Using Synthetic Packet Pairs to Investigate Real-time Latency Variations over Home Broadband Connections

Amiel Heyde
Centre for Advanced Internet Architectures, Technical Report 080530B
Swinburne University of Technology
Melbourne, Australia
amiel@swin.edu.au

***Abstract*—Synthetic Packet Pairs (SPP) is a smart approach to estimating round trip time (RTT) with passive monitoring. SPP gives greater accuracy and resolution than conventional active methods. With the ability to measure RTT as experienced by a particular application, SPP can be used to visualise the effectiveness of QoS implementations and further understand delay characteristics of specific traffic. Additionally, monitoring traffic with SPP can reveal properties of the underlying protocol stack and link configuration. In this report we employ CAIA's free implementation of SPP to examine the network delay experienced by a few applications, as well as to demonstrate queueing and serialisation delay as displayed in a typical ADSL home broadband connection.**

I. Introduction

Broadband Internet services and the associated speed increases bring new possibilities to consumers. Where the Internet was once used solely for data, Voice over IP (VoIP) and other real time applications like online multiplayer games are commonplace. Despite increased speeds, network latency continues to constrain the performance of our Internet connections. As a result, latency measurements play an important role in determining the service quality perceived by the end user.

Round Trip Time (RTT) is a common metric used to determine the delay over a path. It is defined as the time it takes to send a packet to a remote host and receive a response. Multiple components contribute to the overall RTT. These include:

- Processing Delay - The time taken for routers to process packet headers and determine the next hop destination.
- Queuing Delay - The time a packet takes waiting for preceding packets to be processed. Queuing usually occurs where packets traverse a boundary from a fast link to a slow link.
- Serialisation Delay - The time taken for a packet to be clocked onto a link. Serialisation delay = $\frac{size}{rate}$ where size is the total number of bytes transmitted over the wire and rate is the link speed in bytes/sec.
- Propagation Delay - The inherent delay caused by the physical limits of copper or optic fibre transmission media.

While all these components are contributing factors, serialisation delay and queuing delays will be a focus of this report.

RTT measurement techniques fall into two broad categories - active and passive. Active measurements involve sending test probes and measuring the combined time they take to traverse a path in both directions. This method assumes that the delays experienced by the test probes are representative of what an average packet would experience on the link. If regular measurements are required, test probes must be sent constantly, adding to the load on the network. Not only may this cause further delay for other traffic on the link, but estimates may be inflated due to this extra load.

One common tool for active RTT measurement is 'ping'. Many users assume that ping will adequately measure the delay experienced on their link, when in fact, RTT experienced by an application may differ considerably, depending on characteristics of the application traffic. The relevant characteristics of the application's traffic include the packet size, packet rate and variance in inter-arrival times. Additionally, the network may treat certain application traffic differently to traffic generated by ping, especially if prioritisation is configured to increase quality of service.

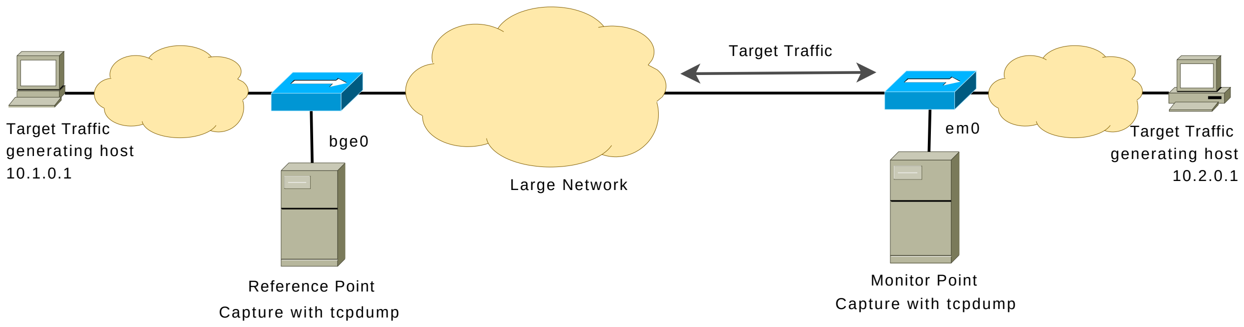

Fig. 1. Using SPP to measure delay between two measurement points.

Passive methods instead utilise traffic already flowing over a path, and therefore implicitly take into account application traffic characteristics, leading to more accurate estimates. Measurements are taken at both ends of a path and round trip time estimates are calculated from these measurements.

Synthetic Packet Pairs (SPP) [1] is an algorithm that creates RTT estimates from data collected with passive monitoring. It can post process files captured at the measurement points or monitor live interfaces to generate estimates in realtime,. SPP provides frequently updated RTT estimates without the need for precise time synchronisation between the measurement points.

This report utilises CAIA's free, open-source implementation of SPP [2] to demonstrate the advantages of SPP when examining latency observed across a consumer ADSL connection. This report demonstrates some of the advantages of SPP and examines latency observed across an ADSL connection.

RTT estimates have been produced with SPP and ping during transmission of a range of traffic types. Results show serialisation delay to be the largest contributing factor, with queueing at the IP layer and underlying ATM layer causing quantisation in RTT values.

The rest of this report is outlined as follows. Section II summarises the operation of SPP, section III describes the experimental environment and section IV looks at the relationship between packet size and RTT. Section V investiages RTT with a few common traffic types and the effect of QoS is considered in section VI. Further work is discussed in section VII and the report concludes in section VIII.

II. SPP Operation

Figure 1 shows a typical configuration for using SPP with post processing. Target traffic is generated by hosts at either end of the path. At two measurement points along the path, traffic is passively captured and written to PCAP files with tcpdump [3]. These two files are then processed with SPP.

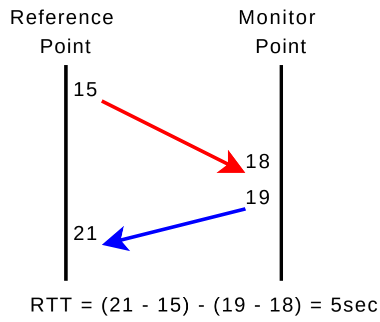

Fig. 2. The RTT calculation process. Arrows represent a packet travelling between measurement points. Values indicate the elapsed time in seconds since the start of measurement.

SPP finds airs of packets that travelled in opposite directions but were observed at almost the same time. It then looks at the times each packet was observed at both measurement points in order to calculate an RTT estimate. This process is repeated, producing a list of RTT estimates. Figure 2 depicts the RTT estimate calculation. For simplicity, values representing elapsed time in seconds have been used instead of real timestamps.

In this example, monitor points are independent of the hosts generating traffic, although this need not be the case. For the experiments in this report, monitor points were placed on the two endpoint hosts.

III. Experimental Setup

Experiments were conducted between Host A (FreeBSD 6.2, located at Swinburne University) and Host B (Linux 2.6, ADSL Internet connection). Figure 3 shows the path between the hosts.

