Some Observations of Internet Stream Lifetimes

Nevil Brownlee

CAIDA, UC San Diego, and
The University of Auckland, New Zealand
nevil@auckland.ac.nz

Abstract. We present measurements of stream lifetimes for Internet traffic on a backbone link in California and a university link in Auckland. We investigate the consequences of sampling techniques such as ignoring streams with six or fewer packets, since they usually account for less than 10% of the total bytes. We find that we often observe large bursts of small 'attack' streams, which will diminish the integrity of strategies that 'focus on the elephants'. Our observations further demonstrate the danger of traffic engineering approaches based on incorrect assumptions about the nature of the traffic.

1 Introduction

Over the last few years there has been considerable interest in understanding the behaviour of large aggregates of Internet traffic flows. *Flows* are usually considered to be sequences of packets with a 5-tuple of common values (protocol, source and destination IP addresses and port numbers), and ending after a *fixed timeout* interval when no packets are observed. For example, Estan and Varghese [1] proposed a method of metering flows which ensures that all packets in *elephant* flows, i.e. those that account for the majority of bytes on a link, are counted, while packets in less significant flows may be ignored.

In contrast, *streams* are *bi-directional* 5-tuple flows, ending after a *dynamic timeout* interval of at least 10s and terminating after a quiet period of ten times their average packet inter-arrival time. Brownlee and Murray [2] investigated stream lifetimes, using a modified NeTraMet [3] meter. By using streams rather than flows, NeTraMet is able to measure various stream distributions at regular intervals (typically five or 10 minutes) over periods of hours or days. In [4] Brownlee and Claffy used this methodology to observe stream behaviour at UC San Diego and Auckland, where about 45% of the streams were *dragonflies* lasting less than two seconds. However, there were also many streams with lifetimes of hours to days, and those *tortoises* carried 50% to 60% of the link's total bytes.

At U Auckland, we use NeTraMet to measure Internet usage (bytes in and out for each user). In recent years the character of our Internet traffic has changed; the total volume has steadily grown, and we now see frequent network-borne attacks. Such attacks frequently appear as short time intervals during which we see large numbers of *dragonfly* streams. With our production NeTraMet rulesets (meter configuration files), attacks like address scans can give rise to tens of

thousands of flows. Such large bursts of flows tend to degrade the performance of our measurement system.

To minimise the effect of bursts of 'attack' streams, we investigated a strategy similar to that proposed by Estan and Varghese [1]. To do that we modified NeTraMet to ignore streams carrying K or fewer packets. That, however, posed the question of choosing a value for K.

In this paper we present some observations of stream lifetimes on a tier 1 backbone in California, which are consistent with earlier work by Brownlee and Claffy [4], and compare them with similar recent observations at Auckland.

We present measurements of the varying population of active streams at Auckland and compare that with the packet rate, using data gathered at one-second intervals over several days.

We investigate the proportion of the total bytes accounted for by streams with K or fewer packets, so as to help determine a suitable value for K. We often see measurement intervals when a high proportion of the total traffic is carried in *dragonfly* streams; for such intervals there are few *elephant* streams.

Lastly, we show that ignoring streams with six or fewer packets can provide effective usage monitoring for U Auckland.

2 Methodology, 'Overall' Traffic Observations

2.1 Understanding Flows and Streams

Traffic Flows were first defined in the seminal paper by Claffy, Polyzos and Braun [5]. A *CPB flow* is a set of packets with common values for the 5-tuple (IP protocol, Source and Destination IP Address and Port Number), together with a specified, fixed inactivity timeout, usually 60 seconds. Note that a CPB flow is unidirectional, with the 5-tuple specifying a direction for the flow's packets. CPB flows are widely used, providing a convenient way to summarise large volumes of Internet traffic data.

The IETF's RTFM architecture [6] provided a more general definition of a traffic flow. RTFM flows are bidirectional, with any set of packet attribute values being allowed to specify a flow. For example, an RTFM flow can be as simple as a CPB flow, or something more complex such as "all flows to or from network 192.168/16."

NeTraMet is an RTFM traffic measuring system that implements an extended version of RTFM flows. Streams were introduced to NeTraMet as a way of collecting data about subsets of a flow. For example, if we specify a flow as "all packets to/from a particular web server," then NeTraMet can recognise a stream for every TCP connection to that server, and build distributions of their sizes, lifetimes, etc.

