Understanding Internet Traffic Streams: Dragonflies and Tortoises

Nevil Brownlee, CAIDA, SDSC, UC San Diego, and the University of Auckland, New Zealand

KC Claffy, CAIDA, SDSC, UC San Diego

ABSTRACT

We present the concept of network traffic streams and the ways they aggregate into flows through Internet links. We describe a method of measuring the size and lifetime of Internet streams, and use this method to characterize traffic distributions at two different sites. We find that although most streams (about 45 percent of them) are *dragonflies*, lasting less than 2 seconds, a significant number of streams have lifetimes of hours to days, and can carry a high proportion (50-60 percent) of the total bytes on a given link. We define *tortoises* as streams that last longer than 15 minutes. We point out that streams can be classified not only by lifetime (*dragonflies and tortoises*) but also by size (*mice and elephants*), and note that stream size and lifetime are independent dimensions. We submit that ISPs need to be aware of the distribution of Internet stream sizes, and the impact of the difference in behavior between short and long streams. In particular, any forwarding cache mechanisms in Internet routers must be able to cope with a high volume of short streams. In addition ISPs should realize that long-running streams can contribute a significant fraction of their packet and byte volumes — something they may not have allowed for when using traditional "flat rate user bandwidth consumption" approaches to provisioning and engineering.

BACKGROUND

MEASURING INTERNET TRAFFIC

The Internet is a global internetwork, sharing information among millions of computers throughout the world. Internet users send packets of information from one machine to another using various Internet protocols; TCP and UDP are the most common transport protocols, but newer standard protocols are starting to appear.

Packets are carried through various links, from user host to regional Internet service provider (ISP), regional to backbone ISP, and so on. Between such links packets are forwarded by routers using IP; IPv4 is most common, but IPv6 is now beginning to be deployed [1].

Typical users are not interested in packets on the Internet; they simply run application programs such as Web browsers, which exchange packets with other computers as they carry out user requests. Groups of packets exchanged in this way are commonly referred to as *traffic flows*.

To measure traffic flows one examines packet headers as they pass by on a given link, determines which flow each packet belongs to using information extracted from its header, and counts packets and bytes for each flow. A system that gathers flow data in this way is called a *traffic meter*. Such a meter may be free-standing or built into a device such as a router.

For our investigations we use RTFM, an Internet standard real-time traffic flow measurement system [2]. The RTFM architecture defines three entities:

- ***Meters*** gather data from packets so as to produce flow data.
- ***Meter readers*** collect flow data from meters.
- ***Managers*** specify real-time data reduction by downloading configuration data (called *rulesets*) to meters, and also specify the intervals at which meter readers read flow data.

We make our flow measurements with NeTraMet [3], an open-source implementation of RTFM. NeTraMet includes an RTFM meter, a combined manager/meter reader, and a compiler for SRL (the Simple Ruleset Language, RTFM's high-level language for specifying rulesets).

TRAFFIC MIX: MICE AND ELEPHANTS

On any Internet link there is always a mix of flows from a variety of applications, carried by various transport protocols, especially TCP and UDP. UDP provides unreliable datagram delivery; that is, an application sends UDP packets, but UDP itself provides no feedback to the sender. UDP is therefore unaware of any network congestion; streaming applications often continue to send data at constant high byte rates.

TCP, on the other hand, not only provides reliable byte stream delivery, but also uses feedback from receiving hosts to control its

0163-6804/02/$17.00 © 2002 IEEE

sending rate. TCP's congestion management algorithms allow a TCP stream to vary its byte rate, seeking to use the highest possible rate but lowering the rate when the network becomes congested. For this reason, TCP is considered *network-friendly*.

Early analyses and simulations of TCP behavior focused on steady state behavior, using "infinite source" workloads (e.g., large file transfers), and assumed that high-volume TCP streams (network *elephants*) would not be significantly affected by the presence of small TCP streams (network *mice*). The fundamental difference between network mice and elephants is that an elephant's TCP session extends past TCP's *slow start* phase, so its behavior, including the way it interacts with other TCP sessions, is controlled by TCP's feedback-based congestion management algorithms. However, mice cannot be controlled by feedback since they are sent and received in their entirety before TCP has an opportunity to apply feedback control.

More recent models of TCP behavior have increasingly focused on interactions between elephants and mice. For example, Joo *et al.* [4] analyzed the expected throughput of TCP streams and how concurrent streams interact. They found that multiple elephants can synchronize with each other, which may cause routers to drop packets. They state, "although elephants are responsible for a major proportion of the bytes on the network, the number of packets generated by mice can be sufficient to create losses from time to time." They also examined the dynamics of packet drops and concluded that mice can break up synchronization effects, leading to more efficient use of network resources. This breakup effect may explain why best effort datagram delivery has served the Internet so well as a lowest common denominator of network service.

As an alternative to classifying flows by size (number of bytes), that is, as *elephants* or *mice*, one can also classify flows by their lifetime (in seconds). Shaikh, Rexford, and Shin [5], using a 60-s timeout, observed flow lifetimes up to 2000 s and found that such "long-lived" flows accounted for a high proportion of bytes on a link. They propose that "load-sensitive" routers might attempt to find better routes for long-lived flows, thus improving overall link utilization.

STREAMS, FLOWS, AND TORRENTS: TRAFFIC KINDS

The term *flow* has various meanings in different contexts. For example, in routing a *flow* is a set of packets with the same source and destination IP addresses, all traveling in the same direction. Internet researchers often use 5-tuples (protocol, source and destination IP address, and port number); they refer to these as *microflows*.

In this article we use the terminology proposed by Brownlee and Murray [6]:

- *Streams* are individual IP sessions (e.g., TCP or UDP) between ports on pairs of hosts.
- *Flows* are sets of packets traveling in either direction between a pair of endpoints (which may be hosts, networks, etc.).
- A *torrent* refers to all the traffic on a given link.