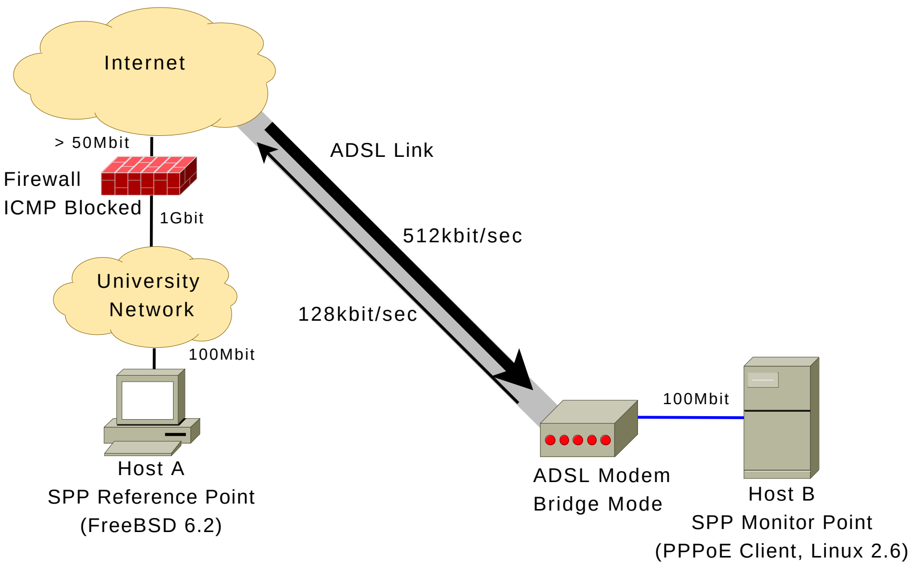

Fig. 3. Path that was traversed in examples in this report

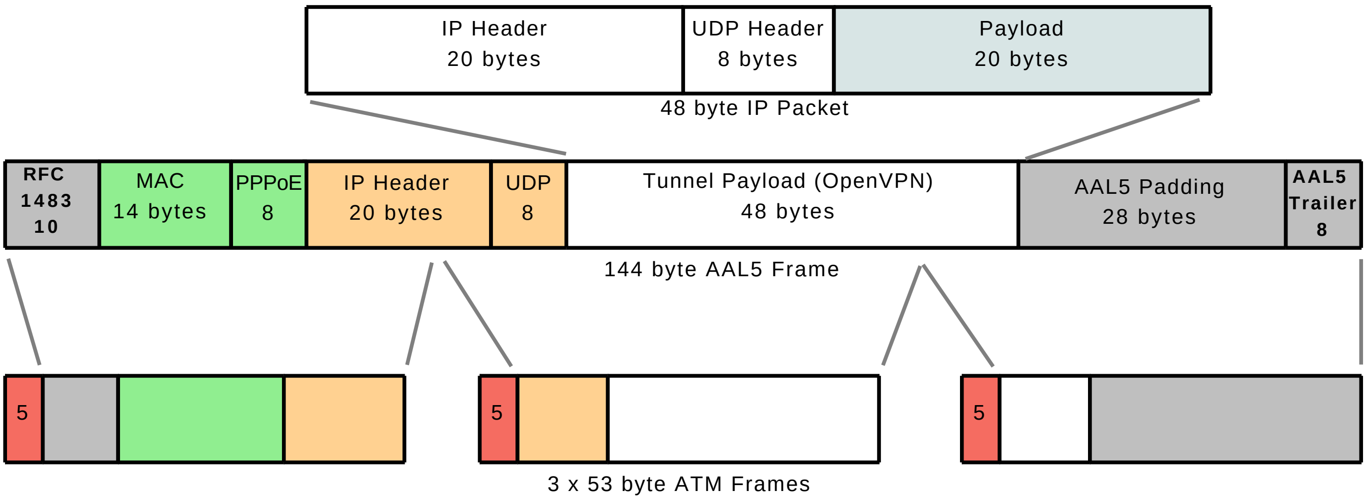

Fig. 4. Protocol breakdown for UDP packet with 20 byte payload

A. *ADSL Link*

The ADSL connection speed was 512kbit/sec downstream and 128kbit/sec upstream, a fairly common speed for a home connection in Australia [4]. This link's relatively low speed emphasises delay characteristics that otherwise may not be as visible over higher speed links.

ADSL-based Internet connections are often established with PPP over ATM (PPPoA) [5] or PPP over Ethernet (PPPoE) [6]. RFC 1483/2684 [7] specifies LLC/SNAP or VC multiplexing as options for encapsulation. The Frame Check Sequence from the MAC layer may be included or excluded when using PPPoE. RTT is affected by the choice of configuration as each options affects the total size of packets on the wire.

The modem used in these tests (D-LINK DSL-500 generation II) provided all the above connection options. Unless otherwise specified, tests in this report used the PPPoE/LLC configuration with the PPPoE session termination on Host B.

B. *Traffic Generation*

A few types of traffic were used for these experiments. IPerf [8] was used to generate fixed size UDP traffic, since it is able to send traffic bidirectionally. Where bitrates are quoted, values refer to the bitrate of ATM cell transmission rather than IP packet transmission.

SSH (Secure Shell [9]) traffic was generated interactively by a user and TCP file transfer traffic was generated with SCP (Secure Copy [10]).

C. RTT Measurement

SPP was used to measure RTT, with measurement points located on Host A and Host B. SPP post processed PCAP files that were captured at these measurement points with tcpdump.

For comparison, ping was used for active probing. By default, ping sends 64 byte IP packet probes once every second. Throughout this report, the default behaviour has been used unless otherwise specified.

Ping samples the path at regular time intervals, unlike SPP where the sample rate of SPP estimates is dependent on the rate at which the algorithm finds packet pairs. As a result, statistical analysis of the estimates from both methods may not give expected results. In particular, cumulative distributions generated from SPP measurements are not representing quite the same thing as those taken from ping. Cumulative distributions from ping are in terms of time, while cumulative distributions from SPP are in terms of pairs (which can loosely be applied to packets). Resampling of SPP results would be necessary to allow precise comparison with ping. This has not be done for this report. Depending on the application of interest, time or packet based distributions may be more relevant.

D. OpenVPN Tunnel

The university firewall blocks all ICMP traffic, which blocks ping probes. For this reason, tests were conducted over an OpenVPN [11] tunnel.

OpenVPN creates a new tun0 interface on both hosts. Packets sent out this interface are encapsulated into UDP frames and sent out the underlying interface. When the packets arrive at their destination, OpenVPN extracts the IP packets and presents them at the tun0 virtual interface.

Tunnelling with OpenVPN causes the firewall to see only the UDP traffic of the VPN rather than the ICMP packets inside the tunnel. The tunnel may introduce slight latency, however relative delays are not the focus of this report and therefore the slight latency introduced by OpenVPN can be ignored.

OpenVPN was configured without any type of compression, encryption or authentication. This was to ensure a fixed known overhead.[1]

E. Protocol overhead

The ADSL link has considerable overheads due to the many protocol layers that reside beneath the application layer [12]. At the top level, UDP headers and IP headers encapsulate the payload adding 8 and 20 bytes respectively. The OpenVPN tunnel adds 28 bytes (8 bytes UDP, 20 bytes IP) while PPPoE adds another 8 bytes.

With a PPPoE/LLC connection profile, 14 bytes of Ethernet header is added plus 10 bytes for the LLC/SNAP encapsulation. An ATM adaption Layer 5 (AAL5) trailer of 8 bytes is added to the end, along with variable length padding, in order to create a frame size which is a multiple of 48. Finally the AAL5 frame is split up into 48 byte chunks and combined with 5 byte headers to form ATM cells. A diagram of all framing overheads and their expected sizes for a UDP packet with a 20 byte payload is shown in figure 4.

IV. Investigating RTT

To get an idea of how RTT varied with a wide range of packet sizes, bidirectional UDP traffic of fixed packet size was tested. Trials were conducted with arbitrary payload sizes evenly spaced from 40 to 1240 bytes[2]

Ping was also tested, set to send probes of the same sizes. Figure 5 shows the cumulative distribution of RTT for each packet size as measured by SPP (coloured) and by ping (grey). There is virtually no difference in the values reported by both methods, confirming that SPP operates as expected. Serialisation delay is evident, with each increase in packet size resulting in increased RTT. Figure 6 shows the individual estimates generated by SPP for a 40 byte payload size.

A. Comparison of measured and calculated RTT

With knowledge of protocol overhead as given in section III-E, expected delay over the ADSL link may be calculated. Table I shows the total number of ATM cells resulting from each of the payload sizes used. Table II shows the calculated delay up and down the ADSL link, the expected RTT over the link (RTT_{ADSL}), the measured RTT of the whole path (RTT_{SPP}), and the difference between RTT_{SPP} and RTT_{ADSL}. Figure 7 shows these estimated and measured delays.

