Internet Stream Size Distributions*

or

Do Mice Matter?

Abstract—**We discuss the *mice vs elephants hypothesis*, i.e. the statement that a high proportion of traffic streams (sometimes referred to as 'flows') on a link account for only a small proportion of the link's traffic load. To test the validity of this hypothesis, we have measured stream distributions in real time on OC3, OC12 and OC48 Internet links.**

We describe a methodology for comparing link traffic loads, using *Complementary Cumulative Stream Size Distributions*, CCSSD. A CCSSD plots percentage of timed-out streams in a measurement interval versus percentage of bytes accounted for by the timed-out streams.

On our three measured links, the OC3 and OC48 links produced similar CCSSD plots but the OC12 link's plot was markedly different. We have determined that this link has a unique traffic mix.

Overall, we conclude that although mice are often insignificant in terms of traffic measured in bytes, they are generally significant in terms of a link's packet load. Furthermore, since the character of traffic on a link can change with time, one should make periodic measurements of the link's stream distributions.

Keywords—**Network Mice and Elephants, CCSSD, NeTraMet**

I. Introduction

Early analyses and simulations of TCP behavior focused on steady-state behavior, using 'infinite source' workloads (e.g. large file transfers), and assumed that long TCP streams (network *elephants*) would not be significantly affected by the presence of short TCP streams (network *mice*). Estan & Varghese [**?**], for example, remark that "a common observation found in many measurement studies is that a small percentage of flows accounts for a large percentage of the traffic." We describe this and similar statements as the *mice vs elephants hypothesis*.

In this paper, we use the terminology proposed by Brownlee & Murray [**?**], i.e.:

- *streams* are sets of packets in one direction between ports on pairs of hosts,
- *flows* are sets of packets traveling in either direction between a pair of end-points (which may be hosts, networks, etc.), and
- a *torrent* refers to all the traffic on a given link.

For example, a TCP session can be described as a single *flow*, made up of two *streams*, each carrying data, acks, or both in one direction between the flow's two TCP hosts.

The fundamental difference between network mice and elephants is that an elephant's TCP session will extend 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. Mice, however, cannot be controlled by feedback since they are sent and received in their entirety before any feedback control can be applied.

Many studies of network stream size distributions have been published. Downey [**?**] presented a model that explains the distribution of file sizes found both in computer systems and in the World Wide Web. Downey's model suggests that such distributions are approximately lognormal and not necessarily heavy-tailed. Zhang et al. [**?**] observe that "Internet traffic is now dominated by mice, i.e. small objects 10-20 kB in size; the average web document is only around 30 kB," but, in contrast, report that "the majority of the packets and bytes belong to elephants." Similarly, Brownlee et al. [**?**] measured stream size distributions and found that TCP data streams have a 95th percentile of approximately 15 kB.

One particularly appealing use of the "mice and elephants" hypothesis, proposed by Duffield et al. [**?**] and Estan & Varghese [**?**], is to concentrate on the elephants when measuring traffic for charging purposes. Since elephants contribute most of the traffic one should sample them often. Since mice do not contribute much to the total load, one need only sample them occasionally. This approach could provide an effective way to cope with measuring the ever-increasing volumes of Internet traffic.

More recent models of TCP behavior have increasingly focused on the interactions between elephants and mice. For example, Zhang & Qiu [**?**] reported on simulations of a mix of short- and long-lived TCP streams and concluded that "RED tends to favor mice, which is desirable because mice typically belong to web browsing sessions"

Joo et al. [**?**] analyzed the expected throughput of TCP streams and the way in which concurrent streams may interact. They found that multiple elephants can synchronize with each other, which may produce packet drops. 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." Joo et al. [**?**] examined the dynamics of packet drops, and concluded that mice can break up synchronization effects, leading to a more efficient use of network resources. This 'break-up' effect may explain why 'best effort' datagram delivery works so well. Guo & Matta [**?**] extended this idea. They analyzed short TCP flow performance and proposed a new, better-than-best-effort TCP service. Their proposed service attempts to enhance the performance of short TCP flows, so as to improve overall utilization of network resources.

To provide a reference point, we rephrase the hypothesis as follows:

> Mice vs Elephants hypothesis:
> *10% of streams on a link account for*
> *90% of its total network traffic.*

If correct, the hypothesis could provide a basis for improvements in datagram forwarding and traffic accounting, as described above. But is the hypothesis valid? If so, under what conditions is it valid? Is it influenced by the traffic composition, e.g. mix of protocols and applications, or by the amount of traffic on a link?

This paper reports on some measurements of stream sizes on

TABLE I

Traffic Kinds USED IN THIS STUDY, I.E. PROTOCOLS AND TCP APPLICATIONS OBSERVED ON OUR OC3, OC12 AND OC48 LINKS. TOTAL LINK TRAFFIC IS THE SUM OF THESE (NON-OVERLAPPING) *kinds*.

traffic kind	Description
UDP	All UDP traffic
web req/ack	TCP traffic sent to port 80, i.e. http requests or TCP acks sent to a web server
web data	TCP traffic from port 80, i.e. data sent to a web browser
KaZaA req/ack	TCP KaZaA file exchange traffic sent to port 1214. See discussion of our OC12 site data for details
KaZaA data	TCP KaZaA file exchange traffic received from port 1214
other TCP	TCP traffic not listed above
other	Protocols other than UDP and TCP

various links and presents some insights as to how mice and elephant populations vary with the character of the link.

We note that previous studies have not to our knowledge separated out the *traffic kinds* within a torrent. Indeed, they have not separated traffic flows into their 'to' and 'from' streams. For applications that move data mainly in one direction, e.g. web and ftp, the data stream will always have many more bytes than its corresponding request/ack stream. As a result, flow distributions will be biased toward a large number of small streams. Using NeTraMet streams provides an effective way to reveal more detail on how the various *traffic kinds* contribute to the overall stream size distributions.

II. MEASUREMENT METHODOLOGY

A. Measuring Stream Size and Lifetime Distributions

We observe streams in real time using the methodology described by Brownlee & Murray [**?**]. We use a NeTraMet meter, modified to observe streams within a flow, to collect data on stream size and lifetime distributions.

Our NeTraMet ruleset (i.e. meter configuration file) classifies packets into flows according to their *traffic kind*. We use the term *kind* to describe both traffic using protocols such as UDP and traffic for various TCP applications. Table **??** lists our various classifications. We refer to these *traffic kinds* throughout this paper.

As listed in the table they are distinct kinds, separating out traffic into non-overlapping groups. However, we observed little KaZaA traffic at our our OC3 and OC48 sites; KaZaA is included in 'other TCP' traffic in our analysis for those sites.

