An empirical comparison of packet loss on end-user perception of performance

Chris Holman*, Jason But, Phillip Branch
Centre for Advanced Internet Architectures, Technical Report 120405A
Swinburne University of Technology
Melbourne, Australia
6963420@student.swin.edu.au, jbut@swin.edu.au, pbranch@swin.edu.au

***Abstract*—In this report we investigate the impact of randomised packet loss on TCP transfers using FreeBSD 8.2. We then evaluate these outcomes with respect to web interactivity and responsiveness. We find that even small rates of packet loss have a significant detrimental impact on perceivable performance of a website, and that avoiding packet loss is likely to be a better strategy than increasing bandwidth. We also investigate the effect of delay, bandwidth bottlenecks and queue size to compare with packet loss.**

I. Introduction

One of the main features of Transmission Control Protocol (TCP)[1] is its guaranteed delivery of data. It does this by defining how data loss should be detected, the data retransmitted and the steps to be taken to avoid further losses. This retransmission, however, takes time and causes (an often significant) drop in goodput, particularly due to the transmission rate being temporarily reduced as a congestion avoidance mechanism. This report investigates the impact of randomised packet loss on TCP, as perceived by the end user, as well as comparing the goodput reductions due to packet loss against goodput gains due to increasing bandwidth.

Previous work in this area [2] investigated long term goodput obtained by TCP steams experiencing packet loss. We expand on this work with a study on short-lived flows of the type typically found in a home user environment.

Google conducted research into the impact of slower page loads on users [3] and found that even a delay of 200ms had a perceivable effect, while 400ms saw a 0.59% decrease in searches - a significant figure when considering the number of requests serviced by Google, a result that demonstrates the impatience of end users. The delay described in their research was inserted during the page processing stage on their servers, however the same conclusions can be applied to delays caused by packet loss or otherwise - a user does not know or care where the delay in a web page appearing occurs, only that it does.

TABLE I
Example bandwidths, packet loss rates and file sizes used in most experiments

Bandwidth	Packet loss	File size
1500kbps	0%	15kB
6000kbps	2%	64kB
15,000kbps	5%	256kB
25,000kbps	8%	1MB
	10%	20MB

Harrop and Armitage [4] propose that a family of five may require up to 113Mb/s of bandwidth for modern applications. This figure will surely benefit long running, high bandwidth streams (such as streaming HD TV) but we will consider whether this recommendation would be noticeable by end users while undertaking typical browsing habits.

We find that transfer time increases exponentially with packet loss, that different queue sizes have negligible impact relative to packet loss, and that increasing available bandwidth has limited returns for typical web browsing habits.

The remainder of the paper is structured as follows: Section II detailing the construction of the test bed and how the data is analysed. Section III details some raw data from the experiment. Section IV explores the meaning behind the data.

II. Experimental method

Three PCs were configured with *FreeBSD 8.2 RELEASE* running the default NewReno congestion control algorithm (see Figure 1). Two of these were configured

*The author was on an internship at CAIA while writing this report

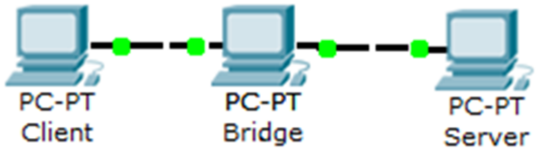

Fig. 1. The configuration of hardware

as endpoints while the third was configured as an Ethernet bridge. We implemented traffic shaping and random packet loss using *ipfw* and *dummynet* on the bridge device.

The benchmark program *nttcp*[5] was used to generate traffic between endpoints. *nttcp* can create a TCP or UDP stream consisting of an arbitrary amount of test data, and then reports basic statistics on the generated flow.

tcpdump[6] was used to capture traffic from the network interface on the client side of the bridge, akin to capturing packets from a modem's LAN interface in a real world setup.

Experiments were conducted using all combinations of a variety of link capacities, packet loss rates and transfer file sizes as shown in Table I. Unless otherwise stated, the network delay was configured to 0ms in the bridge, which resulted in a measured latency of 0.5 ± 0.5ms. This incurred latency is a result of dummynet's internal architecture.

