Reducing US/NZ Web Page Latencies

M. Pearson[a], A. McGregor[b] and J. Cleary[a]

[a]The University of Waikato,
Private Bag 3107, Hamilton, New Zealand

[b]National Laboratory for Applied Network Research (NLANR),
San Diego Super Computer Center,
10100 Hopkins Drive, San Diego, CA 92186-0505, USA

ABSTRACT

Much of New Zealand's Web traffic crosses the Pacific Ocean on sub-oceanic fiber channels. This imposes approximately 60ms of latency in each direction. Because fetching a web page often requires the fetching of many HTML components and each component requires several round trip times this latency is multiplied many times in a typical web page retrieval.

Caches and proxies are often used to reduce the effect of these delays. Providers of Internet service have several choices in how to deploy these components. Depending on the architecture chosen the traffic on the international circuit might be made up of a large number of independent connections or a smaller number of connections carrying aggregated traffic. The appropriate approach is not immediately apparent because there are opposing performance factors.

In this paper we describe a discrete event simulation of the effect of carrying multiplexed HTTP connections over an simulated NZ/US circuit. We show that a high degree of multiplexing mitigates against TCP's bandwidth delay product limits but that carrying every HTTP connection in a seperate TCP causes a significant increase in delay.

The simulation environment used includes a TCP/IP stack derived from a real TCP implementation and is driven by data collected from a large New Zealand web cache.

Keywords: TCP/IP performance, HTTP, cache, proxy

1. INTRODUCTION

Part of the interest in the Internet in New Zealand is because the country is geographically isolated. The Internet allows some New Zealand industries to compete on a more equal footing with those in major northern hemisphere countries, particularly North America. Distance does, however, introduce some special problems in the Internet. Communication from New Zealand to the United States, where most of New Zealand's International traffic is routed, is expensive. This means it is important to utilise the interenation connection well. Secondly the distance imposes extra delay. This paper investigates the use of HTTP proxies to reduce the time required to fetch a web page from the US. The savings reported here are in addition to those that occur because the page is feteched from the proxies local cache.

A New Zealand ISP, that is providing international connectivity, must choose an architecture for the International component of their network. We compare two such architectures. Depending on the architecture chosen the traffic on the international circuit might be made up of a large number of independent connections or a small number of connections carrying aggregated traffic. The former would be the case if each HTTP request is carried directly over the link. The latter is the case if user connections are concentrated between a pair of proxies. More tuning is possible if proxies are used because the TCP connection is terminated at devices under the control of the ISP. The TCP stacks can be tuned to better meet the needs of the international connection. Without proxies TCP connections terminate at the browser and the web server, whose TCP implementations are outside the control of the ISP. These two cases are shown in figure 1. To make the discussion easier this figure and the rest of the paper, are written in terms of an international connection between New Zealand and the United States. The results are, of course, more widely applicable.

E-mail: {m.pearson,t.mcgregor,j.cleary}@cs.waikato.ac.nz

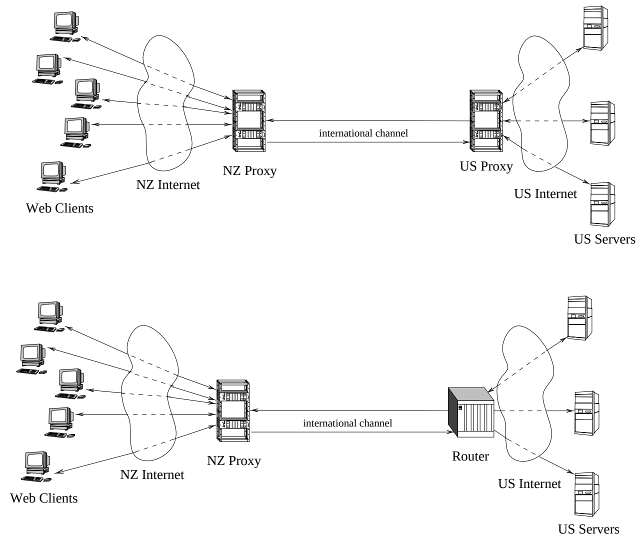

Figure 1. International Architectures

There are advantages to both design approaches described above. The NZ only proxy case is simpler to implement and does not require the ISP to deploy and maintain US based proxy equipment. However the full effect of slow start* will affect every HTTP request. This will be multiplied for most web pages because the pages often consist of multiple HTTP components.