[1]Overhead was confirmed by comparing captures taken on the tun0 interface (showing traffic over the tunnel) and the ppp0 interface (showing tunnel UDP traffic).

[2]Payload sizes were chosen to cover a wide range of sizes and particular values are of no significance.

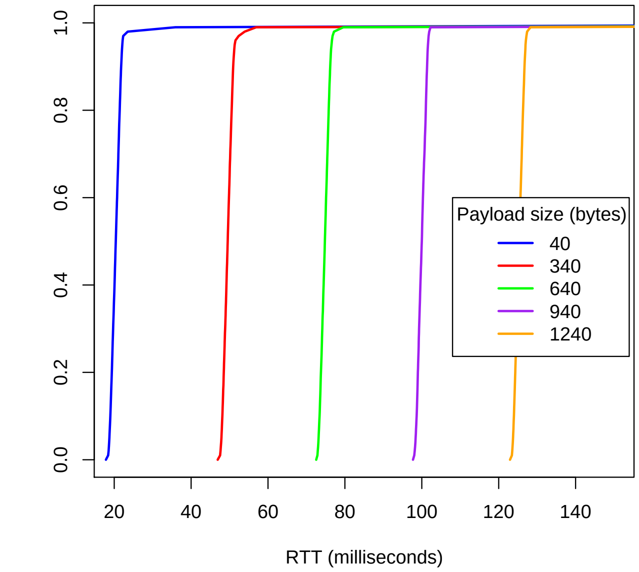

Fig. 5. RTT is directly correlated to packet size

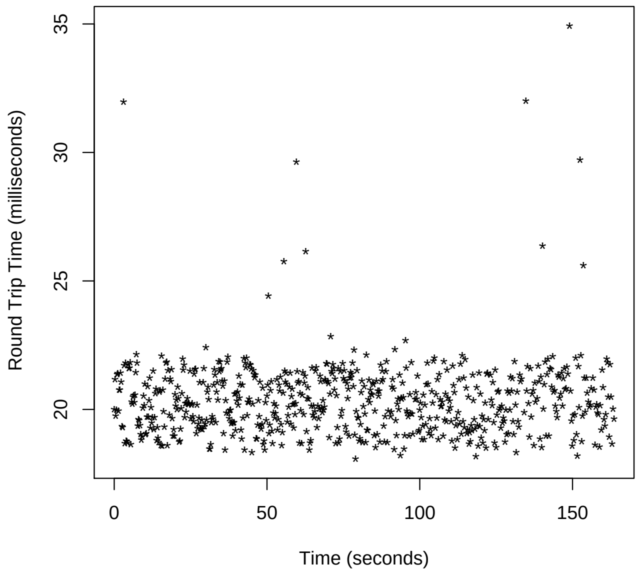

Fig. 6. Round Trip Times for UDP packet with 40 byte payload. Other payload sizes showed similar distribution of RTT values around higher median RTTs.

Data 68 - 1268 bytes	Payload	40	340	640	940	1240
	UDP (8)	48	348	648	948	1248
	IP (20)	68	368	668	968	1268
Tunnel 28 bytes	UDP (8)	76	376	676	976	1276
	IP (20)	96	396	696	996	1296
ADSL 55 - 102 bytes	PPPoE (8)	104	404	704	1004	1304
	MAC (14)	118	418	718	1018	1318
	RFC1483 (10)	128	428	728	1028	1328
	PADDING	+8	+44	+32	+20	+8
	AAL5 (8)	144	480	768	1056	1344
ATM Cells		**3**	**10**	**16**	**22**	**28**

TABLE I
FRAME SIZES AT EACH LAYER AND RESULTING NUMBER OF ATM CELLS REQUIRED FOR EACH PAYLOAD SIZE

UDP Payload (bytes)	40	340	640	940	1240
No. ATM Cells .	3	10	16	22	28
Ex. Delay Down	2.48	8.28	13.25	18.22	23.19
Ex. Delay Up (ms)	9.94	33.13	53.00	72.88	92.75
RTT_{ADSL} (ms)	12.42	41.41	66.25	91.10	115.94
Min RTT_{SPP}	17.84	46.99	72.52	97.66	123.19
Median RTT_{SPP}	20.34	49.52	74.72	100.05	125.37
Diff (Ex. & Med)	7.92	8.11	8.47	8.95	9.43

TABLE II
CALCULATED RTT_{ADSL} COMPARED WITH MEASURED RTT_{SPP} FOR THE ENTIRE PATH

The almost constant difference between the measured RTT and the estimated delay over the ADSL link indicate minimal serialisation delay in the path from Host A to the exchange where the ADSL connection is terminated. This is because serialisation delay of the largest packet size tested would be less than 0.2ms on each of the faster links which make up the remainder of the path.

The approximately constant 8ms of additional delay that is not explained by serialisation can be attributed to OpenVPN and the processing delay in other devices along the path.

B. Confirming Protocol overhead from RTT estimates

As mentioned in section III-A, there may be quite a few variations in encapsulation at the lower layers.

During setup, it was not obvious from the configuration interface of the modem whether the Ethernet Frame Check Sequence was included, but since the protocol overhead is known to be in a small range, testing with

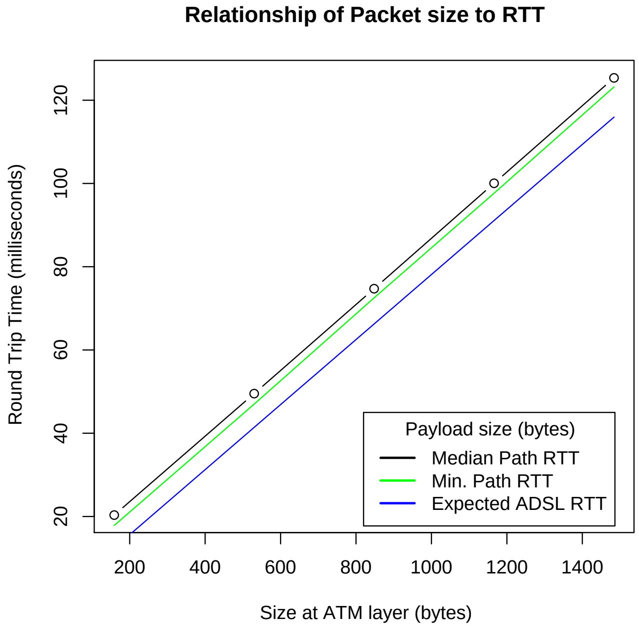

Fig. 7. Linear relationship between RTT and size. RTT_{SPP} is fairly consistently 8-9 ms larger than estimated RTT_{ADSL}.

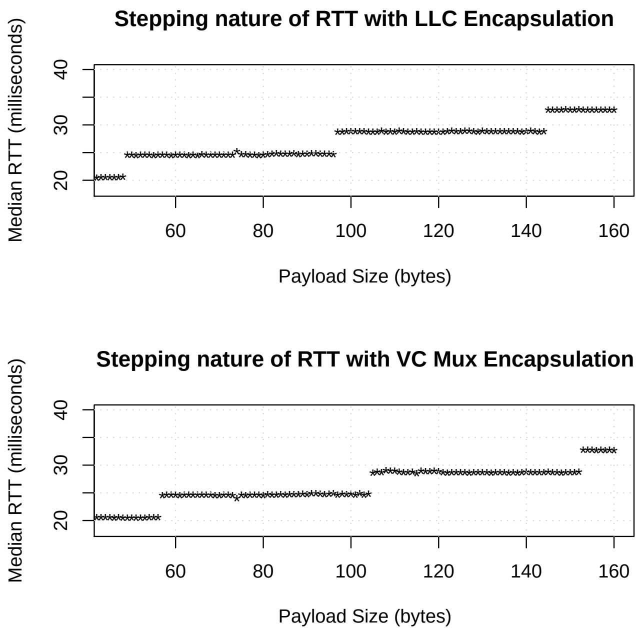

Fig. 8. ADSL link exhibits step increases in RTT as packet size increases linearly. The payload size at which a transition in RTT occurs confirms the size of framing overheads.