Between 5 and 500 iterations of each trial configuration were executed, varying on total payload size, to ensure that adequate packets were transfered to average out the effect of the random packet loss.

This series of tests focused on single TCP streams. The total transfer time is calculated as the time from the first packet with payload data to the final packet with payload data.

III. Results

As shown by Figure 2, an exponential curve fits the data set quite well. This is consistent with all captured data sets. Figures 2 and 3 reinforce this. Other data models were tested and exponential was determined to be the best fit.

At very small payload values, such as 5kB, shown in Figure 4, available bandwidth has very little effect on the results. As 5kB fits inside four packets, a large amount of noise is inherent in the data. At 1.5Mb/s, a packet takes roughly 8ms to be transmitted. When a packet is lost, the sender must wait for the retransmission timer to

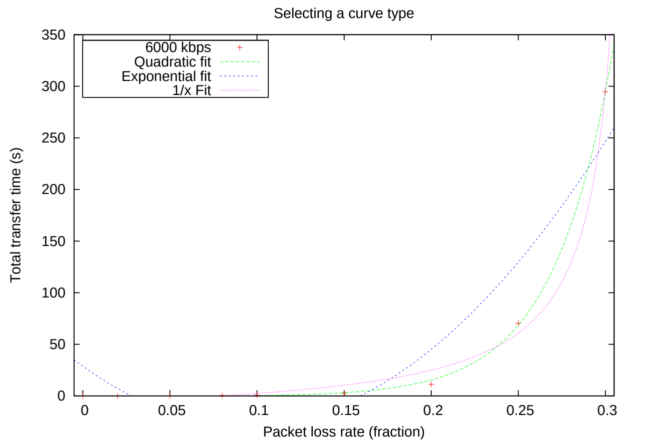

Fig. 2. Fitting different curves to the data set - 64kB filesize at high packet loss rates

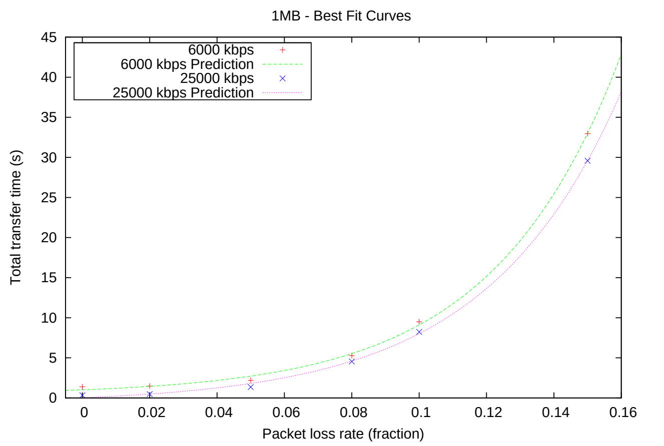

Fig. 3. 1MB TCP stream transfer times, showing best fit curves

expire. [1] Larger payload sizes (see Fig 5) produce data with significantly less noise.

IV. Analysis

We attempted to form a generic equation from the gathered data to predict transfer time given a filesize, bandwidth and packet loss. To avoid introducing extra variables, no artificial delay was configured within dummynet

A. Generalising the data

Using gnuplot's best fit algorithm, the individual data series were fitted to the best fit curves of form shown in Equation (1).

[1]Default settings have TCP fast retransmit engage after four ACKs for the same sequence number. As there are only four packets total, at most three duplicate ACKs could be sent if the first packet was lost. Fast retransmit will not occur