Each stream within a flow is monitored using a dynamic timeout algorithm. The minimum inactive time is two seconds, after which the flow is timed out if it remains inactive for a period equal to 10 times its average packet inter-arrival time. When a stream is timed out, the NeTraMet meter knows its size in packets, size in bytes, and (active) lifetime in microseconds. Each stream's data is used to add a point to their flow's size and lifetime distributions. Since NeTraMet uses RTFM (Realtime Traffic Flow Measurement [**?**]) flows, which are bidirectional, the meter has data for both directions of the flow.

RTFM distribution parameters are: number of bins, upper edge value for lowest and highest bin, and transform to use when binning. For this investigation we use the following parameters:

- Stream size in packets: 50 bins, 1 to 50 packets, linear scale
- Stream size in bytes: 50 bins, 40 Bytes to 40 kBytes, log scale
- Stream lifetimes: 25 bins, 200 ms to 2 s, linear scale

In addition to the specified number of bins, the meter keeps an 'overflow' bin for values above the highest bin. We read flow data (including all the distributions) from our meters at one-minute intervals; the 'reading time' used in our analyses (below) is the time of day when a distribution was read from a meter.

B. Meter Locations

We use NeTraMet to make measurements on high-speed network links at three experimental sites:

OC3: 155 Mb/s ATM link connecting a small University campus to the Internet. Its average aggregate (in + out) data rate is approximately 7 Mb/s.

OC12: 622 Mb/s ATM link connecting a large University campus to the Internet. Average data rate approximately 90 Mb/s.

OC48: 2.4 Gb/s POS Internet backbone link with average data rate approximately 350 Mb/s.

Since RTFM flows provide us with distributions for streams in each direction of a flow, we can distinguish streams for *traffic kinds* to/from servers at each end of our metered links. At our OC3 and OC12 sites one end of the link is physically located inside the site. We describe traffic to/from servers located within our sites as 'inside' traffic and other traffic as 'outside,' and label the traces on our plots accordingly. We separate 'inside' and 'outside' servers in this way so that we can investigate whether traffic patterns to our inside servers differ from patterns for servers elsewhere on the Internet. Preliminary analysis suggests that 'inside' and 'outside' web data distributions can show significant differences, especially at our OC3 site. We will address these differences in a future paper.

C. Data Analysis

C.1 Proportion of traffic by *traffic kind*

We plotted one-minute average packet and byte rates for each *traffic kind* to determine which *kinds* contribute the most bytes and packets to the *torrent*. Stacked plots of traffic rates versus time of day (figures **??**, **??** and **??**) show that the proportions for our *traffic kinds* remain fairly steady over tens of minutes.

C.2 CCSSD plots: testing the 'mice vs elephants' hypothesis

To investigate the contribution of streams of various sizes to a given *traffic kind*, we produce *Complementary Cumulative Stream Size Distributions (CCSSD)* plots using the algorithm described below.

Size distributions indicate the number of streams falling within each bin. We sum a size distribution to produce a cumulative stream-count distribution,

$$S_n = \Sigma_{j=1}^{n} d[j]$$

where j is the count in the jth bin of distribution d.

Within any reading interval, the meter can only count timed-out streams. The total number of these for all the bins (including the overflow bin) is

$$T = S_{b+1}$$

where b is the specified number of bins.

T is only an approximation to the number of streams active within a reading interval. Some streams may have started before the interval and remain active throughout; others may start before the interval and finish during it while some may start during the interval and continue beyond it. If the reading interval is sufficiently long, these 'edge effects' will be minimized (there will be few such streams compared to T).

We integrate the distribution to obtain a cumulative byte or packet distribution,

$$B_n = \Sigma_{j=1}^{n} d[j] \times av[j]$$

where $av[j]$ is the average size, i.e. packet size or geometric mean of bin lower and upper byte sizes for the jth bin of distribution d.

While the meter is counting packets and bytes for each active stream it is simultaneously counting packets and bytes for the flow as a whole, C. C is derived from a single counter in the meter, not from a distribution.

To investigate the way streams of various sizes contribute to the *torrent*, we use cumulative log-log plots of

$$100(1 - S/T) \ \ vs \ \ 100 \times B/C,$$

i.e. *percent timed-out streams uncounted* vs *percent of bytes or packets counted* for each traffic type, and for the torrent as a whole. These are our CCSSD plots.

CCSSD plots (figures **??**, **??** and **??**) show how streams of varying sizes contribute to total byte or packet traffic. Each trace starts near the top left corner, with most streams uncounted and few bytes or packets counted. With increasing values on the x-axis more bytes or packets are counted; at the same time fewer streams remain uncounted. As a result, each trace moves across the plot and decreases on the y-axis.

The size distributions we observe generally appear heavy-tailed, i.e. they have a high peak at a fairly small value, together with a slowly falling tail. UDP flows, for example, typically have a large peak at small sizes with few counts at large values. Web data flows have a peak at small to medium values, and a slowly decreasing tail with significant counts at very large values.

As previously noted, some flows have streams that do not time out within a reading interval. Such streams contribute to the total byte count, C, but not to the timed-out streams count, T. On a CCSSD plot they usually appear as traces that decrease asymptotically toward a byte or packet percentage significantly less than 100%.

We use log-log scales for our CCSSD plots to facilitate identification of features in the heavy tail of the distribution. The byte or packet value for 10% packets uncounted is particularly interesting – flows that satisfy our *mice vs elephants hypothesis* pass through, or to the left of the (10%, 10%) point on a CCSSD plot.

TABLE II

OC3 SITE: TRAFFIC CHARACTERISTICS FOR THE MINUTE ENDING AT 1155, MON 8 OCT 01

Inbound *kind*	Mbit/s	kpkt/s	Byte %	Packet %
other	0.00	0.00	-	-
UDP	0.39	0.09	0.83	4.66
web data	3.72	0.65	29.19	54.54
web req/ack	0.05	0.06	58.71	63.78
other TCP	0.82	0.18	8.22	35.98
total in to site	4.98	0.98	20.83	46.71
Outbound *kind*	**Mbit/s**	**kpkt/s**	**Byte %**	**Packet %**
other	0.00	0.00	-	-
UDP	0.04	0.04	13.09	11.82
web req/ack	0.59	0.53	61.91	72.48
web data	0.32	0.07	44.01	50.56
other TCP	0.65	0.18	15.71	36.83
total out from site	1.60	0.82	29.32	59.00

C.3 Time variation of 90% stream values

Our 'stacked traffic kind' plots show that the traffic mix at each of our sites remains fairly steady over time, suggesting that the CCSSD plots should also remain steady. To investigate variations in our CCSSD plots we use an arbitrary 'streams uncounted' value of 10%. For each one-minute reading interval we determine the percentage of bytes or packets counted and plot these values against time of day (figures **??**, **??** and **??**).

