Working Paper Series
ISSN 1170-487X

High Precision Traffic Measurement By the WAND Research Group

by John Cleary, Ian Graham, Tony McGregor, Murray Pearson, Ilze Siedins, James Curtis, Stephen Donnelly, Jed Martens and Stele Martin

Working Paper 99/17
December 1999

Department of Computer Science
The University of Waikato
Private Bag 3105
Hamilton, New Zealand

High Precision Traffic Measurement by the WAND Research Group

John Cleary*, Ian Graham*, Tony McGregor*, Murray Pearson*, Ilze Ziedins**, James Curtis*, Stephen Donnelly*, Jed Martens*, Stele Martin*

*Department of Computer Science, University of Waikato, Hamilton, New Zealand
**Department of Statistics, University of Auckland, Auckland, New Zealand

1. Introduction

Over recent years the size and capacity of the Internet has continued its exponential growth driven by new applications and improving network technology. These changes are particularly significant in the New Zealand context where the high costs of trans-Pacific traffic has mandated that traffic be charged for by volume. This has also lead to a significant focus within the New Zealand Internet community on issues of caching and of careful planning for capacity.

Approximately three years ago the WAND research group began with a program to measure ATM traffic. We were sharply constrained by cost and decided to start by reprogramming some ATM NIC cards. This paper is largely based on our experience as we have broadened this work to include IP-based non-ATM networks and the construction of our own hardware. We have learned a number of lessons in this work, rediscovering along the way some of the hard discipline that all observational scientists must submit to.

Our work continues its emphasis on cheap reliable equipment using microprocessor technology. Over time our main concerns have shifted from getting any measurements at all to the reliability and accuracy of those measurements and techniques for disseminating and analysing our results.

In the main we gather traces of network traffic store them to disk and then archive and process them. We have not been directly concerned with other possibilities such as real time processing and displaying. Clearly though, the techniques we have been working with are directly applicable to that problem.

The rest of this paper follows the lifetime of data captured from the net right through to processing and analysing it. At each step of this process there are recurring themes that need to be addressed.

The first of these is capacity. The bandwidth of modern networks and the need to capture data over long times imposes stringent demands at every level on both bandwidth and storage capacity. This can be ameliorated by careful use of specialised hardware at critical points and by filtering data so that only what is essential is handed to the next stage.

The next theme is confidence in the results, that they reflect actual network behaviour. An example of the type of problems that can arise is loss of data. This can occur at all stages of the measurement process from the point that data is captured on the wire to transfer to disk. Of more importance even than ensuring that that loss does not occur is having independent checks in place so that if loss does occur it is detected. While loss might not matter very much in normal network operation there is a danger that its effect will become amplified in later analysis steps. For example loss might occur at a

burst in traffic, but it is precisely such moments that are important in assessing overflow statistics in queues. This has the interesting corollary that measurement equipment needs to be more reliable than standard network interface equipment.

Another important aspect of confidence is the need to maintain an audit trail of how data was captured and what has been done to it. Much of the data that we capture is archived and may be reprocessed long after initial capture. Interpretation at this later date requires that careful and detailed information be kept of where and how the data was captured. Also, because there is always the possibility for anomalous and incorrect behaviours in the capturing software and hardware it is necessary to be meticulous about recording which versions of software and hardware were used to capture the data as well as any options that were set during the capturing process. Any later processing of the data also needs to be recorded. For example, in much of the work we do the timestamps recorded on captured data are corrected after the fact. It is essential the software used to do the correction is recorded (and that copies of the software be kept as well).

The final theme that we will consider is access and security. The presence of large amounts of recorded traffic on disk is a juicy target for those who would like to snoop on networks - we have done the hardest part of hacking that is gaining access to the network. Security is dealt with by: omitting sensitive data (for example recording only headers not full packet data); by encrypting parts of the data (for example the IP addresses); and by maintaining secure access to archived data.

The rest of this paper examines each of these themes at the different stages from physical capture on the "wire" in Section 2, measurement hardware in Section 3, general techniques for analysing the data in Section 4 and finally number of case studies and results in Section 5. General information about the WAND project and online access to measured data is available at http://atm.cs.waikato.ac.nz/wand/.