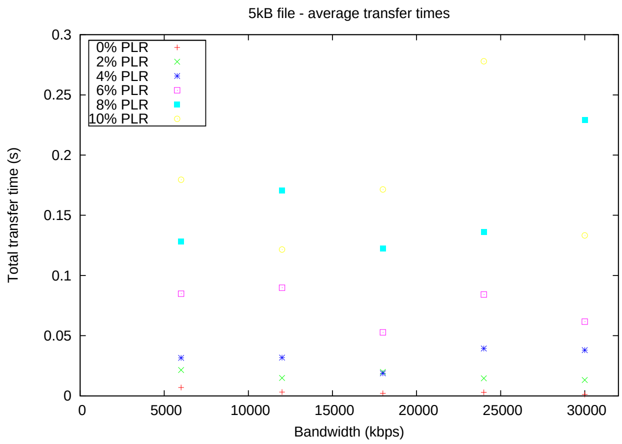

Fig. 4. TCP transfer time for 5kB payload

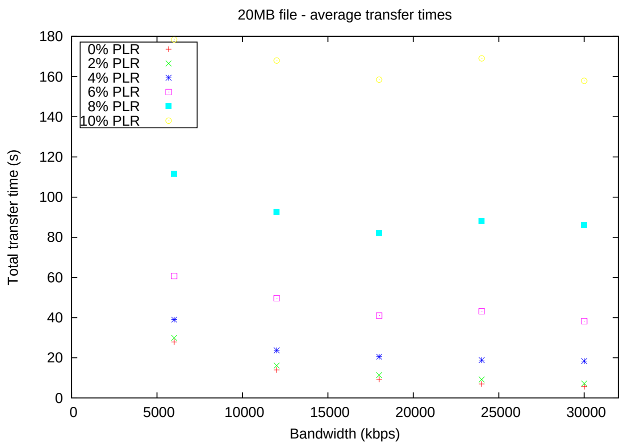

Fig. 5. TCP transfer time for 20MB payload

$$t = a + be^{c*PLR} \tag{1}$$

Figure 3 shows an example best fit curve for a 1MB file plotted against the raw data points. In general, a curve to fit most data series was found. Figure 3 shows the quality of the fit for individual curves.

The curves of best fit for any given file size were similar - the b and c coefficients for the equation 1 were within a very small range, and averaging them in order to generalise did not introduce significant error. The a coefficient served as an offset.

The b and c coefficients for each file size were then plotted to find patterns, shown in Figures 6 and 7. Equations for the two coefficients (b,c) and the offset (a) were found (equations (2), (3) and (4)), their numerical values are tabulated in Table II.

TABLE II
COEFFICIENTS FOR THE BEST FIT EXPONENTIAL CURVES

File size	b Coefficient	c Coefficient
5kB	0.04	21
15kB	0.03	22
256kB	0.2	25.5
1MB	0.6	27
20MB	7	30.5