To locate the 10% point we examine the 'streams uncounted' distribution, looking sequentially through the distribution until we find its first bin with 10% or fewer uncounted streams. Our 90% packets or bytes value is then the cumulative distribution value for that bin. Distributions with low counts in their tails, e.g. those for web request/acks, drop quickly (as observed on their CCSSD plots), which makes their 90% plots more variable than those for traffic with heavier-tailed distributions.

III. EXPERIMENTAL RESULTS

A. *OC3 Meter*

A.1 OC3 Traffic Summary

Table **??** summarizes the characteristics of traffic observed at our OC3 site. We separate the traffic into five different *traffic kinds* by our NeTraMet ruleset, as listed in the table's first column.

Table **??**'s next two columns show the bit rate (Megabit/s) and packet rate (kpacket/s) for the measurement interval used for plots later in this section. The last two columns show the percentage of bytes and packets that account for 90% of the timed-out streams observed in the measurement interval.

Our 'other' traffic uses protocols other than UDP and TCP. It does not use port numbers, hence we are unable to distinguish streams within it. Instead we simply count its total numbers of packets and bytes. In table **??** we use a dash to indicate our lack of stream-related detail for 'other' traffic.

For our OC3 site, inbound traffic (top half of table **??**) is dominated by web data, both for bytes (3.72 Mb/s) and packets (650 p/s) with non-web TCP the next most common type of traffic. 90% of the inbound web streams account for about 29% of the web data bytes and 55% of the web data packets.

Non-web TCP traffic has a higher proportion of larger streams. 90% of non-web TCP streams account for only 8% of their data bytes but approximately 36% of their packets. The effect of combining all *traffic kinds* together is that 90% of the incoming streams account for approximately 21% of their data bytes and approximately 47% of their packets.

We draw two significant conclusions from this data:

- The *mice vs elephants hypothesis* is invalid at this site, since the largest 10% of the streams account for only 79% of the bytes, considerably less than the 90% in the hypothesis.
- There is a large difference in the 90% stream coverage for packets as compared with streams – at this site the largest 10% of the streams account for only 53% of the packets.

A.2 OC3 Variation of traffic over time

In this section we present time-series plots of inbound traffic to our OC3 site for approximately one hour of one-minute meter reading intervals. The plots are 'stacked percentage' rates, i.e. the percentage of each *traffic kind* is the difference between its ordinate value and that for the kind below it.

We do not present plots for outbound traffic since Table **??** shows that there is considerably more inbound than outbound traffic.

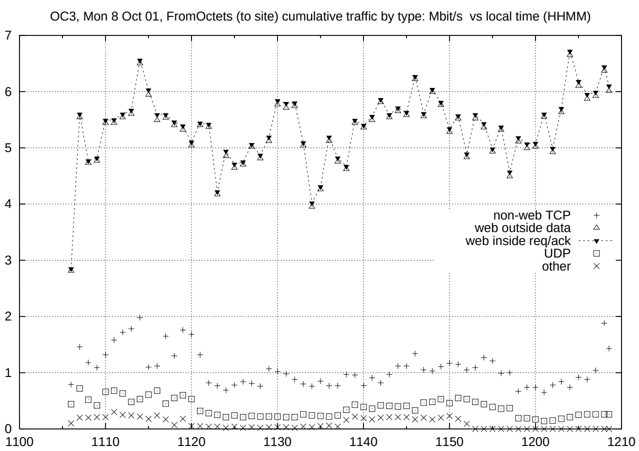

Fig. 1. Inbound byte rates for the various *traffic kinds* observed on our OC3 link. Web traffic is dominant.

From the plots in Figures **??** and **??** we observe that while there is some variation in the data rates for each *traffic kind* from minute to minute, their averages remain fairly constant. In both plots web data is the dominant *traffic kind*, as shown by the large gaps between it (gray triangles) and the next most dominant, non-web TCP (gray pluses). Web requests and acks (gray inverted triangles) appear differently on the two figures. Although they use significant numbers of packets, making them easy to

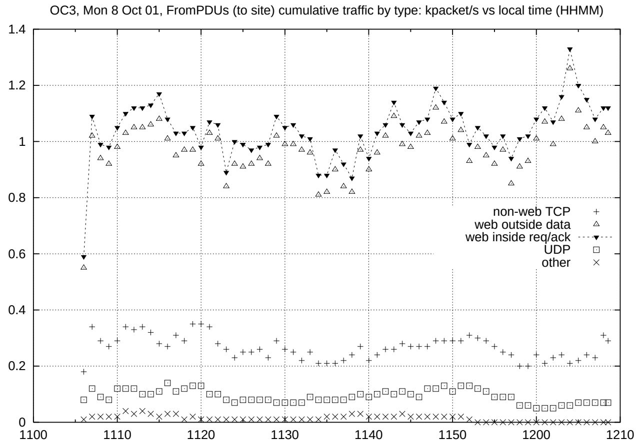

Fig. 2. Inbound packet rates for the various *traffic kinds* observed on our OC3 link. The difference between web request/ack and web data is more apparent for packets than for bytes.

distinguish in Figure **??**, they use almost no bytes in comparison to web data, so they appear superimposed on the web data points in Figure **??**.

A.3 OC3 CCSSD plots

We produced CCSSD plots for several randomly-chosen reading intervals. All these plots were very similar. In this section we present CCSSD plots for the minute ending at 1155, as an example of typical behavior for the link.

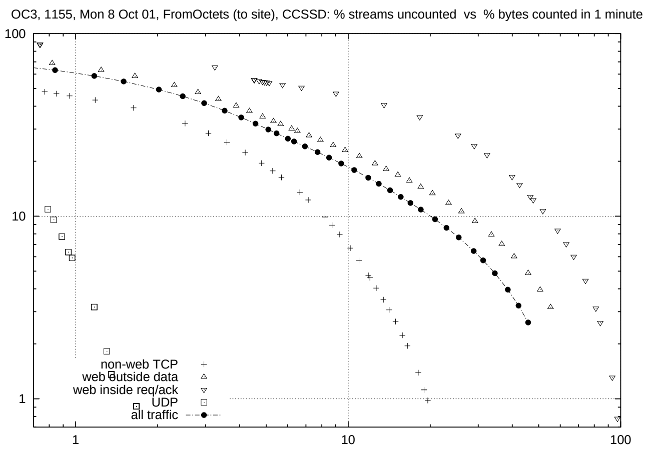

Fig. 3. Byte size distributions for inbound traffic on our OC3 link. The distribution for total traffic lies between web data and 'other TCP'.