For our investigations we classify all traffic within a torrent into four *kinds*, distinguished by transport protocol: *UDP*; *TCP* (the most common); and *other*. Because the Web is a dominant application at some sites, we subdivide *TCP* into *Web* (*TCP*) and *non-Web TCP*. We aggregate all streams for each *kind* of traffic into one of four *flows*; we present data for these flows in Figs. 1 and 4.

MEASUREMENT METHODOLOGY

The NeTraMet implementation of streams creates a data structure for each stream within a flow, and counts each stream's packets and bytes until the stream times out, that is, no packets are observed for a dynamically specified *timeout* interval [6]. When a stream times out, its packet and byte counts are used to add a point to its flow's stream size and lifetime distributions. Streams that remain active for long periods of time make no contribution to the stream distributions.

We have extended the NeTraMet meter to monitor stream lifetimes, and to automatically create flows in the meter's flow table when a stream remains active for more than a specified time. Flows created for such long-running (LR) streams have their packet and byte counters updated for every packet. We collect data for these LR stream flows with every meter reading, so as to obtain time series of packet and byte counts for each LR stream.

We have extended the NeTraMet meter to monitor stream lifetimes, and to automatically create flows in the meter's flow table when a stream remains active for more than a specified time.

Our NeTraMet ruleset produces one flow for each of our four traffic *kinds*. For each flow we build five distributions:

1-2 To/from bytes: Bytes in each direction of streams
41 bins, log scale, 30 bytes to 600 kbytes
3-4 To/from packets: Packets in each direction of streams
41 bins, log scale, 1 to 32,768 packets
5 Flow time: Stream lifetime
41 bins, log scale, 2 s to 15 min

We read our NeTraMet meters every 5 min, producing a set of five distributions for each 5-min interval. The counters for each distribution bin are never reset; instead, we compute the distribution for each reading interval as the difference between successive readings. We can therefore observe variations in the distributions over periods of hours to days, with a time resolution of 5 min.

In order to provide higher resolutions for small and/or short-lived streams we use log scales for the distributions. Streams with size or lifetime greater than a distribution's upper limit are counted in that distribution's overflow bin.

Although we collect byte and packet stream size distributions, in this article we only present "bytes from" distributions, mainly due to space limitations, but also because the other distributions are generally similar to the bytes from ones. Similarly, the percentage of other traffic we observe is negligible compared to our other three traffic *kinds*; we have not shown it on our plots.

Both our NeTraMet meters run on Linux systems located at two sites. Our OC3 meter (Auckland) observes packet headers using a commodity 100BaseT Ethernet card via libpcap [7]; our OC12 meter (UCSD) uses a Dag 3.2 card [8] via direct Linux drivers.

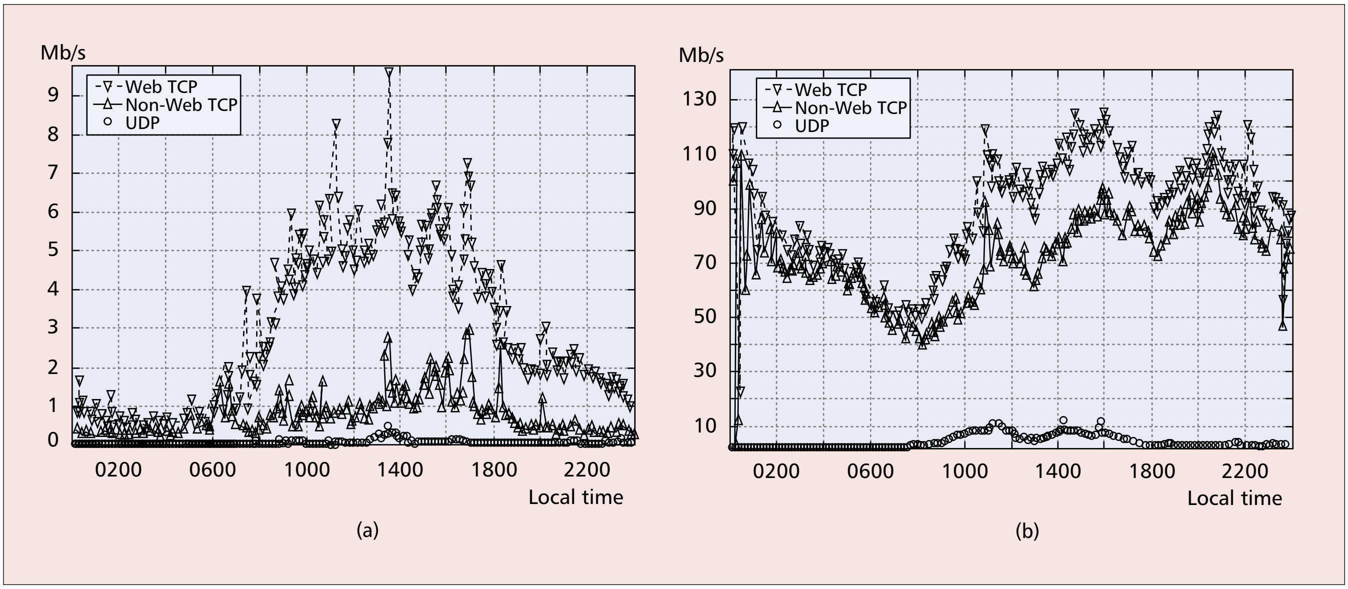


Figure 1. *Cumulative rate (kilobits per second) for various traffic kinds vs. time of day (HHMM) in 5-min intervals at a) UA and b) UCSD for 24 h from midnight local time on Wednesday, 12 June 2002.*

Short vs. Long-Running Streams: Dragonflies and Tortoises

When configuring a NeTraMet meter one must specify a *stream-to-flow* time; the meter creates LR flows for all streams lasting longer than *stream-to-flow* seconds. In choosing a *stream-to-flow* value one must balance the desire to observe streams lasting for shorter periods against the costs of collecting and working with larger data sets. For this investigation we experimented with 5- and 10-min lifetimes before choosing 15 min as the maximum lifetime of a *short* stream; that is, our *tortoises* are LR streams with lifetimes greater than 15 min. We find that 15 min is a reasonable compromise, generating manageable data set sizes that still yield substantial insights into LR stream behavior.