NeTraMet's ability to handle streams in real time allows us to produce stream density distributions (e.g. lifetime and size in bytes or packets) over long periods of time – eight hours or more – while maintaining stream lifetime resolution down to microseconds. Furthermore, NeTraMet can collect such distributions at

5-minute intervals for days, without needing to collect, store and process huge packet trace files.

Although streams are bidirectional, that only means that NeTraMet maintains two sets of counters, one for each direction of the stream. If the meter can only see one direction of the stream, one set of counters will remain at zero. Bidirectional streams are, however, particularly useful for security analysis, where we need to know which attack streams elicited responses from within our network!

2.2 Streams in NeTraMet

From our earlier study of stream lifetimes [4] we know that a high proportion of traffic bytes are carried in *tortoise* streams. We modified the NeTraMet meter to use this fact to cache flow matches for each stream. The meter always maintains a table of active streams; when a new stream appears it is matched so as to determine which flow(s) it should be counted in. The set of matching flows is cached in the stream table, so that later packets can be counted in their proper flows without requiring further matching; we find that for most rulesets, average cache hit rates are usually well above 80%.

Since NeTraMet is now based on stream caching, it is straightforward to collect distributions of byte, packet and stream density, using a set of bins to build histograms for a range of stream lifetimes. We use 36 bins to produce distributions for lifetimes in a log scale from 6 ms to 10 minutes, and read these distributions every ten minutes.

Streams are only counted when they time out, so longer-running streams do not contribute to our distributions directly. Instead we create flows for them, so that they produce flow records giving the number of their packets and bytes every time the meter is read. From those 10-minute flow records we construct two more decades of logarithmic bins, producing lifetime distributions from 6 ms to 30,000 seconds (roughly 8 hours), i.e. nearly seven decades.

2.3 Tier-1 Backbone in California, December 2003

Fig. 1 gives an overview of traffic on a tier-1 OC48 backbone in California over Friday, 6 December 2003. Only one direction is shown, the other direction had about one-quarter the traffic volume. There is a clear diurnal variation from about 450 to 700 Mb/s. Most of the traffic is web (upper half of bars) or non-web TCP (lower half), plus a background level of about 50 Mb/s of UDP and other protocols.

Fig. 2 shows the stream density vs lifetime (upper left traces) for every 10-minute reading interval. There is little variation, and about 95% of all streams have lifetimes less than ten seconds. The lower right traces, however, show stream byte density vs lifetime. Again there is little variation, but only 60% of the bytes are carried by streams with lifetimes less than 1000 s. In other words, most streams are short but the bulk of the bytes are carried in long-running streams.