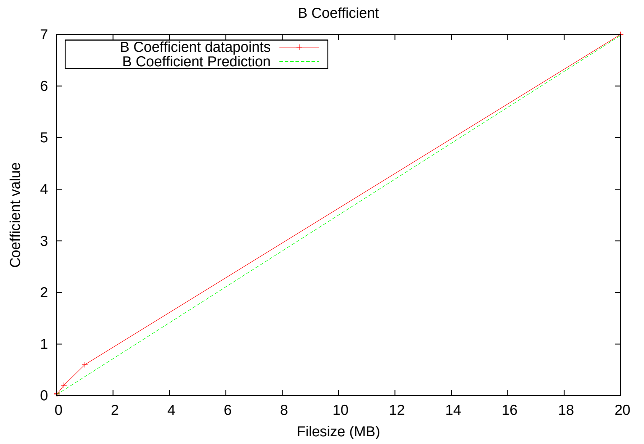

Fig. 6. The b coefficient is linearly proportional to the filesize

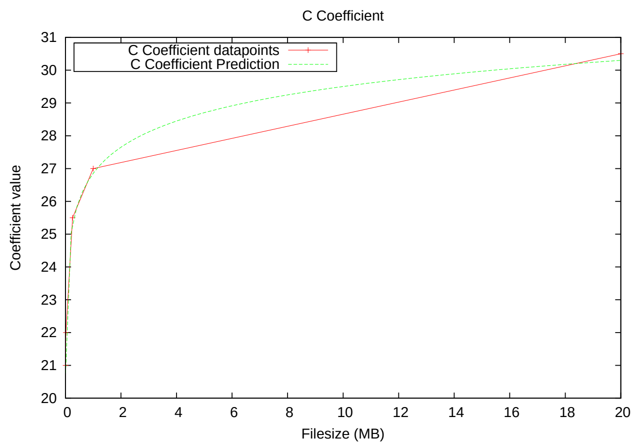

Fig. 7. The c coefficient is logarithmically proportional to the filesize

$$a = size/bandwidth - b \tag{2}$$

$$b = 0.02 + 3.48 * 10^{-7} * size \tag{3}$$

$$c = 0.02 + 1.15 * log((size + 1) * 13600) \tag{4}$$

Figure 8 shows this equation plotted against one set of data. The trend across all sets of data is that the curve

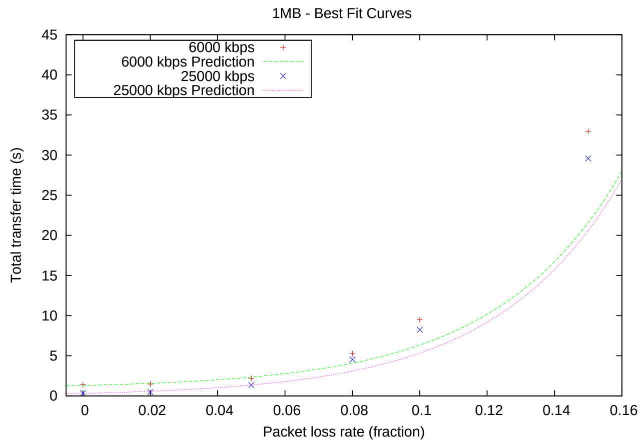

Fig. 8. Original prediction for a 1MB file

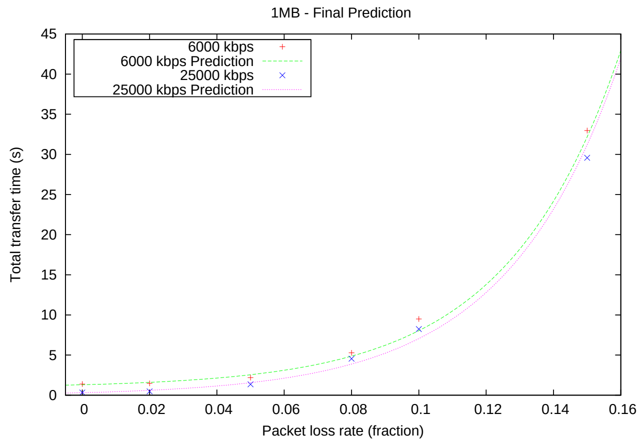

Fig. 9. Final prediction for a 1MB file

does not rise fast enough. An alteration to the equation is made such that the c coefficient is increased according to the size of the file.

$$c = 0.02 + 1.15 * log((size + 1) * 13600) + \frac{40}{log(size * 2)} \tag{5}$$

Figure 9 shows the predicted curves against the experimental data. The added factor based of $40/log(size*2)$ helps to make the prediction more accurate, but it still maintains an average error of around 35%[2]. If the upper bound of the test is kept at 5% packet loss or the 5kB and 15kB file sizes are ignored, the error can be as low as 5%.

Once we determined the generic formulae, we tested its predictiveness using a previously unused file size (500kB) and bandwidth (10Mbps). Figure 10 shows the raw data and prediction for a file size of 500kB. For this set of data points, most errors are <10%, ranging up to 36% for 10% packet loss at 1500kbps. Figures 11 and 12 show the prediction applied against tests running at 10Mbps for two different file sizes.

B. Attempting to integrate delay

We also attempted to integrate delay in to our model. The data obtained showed less of a pattern than previous data sets. Figures 13 and 14 show two sets of experimental data for 30ms. The exponential curve fits the data reasonably well again, but no way to generalise the curves accurately was discovered.