Measuring distributions of shorter flows is complicated by an "edge effect." Since a stream is only counted when it times out, the distribution counts include streams that started in an earlier interval but timed out in a given reading interval. The first bin in our flow time distributions counts flows with lifetimes up to 2 s, which limits the edge effect error to a maximum of 0.6 percent. We find that a high proportion of streams fall in this first bin, so we describe them as *Very-Short* streams (i.e., streams with lifetimes of 2 s or less).

To summarize, we classify streams by lifetime as:

- Very-Short *dragonflies*, lasting up to 2 s
- Short, lasting up to 15 min
- LR *tortoises*, lasting more than 15 min

We emphasize that stream size is independent of stream lifetime.

Observation Sites

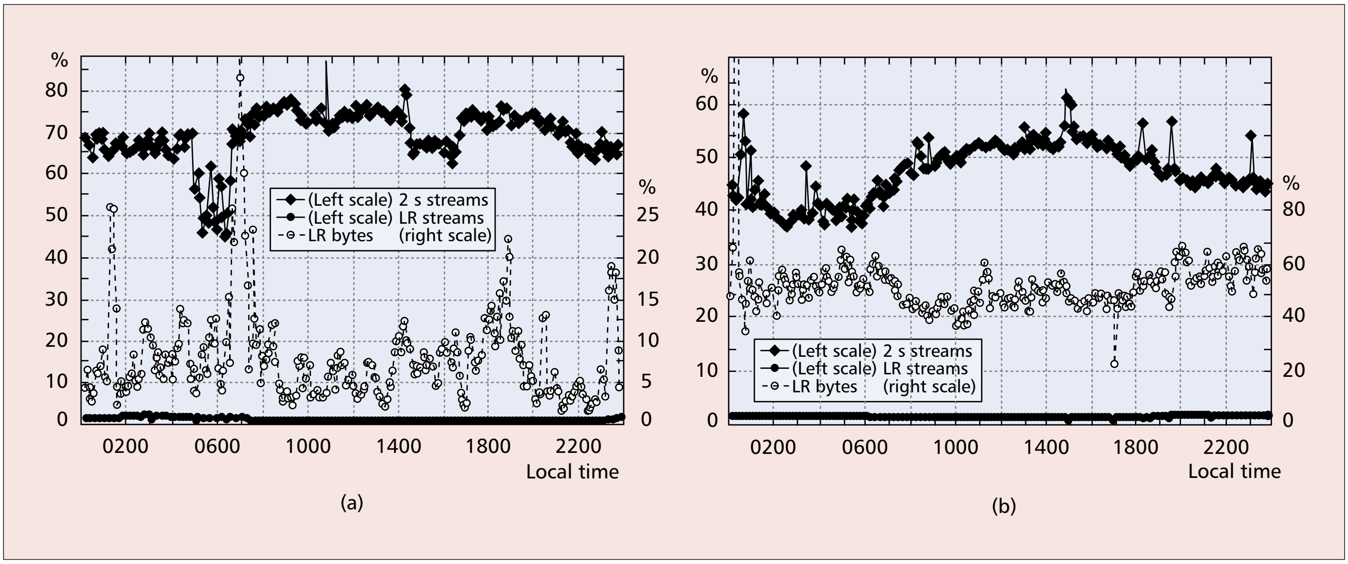
In this section we discuss the Internet traffic observed at two sites, the University of Auckland (UA) and the University of California at San Diego (UCSD). We have collected data at both sites for eight complete days. Traffic patterns vary little over those days; our figures show data for Wednesday, 12 June 2002.

UA runs a campus network serving about 35,000 users. The campus is connected to the Internet via an OC3 (155 Mb/s) asynchronous transfer mode (ATM) link; however, this link is rate-limited to 9 Mb/s by the university's Internet provider. Our NeTraMet meter is connected to a 100 Mb/s Ethernet hub located between the university's access router and its firewall.

Figure 1a shows stacked bar plots of *Web* TCP, *non-Web TCP*, and *UDP* traffic at UA vs. local time. We read our meters every 5 min, then plot the average bit rate for each 5-min interval. The bit rate for each *kind* of traffic is indicated by the distance between its trace and the one below it. Furthermore, the topmost trace shows the whole torrent's traffic. Figure 1a shows a typical day for a small enterprise site. There was little traffic in the early morning hours, except for a few brief spikes; these probably indicate periods during which mirroring servers at UA were updating their content. From about 7 a.m. the load grew; there was a slight dip around lunchtime, then the load remained steady through the afternoon. At about 6 p.m. the load decreased sharply, rose again slightly in the early evening, then decreased toward midnight. On this link Web traffic was dominant, the ratio of Web to non-Web TCP traffic remaining fairly steady at about 80 percent. UDP traffic contributed few bytes to the load, and there was almost no traffic other than TCP and UDP.

One distinctive feature of this site is that UA recovers the costs of Internet connectivity directly from its users, charging for each megabyte of data sent or received. Departments are billed monthly for staff usage, and students are charged by a real-time access control and billing system. Clearly the Internet usage patterns of UA users are influenced by the knowledge that they are paying for it. For example, UA has not (yet) seen widespread use of peer-to-peer file sharing.