The US-proxy case improves slow start behavior because the inital slow start only occurs once for each inter-proxy TCP connection not for each HTTP request. Slow start will still occur locally within NZ and the US but it has less impact at these points because the latency is lower. Further performance may be gains are possible in the proxy-to-proxy case because the TCP stacks operating over the international link are under the control of the ISP and may be tuned. In particular a large buffer size may be selected. Finally the aggregation of several HTTP connections over a single TCP connection may allow TCP to better package the data and to carry more piggy-backed acknowledgments. Opposing these performance gains for the proxy-to-proxy case performance may be limited by the number of TCP connections available between the proxies. The performance of TCP connections in networks where the poduct of delay and bandwidth is high is known to be limited.

In this paper we describe a discrete event simulation that investigates the effect of carrying multiplexed HTTP connections over a high delay bandwidth product circuit. The simulations include a real TCP/IP protocol stack and are driven by a trace of HTTP activity collected from the NZ international exchange (NZIX). The simulations indicate that multiplexing HTTP connections between proxies at both ends of the link reduces international latency provided sufficient TCP connections are available between the proxies.

The rest of this paper is organised as follows. Section 2 describes the network being simulated, including the capacities of the links and transmission delays. Section 3 describes the workload including the main characteristics and how heavier workloads were formed to simulate high load on the links. The simulator design is explained in section 4 and the results of the simulations are shown in section 5. The paper ends with the primary conclusions we draw from the results.

2. SIMULATED NETWORK

There are may ways in which an International NZ/US connection might be deployed as part of an ISP's network. This paper considers two architectures.

*To reduce the chances of catastrophic network congestion TCP has a number of congestion control mechanisms including slow start. Under slow start traffic is sent into the network at an exponentially increasing rate until congestion is detected at which stage the sender re-enters slow slow start but with the additional knowledge of where congestion occurred previously it reduces the rate of increase before that point is reached. See RFC2001[1] and Internet Draft tcpimpl-cong-control-02[2]

US and NZ proxies

The main elements of the first of these architectures is shown in figure 1. The diagram shows web clients located in NZ connecting to a NZ proxy. This proxy in turn connects to a US based proxy which, in its turn, connects to web servers based in the US. There are three TCP connections involved in fetching a web page. The first connects the web client to the NZ proxy, the second is between the two proxies and the third is from the US proxy to the US server.

Multiplexing is implemented between the proxies. That is the data for different replies may be interleaved on a single TCP connection between the proxies. The overhead of multiplexing is assumed to be, on average, 20 bytes per HTTP reply segment received from an HTTP server. It is expected that multiplexing will improve the efficiency of the international link. Because TCP does not maintain the boundaries between application requests the data from (possibly different) HTTP reply packets may be repackaged for more efficient TCP transmission. In most cases the TCP segments will be the maximum MSS size.

Because the connections between the proxies persist indefinitely the effect of TCP slow start is greatly reduced over the international component of the network, which is where slow start would otherwise have the greatest effect.

The number of concurrent TCP connections between the proxies may be limited. Each active TCP connection consumes resources and this imposes a maximum value. This parameter may als be tuned to improve performance. if a very large number of concurrent connections are available some of the performance advantages described previously will not occur. For example the effect of slow start will be felt by a larger proportion of HTTP requests as the number of TCP connections between the proxies increases. We investigate this issue by simulating differeing number of connections between the proxies.

NZ only proxy

A second, simpler, case is also considered. In this second case the US proxy is omitted. Only two TCP connections are involved with fetching a web page. The first, from the web client to the NZ proxy, is the same as in the US-proxy case. The second connection is from the NZ proxy to the US server providing the web page. In this case there is a TCP connection across the international link for each HTTP request.