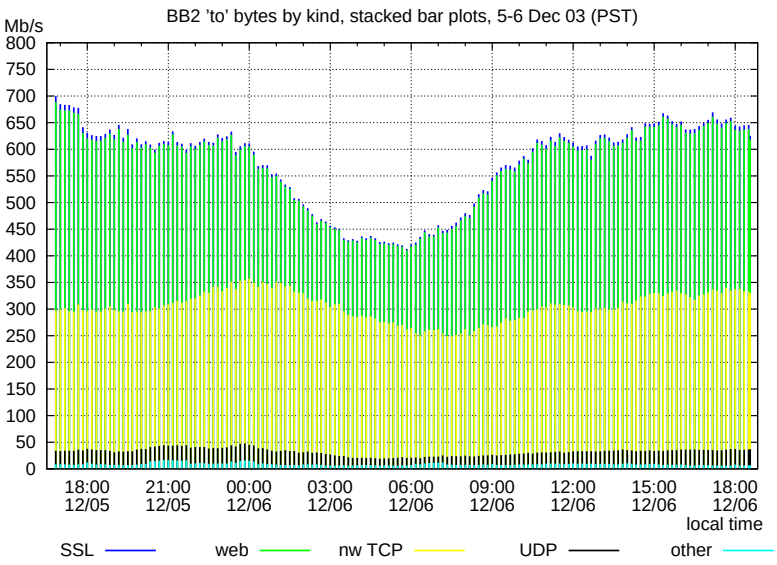

Fig. 1. Stacked-bar plot of traffic on an OC48 backbone in California

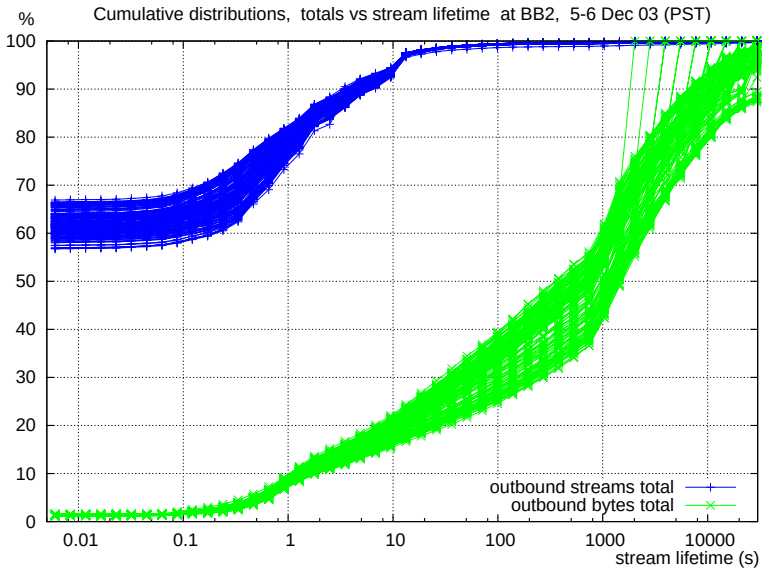

Fig. 2. Stream lifetimes for traffic on a tier-1 backbone in California

2.4 U Auckland Gateway, October 2004

Fig. 3 shows the traffic on U Auckland's 100 Mb/s Internet gateway for Friday-Saturday 1-2 October 2004. There is only around 15 Mb/s of traffic, and it is rather bursty, probably because the total rate is low. During the day web traffic dominates, especially on Friday. In the evenings there are periods of high non-web TCP usage when we update local mirrors for databases outside New Zealand.

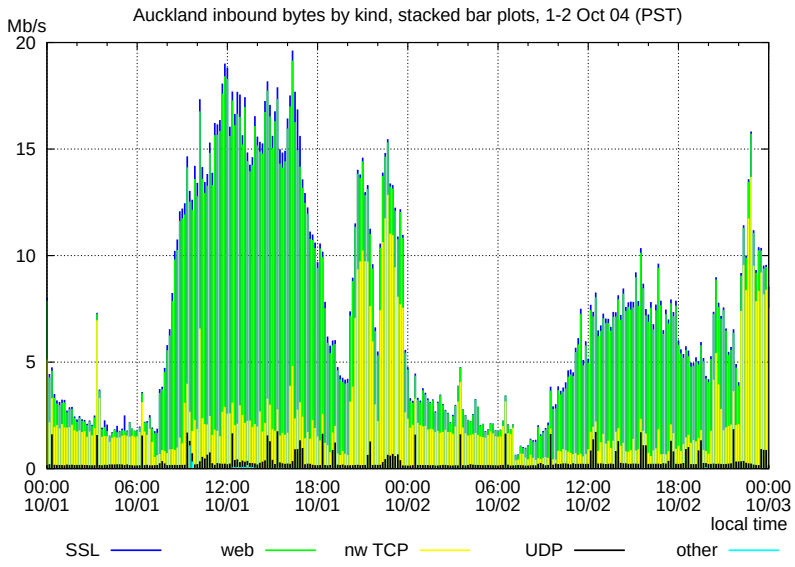

Fig. 3. Stacked-bar plot of traffic on the U Auckland (100 Mb/s) gateway

Fig. 4 shows the stream density vs lifetime as for fig. 2. Here the stream lifetime and byte densities vary greatly, again reflecting the low traffic levels at Auckland. Stream lifetimes are similar at Auckland and California, with 70% to 95% of the streams again lasting less than 10 seconds. However, at Auckland up to 60% of the bytes are carried in streams lasting only 10 seconds; probably reflecting the high proportion of web traffic at Auckland.

3 Streams and Packets at Auckland

We modified NeTraMet to write the packet rate and number of active streams and flows to a log file every second. Fig. 5 shows the packet rate (lower trace) and number of active streams (upper trace) for each second during Friday 1 and Saturday 2 October 2004.