SPP revealed the precise overhead due to the properties of ATM.

The fact that data transferred on ADSL links must be encapsulated in 53 bytes ATM cells causes linear gradation in packet size to result in step increases in size at the ATM layer. Figure 8 shows linear increases in payload size result in stepping RTT in 48 byte intervals. Although LLC encapsulation is used throughout this report, VC multiplexing is shown to demonstrate the effect of the ADSL connection options on RTT.

With LLC encapsulation, the same RTT value is reported for payload sizes ranging anywhere within the 48 byte range of 97 to 144. This confirms that packets are in fact encapsulated in fixed sized cells with 48 byte payloads.

Knowing that a payload size of 96 bytes results in an AAL5 frame with no padding (when using LLC encapsulation), the total overhead must be a multiple of 48. (Since three whole ATM cells are filled with payload, headers are an integer multiple of the ATM cell payload size). Consulting relevant specification documents [7] [6], we can conclude that the Ethernet Frame Check Sequence is not included in the overhead of this link.

V. Real World Scenarios

This section will cover a few common applications and how SPP can be used to highlight delay characteristics of their traffic and the ADSL link.

A. *SCP transfer up ADSL link*

Although not time sensitive, file transfer using TCP presents an interesting case for delay analysis. An 11MB test file was transferred from the Host B to Host A.

RTT estimates by ping and SPP taken during the transfer are shown in figure 9. Initially obvious is a high average RTT caused by queuing. Also interesting is the appearance of quantisation in the estimates produced by SPP. Figure 11 gives a closer look. The RTT_{SPP} histogram, shown in figure 12 clearly depicts high concentrations of RTT estimates spaced at regular intervals. The spacing between the peaks is 106ms.

This quantisation in RTT is an artifact of queuing. Figure 10 shows the packet size distribution of the transfer. The length of the IP packets travelling up the link were mostly 1450 bytes long.[3] Adding overheads, this packet size results in 32 ATM cells (1696 bytes) on the link. Serialisation delay for 32 ATM cells up the link amounts to 106ms, corresponding to the distance

[3]Learned by inspection of the raw PCAP files used by SPP

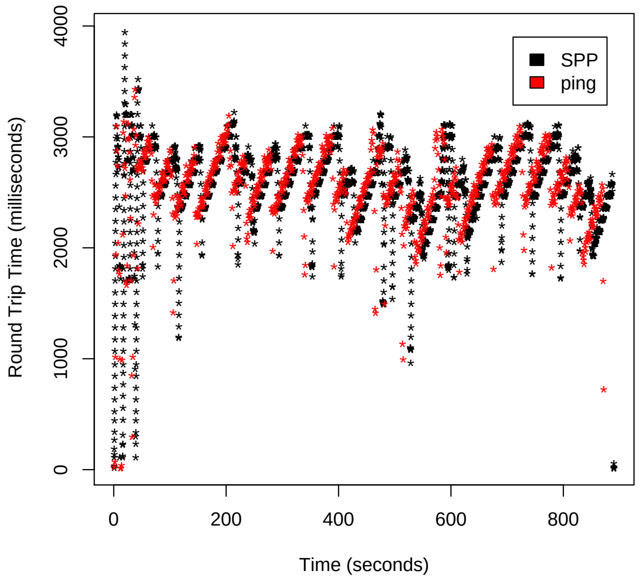

Fig. 9. Estimates generated during SCP transfer up the ADSL link. RTT estimates from both SPP and ping show a sawtooth trend as would be expected from TCP. Queuing at the ADSL link causes quantisation.

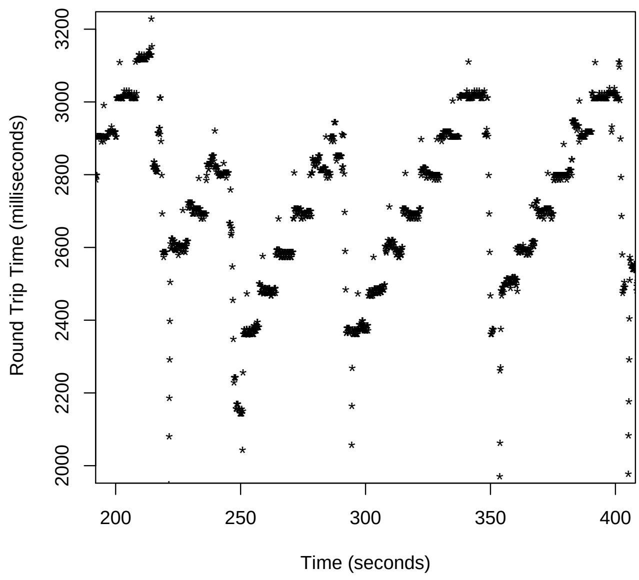

Fig. 11. A closer look at SPP estimates reveals single estimates stepping down as the queue length decreases rapidly.

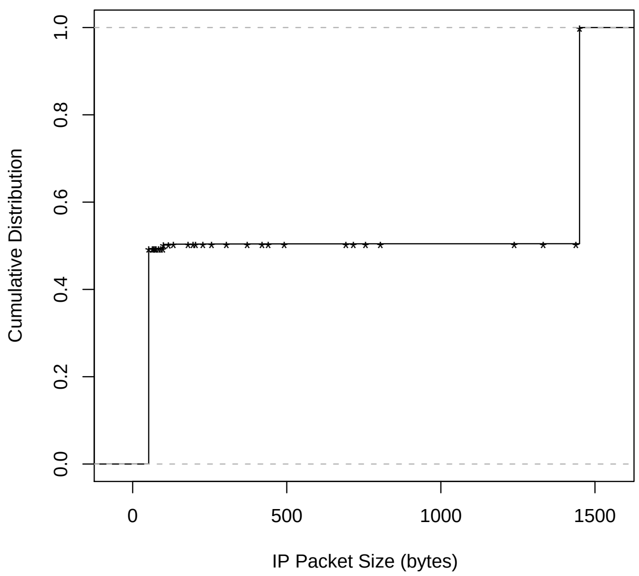

Fig. 10. Cumulative distribution of packet size for SCP transfer up the ADSL link. Packets were generally one of two sizes, corresponding to data packets and ACKs flowing in opposite directions. IP packets travelling up the ADSL link were 1450 bytes while the ACKs travelling down the link were 52 bytes.

between peaks in the RTT distribution. Therefore, the quantisation level of an RTT estimate was determined by the number of packets in the queue when it arrived at the ADSL modem for transmission.

Counting the bands up the left hand side of figure 9 gives an indication of the queue length at the ADSL modem. Accordingly, the queue length ranged from approximately 20 to 30 packets throughout the majority of the transfer.

The overall sawtooth trend of the estimates look like the typical fluctuations observed in TCP throughput due to the varying size of the congestion window. As the transmission rate increased, queue size and RTT increased also. When packets were lost and the rate reduced, the queue length decreased and RTT was reduced. In figure 11, individual estimates can be seen at each RTT quantisation level as the queue size reduced rapidly.

Figure 13 shows that SPP and ping report comparable RTT distribution.