2. Wire Level

The networks that are most interesting to analyse are those that carry large amounts of aggregated traffic. These links are also the most important to analyse from the point of view of telecommunications and service suppliers because they are costly and central to their operations. This means that measurement is being done on critical physical interfaces. Thus even the act of tapping into the wire or fibre can be come a serious operation. This requires a good relationship with the owner of the physical interface.

There are typically two ways of tapping into high-speed links. One is to use a switch or router to duplicate the traffic to an output port that is used solely for measurement. The other is to tap into the physical wire directly. Diverting traffic has the disadvantage that in some cases it distorts traffic by increasing the demands on internal communication channels, so that it may not accurately reflect what is on any actual physical link. It has the advantage that it can be done without disrupting the traffic. The main disadvantage of physically tapping into a link is that it requires disrupting the link briefly, and that it may alter signal levels on the link. The main advantage is that it is guaranteed to be measuring what is on an actual link. We have used both techniques depending on circumstances.

As described below we like to make use of GPS for precise timing signals on our measurement hardware. This usually requires that there be an antenna installed on a roof and that a cable be taken from the GPS antenna to the machine room. This turns out to be remarkably difficult. In some cases, for security reasons, it is impossible. In others it is just expensive and awkward. This issue has become so difficult in practice that we are currently investigating alternatives to GPS including: using more accurate oscillators (including temperature controlled oscillators); synchronising with SONET clocks; and using timing signals from cell phone base stations.

3. Measurement Hardware

Our initial experiments in ATM data capture were made with re-programmed network interface cards (NICs). However, we found that this was not an ideal solution. The principal problem is that NICs are not designed for the accurate time-stamping of cell or packet arrivals. Our design aim was to time ATM cells with an accuracy and resolution that was approximately one tenth of a cell time, that is better than 275 nano-seconds at OC3, and 70 ns at OC12. With commercial NICs we were not able to get to within two orders of magnitude of this accuracy. The other major problem was that no single manufacturer produced a range of NICs that would give us consistent results for different network speeds and protocols. So, we took the decision to design and build our own hardware.

Our design aims in the Dag series were:

- a range of data capture boards with a consistent architecture, capable of handling data rates from 10 Mbps Ethernet to ATM and POS at OC48 and beyond;
- on board intelligence to filter the data before it is passed to the host processor;
- industry-standard interfaces — PCI and Compact PCI;
- a programmable and re-configurable structure to give the greatest flexibility to each design;
- highly accurate timing of cell or packet arrivals, referenced to a universal time standard;
- low cost.

The general architecture of the Dag series is shown in Fig.1. In the lower speed Dags we have used ARM7 processors, the higher speed boards, Dag 3 and up, use a 233 MHz StrongARM. However even a 233 MHz processor cannot execute many instructions in the approximately 170 ns cell time of an OC48 network!

Each board has the ability to receive periodic timing pulses. When used with a GPS antenna these pulses enable data to be time-stamped to an accuracy of ±250ns to UTC. This is sufficient for us to time delays through ATM switches and to see timing jitter caused by SONET framing. In some cases the timing pulses are used only for local synchronisation. One board is designated the master and sends pulses to the other boards. One important use of this is to synchronise boards measuring the two directions of bidirectional ATM circuits. The relative timing between the boards is then accurate

although they drift by approximately ±0.5ppm to UTC which means a drift of about 1ms in a 15 minute run.

One typical example of data filtering is where we need only to record packet headers for AAL5 encapsulated IP packets, on an ATM link. In almost all cases the entire header is contained with the first cell of each AAL5 pdu. This first cell is not marked in any way, but the last cell in each AAL5 pdu is marked by having a particular bit set in the *payload type* field of the cell header. AT OC3 and OC12 the StrongARM processor is quite fast enough to maintain state for each active virtual circuit, and to select only the first cell in each pdu for transfer to the PC. AT OC48 the processor is not fast enough, and has to be assisted with table lookup firmware within the FPGA.