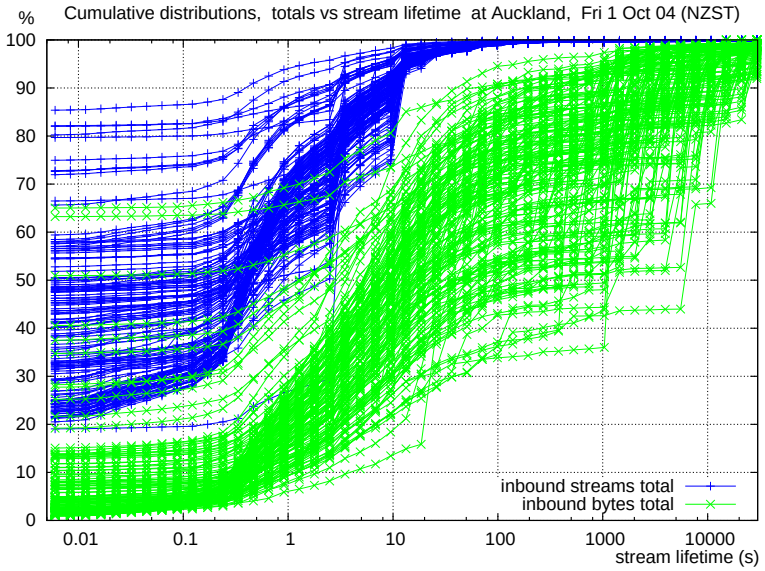

Fig. 4. Stream lifetimes for traffic on the U Auckland (100 Mb/s) gateway at ten-minute intervals for Friday, 1 October 2004

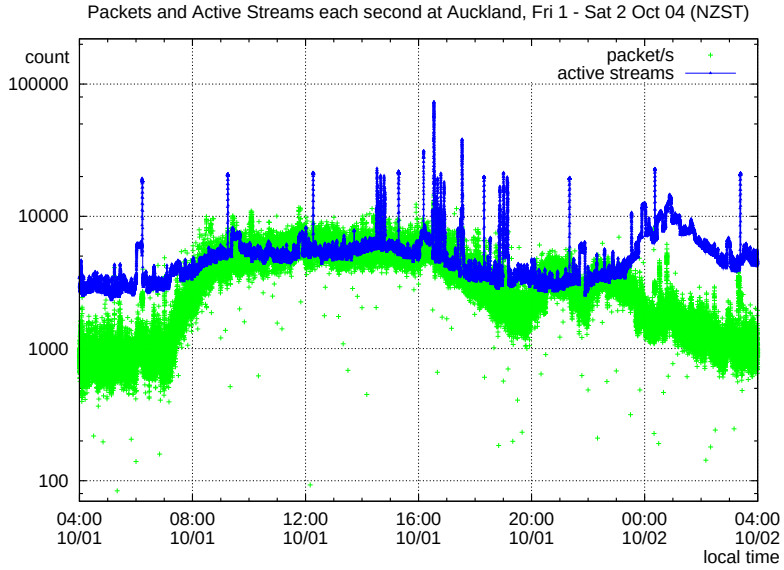

Fig. 5. Packet rate and number of active streams at one-second intervals at Auckland for Friday 1 October 2004

The diurnal variation in stream numbers generally follows the variation in packet rate, i.e. it rises from about 0600 to 0900, falls from about 1700 to 2000, then rises again in the evening. Unlike the packet rate, however, the number of streams rose while the traffic rate fell around midnight on Friday 1 Oct 04. That rise was not repeated over the weekend; it appears to have been a one-off event (e.g. a database replication job copying many tiny files) rather than part of the diurnal pattern.

At regular three-hour intervals we see a short, high step in the number of streams. Our network security team were well aware of this; they are investigating. We believe that such steps are caused by some sort of network attack. Similarly, every day at 1630 we see a bigger spike. We have also observed other, less regular, spikes taking the number of active streams as high as 140,000. Fig. 6 shows more detail for two of these spikes.