B. SCP transfer down ADSL link

The same 11MB test file was transferred down the link with SCP. Figure 15 shows estimates generated by SPP and ping. Ping reported a large range of fairly dispersed RTT values, while SPP estimates fell into well defined bands. Like in the previous transfer, serialisation delay

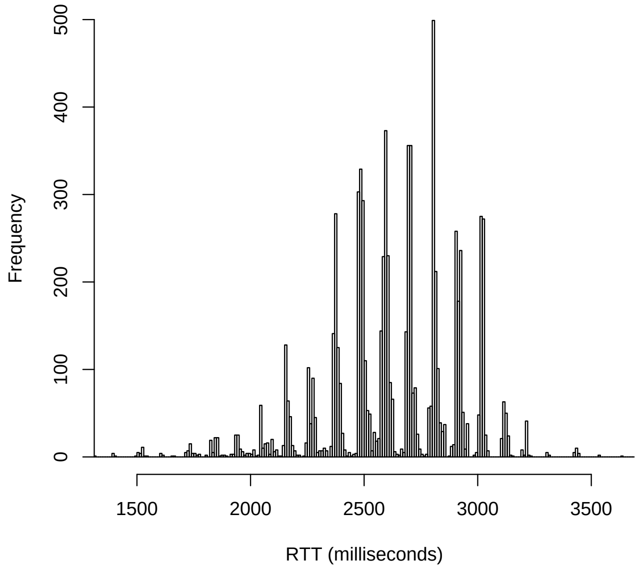

Fig. 12. Histogram of RTTs experienced by the SCP transfer up the ADSL link. SPP reports that the session experiences RTTs centered around 2610ms with quantised values spaced 106ms apart; due to queuing at the ADSL link.

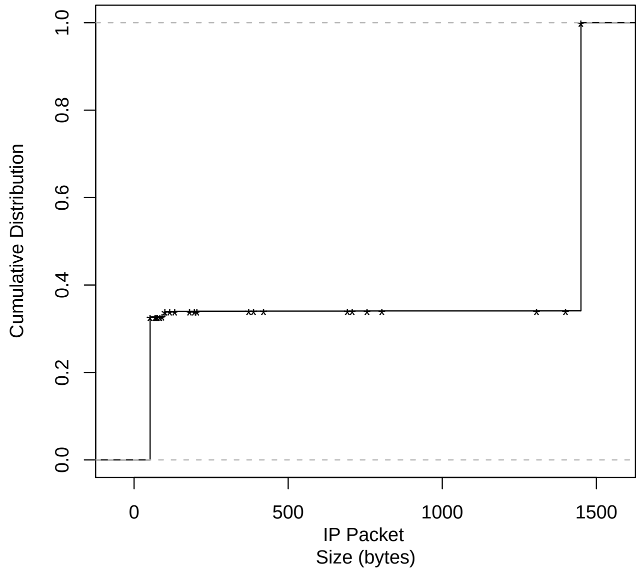

Fig. 14. Cumulative distribution of packet size for SCP transfer down the ADSL link. Packets were generally one of two sizes, corresponding to data packets and ACKs flowing in opposite directions. IP packets travelling down the ADSL link were 1450 bytes while the ACKs travelling up the link were 52 bytes.

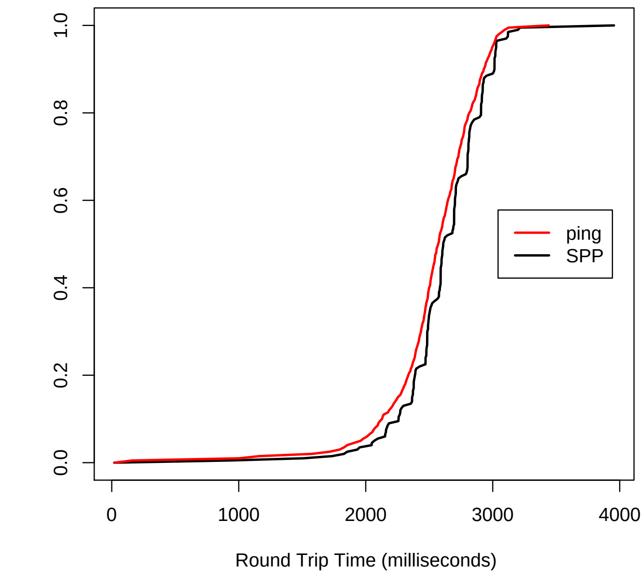

Fig. 13. Both ping and SPP show similar cumulative distributions for the SCP transfer up the link. An exception is the quantisation shown by SPP.

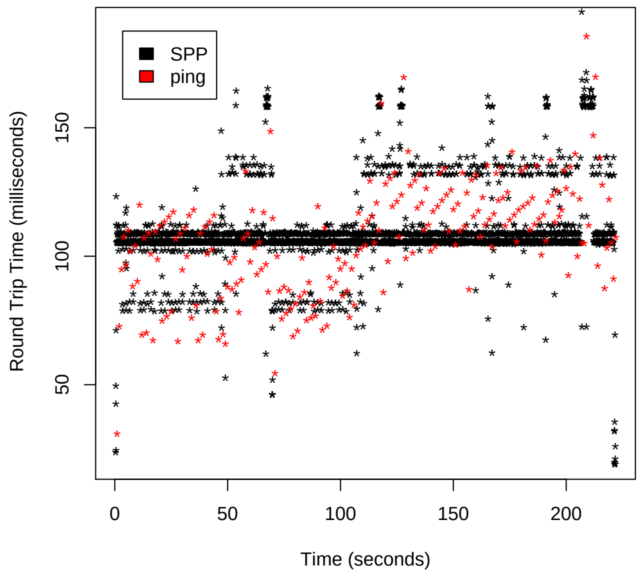

Fig. 15. RTT estimated by ping and SPP during SCP transfer down the ADSL link.

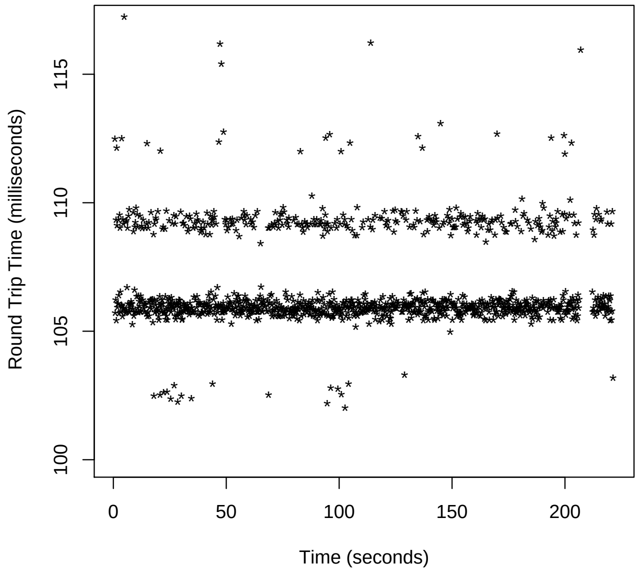

Fig. 16. RTT estimates generated by SPP during SCP transfer down the ADSL link showing 3.3ms banding due to ADSL/ATM interactions.

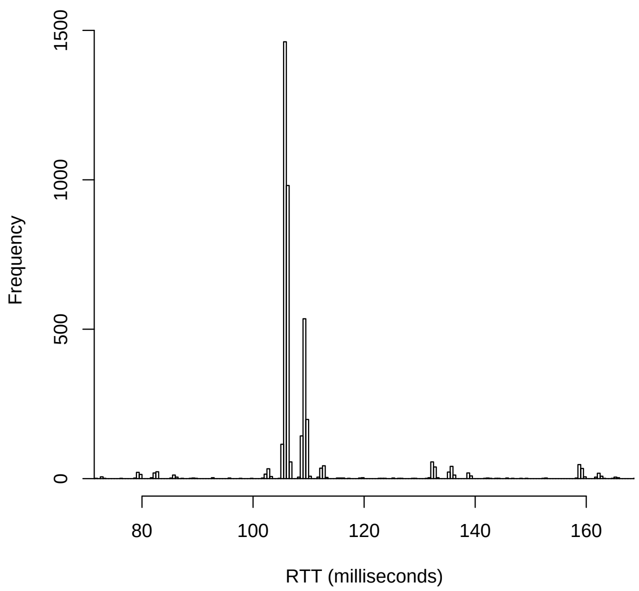

Fig. 17. Histogram of RTTs during SCP transfer down the ADSL link. RTT distribution is concentrated around 105.8ms