The board is programmable in two ways – in the code that runs on the on-board microprocessor, and in the configuration of the FPGA. This allows a single board to serve very different functions. For example, the Dag3.2 was initially intended to receive and analyse data on OC3 and OC12 ATM links. A simple reprogramming of the FPGA allows the board to measure both ATM and POS traffic. Recently, we have also programmed the board to act as a data source for network testing.

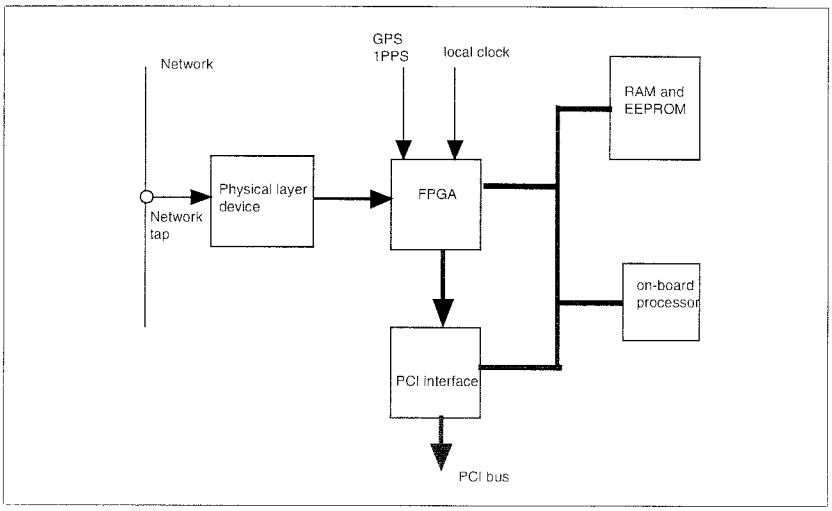

Figure 1. Dag Architecture.

4. Analysis

The WAND group is actively involved in analysing the data that we capture. In this section the types of analysis we are attempting are outlined and following this a number of case studies are described in more detail.

Because we capture timestamps that are accurate and globally synchronised to UTC we can make a number of important types of measurement that would otherwise be impossible. For example, we can make unidirectional delay measurements over inter-continental distances. Thus we have analysed delay times between the United Kingdom and New Zealand. Interestingly the round trip delay times looked quite constant but analysis of the unidirectional times showed strong diurnal variations that cancelled out in

the two directions. We have also analysed the unidirectional delay, delay variation and cell loss for voice over IP between the United States and New Zealand. One problem with such measurements is how to match the different packets or cells as the ends of the measurements. One way to do this is to search for matching packets whose contents are the same (but whose headers my have been transformed in transit). This unfortunately requires complete capture of all traffic. To alleviate this we compute CRC sums over the payloads of the packets and then record just the CRCs not the complete payloads. This introduces a small possibility of spurious matches. However, we have found that in practice we can achieve excellent matches with 32 bit CRCs with very low levels of spurious matches (we have checked the process by doing complete capture and then simulating the CRC).

In principle the measurements that we make are passive, that is, we just observe actual traffic that is passing and do not need to inject our own traffic into the network. This has the advantage that the measurements do not disturb the network. However, we can also add significant value to active measurements as well. The timing measurements we obtain are often much more accurate and reliable than those reported back to the application level of active programs. Thus passive measurement can be used to enhance active measurements and also to debug and calibrate them. We are currently setting up measurements to calibrate the work done by the Skitter, AMP, Surveyor and RIPE projects [1], [2], [3], [4].

Analysis of the traffic can proceed at a number of levels. For example, at the level of aggregate traffic it is possible to do studies of queuing behaviour. This is often done by fitting statistical models to the data. We currently have three different statistical analysis projects seeking to describe aggregate traffic volumes using wavelets, inter-arrival times of cells using mixtures of exponentials and non-parametric models.

A different approach is to separate out different types of traffic and generate separate models for each type of traffic. Thus we are currently separating out voice over IP traffic (see Section 5.2) and hope to extend this to BGP and games traffic soon. At the TCP level we are extracting separate measurements for the different packet type that occur in a session (see Section 5.1).