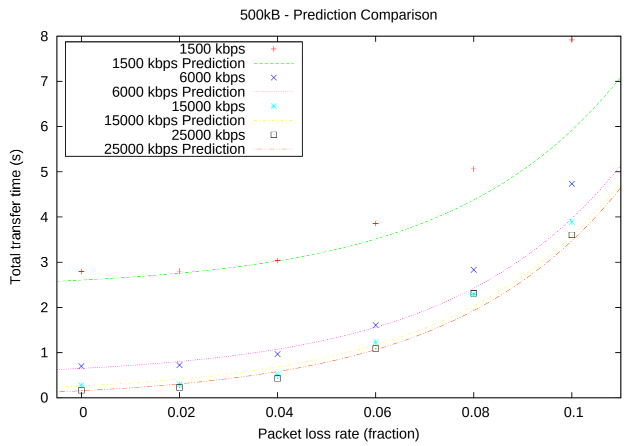

Fig. 10. Applying the prediction to different data - 500kB file size

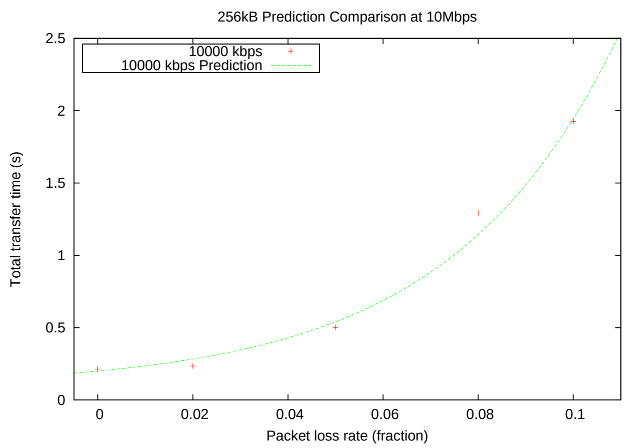

Fig. 11. Applying the prediction to different data - 10Mbps

[2]The small file sizes contribute most of the error. If 5kB and 15kB are ignored, an average error of 6% is seen