UCSD is connected to the Internet via an OC-12 (622 Mb/s) ATM link to the San Diego Supercomputer Center (SDSC). SDSC has high-



■ **Figure 2.** *Percentage of Short and LR streams (black symbols using left axis), and percentage of bytes in LR streams (gray symbols using right axis) vs. time of day (HHMM) in 5 min intervals at a) UA) and b) UCSD for 24 h from midnight local time.* ***Note that*** *40–70 percent of streams are* dragonflies *(black diamonds) lasting 2 s or less. About 1.5 percent of streams (black dots) are* tortoises *(LR streams lasting more than 15 min)s. They contribute 5–50 percent of a torrent's bytes (gray circles).*

speed links to three research networks as well as a lower-speed connection to the commodity Internet. Our NeTraMet meter is attached to UCSD's OC-12 link via a passive optical splitter.

Figure 1b shows stacked bar plots of UCSD traffic. UCSD does not attempt to recover per-byte Internet usage costs, nor does it impose rate limits on individual Internet connections. In this environment Internet usage is limited only by congestion, which will increase as the university's total Internet usage increases. Occasionally UCSD upgrades its commodity Internet capacity, historically allowing the load and congestion cycle to build up again.

UCSD's total traffic byte volume is 16 times greater than UA's nearly all the time, but still utilizes only about 15 percent of their OC-12 link's maximum capacity. UCSD also has more UDP traffic than Auckland; 1 Mb/s at night, rising to around 10 Mb/s during the day.

At UA Web was the dominant TCP application, but UCSD's Web to non-Web TCP ratio is only about 50 percent (much less than UA's 80 percent), indicating that at UCSD a higher proportion of TCP traffic is generated by non-Web applications.

Overall, the diurnal variations for Web and UDP traffic suggest that these kinds of traffic follow human activities. Non-Web TCP, however, has a fairly high background level (about 0.5 Mb/s at Auckland and 70 Mb/s at UCSD); it varies about this level, with its UCSD minimum around 8 a.m. and its maximum around 10:30 p.m.

Percentage of Streams and Bytes in a Torrent

As well as producing stream packet and byte size distributions, our NeTraMet ruleset records the total number of packets and bytes observed in each traffic flow. For each meter reading we sum the LR stream byte counts; from these sums we compute the percentage of bytes in LR streams. In this section we discuss the LR byte percentages, together with the percentages of Very-Short (≤ 2*s*) and LR (> 15 *m*) streams at our two observation sites. Figure 2 plots these three measures at 5-min intervals over a day for UA and UCSD.

First, the percentage of LR streams is plotted with black dots (bottom trace) in Fig. 2, using the left-hand y-axis scale. At UCSD (Fig. 2b) about 1.5 percent of the streams were LR, and this level varied little during the day. At UA (Fig. 2a) there were only about 0.5 percent LR streams during the day, but nearly 1.5 percent from midnight until dawn.

Second, the percentage of Very-Short streams is plotted with black diamonds (top trace) in Fig. 2, also using the left-hand y-axis scale. At UCSD (Fig. 2b) there was a clear diurnal variation of the Very-Short stream percentage from 38 percent around 4 a.m. to about 55 percent around 3 p.m., corresponding well with UCSD's diurnal variation in Web traffic shown in Fig. 1b. At UA, however (Fig. 2a), about 70 percent of the streams were Very-Short nearly all day. Between 5 a.m. and 7 a.m. the Very-Short stream percentage dropped to 50 percent; during that interval there were considerably fewer TCP Web bytes than non-Web bytes.

Third, the percentage of bytes in LR streams is plotted with open circles (middle trace) in Fig. 2, using the right-hand y-axis scale. At UCSD (Fig. 2b) about 50 percent of all bytes were in LR streams, and this percentage varied little during the day. At UA (Fig. 2a) only about 5 percent of all bytes were in LR streams, most likely because UA had much less non-Web TCP traffic than UCSD.

To summarize:

- About 1.5 percent of UA and UCSD streams were LR.
- 40 percent (UCSD) to 70 percent (UA) of the streams were Very-Short, and most of them appear to be Web traffic.