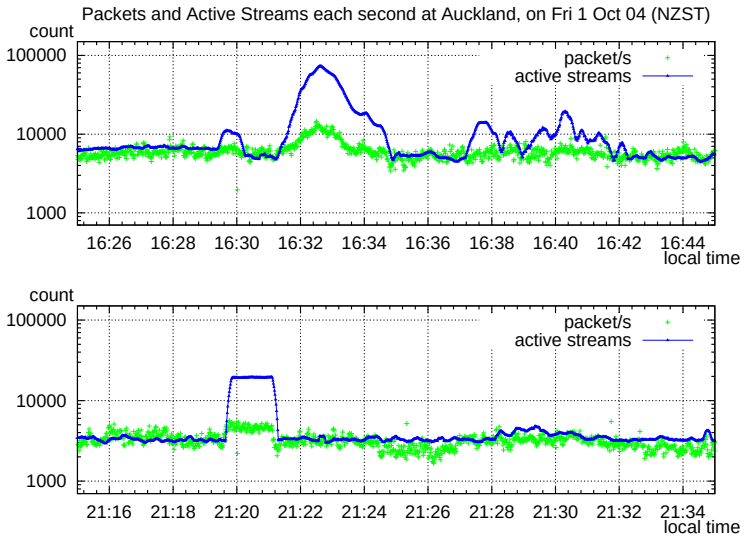

Fig. 6. Details of fig. 5 showing spike in streams at 1633, and step at 2120

4 Usage Metering at Auckland

For usage accounting at Auckland we want to ignore streams with K or fewer packets. To help select a K value, we plotted distributions of byte density vs stream size (packets). Fig. 7 shows distributions for inbound (lower traces) and outbound (upper traces) byte-percentage distributions for ten-minute sample intervals from three hours from 2100 on Friday 1 October 2004. For most of those intervals it seems that we could ignore streams with six or fewer packets in

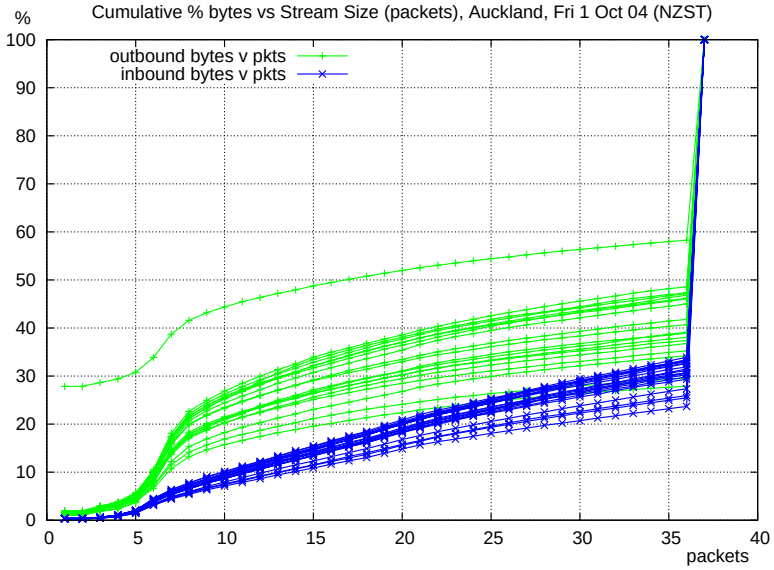

Fig. 7. Byte density vs packets in stream for three hours at Auckland, from 2100 on Friday, 1 Oct 2004

either direction. However, there is one *outbound* trace, for the interval ending at 2120, which has 29% of its bytes in streams with only one or two packet. Fig. 6 shows that at 2120 the number of active streams had risen sharply.

Table 1 shows that the interval ending at 2120 had two unusual features: a high inbound UDP traffic rate, and a low outbound non-web TCP traffic rate. We examined the ten intervals with their highest proportion of bytes in short streams. Few of those had low outbound non-TCP rates, but all had high UDP inbound rates. We hypothesise that the step in streams was caused by an inbound address or port scan, i.e. a flood of single-packet UDP streams. Although few of those inbound UDP probe packets elicited any response, those that did increased the proportion of bytes in small streams enough to dominate the outbound traffic.

Since U Auckland has about five times as many inbound traffic bytes as it does outbound, we plotted the total (inbound+outbound) byte-percentage distributions for every ten-minute interval over 1-2 October 2004, producing fig. 8. We often see intervals when the short streams contribute a significant proportion of the total link bytes, suggesting that we should not simply "focus on the elephants" for our usage measurements.