Figure **??** shows how streams of various *traffic kinds* contribute to the bytes flowing out from our OC3 site. UDP streams (gray squares) are generally short, but their trace reaches 1% uncounted streams that account for only 2% bytes, implying that there must also be some long-running large streams. Non-web TCP streams (gray pluses) have a higher proportion of large streams. The largest 1% of them carry 80% of the bytes.

Streams of web data flowing to the site (gray triangles) are generally smaller, with 3% carrying 55% of the bytes. Web request/ack streams (gray inverted triangles), responding to web data being sent out from the site, are much smaller. The majority of them time out during the reading interval (their trace heads almost vertically down near 100%).

By combining all flows together (black dots) we see that the total traffic behavior lies between that for non-web TCP and TCP data.

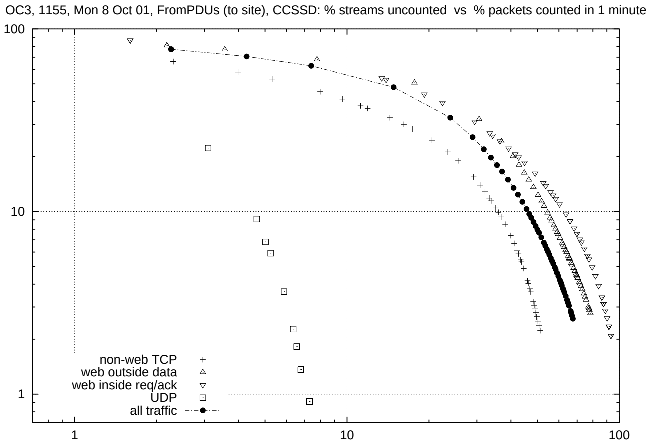

Fig. 4. Packet size distributions for inbound traffic on our OC3 link.

Figure **??** shows how streams of various *traffic kinds* contribute to the packets flowing out from the OC3 site. The traces for each *traffic kind* are similar in shape to those for bytes in figure **??**. However each *traffic kind's* trace is displaced to the right, indicating that each *kind's* larger streams account for a higher percentage of packets than they do for bytes.

A.4 OC3 bytes/packets for 90% streams

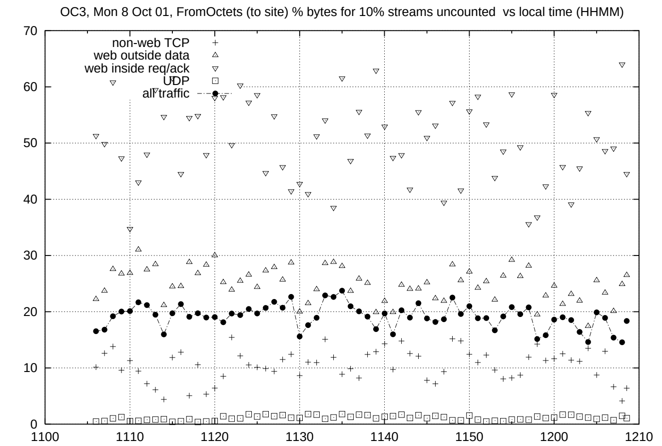

Fig. 5. Byte coverage for 90% of the inbound streams on our OC3 link for each *traffic kind*. Although the percentages vary from minute to minute, their long-term averages are fairly stable

Figures **??** and **??** show the percentage of bytes and packets accounted for by 90% of the timed-out streams in each one-minute reading interval.

For *traffic kinds* other than web requests and acks, the byte and packet percentages are fairly steady. This steadiness is particularly obvious for the total traffic trace (black dots).

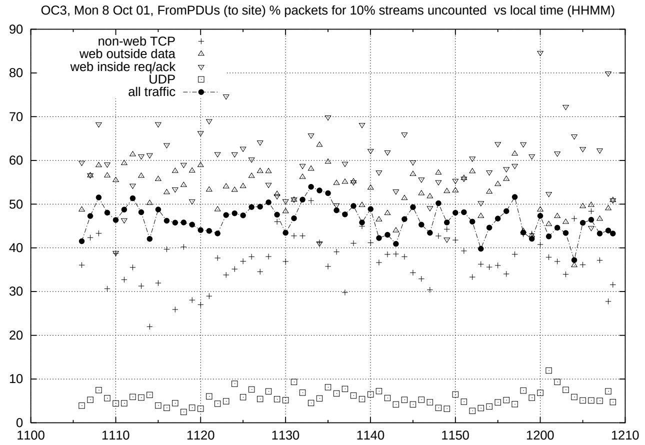

Fig. 6. Packet coverage for 90% of the inbound streams on our OC3 link for each *traffic kind*.

TABLE III

OC48 SITE: TRAFFIC CHARACTERISTICS FOR THE MINUTE ENDING AT 1407, SUN 7 OCT 01

Inbound *kind*	Mbit/s	kpkt/s	Byte %	Packet %
other	1.18	1.88	-	-
UDP	10.27	6.88	5.46	15.10
web data	126.71	18.46	10.29	43.99
web req/ack	20.36	23.94	49.42	50.86
other TCP	75.51	22.66	2.50	15.10
total in to site	234.03	73.82	9.39	33.05
Outbound *kind*	**Mbit/s**	**kpkt/s**	**Byte %**	**Packet %**
other	3.86	0.95	-	-
UDP	18.11	4.49	2.27	18.08
web req/ack	5.75	6.22	61.97	83.05
web data	99.67	15.46	3.94	43.99
other TCP	41.21	8.64	1.66	19.13
total out from site	168.60	35.76	4.67	31.46

Web requests and acks are much more variable, most likely because their streams carry little data, e.g. http GET requests, and TCP ACKs with no data, producing distributions with sparse tails.

B. OC48 Meter

B.1 OC48 Traffic Summary

Table **??** summarizes the characteristics of traffic observed at our OC48 site. We segregate the traffic into five different *traffic kinds* by our NeTraMet ruleset. The final two columns show the bit rate (Megabit/s) and packet rate (kpacket/s) for the measurement interval used for plots later in this section. The last two columns show the percentage of bytes and packets that account for 90% of the timed-out streams observed in the measurement interval.

Our OC48 site is located within a backbone network where our meter taps the link's two fibers. For this link there is no obvious 'site' to assign the meter ruleset's *to* and *from*. As a result, the *to* and *from* directions in table **??** are initially chosen arbitrarily.

For this site, the traffic in both directions is dominated by web data (127 and 100 Mb/s) with non-web TCP the next most common. In this respect the link is similar to our OC3 link. On our OC48 link, however, the traffic rates are much higher, the aggregate rate (in + out) is 403 Mbps and 110 kp/s.