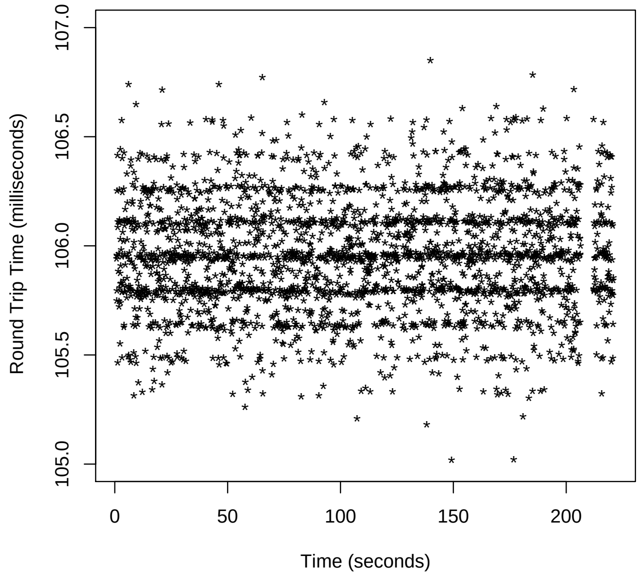

Fig. 18. A close look at the central band of RTT estimates by SPP. Quantisation is evident at 15 microsecond intervals.

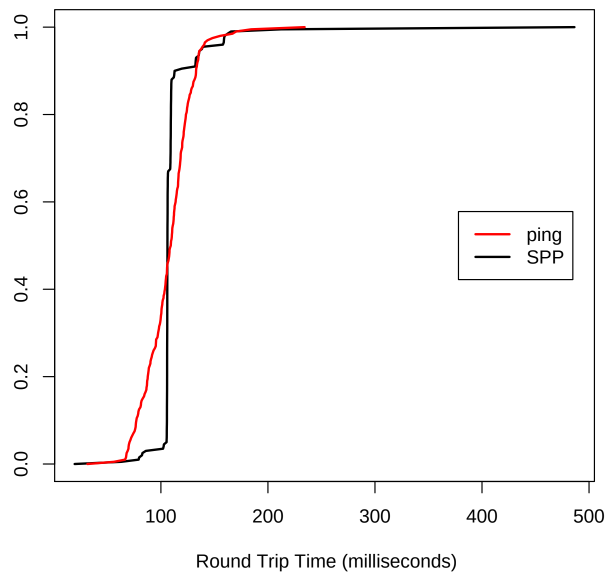

Fig. 19. Cumulative distribution of RTT during SCP transfer down the ADSL link. Ping shows a much larger spread.

of the large packets gave rise to quantisation. For the packet size of 1450 bytes (32 ATM cells) at 512kbits/sec the delay down the link was 26.5ms. This is reflected in the bands that are visible around 79, 105, and 132 milliseconds.

Further banding is clearly visible in figure 16, with intervals of 3.3ms. This quantisation comes from the underlying ATM cells. On the 128kbit uplink, serialisation delay of the 53 byte ATM cell is 3.3ms. These bands are evident because the ADSL standard specifies that ATM cells are sent constantly back to back in superframes [13], causing cell transmission times to be quantised.

The distribution of estimates by SPP can be seen in figure 17. Quantisation in both 3.3ms and 26.5ms intervals can be observed.

Figure 18 shows yet another level of quantisation, this time around 15 microseconds apart. This may be due to buffers or scheduling artifacts in the network interface card hardware or drivers of the end hosts. Further investigation is necessary to understand these bands. Experiments in a more highly controlled network environment will be required to achieve this.

Finally, the cumulative distribution in figure clearly shows the high density of SPP estimates at 105.8ms.

C. Interactive SSH session

SSH (Secure shell) is a commonly used tool remote administration. Due to its real time nature, high delays can impede user productivity. SPP can be used to investigate delay experienced by the user.

Figure 20 shows the delay experienced along the path during the SSH session as estimated by ping. Figure 21 shows the output from SPP. Since SPP generates estimates from most packets that make up the SSH session, the results are more detailed as a significantly greater number of RTT estimates are generated. SPP only produces estimates when observing traffic and therefore does not report estimates which are not pertinent to the application. As an example, figure 21 reveals that the session was completely idle during the period from about 80 to 120 seconds.

The cumulative distribution in figure 22 shows quite a difference between the two methods. Since ping is not measuring what is actually experienced by the SSH session, it significantly underestimates the fraction of packets of the session experiencing higher delay.

D. X11 application over SSH

The performance of remote X11 applications are highly sensitive to network delay. Once again SPP, can

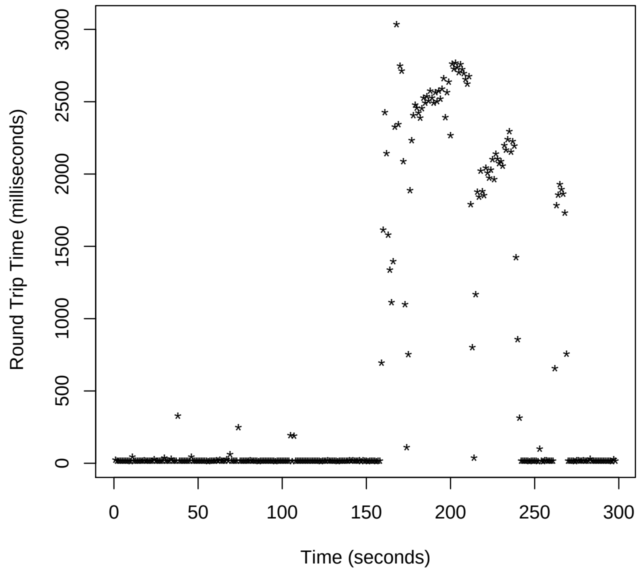

Fig. 20. RTT estimated by ping during an interactive SSH session.

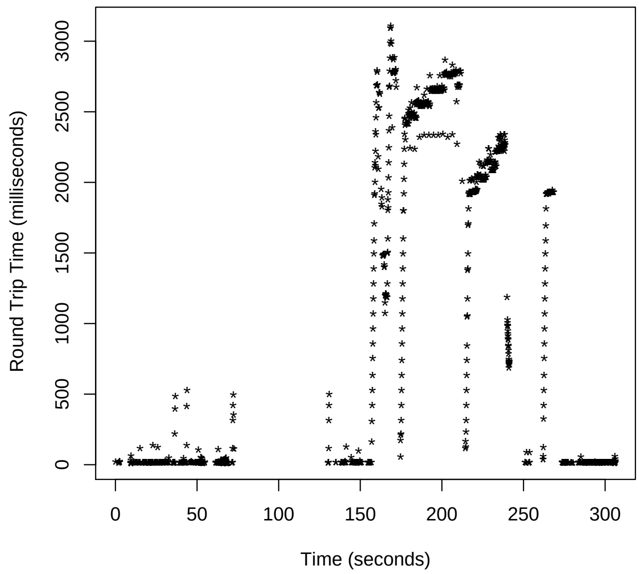

Fig. 21. RTT estimated by SPP during SSH session. Compared with ping, SPP gives a better understanding of the delay characteristics of the session.