Table 1. Main Network Parameters

US/NZ Bandwidth	34.368Mbps (E3)
US/NZ Delay	320ms[3]
TCP buffer size Proxies Servers	 32767 as measured
Maximum Segment Size Between proxies Elsewhere	 1460 as measured
Delays in US cloud	as measured
Delays in NZ cloud	not simulated

The main parameters of the network are shown in table 1. The values have been chosen to match real network parameters where possible.

As noted above the bandwith delay product of the network has an impact on TCP performance. The bandwidth delay product for this network is given by:

$BDP = D * 2 * B$

where:	$DELAY$	is the trans-pacific latency
	B	is the link bandwidth

so

$$\begin{aligned} BDP &= (0.060 * 2) * 34.368 \\ &= 4.1 \text{ megabits} \end{aligned}$$

This means that the TCP retransmit buffers need to be larger than 516kb to allow this link to be filled by a single TCP connection. If smaller buffers are used the buffer will be filled before the first data sent has been acknowledged and the flow of data into the link will be suspended while the transmitter waits for an acknowledgment. Standard TCP limits the window size to 64kb.[4] The big window extension for RFC1323[5] extends this limit to 2^{32}. Use of this option is not widespread and is outside the control of the ISP.

If an implementation is limited to 32767 bytes (a common implementation maximum) then the maximum bit rate for a single TCP connection over this link is:

$$\begin{aligned} MBR &= S/(D * 2) \\ &= 32767 * 8/(0.060 * 2) \\ &= 2.2 Mbps \end{aligned}$$

where:	D	is the trans-pacific latency
	S	is the buffer size

This effect is seen in simulation of the link and is shown in figure 4. This graph shows that the bandwidth consumed by a single TCP connection as the offered load increases plateaus at around 2.2Mbps. This bandwidth limitation is expected to impact negatively on the performance of the international connection, especially where a US based proxy is used and the number of connections between the US and NZ proxies is small.

Real International architectures would be not be as simple as the one described in this paper. Most will need more than a single proxy at each end of the international link to support the required load. The NZ proxy would almost certainly include a cache that satisfies some of the HTTP requests locally. There are many routers not shown. The simpler architecture used in this paper makes the simulation easier and shows the differences between the international implementations without interference from the full range of factors that would impact the performance of a real system.

3. SIMULATED WORKLOAD

Most of the information required to generate the simulation input files (described in the next section) was gathered from HTTP logfiles collected from the New Zealand Internet exchange (NZIX). The trace files used were collected from 3:00pm to 3:10pm in July 1997.

There were, on average, 421 requests† per interval.

To generate higher loads than those experienced when the trace files were collected, traces for the same time on successive days in July were integrated into a single trace. When higher still loads were required more than one copy of each trace was integrated into the logfile. Each copy was offset in time to minimise the effect of the artificial self correlation of the trace generated in this way.

The TCP MSS and server buffer sizes are not recorded in the HTTP traces we used. To discover these parameters a connection was established to each host while the network traffic was monitored using tcpdump. From the tcpdump output the MSS and window size was discovered for most hosts. Some hosts did not advertise their MSS. In this case the most common MSS (1480) was used.

†The complete trace includes requests that were satisfied by the cache or were not successful. There were 421 successful international requests that were not satisfied by the cache hierarchy