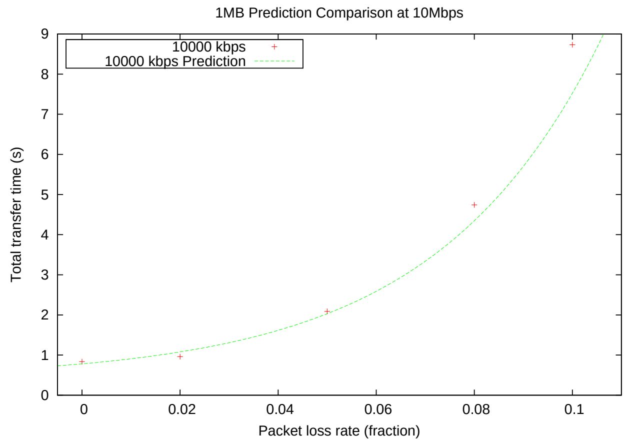

Fig. 12. Applying the prediction to different data - 10Mbps

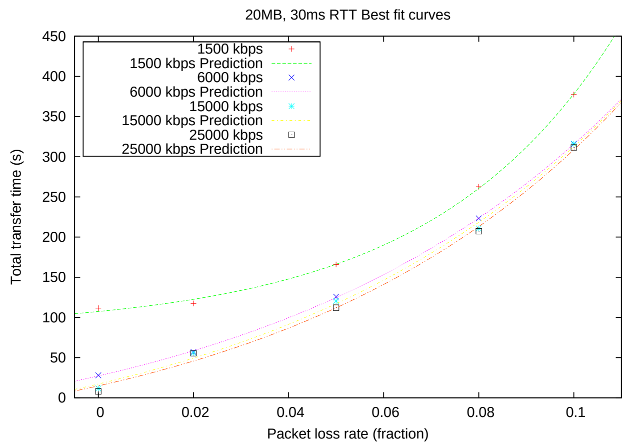

Fig. 13. A 20MB file with a 30ms RTT

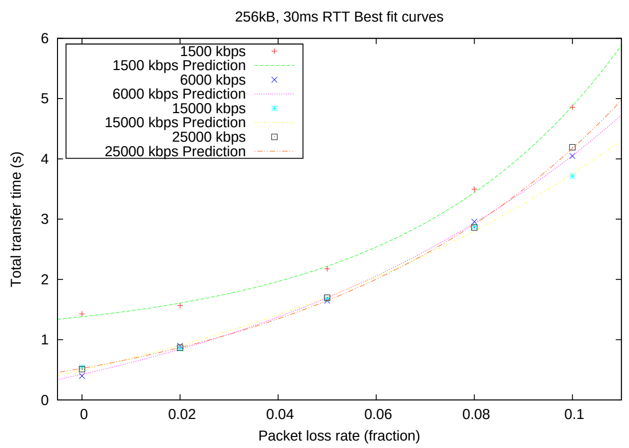

Fig. 14. A 256kB file with a 30ms RTT

TABLE III
DELAYS IN WEB PAGE LOAD TIMES INDUCED BY PACKET LOSS

Element	Size (kB)	Base (s)	2%	5%
HTML	15kB	0.069	+0.010	+0.060
Images or JS	256kB	0.399	+0.501	+1.25
Total		0.468	0.979	1.778

C. Implications for web browsing

Google's research into user responsiveness[3] demonstrates users begin to respond negatively if they were kept waiting as little as 200ms longer than normal. As packet loss induces extra delay on flow completion, this research is relevant to analysing how packet loss impacts the perception of network performance.

For the purposes of this comparison, we use 6Mb/s of bandwidth and an RTT of 30ms to the website - the lower bound for an ADSL connection accessing websites hosted in the same region.

An average web page is 784kB in size [7], consisting of around 450kB of images, 150kB of javascript and 36kB of HTML. Since the HTML must be retrieved before any further elements can be requested [3] or rendered, it is the most time critical piece. Using the closest file sizes from our experiments - 15kB - an extra 10ms is required for the transfer to complete at 2% packet loss as compared to the ideal case. In the case of 5% packet loss, an extra 60ms is required. Assuming an average sized image or JavaScript file is around 256kB, 2% packet loss increases the total download time by 400ms while 5% loss rate increases this by a further 750ms. This is summarised in Table III

Additionally, delay also has an effect on total transfer times. Using otherwise identical circumstances, when the RTT is increased from 30ms to 100ms, the 2% loss figures rise from 10ms to 70ms for a 15kB HTML transfer and from 400ms to 2000ms for a 256kB transfer.

The majority of Australian consumer Internet connections are above 5Mb/s, with 92.7% above 2Mb/s. [8] As the NBN is rolled out to Australian homes, this figure will rise significantly - NBN service is expected to provide a minimum of 12Mb/s link capacity [9] to 93% of residences [10].

Our previous results indicate minimal changes in total download times at these higher rates. For the average Internet user[4], we do not expect this increase in available

[3]Google's SPDY protocol alleviates this by giving the server the ability to pre-emptively push content to the client before the client asks for it

[4]Assumed to be someone who browses Facebook, checks their GMail and watches a few videos on YouTube

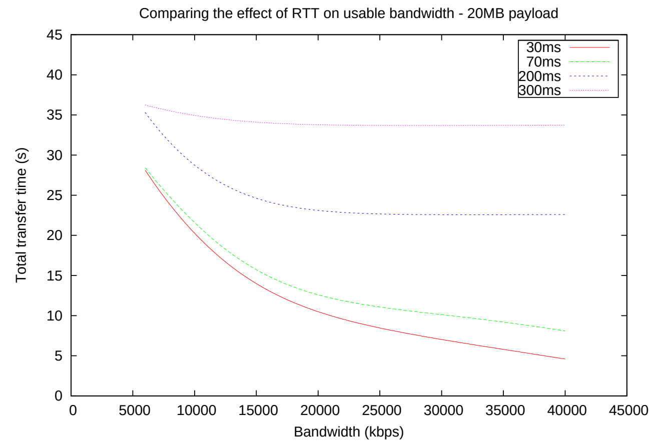

Fig. 15. Transfer times vary a lot more under lower RTTs than higher

bandwidth to result in a noticeable improvmeent in access times.