The final step in analysis is to do simulations of network traffic. At the moment we are doing simulation studies of the effect of voice over IP traffic on TCP traffic. The study is aimed at assessing to what extent aggressive UDP traffic will interfere with the backoff and flow control of TCP. The measured voice sessions are used to generate traces which are then fed into the simulator. The advantage of this is that it is possible to investigate scenarios where there are much higher levels of voice over IP traffic than actually occur at the moment. Much of the analysis work that we are doing can be used in this way either by generating traces that are fed directly to the simulator or by providing the parameters for statistical models used in the simulator.

One of the important aims of our work is to keep traces of traffic over long periods of time so that trends in traffic volumes and changes in the traffic mix can be tracked. This requires that we are consistent in how measurements are done and recorded. The longest series we have at the moment began in March 1998 and records about 10minutes of data twice a day on a link that sees a substantial fraction of New Zealand's Internet traffic.

We are co-operating with the CAIDA group that produces the CoralReef software for analysing and displaying network traffic [5]. Currently we are able to take disk files and input them to CoralReef and are actively working on real time interfaces from our measurement hardware so that CoralReef can be used to make real time displays of data.

5. Case Studies

The following sections give some examples of data analysis carried out on traffic measurements collected using WAND measurement systems.

5.1 Decomposing WWW Response Times

Delays fetching a web page come from a number of different sources that can not all be attributed to simple round trip times (RTT). A question that is often asked is "Is the web slow because of the network or the servers?" The answer to this is important not only from a user point of view but also to know how improve the performance of the web. There is, for example, no point in adding networks capacity if servers are the main component of WWW delay. This analysis is an attempt to answer that question.

Delays in fetching a web page come from four areas:

- server and stack processing delay;
- congestion and queuing delay;
- retries after message loss;
- physical network delay.

Estimating these components of delay individually relies on the observation that an HTTP transfer involves a number of different steps, which occupy the server CPU in varying amounts. Fig. 2 shows a diagram of a simple HTTP transfer and the packets involved in it.

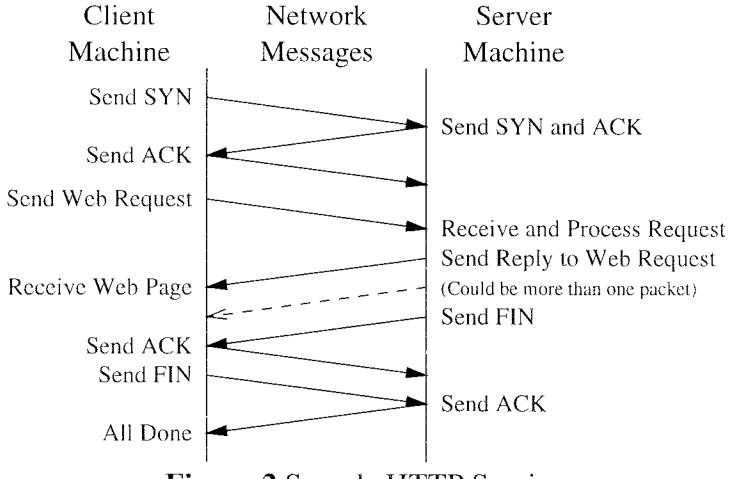

Figure 2 Sample HTTP Session

If a trace with accurate timestamps and many HTTP requests is collected it is possible to identify the separate components of WWW page fetching delay. Physical network delay (which includes speed of light delays as well as router and switching

delays) can be estimated by looking at the minimum delay time for packets such as the ACK/SYN pairs. These pairs involve little CPU processing on the server. The physical network delay can then be subtracted from the other delay times. The time attributable to retries after message losses can be separated out because all packets both sent and received can be seen. Consequently any packet losses can be detected and separately accounted for. Finally the effects of server CPU delays can be separated out by using the fact that some messages involve server CPU and disk activity and others do not. This allows the delays for the different classes to be counted separately and allows queuing and congestion to be estimated as well as server delay.