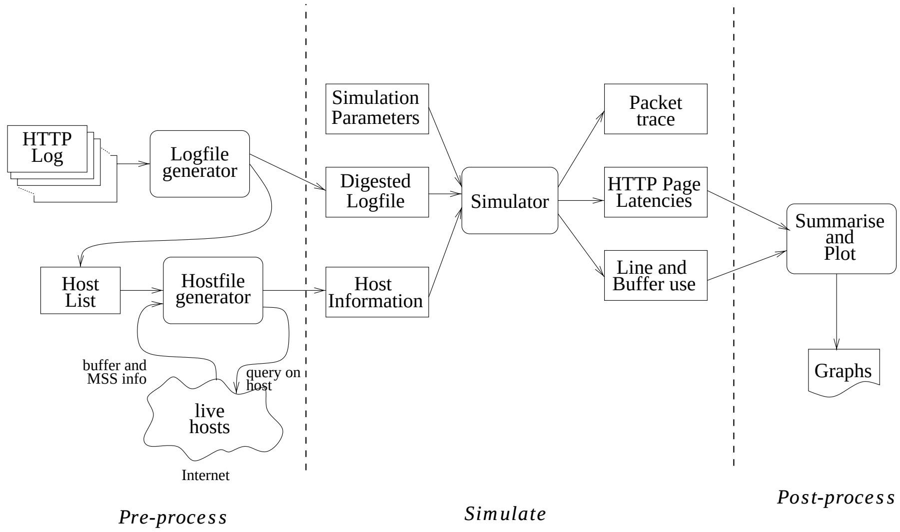

Figure 2. Simulator Design

4. SIMULATOR DESIGN

The simulation process can be considered as three interlinked processes, pre-processing, simulation and post-processing.

Pre-Processing

In the preprocessing stage the input files for the simulator are prepared. These are:

- A "hostfile" which contains an entry for each server accessed during the simulation. The entries contain: a unique ID for each host, the DNS name of the host, the maximum window size for the host and the TCP maximum segment size (MSS) for connections to the host.
- A "logfile" which contains an entry for each HTTP request. The entry contains the host ID for the server the request is fetched from, the size of the HTTP GET request, the size of the HTTP response and the time taken in the US component of the network.
- The simulation parameters including: the buffer sizes used by the proxies, the presence or absence of a US proxy, the number of connections between the proxies and the international link speed and delay in each direction.

775 simulations (each requiring from around a minute to 15-20 minutes on a 450Mhz Pentium system running Linux) were run to produce the graphs reported in this paper.[‡].

Post-Processing

Post processing is mostly a matter of collecting the results of interest from many simulation runs into a single set of plots. This was done with a set of perl scripts. GNU plot was used to draw the plots.

Simulation

The simulator used in this study was based on the ATM-TN simulator[6] with modifications for this problem. The changes include replacing the ATM infrastructure with a simpler and more general bit serial interface.

The main two components used from ATM-TN are the conservative (as opposed to parallel) simulation engine and the TCP model. ATM-TN's TCP model includes the actual TCP code from 4.4 BSD Lite, modified to suit the

[‡]For the curious that's around 3.5 million million CPU cycles

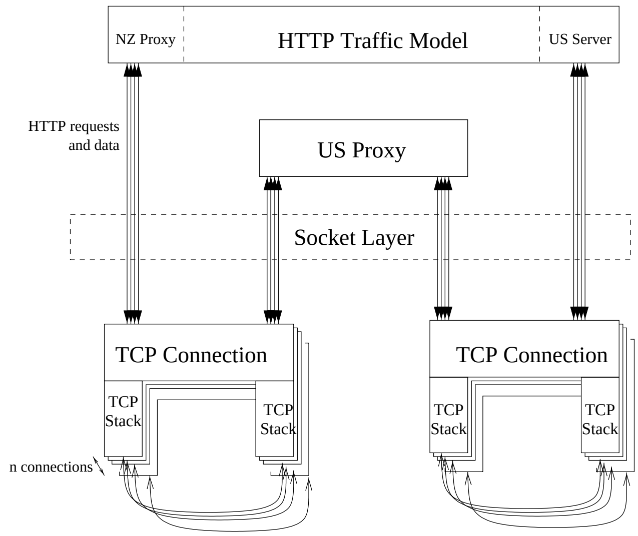

Figure 3. Simulator Design

simulation environment. Connections are simulated on a packet by packet basis and include slow start, congestion control, fast retransmit, and fast recovery algorithms.[1]

The simulator design for the US-proxy case is shown in figure 3. The simulator simulates the connections between the NZ proxy and the servers in the US. It does not include the NZ proxy to web client component of the network because this is not significant to the study. Additional delays that are dependent on the type of connection (e.g. modem, ADSL or direct connect) will be incurred in the NZ component of the real network.

The non-US-proxy case is similar to the US-proxy one with the omission of the US proxy and the replacement of the two TCP connection modules with a single TCP connection module and a single set of end-to-end TCP connections.