Table IV shows the average total transfer times for TCP streams at 100ms. As shown, with a 100ms Round Trip Time (Eastern Australia to Japan, or half that of the RTT from eastern Australia to Western USA) [5] , benefits for bandwidth increases are negligible over 6Mb/s where the file size is typical of a web page element. Conversely, a noticeable improvement is seen over NBN equivalent speed increases for larger flows (>1MB) from local data sources (30ms RTT [6])

Figure 15 plots the impact of RTT delay on total transfer times using a range of bandwidth values. These results re-affirm that as RTT increases, any extra available bandwidth will not have a major impact on the time taken to transfer data. This indicates that bandwidth is not the limiting factor when communicating with distant servers. On the other hand, when communicating with geographically local data sources, any increase in the available bandwidth will be used by TCP to deliver appreciably quicker download times.

D. Queue size investigation

As a tangent to the previous research, the impact of changing the available queueing in the network was investigated. Larger queue sizes can increase the RTT and affect packet loss rates in a non-random manner. When a queue fills, it drops packets to let the TCP control algorithms know that it needs to back off. We combine this non-random packet loss with random loss rates and observe the results for different queue capacities.

For unrealistically low queue sizes (eg, 1 packet), large variations in transfer times are observed due to unavoidable packet loss. Otherwise, very little impact on transfer time was seen with low RTT. Increasing the RTT from 30ms to 100ms saw the impact rise.

Figures 16 and 17 show how queue size impacts a 1MB file at 30ms and 100ms RTT respectively - or rather, how the queue size does not have a *significant* impact on total transfer times. There is a very slight trough in the data at 50 packets, but even 1% packet loss has a more significant impact than all tested queue sizes except 1. Our results indicate that buffer bloat may not be as important an issue under certain circumstances where the flow is not exceptionally long-lived.

[5]RTT to fremont1.linode.com from a TPG home internet connection, 30 packets: rtt min/avg/max/mdev = 204.533/205.180/206.522/0.690 ms

[6]RTT to gmail.com from a TPG home internet connection, 30 packets: rtt min/avg/max/mdev = 29.305/30.758/43.294/2.424 ms

TABLE IV
FLOW DURATIONS (S) WITH 0% PACKET LOSS AT 100MS RTT

Bandwidth	15kB file	256kB file	1MB file
1.5Mb/s	0.2310	1.54	5.69
6Mb/s	0.2090	0.68	1.79
15Mb/s	0.2040	0.65	1.75
25Mb/s	0.2040	0.61	1.70

TABLE V
FLOW DURATIONS (S) WITH 0% PACKET LOSS AT 30MS RTT

Bandwidth	15kB file	256kB file	1MB file
1.5Mb/s	0.0900	1.43	5.58
6Mb/s	0.0690	0.542	1.44
15Mb/s	0.0640	0.511	1.15
25Mb/s	0.0640	0.399	0.508

V. CONCLUSION

This report investigated completion times for file transfers, while varying size, bandwidth and packet loss. The total transfer time was modeled successfully using exponential functions, as derived in section IV-A. There was found to be an exponential relationship between

TABLE VI
FLOW TIMES (S) FOR A 1MB FILE OVER 0% PACKET LOSS

Queue size	30ms RTT	100ms RTT
1	15.86	43.33
10	2.66	7.013
50	1.146	3.633
100	0.6342	1.230
200	0.6350	1.231
500	1.1467	3.639

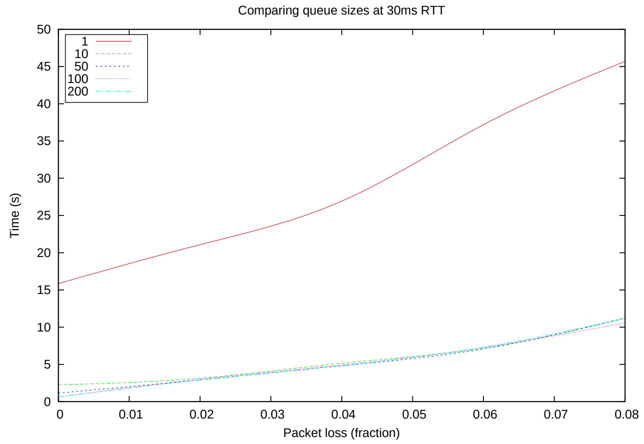

Fig. 16. Transfer times for a 1MB file given a variety of queue sizes and packet losses with 30ms RTT. Note: all data series except '1' are overlapping