Although this work is still in progress early results shown in Fig. 3 indicate that for the NZIX the delays experienced by most HTTP requests are dominated by server delays. Each point plotted shows the breakdown as a percentage between server delay and network (queuing) delay. The third component, the physical delay, can be reconstructed because the three values sum to 100%. Points at the top left corner of this graph indicate that queuing delay formed the major component of the time to fetch the request, while points at the bottom right corner indicate server delay as the major component. In only 3 of the 40 connections analysed is the queuing delay greater than the server delay. This data strongly suggests that the bulk of experienced network delays are caused by delay on servers. This result is even more surprising because New Zealand is physically remote from many web servers and it might be expected that many sites would be dominated by physical delay and queuing delay from the intermediate hops.

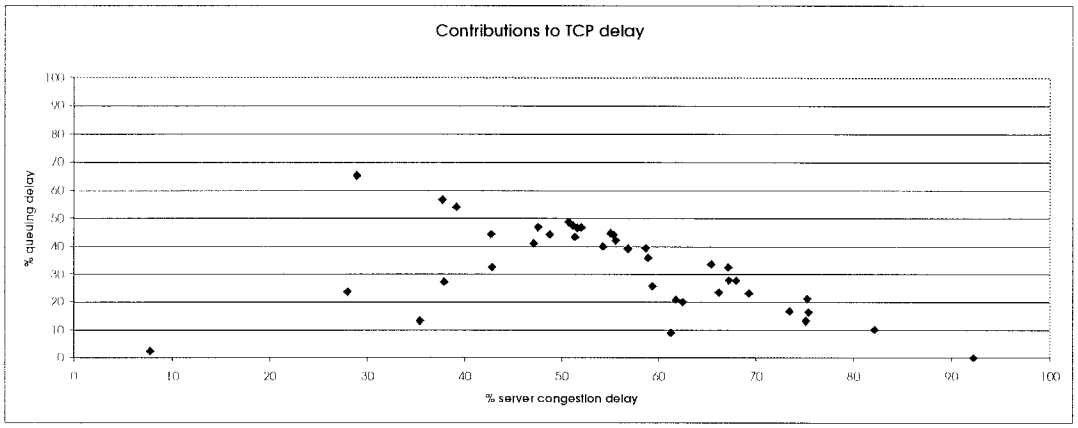

Figure 3. Breakdown of Delays in TCP Traffic.

5.2 Detection of Voice over IP

The use of the Internet to carry voice over IP (VoIP) communication has been increasing recently . There is concern that interaction between VoIP, which uses UDP, and other Internet traffic, most of which uses TCP, may cause congestion for TCP users or even congestion collapse. This problem can occur because TCP reduces its transmission rate when congestion is detected while UDP does not.

This study focuses on H.323, the ITU standard for VoIP [ITU Standards H.323 2/1998]. Multiple vendors already use this protocol, and many other VoIP systems are being converted to use H.323. The goal is to create an accurate simulation model of a mixed TCP/UDP environment. To build such a model it is necessary to capture sample

VoIP sessions and either create a synthetic data source with the same characteristics, or use the captured data to generate a trace that can drive the simulator. Two options are available for obtaining sample data for the simulation. The first is to measure sample conversations in a lab environment. The advantages of this are complete control of the systems and protocols used. The second option is to capture a set of IP header traces and, using the signature of H.323 sessions, to identify data that is likely to be voice traffic. This approach will give a more varied and realistic picture of VoIP traffic for a number of reasons. First there are multiple implementations of both H.323 compliant programs, and voice compression systems. Second, different users of VoIP applications have different connection and machine speeds. Each user will therefore use different data rates and compression techniques.

Identifying H.323 completely accurately without any IP payload and no prior knowledge is impossible. Without being able to view the contents of the connection setup and call control packets it is impossible to tell exactly what the two machines are doing, and what the data passing between them actually means. However H.323 has a distinctive enough signature to make a heuristic approach possible. Currently a two-stage identification system is used. The first step looks for pairs of IP addresses that communicate on two known TCP ports: 1503; and 1720. These ports are the listen ports specified by H.323. The next step relies on recognising the use of the Real Time Protocol (RTP)[RFC1889] whose use is also specified by H.323. This protocol is responsible for the transfer of the voice data itself. H.323 sets up dynamic UDP ports for RTP to run over, but RTP specifies the use of two adjacent UDP ports. The first port will be an even number, and will contain the data connection. This port will therefore have a large number of regular (often fixed) sized packets. The next (odd numbered) port will contain a small number of packets used for call control information. If two IP addresses are communicating in these ways it is likely, although not certain, that the data represents an H.323 session.