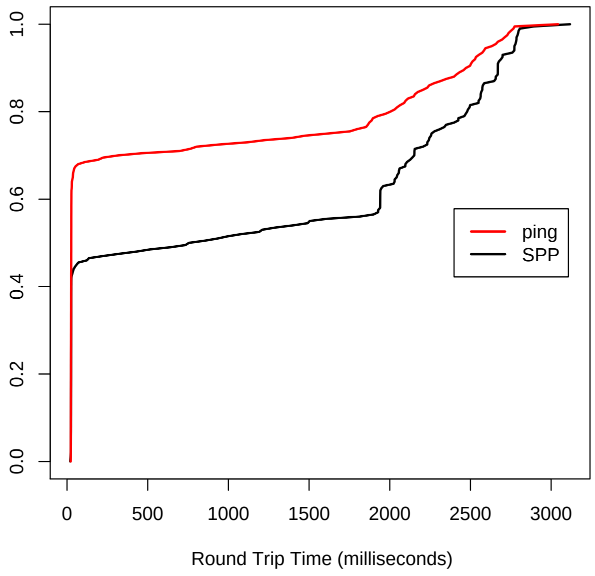

Fig. 22. Cumulative distribution of estimates generated during SSH session. Distribution as estimated by SPP is significantly different to that estimated by ping, since ping includes samples from periods when the session is idle and SPP does not. (See section III-C)

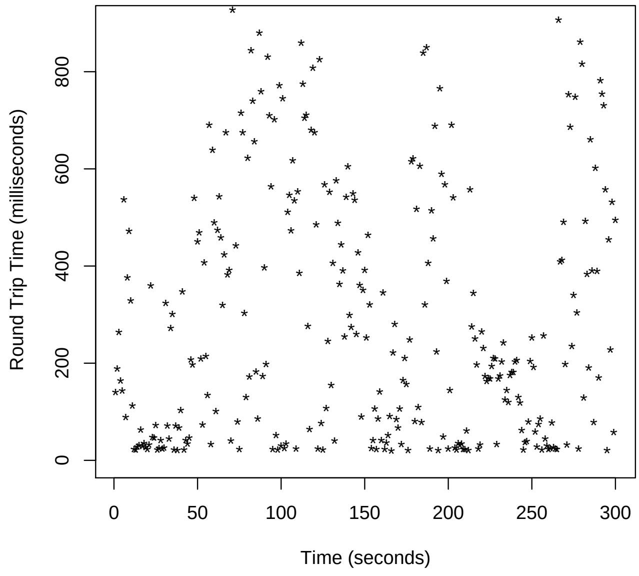

Fig. 23. RTT estimated by ping during use of an X11 application over SSH.

be used to look into the dynamics of delay experienced by the the application.

The Amarok music jukebox was run compressed over SSH with the command `ssh -C -X host amarok`. It was set to play a song, showing its small visualisation area. Songs were skipped a few times during the test period.

Figure 23 shows RTT_{PING} varying widely from around 25ms to over 800ms with no obvious trend.

SPP reports with much more clarity in figure 24. While RTT_{SPP} estimates still range from approximately 25ms to over 800ms, there are some interesting artifacts that are not revealed when using ping. Most notable are the distinct periods around 40, 160 and 250ms. RTT is fairly low during these periods. The behaviour during these periods was caused by skipping a song in Amarok. The time periods where RTT is spread over a large range is during normal playback, where the visualisations are being updated constantly. It appears that SPP is showing much lower RTT on average. Figure 25 confirms this, showing that a large number of packets experience less delay than would be expected, going by estimates from

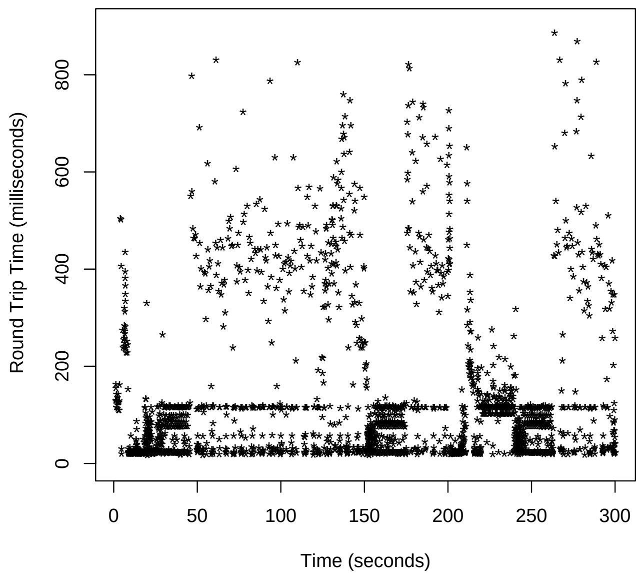

Fig. 24. SPP gives a better understanding of the delay characteristics during use of an X11 application over SSH.

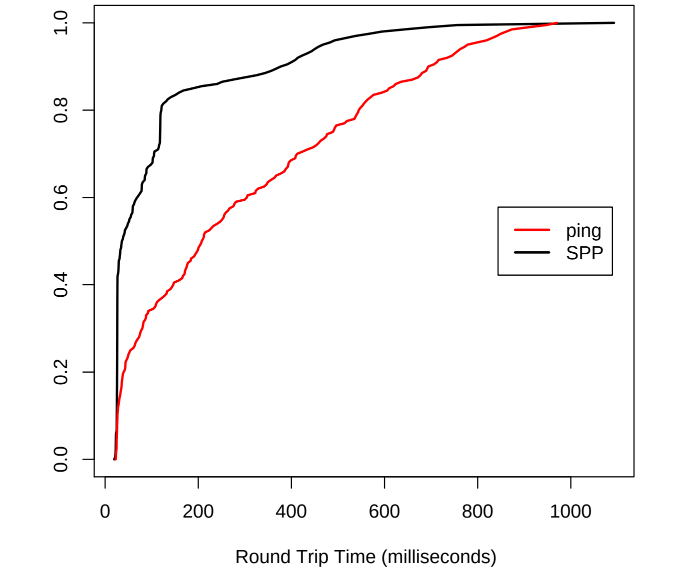

Fig. 25. Cumulative distribution of RTT during use of an X11 application over SSH. Distributions vary considerably. (See section III-C)

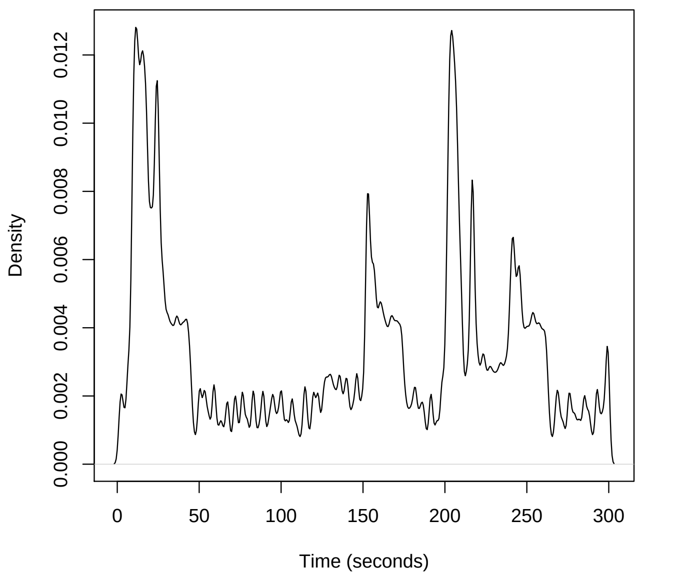

Fig. 26. The varying synthetic packet pair rate over time during the use of an X11 application over SSH.

ping.[4]

Since SPP is reporting a time and result for every pair observed, packet rate can be inferred as well. Figure 26 shows the pair rate observed throughout the trial. Interestingly, the times surrounding 40, 160 and 250ms show a pair rate which is above average. Given the low RTT during these periods, we can predict that either packet sizes were smaller or the interarrival time of these packets were more consistant. The first would reduce serialisation delay and the second would reduce queueing.

VI. Impact of QoS

SPP is especially useful for measuring RTT where different types of traffic do not traverse the same path or are treated differently by the network equipment along the path. A common cause of this is when routers are configured to set different priority levels for various classes of traffic, or reserve bandwidth for certain types of traffic. In previous examples ping gave a reasonable approximation to RTT but the following results show that when QoS is involved SPP is required, as active probing cannot give meaningful results.