5 Ignoring Short Streams at Auckland

Our observations in section 4 suggest that on our link, intervals when traffic is dominated by short streams are caused by network attacks *(plagues of dragon-*

Table 1. Inbound and outbound traffic rates (Mb/s) for various Kinds of traffic on Friday 1 October 2004

Inbound rate	UDP	non-web	web	SSL	other
2110	0.15	2.91	8.85	0.51	0.03
2120	*1.66*	2.23	10.15	0.52	0.04
2130	0.21	1.37	9.86	0.50	1.09

Outbound rate	UDP	nonweb	web	SSL	other
2110	0.10	1.47	3.31	0.73	0.03
2120	0.10	*0.92*	3.34	0.850	0.03
2130	0.10	3.71	3.54	0.859	0.07

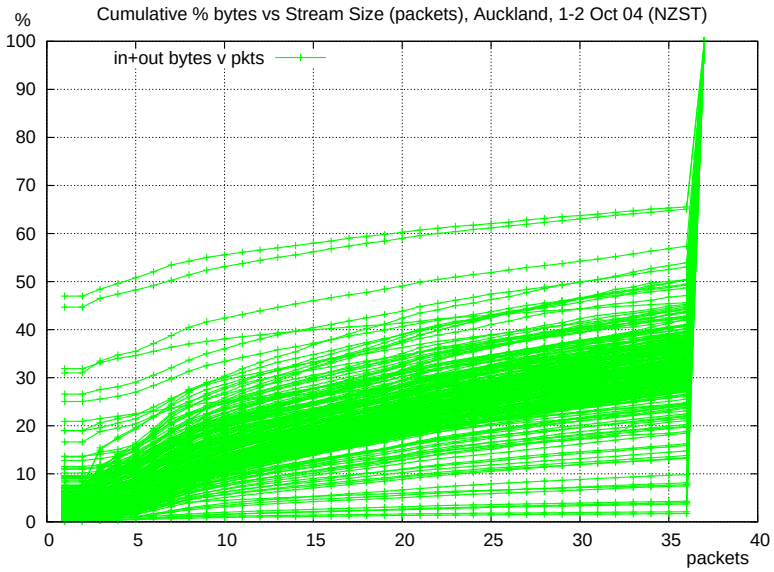

Fig. 8. Total (inbound+outbound) byte density vs packets in stream at Auckland, 1-2 October 2004

flies). Although we need to know about those for security monitoring, they are less important for usage accounting. We decided to try metering while ignoring streams with six or fewer packets (total in both directions).

We ran the meter with $K = 6$ for five days, using our normal 'usage accounting' ruleset. All five days were similar (including regular three-hourly spikes and a daily spike at 1640); fig. 9 shows the packet rate (lower trace), active streams (middle trace) and flows (upper trace) for every second of Thursday, 7 October 2004. The packet rate and streams traces are similar to those in fig. 5; the number of flows is stable and tracks the packet rate.

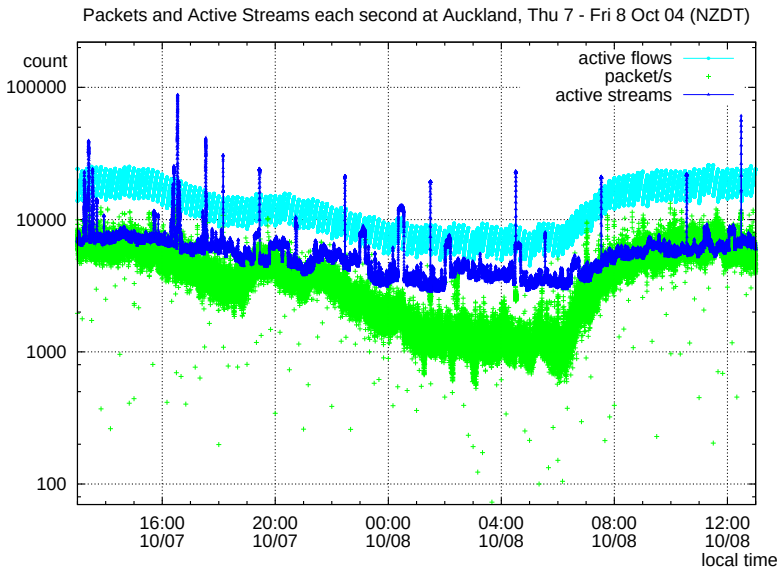

Fig. 9. Packet rate, active streams and active flows at one-second intervals at Auckland for Thursday, 7 October 2004. Our NeTraMet meter used $K = 6$, i.e. streams with six or fewer packets were not matched to flows

Fig. 10 shows the three hours from 2200 in more detail. The number of active flows rises steadily as new flows appear, and falls rapidly when flows are read every ten minutes, allowing the meter to recover flow table space for newly-inactive flows. The 'sawtooth' behaviour, clearly visible in the plot, is thus an artifact of the RTFM architecture. (The important point here is that when we ignore small flows, the average number of active flows remains stable over long periods, minimising the load on the flow data collection system.)