5.3 Analysis of Laboratory Based VoIP

To gain an understanding of the nature of VoIP over heavily used links we conducted another experiment where VoIP traffic was sent between the University of Waikato and the University of California, San Diego. GPS based DAG systems were deployed at both ends of the link while two speakers attempted a conversation. The results are shown in Fig. 4. Every voice packet is plotted with the real time it was sent plotted along the X axis and the one-way delay plotted on the Y axis. As well as the VoIP packets background traffic is also plotted (this was done to check and see if the VoIP packets were being treated differently by routers and had different characteristics from bulk traffic - it seems that they are indistinguishable). Qualitatively the speakers reported that voice quality in the NZ to US direction was acceptable while in the US to NZ direction it was unintelligible most of the time. This matches the measurement results that show significantly longer delays and much higher delay variation on the US to NZ link compared with the NZ to US link. This asymmetric congestion is caused by a 5:1 imbalance in traffic between the two countries. During these measurements significant loss was seen on the US to NZ link while no loss was experienced in the other direction.

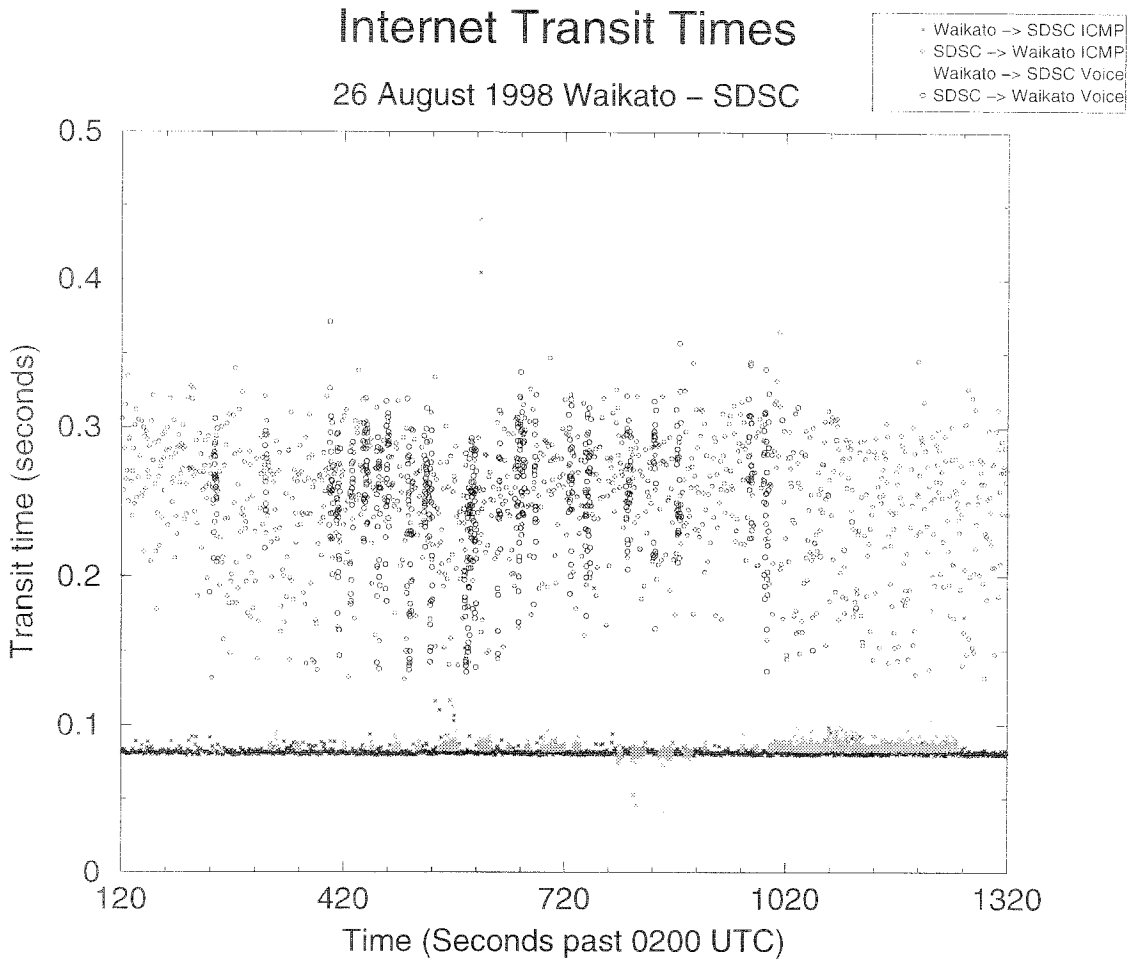

Figure 4. Unidirectional Delays of Voice over IP Packets.

5.4 NZIX

Data for case studies 1 and 2 above was collected from the New Zealand Internet Exchange (NZIX). The NZIX was originally the primary peering point in New Zealand for international bandwidth providers, ISPs, and universities. Although other exchanges now exist and most universities have separate Internet connections, the NZIX still supports a number of ISPs, universities and bandwidth providers.

To facilitate measurement the central switch of the exchange provides a "span port", that is, a single 100baseT port that attempts to provide a copy of all of the traffic on the switch. The traffic aggregate is less than 100Mbps, so the span port is not a limiting factor in normal operation. We use our own libpcap based program to capture a fixed number of bytes from each packet. The capturing PC is also connected to a GPS time source via its serial port, and records timestamps each second. These are used for later clock correction.

For privacy reasons an anonymising program replaces the original IP addresses, and zeros any payload bytes that may have been inadvertently captured.

5.5 Long Range Dependence

In this section we analyze two traces taken from a longer one-week trace of an ATM link at Auckland University. Both traces contain timestamps of packets arriving over two hours of heavy activity, one trace for each direction of the link. We concentrate on examining the timestamps for evidence of long-range dependence.

