapidshare. `http://www.rapidshare.com/`, 2009.

[43] Rigney, C., Willens, S., Livingston, Rubens, A., Merit, Simpson, W., and Daydreamer. Remote Authentication Dial In User Service (RADIUS). RFC 2865, 2000.

[44] Schneider, F., Agarwal, S., Alpcan, T., and Feldmann, A. The New Web: Characterizing AJAX Traffic. In *Proc. Passive and Active Measurement Conference* (2008).

[45] Schulze, H., and Mochalski, K. Ipoque: Internet Study 2007.

[46] Schulze, H., and Mochalski, K. Ipoque: Internet study 2008/2009.

[47] Siekkinen, M., Collange, D., Urvoy Keller, G., and Biersack, E. W. Performance Limitations of ADSL Users: A Case Study. In *Proc. Passive and Active Measurement Conference* (2007).

[48] Siekkinen, M., Urvoy Keller, G., Biersack, E. W., and Collange, D. A Root Cause Analysis Toolkit for TCP. *Comuter Networks 52*, 9 (Jun 2008).

[49] Sripanidkulchai, K., Maggs, B., and Zhang, H. An Analysis of Live Streaming Workloads on the Internet. In *Proc. ACM Internet Measurement Conference* (2004).

[50] Tomlinson, D. Plusnet report: More Record Breaking Streaming and the Latest iPlayer News. `http://community.plus.net/blog/2008/07/17/more-record-breaking-streaming-and-the-latest-iplayer-news/`.

[51] Veal, B., Li, K., and Lowenthal, D. New Methods for Passive Estimation of TCP Round-Trip Times. In *Proc. Passive and Active Measurement Conference* (2005).

[52] Wikipedia. Zwangstrennung (in German). `http://de.wikipedia.org/w/index.php?title=Zwangstrennung&oldid=56733242`, 2009.

[53] Wills, C., and Mikhailov, M. Studying the impact of more complete server information on Web caching. In *Proc. of the 5th International Web Caching and Content Delivery Workshop* (2000).

[54] Xie, Y., Yu, F., Achan, K., Gillum, E., Goldszmidt, M., and Wobber, T. How Dynamic are IP Addresses? In *Proc. ACM SIGCOMM* (2007).

[55] Yu, H., Zheng, D., Zhao, B. Y., and Zheng, W. Understanding User Behavior in Large-Scale Video-on-Demand Systems. *ACM SIGOPS Operating System Review 40*, 4 (2006).

[56] Zhang, Y., Breslau, L., Paxson, V., and Shenker, S. On the Characteristics and Origins of Internet Flow Rates. In *Proc. ACM SIGCOMM* (2002).

[57] Zink, M., Suh, K., Gu, Y., and Kurose., J. Watch Global, Cache Local: YouTube Network Traces at a Campus Network - Measurements and Implications. In *Proc. ACM Multimedia Computing and Networking* (2008).