On this link, 90% of the web data streams accounted for 10% *from* and only 4% *to* for bytes. This difference suggests that the *to* web data carried a higher proportion of large web objects. Non-web TCP traffic ('other TCP' in the table) has a higher proportion of large streams, bringing its 90% stream coverage to 2.5and 1.7% *to* for bytes.

For packets, however, the 90% coverage figures are reduced, being 15% *from* and 19% *to* for non-web TCP, and poorer still for web data.

For this site we conclude that:

- The *mice vs elephants hypothesis* holds for bytes for the total link traffic, with 90% of the streams accounting for only 9.4% *from* and 4.7% *to* of the total traffic bytes.
- The hypothesis does not hold for bytes for some *traffic kinds*, as shown in table **??**
- The hypothesis does not hold for packets for *any* kind of traffic.

B.2 OC48 Variation of traffic over time

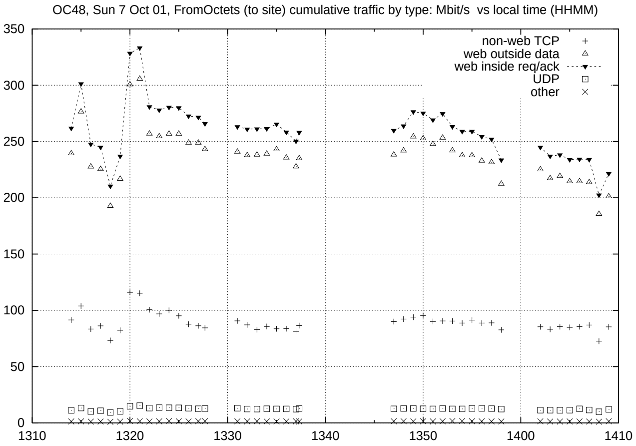

Fig. 7. Inbound byte rates for the various *traffic kinds* observed on our OC48 link. Web traffic is dominant.

The plots in Figures **??** and **??** show how traffic rates vary for the various *traffic kinds*. The meter on this link uses prototype OC48 measurement cards and drivers for them [**?**]. We have carefully optimized the meter code to handle high packet rates, but at the time we made these measurements, the meter was unable to cope with the load for more than between 10 and 15 consecutive minutes, depending upon the amount of traffic. Whenever the meter was overwhelmed, we restarted it. This cycling produces the noticeble gaps in data collection, reflected in our plots. We are continuing to refine the code to allow the meter to gather longer continuous runs of data.

The data rates for the various *traffic kinds*, i.e. the vertical separations between traces on the plots, are steady, somewhat more

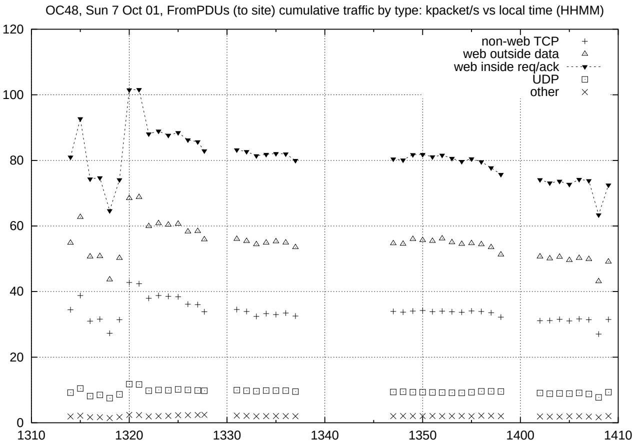

Fig. 8. Inbound packet rates for the various *traffic kinds* observed on our OC48 link.

than those for our OC3 link (figures **??** and **??**). The difference between web data and web request/ack traffic for packets and bytes is clearly visible. Figure **??** shows that the web data byte rate is much higher than the web request/ack data byte rate, whereas in figure **??** we see that these two kinds have similar packet rates.

The difference between packet and byte stream sizes for web traffic arises because the TCP protocol requires at least half its data packets to be acknowledged, setting a limit to the difference in data and request/ack stream sizes in packets. There is no such limitation on byte sizes, however, and web traffic carries data to a browser, but no data – only ack flags – back to a server. As a consequence, we see big differences in stream byte size for web data as compared to web request/acks.

In the following section we analyze CCSSD plots for the minute ending at 1407 (after the meter had been running for six consecutive minutes and two minutes before it was overwhelmed). These data reflect typical behavior for this link.

B.3 OC48 CCSSD plots

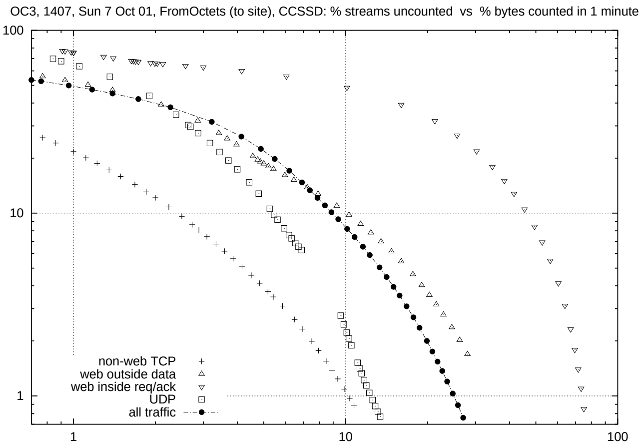

Fig. 9. Byte size distributions for inbound traffic on our OC48 link.

Figure **??** shows how streams of various kinds contribute to the overall traffic on the link. The UDP trace (gray squares) starts at about 70% streams uncounted, implying that 30% of the UDP streams account for less than 1% of the UDP bytes. The UDP trace falls moderately fast, suggesting that UDP traffic on this link has a higher proportion of large streams than that on our OC3 link (figure **??**).

Non-web TCP (gray crosses) and web data (gray triangles) have a higher proportion of small streams than UDP since their traces start further down the plot. The shape of traces in figure **??** is similar to that of the corresponding traces for our OC3 link, indicating that the distribution of large streams for these *traffic kinds* is similar to that on the OC3 link.

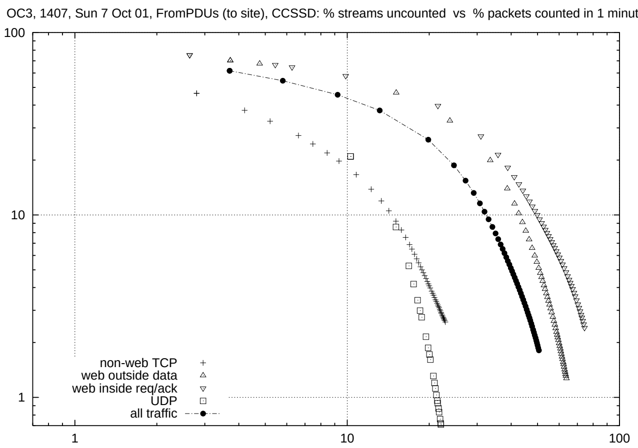

Fig. 10. Packet size distributions for inbound traffic on our OC48 link.