HTTP traffic model

The HTTP traffic model is responsible for creating TCP connections, sending HTTP GET requests, receiving the request at the destination and returning and results and for recording the time required to complete the HTTP requests. The HTTP model makes use of the hostfile and the logfile to control the simulation. The delays in the US part of the network are simulated by the HTTP traffic model which releases the packets that make up the HTTP response at a regulated rate so that the complete response arrives at the US proxy at the same mean rate as it did when the page was fetched on the real network.

TCP Connection

The TCP connection model simulates an end-to-end TCP connection and is based on a pair of TCP stacks, one for each end of the connection. It is assumed that the effect of line errors is negligible.

US Proxy Model

The US proxy accepts HTTP GET requests across the TCP connections from the NZ proxy and forwards the request to the US server over a new connection to that server (HTTP 1.1 is not simulated). When the first packet of the reply arrives from the server a TCP connection is chosen to carry the HTTP reply to the NZ proxy. The connection with the smallest number of bytes awaiting transmission or transmitted but not acknowledged is chosen. As the packets of the reply arrive they are queued for transmission over the chosen TCP connection. Because multiplexing is used the data from different HTTP replies can be intermixed on a single TCP connection between the proxies.

The simulation assumes that the proxy has sufficient CPU and memory to manage the workload and that the delays at the proxy, other than TCP queuing and transmission delays, are negligible.

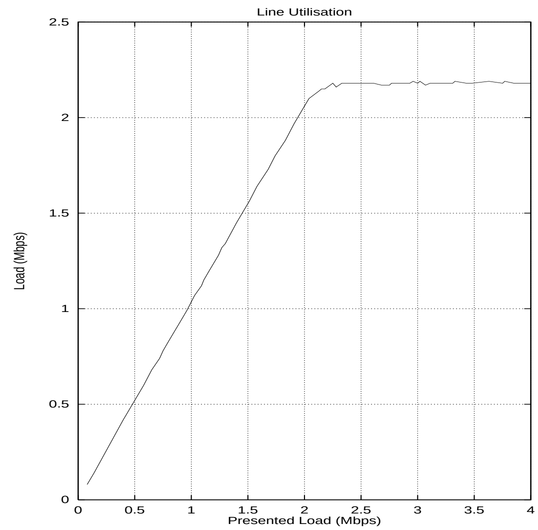

Figure 4. Link usage against load for a single TCP connection

5. RESULTS

The results of the simulations are presented in the following graphs. Each point on a graph represents a simulation run. The points have been joined with (straight) lines.

As discussed in section 2 the bandwidth delay product of the network combined with the 32Kb maximum window size limits the throughput that can be obtained by a single TCP connection. A simulation of a single TCP connection over the international link is shown in figure 4. Visual inspection shows that the curve is asymptotic to about 2.2Mbps. This corresponds closely to the calculated maximum which is also approximately 2.2Mbps.

Link Utilisation

In figure 5(a) the load to NZ is plotted against the total NZ bound traffic presented to the US proxy or, in the no-US-proxy case the US router. The presented load already includes the TCP headers required to deliver it to the US proxy or router from the HTTP server.

Although each TCP connection is limited to 2.2Mbps there are many concurrent TCP connections in the no-US-proxy case (see figure 7(a)). This allows the link to saturate under high loads. The slope of the line through most of the graph is about 1.03 indicating that there are very few retransmissions occurring. The graph does not tail off until the link is within 0.5% of being saturated.[§] Figure 5(b) shows that page latencies increase dramaticly at this time.

If there are a large number of concurrent connections between the proxies the US-proxy line ustilisation behaves in a very similar way to the no-US-proxy case. The lines for 35 and above connections have a slightly smaller slope than the no-US-proxy case indicating a small efficiency gain through repackaging the load on the more heavily used TCP connections. The efficiency gain is small and is probably not a significant saving.

For smaller numbers of connections (15 and below) the link does not reach saturation. Instead the TCP connections reach their saturation point and they limit the flow of packets to the international link.