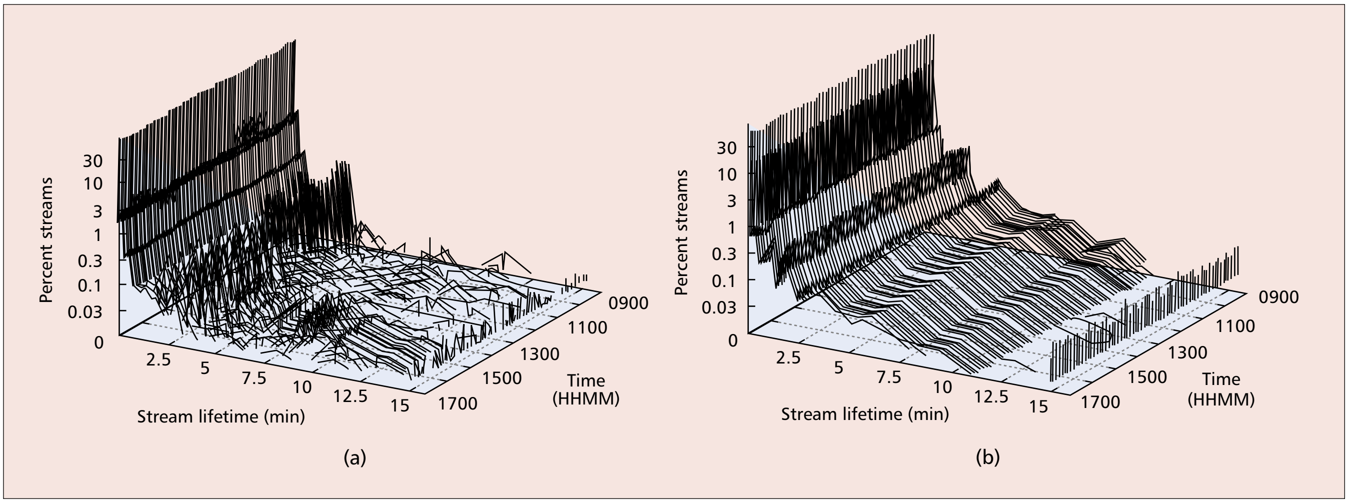


Figure 3. *Stream lifetime distributions at a) UA and b) UCSD. Percent streams vs. lifetime (minutes) for 8 h, 9 a.m. to 5 p.m. local time on Wednesday, 12 June 2002, in 5-min intervals. Note that although the link data rates differ, the shape of the distribution is similar for both sites.*

- 5 percent (UA) to 50 percent (UCSD) of all bytes were in LR streams, and most of them appear to be non-Web traffic.

Short Stream Behavior

Lifetime distributions

Figure 3 shows lifetime distributions for short streams, that is those with lifetimes up to 15 minutes, at UA (Fig. 3a), and UCSD (Fig. 3b). In both these plots at least 45 percent of the counted streams lie in the first bin; that is, they had lifetimes of 2 s or less. We use a logarithmic scale for the distribution's y-axis, to reveal the whole range of percentages. The rest of the distribution falls away quickly at UA, where there are few streams with lifetimes above 2.5 min. At UCSD the lifetime distribution falls away more slowly, but there are few streams with lifetimes above 5 min. However, above these lifetimes, the lifetime distributions slope down gently toward 15 min, our maximum short stream lifetime.

As defined in an earlier section, our LR streams have lifetimes greater than 15 min. When an LR stream times out, the meter increments the overflow bin for its distributions (i.e., the distributions for that stream's flow). The lifetime distribution overflow counts appear as the high values plotted for y values above 15 min (i.e., the spikes at the right edge of Figs. 3a and 3b). As we saw on the LR byte percentage plots (Fig. 2), although there are few LR streams, they can account for a high percentage of a flow's total bytes throughout the day.

The most striking feature of the distributions in Fig. 3 is that their shapes are similar. At both sites we observe that the distributions do not change rapidly with time. At UCSD (Fig. 3b) there was little change in the shape of the distributions over the eight hours shown. At UA (Fig. 3b), where the link capacity is lower, the proportion of streams with lifetimes between 7.5 and 12.5 min increased during the afternoon. This increase could indicate that when the UA link's byte load is high, not only are there more users, but those users are working with larger files.

Byte Size Distributions

Figure 4 shows byte size distributions for short streams at UCSD; byte size distributions at UA (not shown) are similar. The distributions are collected using 41 bins in a log scale from 30 bytes to 600 kbytes. The jagged appearance for small stream sizes (i.e., below about 300 bytes) is an artifact of the limited bin resolution for streams with only a few packets.

For short UDP streams (Fig. 4a), the byte size distribution has peaks at about 30 and 80 bytes, indicating that a high proportion of UDP streams have only one or two packets in their from (i.e., destination to source) direction. Short UDP stream sizes fall steadily (on a log scale) from about 300 bytes to 20 kbytes; they also have a noticeable percentage of streams in the overflow bin (i.e., with sizes above 600 kbytes). Successive 5-min distributions vary somewhat over periods of about 15 min, producing the corrugated effect on the plot.

Short Web streams (Fig. 4b) have high peaks at 30, 50, 100 and 200 bytes, a local maximum from about 300 to 800 bytes, a plateau from 1 kbytes to about 40 kbytes, and a fairly steep fall from there. Since Web streams use TCP, they require at least two packets in each direction; hence, the size distribution's peaks below 300 bytes indicate streams that most likely failed to establish a TCP session. The local maximum from 300 to 800 bytes is probably for streams carrying small Web objects (buttons, "file not found" messages, etc.), and the plateau suggests that Web objects have a fairly flat file size distribution below about 40 kbytes, with a power law fall (linear on the log plot, Fig. 4b) for large files. The 5-min Web stream size distributions are remarkably steady, suggesting that Web usage patterns at UCSD are stable over long periods.

Short non-Web TCP stream distributions (Fig. 4c) have a stream size peak at 30 and 50 bytes, smaller peaks at 180 and 1500 bytes, and a steady fall from there to 600 kbytes. The steady fall suggests that non-Web stream sizes also have a power law distribution, at least for stream sizes from about 1500 bytes to 300 kbytes.

There are also noticeable local maxima at 5 kbyte for Web streams and 90 kbytes for non-Web streams, indicating transmission of many objects this size; we have not yet determined what they were.

To summarize:

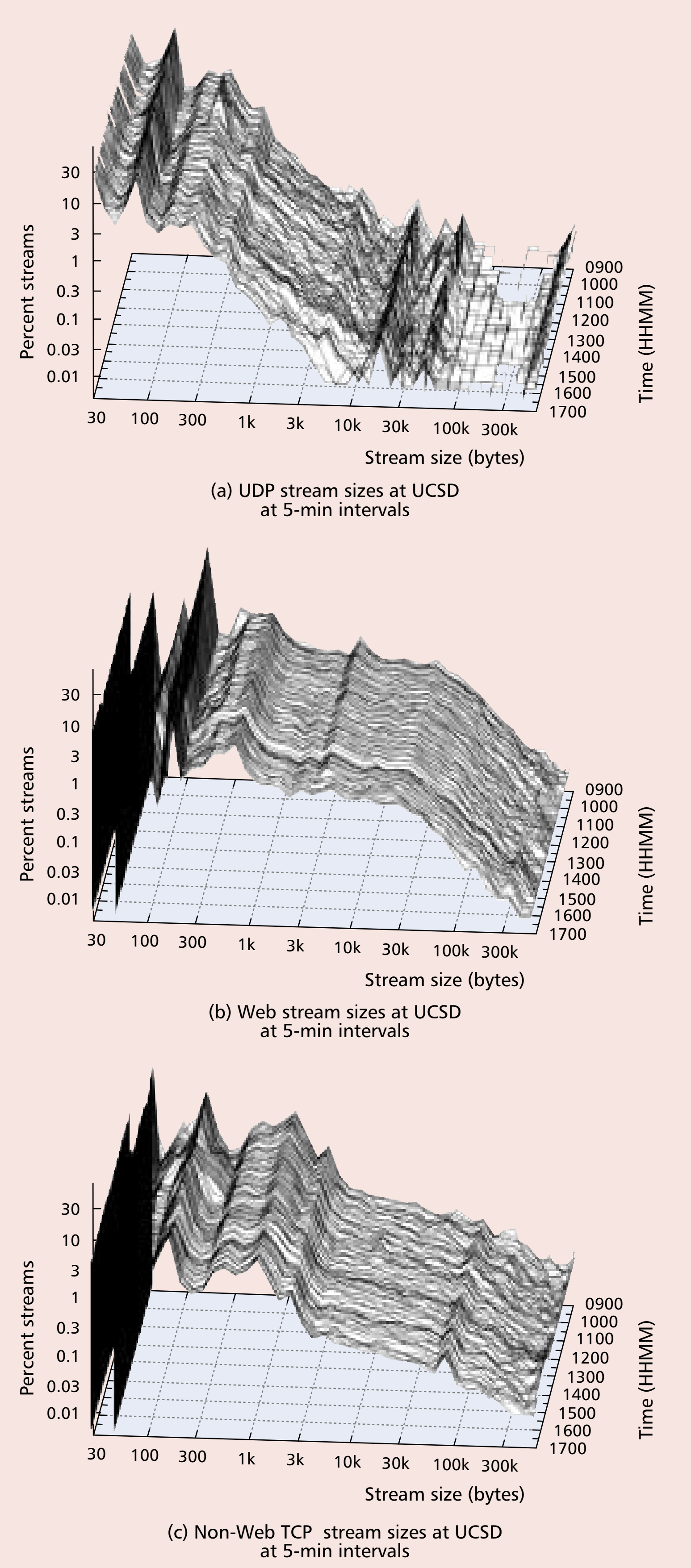
- Short stream size distributions for UDP, non-Web, and Web TCP traffic are distinctly different and stable over periods of hours.
- Our study only reflects data collected at UA and UCSD, but at this stage we believe these sites are representative examples of medium to large Internet-edge networks.

LR Stream (Tortoise) Behavior

We find that LR streams (i.e., streams lasting more than 15 min) occur frequently at both UA and UCSD. To gain a better understanding of their behavior we selected a representative set of LR streams and produced thumbnail plots of their bit rates (kilobits per second) vs. time (minutes), as shown in Fig. 5. In this section we comment on the behavior of these streams.

- Figure 5a shows three streams with bit rates below 60 b/s. At such low bit rates our bit rate resolution is poor, producing the stepped effect in the traces. The top trace was an NTP stream. These are always present, serving to keep end system clocks synchronized. The middle trace was an SSH stream that was mostly quiescent but had occasional intervals reflecting user activity. The lower trace was a DNS stream with a diurnal bit rate pattern, highest during the afternoon.
- Figure 5b shows a 24-h Web stream that consisted entirely of 20 b/s bursts at half-hour intervals, suggesting that it was carrying a Web page that was "refreshed" (redisplayed) every half hour. The apparent size of the bit rate bursts depends on whether they happen to fall in one or two of our 5-min reading intervals.
- Figure 5c shows an FTP file transfer, taking 5 h at a high but variable rate, 50–200 kb/s. Its high bit rate variance suggests that this stream was traversing severely congested Internet paths.
- Figure 5d shows two streams that ran at 180 and 320 kb/s nearly all day, moving about 2.9 and 1.4 Gbytes. Although they were TCP streams their data rates did not vary much, indicating that there was little congestion on their Internet paths. These streams effectively reduced the link's available capacity by several hundred kilobits per second.
- Figure 5e shows two near identical Doom streams that ran for 9 h at 80–200 kb/s. They were UDP streams, hence their steady rate for long periods. Their sudden rate changes presumably correspond to changes in the state of the game.
- Figure 5f shows a RealAudio stream that ran at 180 kb/s for 15 h, transferring 1.2 Gbytes. This was a UDP stream; its rate varied only about +/– 5 kb/s, much less than the TCP streams in Fig. 5d.

To summarize:

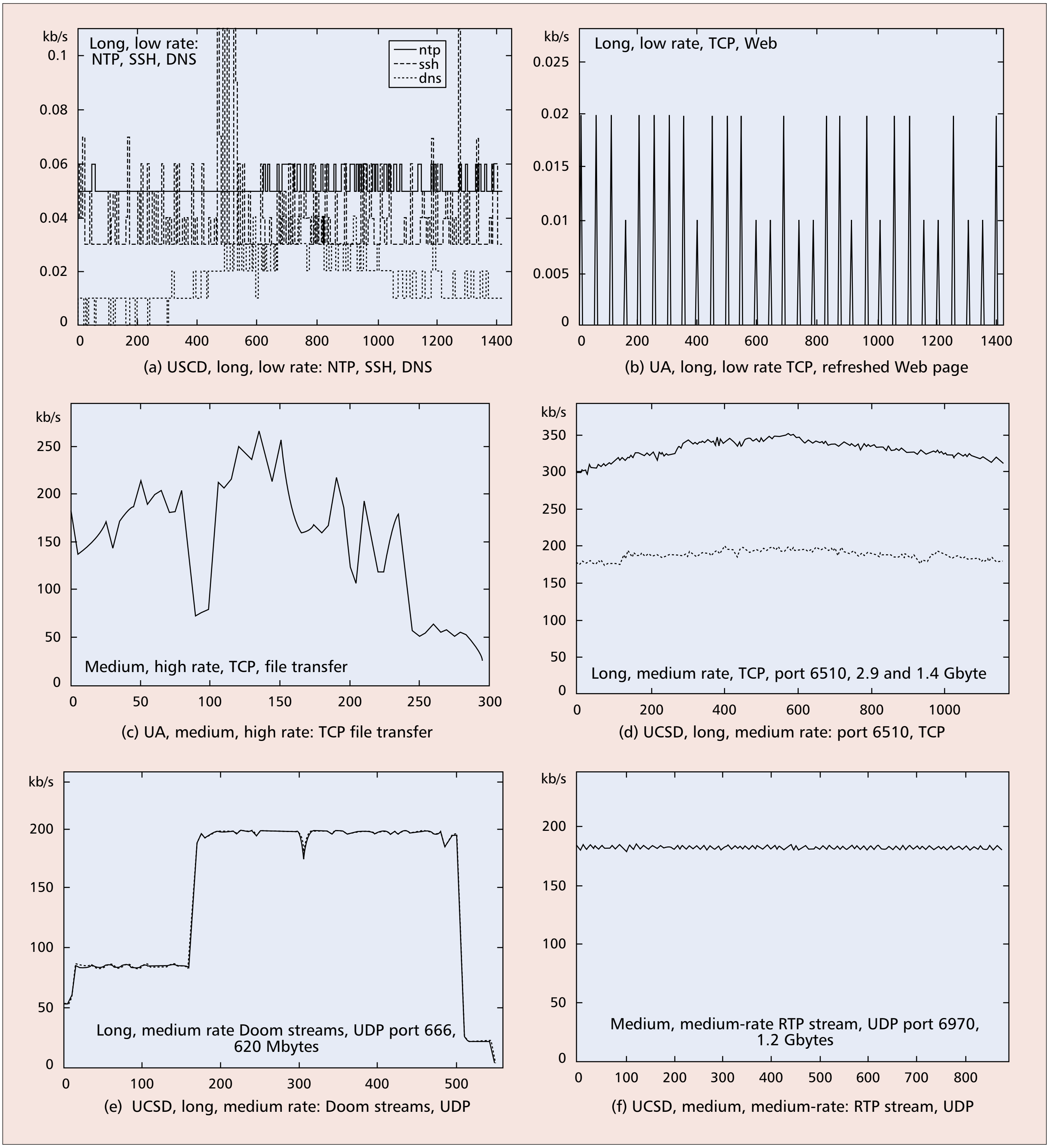


(a) UDP stream sizes at UCSD at 5-min intervals

(b) Web stream sizes at UCSD at 5-min intervals

(c) Non-Web TCP stream sizes at UCSD at 5-min intervals

Figure 4. *Size distributions for short streams at UCSD. Percent of streams vs. stream size (kbytes) for eight hours, 9 a.m. to 5 p.m. local time on Wednesday, 12 June 2002. Note the clear differences between the three flow kinds.*



(a) USCD, long, low rate: NTP, SSH, DNS

(b) UA, long, low rate TCP, refreshed Web page

(c) UA, medium, high rate: TCP file transfer

(d) UCSD, long, medium rate: port 6510, TCP

(e) UCSD, long, medium rate: Doom streams, UDP

(f) UCSD, medium, medium-rate: RTP stream, UDP

Figure 5. *LR stream histories, UA on Thursday 4 April 2002 (b, c) and UCSD on Thursday 28 March 2002 (a, d, e, f). Bit rate (kilobits per second) vs. elapsed time (minutes).*

- Long, continuous LR streams may be low-rate (service support or user interaction) or high-rate (audio/video data streams).
- Brief and medium-duration LR streams tend to be high-rate, running until some user-initiated activity is completed.
- TCP LR streams show rate variation as Internet congestion changes over time, with rate variations similar for streams sharing congested links in their Internet paths.
- UDP streams tend to run at fairly constant bit rates, but these rates change in response to application dynamics.

LR Stream Lifetimes

Figure 6 is a log-log scatter plot showing percentage of LR streams vs. LR stream lifetime (minutes) for Auckland and UCSD. The two plots cluster around a line, suggesting that stream sizes follow a power law distribution. The spread of points around this line is narrow for lifetimes from 20 to about 100 min; the spread increases for longer lifetimes.

More points are plotted for UCSD than for UA, providing more detail for lifetimes above 500 min. Similarly, because there are fewer points for UA (reflecting the lower traffic rate at UA), the y-axis has lower resolution for the UA plot, producing the line at stream percentage 0.06 percent. The two plots are nonetheless similar, indicating that users at the two sites are running similar application cross-sections.

CONCLUSION

As recently as July 2000, Zhang *et al.* [9] observed that "Internet traffic is now dominated by mice, that is small objects 10-20 kB in size; the average web document is only around 30 kB," but in contrast reported that "the majority of the packets and bytes belong to elephants." Similarly, in April 2001 Brownlee *et al.* [10] measured stream byte size distributions and found that TCP data streams had a 95th percentile of approximately 15 kbytes.

Since 2000 Internet link speeds have increased as users migrated to cable modem and DSL connections, backbone links were upgraded from OC-3 (155 Mb/s) toward OC-48 (2.4 Gb/s), and ISPs installed newer faster routers to handle increasing packet loads. At the same time computer hardware improved; systems with 1 GHz processors, 512 Mbytes memory, 20 Gbytes disk drives, and ever-increasing I/O bus speeds became common. This dramatic increase in network and computer capability has allowed users to work with ever larger files. As a result we now observe that the average size of Web objects has increased considerably, with Web objects up to 50 kbytes becoming common.