The overall traffic trace (black dots) again lies mostly between those for non-web TCP and web data.

Figure **??** shows how streams of various *traffic kinds* contribute to the packets flowing out from the OC3 site. As in figure **??**, most of the traces are displaced to the right, indicating that large streams account for a higher proportion of packets than they do for bytes.

B.4 OC48 bytes/packets for 90% streams

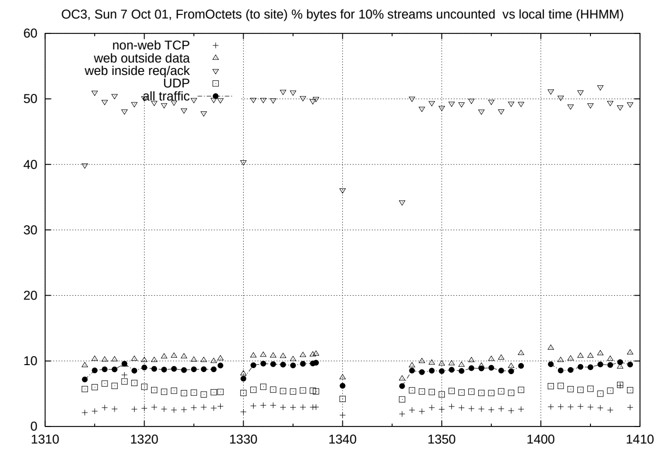

Fig. 11. Byte coverage for 90% of the inbound streams on our OC48 link for each *traffic kind*.

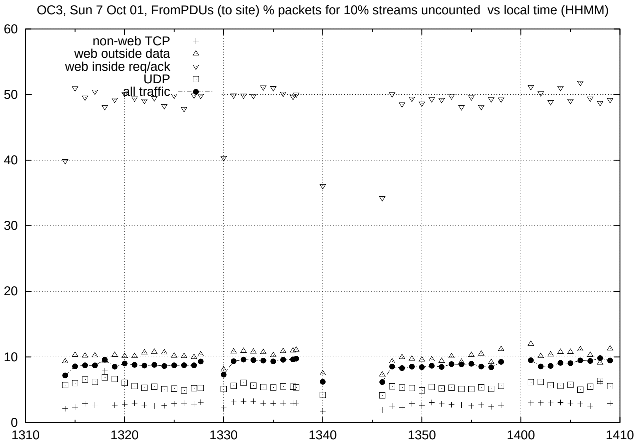

Fig. 12. Packet coverage for 90% of the inbound streams on our OC48 link for each *traffic kind*.

Figures **??** and **??** show the percentage of bytes and packets accounted for by 90% of the timed-out streams in each one-minute reading interval. Overall the traces are steady, more so than those for our OC3 link. As for the OC3 link, the web request/ack traffic shows the greatest variation in byte and packet coverage by 90% of the timed-out streams.

C. OC12 Meter

C.1 OC12 Traffic Summary

Traffic at this site is qualitatively different from our other two sites. Early measurements for this site produced ‘stacked rate’ plots that were dominated by non-web TCP, with little web traffic, and CCSSD plots which appeared different from those for our OC3 and OC48 sites.

We ran a test ruleset on our OC12 meter to determine which applications contributed most to the non-web TCP traffic, and found that a significant portion of the non-web TCP uses port 1214. Port 1214 is used by KaZaA [**?**] servers. KaZaA is a peer-to-peer file sharing protocol, designed as a replacement for Napster, but more general in that it handles many file types and more distributed in that it does not rely on a single central directory server.

We re-wrote our distribution-gathering ruleset to treat KaZaA as a separate *traffic kind*, and collected a new set of flow data. We have observed periods when KaZaA traffic has accounted for 60% of the bytes in our OC12 torrent, but such periods are unusual. The data presented in this section covers about three hours on a weekday.

Table **??** summarizes characteristics of traffic observed at our OC12 site. We segregate traffic into seven different kinds by our NeTraMet ruleset, as indicated in the table’s first column. The table’s next two columns show the bit rate (Megabit/s) and packet rate (kpacket/s) for the measurement interval used for plots later in this section. The last two columns show the percentage of bytes and packets that account for 90% of the timed-out streams observed in the measurement interval.

On this OC12 link only 20% of the traffic is web, and approximately 10% of it is KaZaA. However, nearly half the link’s traffic is TCP for applications other than web and KaZaA. Since

TABLE IV

OC12 SITE: TRAFFIC CHARACTERISTICS FOR THE MINUTE ENDING AT 1430, MON 29 OCT 01

Inbound *kind*	Mbit/s	kpkt/s	Byte %	Packet %
other	0.19	0.15	-	-
UDP	11.74	3.57	1.08	5.16
KaZaA data	3.39	0.95	2.99	36.71
KaZaA req/ack	3.03	0.76	2.31	48.31
web data	11.93	1.90	24.82	48.37
web req/ack	0.32	0.53	27.76	78.82
other TCP	30.65	5.16	0.64	10.95
total in to site	61.25	13.02	2.31	20.98

Outbound *kind*	Mbit/s	kpkt/s	Byte %	Packet %
other	0.16	0.12	-	-
UDP	0.90	0.86	10.04	21.41
KaZaA req/ack	3.02	0.78	2.18	23.73
KaZaA data	0.47	0.60	19.83	46.75
web req/ack	2.30	2.24	56.99	66.21
web data	2.94	0.44	4.14	53.76
other TCP	17.97	4.97	1.02	10.95
total out from site	27.76	9.99	6.02	27.00

there is more traffic inbound from this site, we focus on its *from* traffic in the following sections.

For 'other TCP' traffic, 90% of the timed-out streams account for less than 1% of the bytes and nearly 11% of the packets. These low percentages suggest that this *traffic kind* has a high proportion of large streams.

Conversely, web data depends much more upon small streams, 90% of its timed-out streams account for approximately 25% of the bytes, and 48% or more of the packets.

KaZaA's 90% values lie closer to those for 'other TCP' than for web, suggesting that larger streams contribute a high proportion of the total KaZaA bytes,

From our OC12 link data we draw the following conclusions:

- 'Other TCP' is the dominant traffic kind.
- KaZaA and other non-web *traffic kinds* satisfy the *mice vs elephants hypothesis*.
- Web traffic does not satisfy the hypothesis.