Latency

Perhaps the most interesting result of the study is shown in figure 5(b). The graph shows the average time required to fetch a set of sample pages that were present in all simulations. The result for the US-proxy case, with a large number of connections, is around 17% lower (0.5s per HTTP request) than the no-US-proxy case. This is cause my

[§] Note that this graph shows presented load against load carried, including protocol overheads in both cases *not* presented load against useful data transmitted.

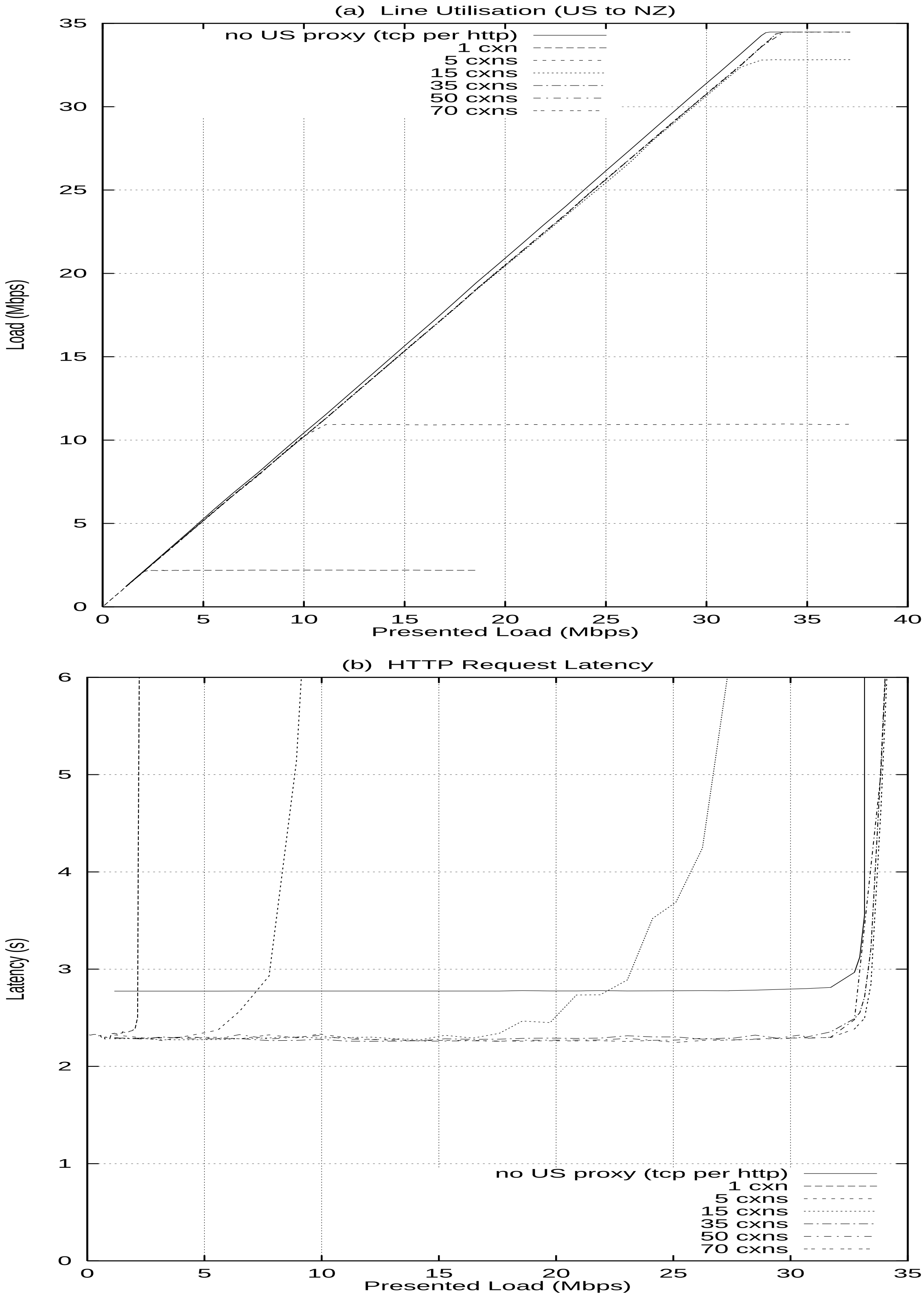
(a) Line Utilisation (US to NZ)
Load (Mbps)
Presented Load (Mbps)
no US proxy (tcp per http)
1 cxn
5 cxns
15 cxns
35 cxns
50 cxns
70 cxns
(b) HTTP Request Latency
Latency (s)
Presented Load (Mbps)
no US proxy (tcp per http)
1 cxn
5 cxns
15 cxns
35 cxns
50 cxns
70 cxns