In this example, Host B was configured with quality of service controls[5], reserving bandwidth and giving priority to SSH traffic. RTT was measured with SPP and ping during the course of an SSH session. A file download from host B to host A provided cross traffic during some periods of the session.

RTT_{SPP} as measured by SPP is shown in figure 27. The red lines along the top of the graph denote the periods when cross traffic was present. The blue line indicates increased activity in the SSH session. [6]

The SSH session was affected by the cross traffic. While RTT of the SSH session (RTT_{SPP}) is usually around 25ms, the figure depicts upward smearing during the presence of cross traffic. In contrast, figure 28 shows RTT_{PING}. At the beginning of the test, RTT_{PING} is low, much the same as experienced by the SSH session, but when the cross traffic appears, the RTT estimates increase by many seconds almost immediately. For the entire time the cross traffic is present, the ping times are not representative of the RTT experienced by the SSH

[4]Refer to section III-C for differences between cumulative distributions of ping and SPP estimates.

[5]The specific QoS implementation used was Hierarchical Token Bucket Queuing [14], with reserved bandwidth and priority given to SSH traffic.

[6]The cat command was used to echo a file to the command line during this period.

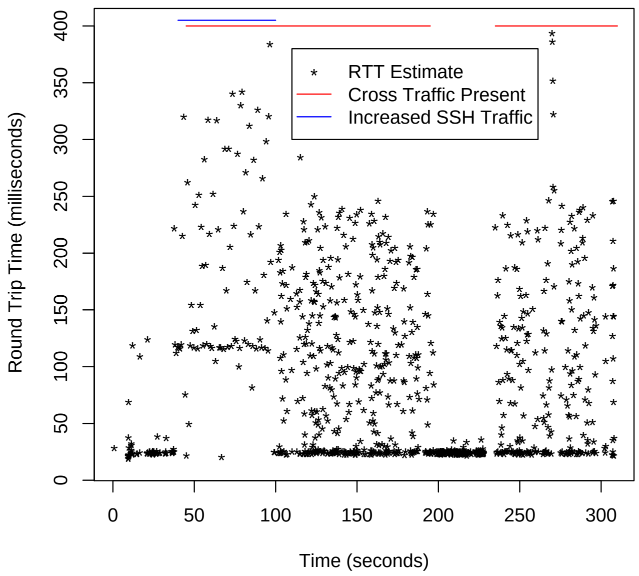

Fig. 27. RTT estimated by SPP during a prioritised SSH session. SPP shows a generally low RTT with upward smearing during periods where cross traffic is present

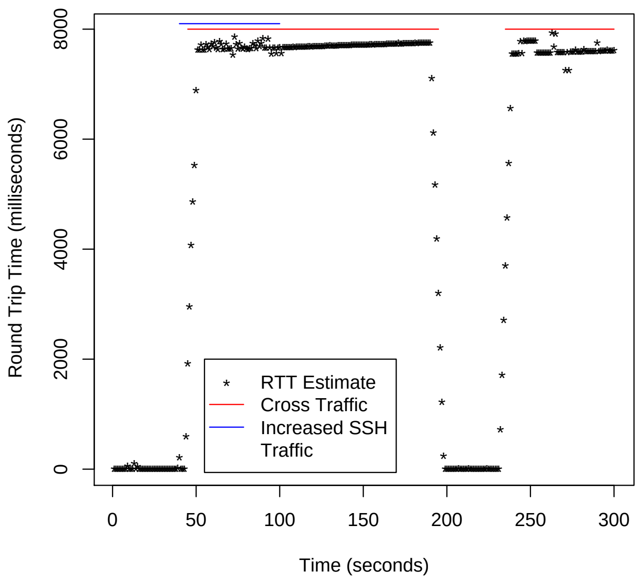

Fig. 28. RTT estimated by ping during a prioritised SSH session. ICMP traffic did not recieve priority. Ping reports an extremely high RTT of around 8 seconds when cross traffic is present.

session. This is because the ping probes are included in the same QoS class as the cross traffic, while the SSH session has its own reserved bandwidth, ensuring a lower RTT despite the cross traffic.

This example highlights what is perhaps the most useful application of SPP, that is determining the delay actually experienced by an application when traversing paths that treat each traffic type differently.

VII. Further Work

Since our SPP implementation has access to details of whole packet, it would be advantageous to make use of the extra information available. Pair size could be added as another field in SPP output, as well as a moving average of pair rate. These values could be used to infer further information about the path that would not otherwise be possible. Extra statistics may also be generated taking into consideration the packet size, such as an estimation of serialisation delays based on differences in the delay of different sized packets.

It would be advantageous to develop a better understanding of how the characteristics of target traffic affect the ratio of pair rate to packet rate, and to what extent relative pair rate is representative of relative packet rate. With this knowledge, a pair rate meassurement may help to determine whether variations in RTT are likely due to the target traffic, or other traffic on the same path.

VIII. Conclusion

Synthetic Packet Pairs have proven beneficial in the investigation of round trip time over an ADSL link. SPP consistently gave more detail and further insight than did active probing with ping. Queuing at the packet layer and ATM layer have been observed and SPP has been shown to report RTT experienced by an application when QoS is in use, where active probing could not. Overall, SPP proves to be a useful tool for network, protocol and application latency analysis.

References

[1] S. Zander, G. Armitage, T. Nguyen, L. Mark, and B. Tyo, “Minimally Intrusive Round Trip Time Measurements Using Synthetic Packet-Pairs,” Tech. Rep., July 2006. [Online]. Available: http://caia.swin.edu.au/reports/060707A/CAIA-TR-060707A.pdf

[2] A. Heyde, “Synthetic Packet Pairs software 0.2.0.” [Online]. Available: http://www.caia.swin.edu.au/tools/spp

[3] “Tcpdump.” [Online]. Available: http://www.tcpdump.org

[4] ABS, “Internet Activity Survey,” December 2007. [Online]. Available: http://www.abs.gov.au/ausstats/abs@.nsf/mf/8153.0/

[5] G. Gross, M. Kaycee, A. Li, A. Malis, and J. Stephens, “PPP Over AAL5,” RFC 2364 (Proposed Standard), Jul. 1998. [Online]. Available: http://www.ietf.org/rfc/rfc2364.txt

[6] L. Mamakos, K. Lidl, J. Evarts, D. Carrel, D. Simone, and R. Wheeler, “A Method for Transmitting PPP Over Ethernet (PPPoE),” RFC 2516 (Informational), Feb. 1999. [Online]. Available: http://www.ietf.org/rfc/rfc2516.txt

[7] D. Grossman and J. Heinanen, “Multiprotocol Encapsulation over ATM Adaptation Layer 5,” RFC 2684 (Proposed Standard), Sep. 1999. [Online]. Available: http://www.ietf.org/rfc/rfc2684.txt

[8] A. Tirumala, F. Qin, J. Dugan, J. Ferguson, and K. Gibbs, “Iperf, the TCP/UDP bandwidth measurement tool.” [Online]. Available: http://dast.nlanr.net/Projects/Iperf/

[9] “Secure Shell - OpenSSH.” [Online]. Available: http://www.openssh.org

[10] “Secure Copy - SCP.” [Online]. Available: http://nixdoc.net/man-pages/FreeBSD/scp.1.html

[11] J. Yonan, “OpenVPN, An Open Source SSL VPN solution.” [Online]. Available: http://openvpn.net/

[12] D. V. Aken and S. Peckelbeen, “Encapsulation Overheads in ADSL Access Networks,” June 2003, accessed 13 May 2008. [Online]. Available: http://www.oplnk.net/files/WhitePaper_EncapsOverheads.pdf

[13] R. H. Hafez, “ADSL and DMT,” 1997. [Online]. Available: http://www.sce.carleton.ca/courses/sysc-5403/f06/DSL_ST.pdf

[14] M. Devera, “Hierachical token bucket theory,” 2002. [Online]. Available: http://luxik.cdi.cz/~devik/qos/htb/manual/theory.htm