C.2 OC12 Variation of traffic over time

The plots in Figures **??** and **??** show how traffic rates vary for the various *traffic kinds*. 'Other TCP' traffic (gray pluses) is the dominant *kind*, KaZaA appears above it (gray diamonds), and web traffic appears at the top. The most striking feature of these plots is that KaZaA (gray circles) contributes a significant proportion of the link's byte rate and packet rate.

During this measurement run, the proportions of each traffic kind show short-term variations over periods of one to five minutes but overall they remain steady. The UDP and 'other' traffic is steady, and TCP traffic varies more, showing perturbations over periods of five to ten minutes.. In the following section we show CCSSD plots for the minute ending at 1430. These show typical behavior for this link.

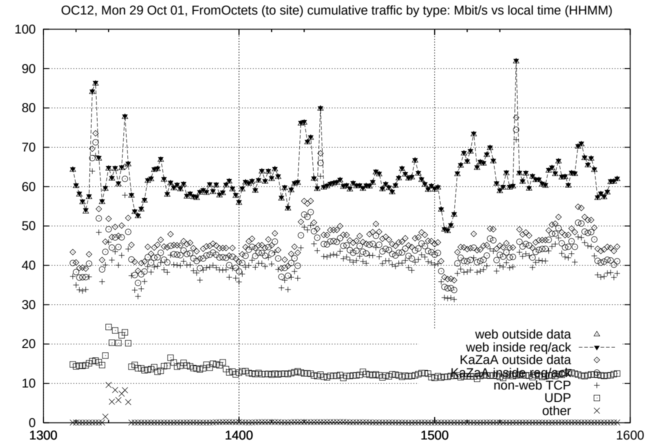

Fig. 13. Inbound byte rates for the various *traffic kinds* observed on our OC12 link. 'Other TCP' traffic is dominant, followed by web and KaZaA.

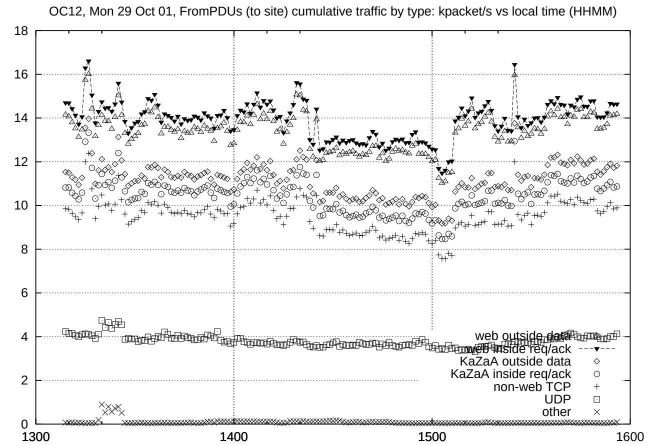

Fig. 14. Inbound packet rates for the various *traffic kinds* observed on our OC12 link.

C.3 OC12 CCSSD plots

Figure **??** shows how streams of various kinds contribute to the overall byte traffic flowing in to our site over the OC12 link.

Non-web TCP (gray pluses) has a high proportion of small streams; it appears on the plot with 10% of streams covering less than 0.7% of the bytes. Its trace declines steadily, indicating that it also has a high proportion of larger streams.

Web data (gray triangles) with about 45% of the timed-out streams covers only 0.7% of its bytes. Web data also has many larger streams, so its trace meanders across the plot until it turns down toward only 50% of the web data bytes.

KaZaA data traffic appears at the left of the plot (gray diamonds), at the same point as the web data. However, the KaZaA data trace turns down earlier than web data, with only about 15% of its bytes covered. This early downturn indicates that KaZaA's distribution has a heavier tail than that for web data.

KaZaA req/ack traffic is similar, except that it turns down earlier than its data, reaching only 7% byte coverage. KaZaA traffic entering the KaZaA servers at this site has a heavier tail than its data coming into the site from external KaZaA servers.

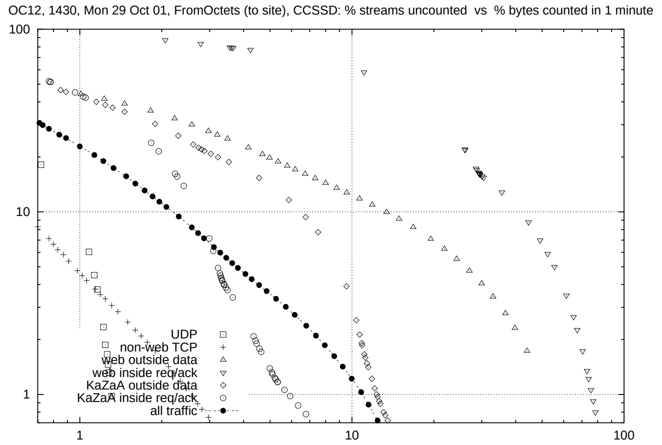

Fig. 15. Byte size distributions for inbound traffic on our OC12 link. KaZaA data and request/ack traces remain closer together than web data and request/ack, suggesting that KaZaA moves data both in and out, unlike web which only moves data in.

From this analysis of the KaZaA traces we believe that KaZaA traffic is less asymmetric than web traffic, i.e. it carries a high byte volume in both directions. (Compare this behavior with web traffic, which carries high volumes of data from server to client, but only small volumes (requests and acks) from client to server.

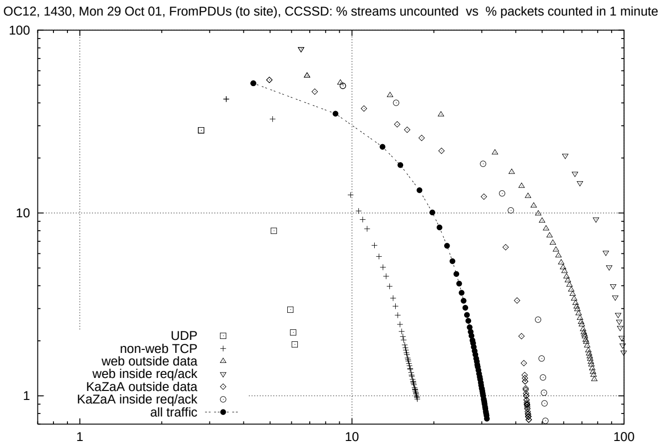

Fig. 16. Packet size distributions for inbound traffic on our OC412 link.

Figure **??** shows how streams of various *traffic kinds* contribute to the packets flowing into our OC12 site. Although their traces for each kind are similar in shape to those for bytes in figure **??**, they are shifted to larger values for both x- and y-axis values, suggesting that their small streams account for a smaller percentage of their packets than for their bytes, and also that their packet size distributions are more heavily weighted in their tails than are their byte size distributions.