Along with increasing file size, the last few years have seen the rapid growth in usage of an ever increasing set of peer-to-peer file sharing systems (e.g., Napster, Gnutella, E-Donkey). These peer-to-peer applications have significantly changed the traffic mix, so a higher overall proportion of their streams have large numbers of bytes. In addition to streaming protocols carrying audio and video programs, voice over IP or multimedia conferencing are increasingly common. Clearly these trends will continue.

Our current observations confirm that most streams are very short. At least 45 percent of streams have lifetimes of 2 s or less (*dragonflies*), and about 98 percent of them last less than 15 min. However, the remaining 1 or 2 percent, which we call Long-Running streams (*tortoises*), have lifetimes of hours to days and can carry a high proportion (50 to 60 percent) of the total bytes on a link.

We submit that ISPs need to be aware of the behaviors of short streams. In particular, any forwarding cache mechanisms in Internet routers must be able to cope with the high volume, both absolute and as a percentage, of short streams. In addition, ISPs should realize that LR streams can contribute a significant fraction of their packet and byte volumes, reducing the available bandwidth of their Internet links.

Lastly, we emphasize that streams can be classified not only by their size (*mice and elephants*), but also by their lifetime (*dragonflies and tortoises*). Furthermore, stream size and lifetime are independent dimensions; each is of interest in understanding the overall behavior of streams in a torrent.

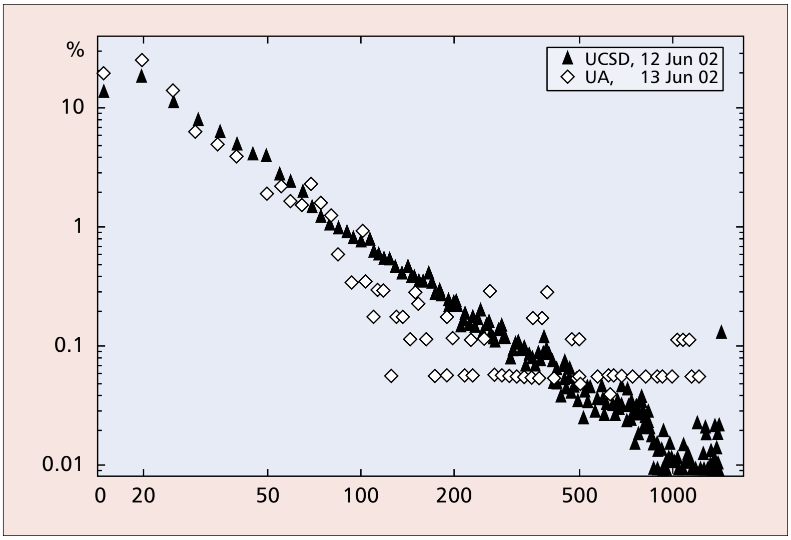


Figure 6. *Stream lifetime distributions, UCSD and UA. Percentage of LR streams in torrent vs. stream lifetime (minutes).*

REFERENCES

[1] J. Bound, IPv6 Implementation, ISOC Member Briefing #4, Sept. 2001; http://www.isoc.org/briefings/004/

[2] N. Brownlee, C. Mills, and G. Ruth, "Traffic Flow Measurement: Architecture," RFC 2722, Oct. 1999.

[3] N. Brownlee, "Using NeTraMet for Production Traffic Measurement," *IM 2001*, May 2001.

[4] Y. Joo *et al.*, "On the Impact of Variability on the Buffer Dynamics in IP Networks," *Proc. 37th Annual Allerton Conf. Commun., Control Comp.*, Sept. 1999; http://www.dsp.rice.edu/publications

[5] A. Shaikh, J. Rexford, and K. G. Shin, "Load-Sensitive Routing of Long-Lived IP Flows," *Proc. ACM SIGCOMM*, Sept. 1999.

[6] N. Brownlee and M. Murray, "Streams, Flows and Torrents," *PAM 2001 Wksp.*, Apr. 2001

[7] tcpdump/libpcap website, http://www.tcpdump.org/

[8] Dag Website: http://dag.cs.waikato.ac.nz/

[9] Y. Zhang and L. Qiu, "Understanding the End-to-End Performance Impact of RED in a Heterogeneous Environment," Cornell CS tech. rep. 2000-1802, July 2000; http://www.aciri.org/floyd/red.html

[10] N. Brownlee *et al.*, "Methodology for Passive Analysis of a University Internet Link," *PAM 2001 Wksp.*, Apr. 2001.

BIOGRAPHIES

NEVIL BROWNLEE (nevil@caida.org) co-chaired the IETF's Realtime Traffic Flow Measurement (RTFM) Working Group, and is now co-chair of the IP Flow Information eXport (IPFIX) Working Group. He created NeTraMet, an open-source implementation of the RTFM (Internet Standard) architecture, at the University of Auckland late in 1992. He now works half time in Auckland overseeing technology developments, particularly those relating to networks, and teaching in computer science. He spends the other half of his time at CAIDA in San Diego, California, where he pursues research into the behavior of Internet traffic, and continues to develop NeTraMet so that it can handle higher-speed networks.

KC CLAFFY (kc@caida.org) is principal investigator for CAIDA, the Cooperative Association for Internet Data Analysis, based at the University of California's San Diego Supercomputer Center. Her research interests include: data collection, analysis, and visualization of Internet workload, performance, topology, and routing behavior. She also works on engineering and traffic analysis requirements of the commercial Internet community, often requiring ISP cooperation in the face of commercialization/competition.