It has long been known that traffic streams are not adequately modelled by traditional models such as the Poisson process, and that they exhibit the kind of scaling associated with long-range dependence. Long-range dependence occurs when correlations decay as a power law rather than exponentially. Long-range dependence, if present, has serious implications for the design and control of networks.

The first trace consists of 2,066,749 timestamps and the second of 2,153,602 timestamps. Fig. 5 plots the number of packets in intervals of 0.1 second for trace 1. The mean number of packets in an interval is 28.70, the minimum 0, the maximum 288, with standard deviation 11.31. We see that there are no marked changes in the arrival rate, although it is slightly lower in the second hour of the trace. A similar plot of trace 2 displayed more variation over time. For trace 2, the mean number of packets in an interval is 29.91, the minimum 1, maximum 205 and standard deviation 11.04.

Both traces were examined for long-range dependence using the non-parametric wavelet estimators developed by Abry and Veitch [6]. Abry and Veitch point out that the estimates of the scaling parameter α obtained using this method are unbiased and computationally inexpensive. The method can also be used to detect and decrease the influence of deterministic trends. To apply the method, the logscale diagram is first examined for regions of alignment. (The logscale diagram is a plot of the logarithm of variance estimates of the wavelet details at each scale against the scale at they are estimated.) A region of alignment at large scales, with slope between 0 and 1, indicates the presence of long-range dependence. An estimate of the scaling parameter α is then obtained by taking a weighted regression over this region.

This method was applied to both traces in two ways. The data were first aggregated into counts over intervals of 0.01 seconds. The estimate of the scaling parameter, α, obtained for trace1 was 0.798, with 95% confidence interval (0.663,0.934). For the second trace it was 0.793, with 95% confidence interval (0.793,0.912). In both cases, the number of vanishing moments of the wavelet was equal to 2, indicating the presence of some trend or level shift. The method was then applied to the inter-arrival times of the packets. Preliminary QQ plots and CD plots of the inter-arrival times indicated that they were *not* heavy-tailed. For the first trace, the estimate of α, was 0.734, with 95% confidence interval (0.685,0.784) and for the second trace was 0.847 with 95% confidence interval (0.744,0.949). These estimates of α indicate the presence of long-range dependence in both traces.