C.4 OC12 bytes/packets for 90% streams

Figures **??** and **??** show the percentage of bytes and packets accounted for by 90% of the timed-out streams in each one-minute reading interval. Overall the values for each *traffic kind*

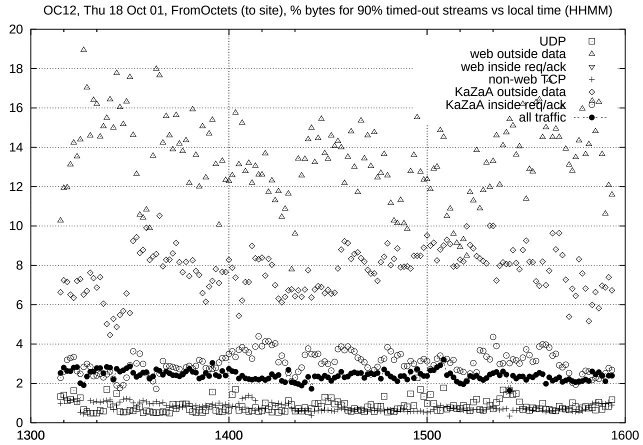

Fig. 17. Byte coverage for 90% of the inbound streams on our OC12 link for each *traffic kind*.

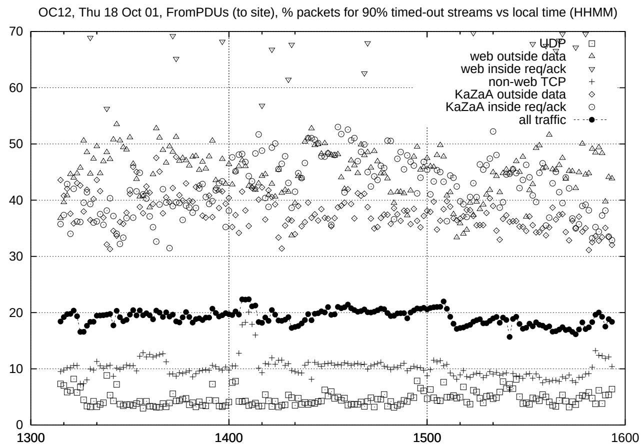

Fig. 18. Packet coverage for 90% of the inbound streams on our OC12 link for each *traffic kind*.

are steady, particularly the value for the total traffic. KaZaA and web traffic show greater variation in their 90% coverage levels, reinforcing our hypothesis that their distributions have less weight in their tails, i.e. fewer large streams.

IV. Conclusions and Future Work

CCSSD plots provide a useful overview of stream behavior on a link. More work is needed to develop a better understanding of how to interpret the traces for each kind of traffic. It would be useful to understand stream distributions directly from the traces, for example by making CCSSD plots for simulated traffic mixes.

Our measurements show that the *mice vs elephants hypothesis* holds at some sites (OC48 and OC12) for bytes, but not at our OC3 site. The percentage of packets covered by 90% of timed-out streams is much higher than for bytes. For packets, the hypothesis failed for total traffic at all three sites. UDP traffic at our OC12 site was the only *traffic kind* for which the hypothesis is valid.

New protocols can rapidly change the overall character of traffic on a link, as observed on our OC12 link. Such protocols may have different stream distributions from the link's previous traffic. In the case of the OC12 data, the 'other TCP' traffic has a high proportion of large streams, KaZaA's traffic has a less heavy tail, and web traffic has comparatively few large streams.

Overall, 'total traffic' stream distributions are strongly influenced by the mix of *traffic kinds*. Although the *mice vs elephants hypothesis* is a useful approximation, researchers should determine the stream behavior on any link by measuring it prior to committing to any monitoring or sampling system that relies on the hypothesis.

Finally, do mice *really* matter? In terms of bytes our answer is equivocal. In terms of packets, however, they *do* matter. On the three links we have measured, 90% of the streams account for as much as 50% of the total packets. This high proportion of mice within the total packet load on a link supports the notion that network queue management should perhaps try to give mice priority on the Internet, so as to allow the mice to break up synchronization effects, leading to a more efficient use of network resources.

Mice could also be important for billing. Where link costs are high, providers tend to charge for bytes transferred, since these relate directly to bandwidth consumed. However, in situations where link costs are low, for example large campus networks using 100 Mbps or 1 Gbps Ethernet, the purchase and maintenance of network switches can easily dominate the network's operating costs. One possible way to recover those costs would be to charge for packets transferred. Since 50% of the packets can be mice, they could not be safely ignored under such a cost recovery model.

We are continuing to develop our traffic meter, especially to improve its performance on very high speed links. We will periodically repeat these measurements in order to track medium- and long-term changes in the character of traffic on these links.

REFERENCES

[1] N. Duffield, C. Lund, and M. Thorup, *Charging from Sampled Network Usage*, ACM SIGCOMM Internet Measurement Workshop, November 2001

[2] Dag website, http://dag.cs.waikato.ac.nz

[3] A. B. Downey, *The structural cause of file size distributions*, MASCOTS Symposium, 2001, available at http://rocky.wellesley.edu/downey/filesize/

[4] N. Brownlee, *Using NeTraMet for Production Traffic Measurement*, Intelligent Management Conference (IM2001), May 2001

[5] KaZaA website, http://www.kazaa.com/

[6] N. Brownlee, kc. claffy, M. Murray and E. Nemeth, *Methodology for Passive Analysis of a University Internet Link*, PAM2001 Workshop, April 2001

[7] C. Estan and G. Varghese, *New Directions in Traffic Measurement and Accounting*, ACM SIGCOMM Internet Measurement Workshop, November 2001

[8] Y. Zhang and L. Qiu, *Understanding the End-to-End Performance Impact of RED in a Heterogeneous Environment*, Cornell CS Technical Report 2000-1802, July 2000, available from http://www.aciri.org/floyd/red.html

[9] N. Brownlee, C. Mills and G. Ruth, *Traffic Flow Measurement: Architecture*, RFC 2722, October 1999

[10] N. Brownlee and M. Murray, *Streams, Flows and Torrents*, PAM2001 workshop, April 2001

[11] Y. Joo, V. Ribeiro, A. Feldmann, A. C. Gilbert and W. Willinger, *On the impact of variability on the buffer dynamics in IP networks*, Proc. 37th Annual Allerton Conference on Communication, Control and Computing, September 1999, available from http://www.dsp.rice.edu/publications

[12] L. Guo and I. Matta, *The War Between Mice and Elephants*, Technical Report BU-CS-2001-005, Boston University, Computer Science Department, May 2001, available at http://www.cs.bu.edu/techreports/2001-005-war-tcp-rio.ps.Z