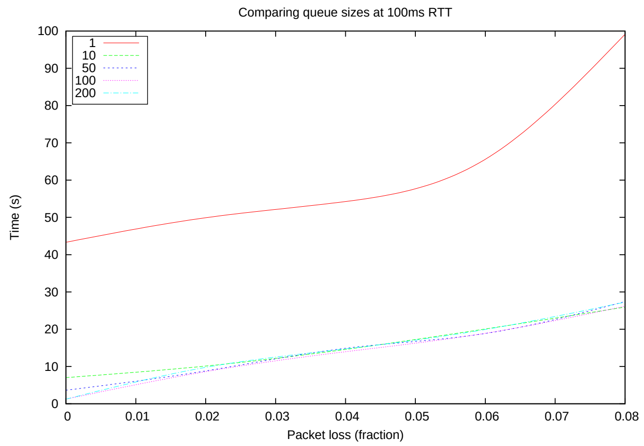

Fig. 17. Transfer times for a file given a variety of queue sizes and packet losses with 100ms RTT. Note: all data series except '1' are overlapping

packet loss and transfer times. Packet loss had a more severe impact on transfer times than increasing bandwidth and optimising queue sizes, with all three amplifying the effects of the others.

Additionally, varying the queue size had almost no perceivable impact on transfer times at realistic RTTs, except when the queue size was set unrealistically low - in our experiment, a queue size of 1 packet.

References

[1] J. Postel, "Transmission Control Protocol," RFC 793 (Standard), Internet Engineering Task Force, Sep. 1981, updated by RFCs 1122, 3168, 6093, 6528. [Online]. Available: http://www.ietf.org/rfc/rfc793.txt

[2] M. Mathis, J. Semke, J. Mahdavi, and T. Ott, "The macroscopic behavior of the tcp congestion avoidance algorithm," *SIGCOMM Comput. Commun. Rev.*, vol. 27, no. 3, pp. 67–82, Jul. 1997. [Online]. Available: http://doi.acm.org/10.1145/263932.264023

[3] J. Brutlag, "Speed Matters for Google Web Search," Accessed Feb 2012, original location returns *file not found*. Mirror of original can be found at http://tinyurl.com/7avrcs7. [Online]. Available: http://code.google.com/speed/files/delayexp.pdf

[4] W. Harrop and G. Armitage, "Quantifying the broadband access bandwidth demands of typical home users," in *Australian Telecommunication Networks and Application Conference 2006*, Melbourne, Australia, 4-6 December 2006, pp. 16–20. [Online]. Available: http://caia.swin.edu.au/pubs/ATNAC06/Harropm.pdf

[5] "NTTCP." [Online]. Available: http://www.freebsdsoftware.org/benchmarks/nttcp.html

[6] "tcpdump." [Online]. Available: http://www.tcpdump.org/

[7] Pingdom, "Web pages are getting more bloated, and here's why," Accessed Jan 2012. [Online]. Available: http://royal.pingdom.com/2011/11/21/web-pages-getting-bloated-here-is-why/

[8] Whirlpool, "Australian Broadband Survey 2011 Report," Accessed Jan 2012. [Online]. Available: http://whirlpool.net.au/survey/2010/

[9] iiNet Pty Ltd, "iiNet NBN Business Plans," Accessed Jan 2012. [Online]. Available: http://www.nbn.gov.au/wp-content/uploads/2011/05/august-update.html

[10] NBN Co Ltd, "National Broadband Network: Progress Update August 2011," Accessed Jan 2012. [Online]. Available: http://www.nbn.gov.au/wp-content/uploads/2011/05/august-update.html