Fig. 10 also shows that when the number of active streams increases sharply (showing spikes and steps), the number of flows is not affected. That supports our hypothesis that such 'attack' increases are caused by bursts of short-lived streams.

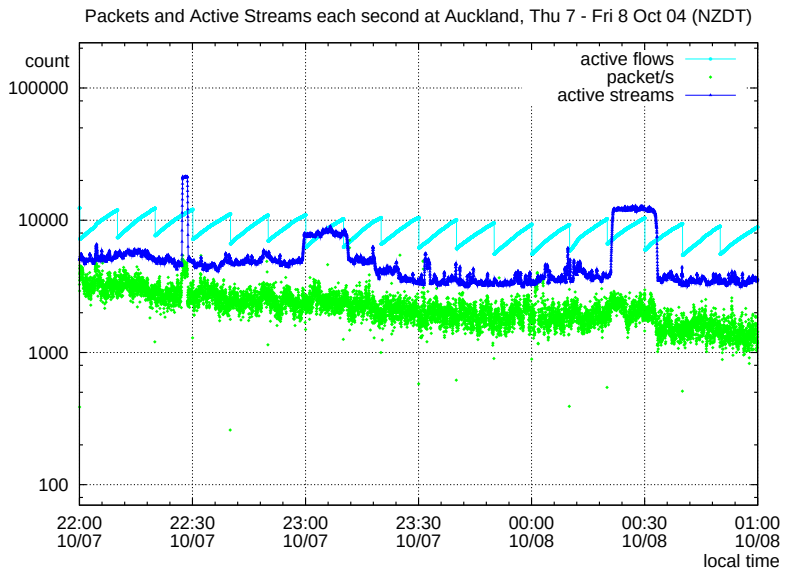

Fig. 10. Detail of fig. 9 showing spike and steps in streams, and sawtooth variation in flows

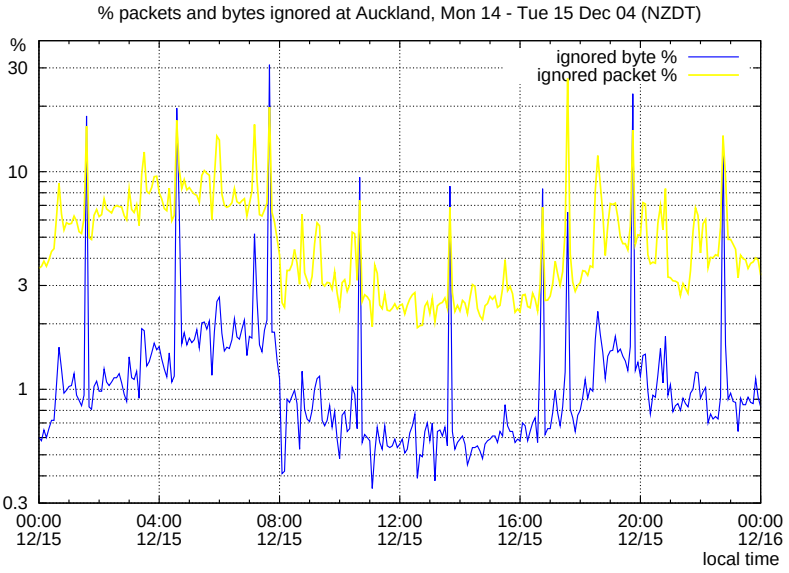

Fig. 11. Percent of bytes and packets ignored at five-minute intervals at Auckland, measured using $K = 6$, i.e. ignoring streams with six or fewer packets

To verify our estimate that ignoring streams with six or fewer packets would exclude between 5% and 10% of the total bytes, we modified NeTraMet so as to collect distributions of the ignored packets and bytes as a function of stream size. One day of typical 'ignored' data, collected at 5-minute sample intervals, is shown in fig. 11.