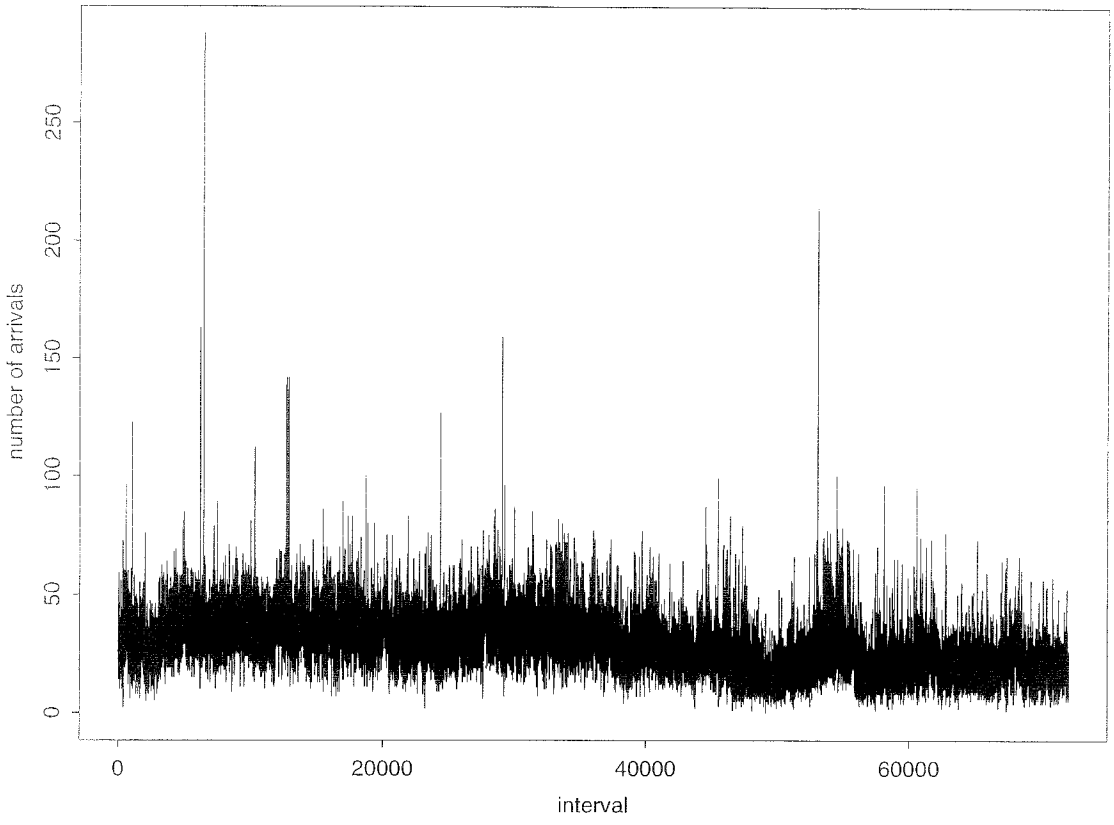

Figure 5 Number of Arrivals per 0.1s Interval for Auckland ATM Link

6. Summary

The WAND group has wide interests in the measurement and analysis of Internet traffic. To date our focus has been on building measurement systems (including custom hardware where necessary) that perform verifiably accurate measurements. Using these measurement systems we have been to collect the raw data that has allowed the analyses described above to be performed. We are continuing to deploy measurement stations and increase the range of analyses available.

7. References

[1] Skitter Home Page, http://www.caida.org/Tools/Skitter/
[2] AMP Home Page, http://amp.nlanr.net/AMP/
[3] Surveyor Home Page, http://www.advanced.org/csg-ippm/
[4] RIPE Home Page, http://www.ripe.net/test-traffic/index.html
[5] Coral Reef Home Page, http://www.caida.org/Tools/CoralReef/
[6] Abry, P. and Veitch, D. (1998) Wavelet analysis of long-range dependent traffic. IEEE Transactions on Information Theory, 44, pp 2-15.