The 'ignored packet' percentage (upper trace) generally varies between about 2.5% and 8%. Furthermore, its average varies inversely with the average packet rate, suggesting that small *(dragonfly)* streams provide a more or less constant background all the time, with their 'ignored' percentage more obvious when the packet rate is low.

For about 95% of our sample intervals, between 0.5% and 2% of the bytes were ignored (lower trace). In the other 5% of the intervals we saw large spikes in the packet rate and in the number of active streams, as shown earlier in fig. 9. During those spikes, 10% to 30% of the bytes were ignored. Overall, the percentage of bytes ignored is acceptably low, with high 'ignored byte' percentages only occurring during attack events.

6 Conclusion

At Auckland we see frequent bursts of incoming 'attack' streams, which can dominate the traffic mix on our Internet gateway. We believe that traffic engineering and accounting approaches that ignore streams with six or fewer packets ($K = 6$) means that in the long term only about 2% of our total bytes are not measured as 'user' traffic. In return we achieve a significant reduction in the number of flows we have to create, read, store and process.

However, for some traffic mixes, this sampling bias against small flows can radically warp the inferences one makes about the aggregate traffic.

We are continuing our investigation of stream behaviour, especially that relating to the 'attack streams' *(plague of dragonflies)* events. We have not observed these on the California backbone link, where traffic levels are much higher and there is more statistical mixing, but such 'attack streams' probably do appear there.

An alternative approach is the adaptive one proposed in [7], which adapts its sampling parameters to the traffic in real time That approach avoids the bias against small flows and should give a true picture of the actual traffic load, within its sampling limitations. Our approach, however, may be more useful for accounting applications, since we are not sampling. Instead we preserve detail for all the larger streams (which we can bill to a user) while ignoring the small 'attack' streams (which are overhead, not billable to a user).

6.1 Future work

We are continuing to investigate the *plague of dragonflies* events at Auckland. We would like to improve our network attack detection ability by recognising and reporting frequently-occurring attack patterns. The ability to summarise

large groups of small streams would also reduce the number of packets we ignore in our traffic monitoring.

At this stage it is clear that NeTraMet can handle our network's data rate at 100 Mb/s. We are confident that this can be done – without having to use sampling techniques – at 1 Gb/s.

6.2 The Need for Ongoing Measurements

At U Auckland we use NeTraMet for usage accounting and traffic analysis, *Snort*[1] and *Argus*[2] for security monitoring, and *MRTG*[3] for traffic engineering. Each of these tools is specialised so as to perform its intended function well, but there is little overlap between the tools. Indeed, when an unusual event occurs on the network, it can be useful to have data from many tools, providing many different views of that event.

We believe, therefore, that *every large network should collect traffic data on an ongoing basis, using several different tools.* The work required to support such monitoring is well justified by the ability it provides to investigate incidents soon after they occur. In addition, the understanding gained about the network, its traffic, and the ways that traffic changes over time, provides a sound basis for long-term improvements in the network's performance and in service to it's users.

Acknowledgement

Support for this work is provided by DARPA NMS Contract N66001-01-1-8909, NSF Award NCR-9711092 'CAIDA: Cooperative Association for Internet Data Analysis,' and The University of Auckland.

References

1. C. Estan and G. Varghese, *New directions in traffic measurement and accounting: Focusing on the Elephants, Ignoring the Mice,* ACM Transactions on Computer Systems, August 2003
2. N. Brownlee and M. Murray, *Streams, Flows and Torrents,* PAM2001, April 2001
3. N. Brownlee, *Using NeTraMet for Production Traffic Measurement,* Intelligent Management Conference, IM2001, May 2001
4. N. Brownlee and K. Claffy, *Understanding Internet Stream Lifetimes: Dragonflies and Tortoises,* IEEE Communications magazine, October 2002
5. K. Claffy, G. Polyzos and H-W. Braun, *A parameterisable methodology for Internet traffic flow profiling,* IEEE Journal on Selected Areas in Communications, 1995.
6. N. Brownlee, C. Mills and G. Ruth, *Traffic Flow Measurement: Architecture,* RFC 2722, October 1999.
7. C. Estan, K. Keys, D. Moore and G. Varghese, *Building a better NetFlow,* SIGCOMM, September 2004

[1] Security Monitoring, http://www.snort.org/
[2] Network Auditing, http://www.qosient.com/argus/
[3] Traffic Rate Monitoring, http://people.ee.ethz.ch/ oetiker/webtools/mrtg/