Figure 5.

the reuse of the international TCP connections saving most of the cost of slow start. The saving for an HTML page with multiple components may be even greater.

For smaller numbers of connections the latency rises rapidly as the TCP throughput limit is approached. Comparison of figures 5(b) and 5 indicates that this begins to occur when the TCP connections reach about 60% of their capacity. To achieve the best HTTP latency performance more connections are required than are needed to saturate the international link.

Figure 6 shows the buffer space needed in the routers which feed each end of the international link. Figures 6(a) and (c) show the mean usage while figures 6(b) and (d) show the peak usage. Note that the graphs have different scales. The peak usage is more erratic than the mean because of subtle interactions between connections.

In the no-US-proxy case the buffer space required to avoid packet loss becomes very large as the link to NZ saturates. This is also true of the US-proxy case if the number of connections is large enough to allow the link to saturate. If there are too few TCP connections to carry the load the mean buffer usage reduces as the TCP connections throttle their use of the link.

Buffer (or link) usage is never heavy in the NZ to US direction in the simulation.¶

Figure 7(a) shows the number of connections between the US proxy and servers for the US-proxy case. In the no-US-proxy case it shows the number of connections from the NZ proxy to US servers. In the latter case this increases rapidly when the international link is saturated because the HTTP requests take a long time to complete (see figure 5(b)). In general the no-US-proxy case uses more connections than the US-proxy case because the connections take longer to complete.

The typical relationship between in-bound and out-bound traffic can be seen in figure 7(d). When there are sufficient TCP connections to carry the load this shows an in-bound to out-bound ratio of about 1:19.

The difference between the no-US-proxy case and the US-proxy case in figures 7(c) and (d) indicates the saving made by repackaging HTTP requests into a smaller number of larger TCP packets. This has a more significant effect than in the US to NZ direction because HTTP requests are smaller than HTTP replies. The effect is probably not useful in current practice because NZ to US links are not normally saturated. This is because of the requirement to purchase symmetric terrestrial connections. In the longer term the saving may be valuable if the asymmetry introduced by unidirectional satellite links causes the NZ to US links to saturate.

6. CONCLUSIONS

Multiplexing HTTP over, standard window size, TCP connections between international proxies, offers significant performance advantages.

The number of TCP connections between the proxies is important. To avoid the link being under utilised around 20 connections were required for an E3 connection from NZ to the US. However additional connections are needed if the best page latency is required. In this case around 50 connections were required.

REFERENCES

1. W. Stevens, "TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms," Tech. Rep. RFC2001, IETF, Jan. 1997.
2. W. S. M. Allman, V. Paxson, "Internet draft: TCP congestion avoidance," Tech. Rep. draft-ietf-tcpimpl-cong-control-02, IETF, Dec. 1998.
3. W. L. Morgan and G. D. Gordo, *Communications satellite handbook*, Wiley, New York, 1989.
4. J. Postel, "Transmission control protocol," Tech. Rep. RFC793, DARPA, Sept. 1981.
5. V. Jacobson, R. Braden, and D. Borman, "TCP extensions for high performance," Tech. Rep. RFC1323, IETF, May 1992.
6. M. Arlitt, Y. Chen, R. Gurski, and C. Williamson, "Traffic modeling in the ATM-TN TeleSim project: Design, implementation, and performance evaluation," in *Proceedings of the 1995 Summer Computer Simulation Conference*, (Ottawa, Ontario), July 1995.

¶A real US/NZ link would be more heavily used in the US direction because of requests on NZ servers from US clients. These are not simulated here. We assume that sufficient NZ/US capacity exists to carry the client requests to the US.

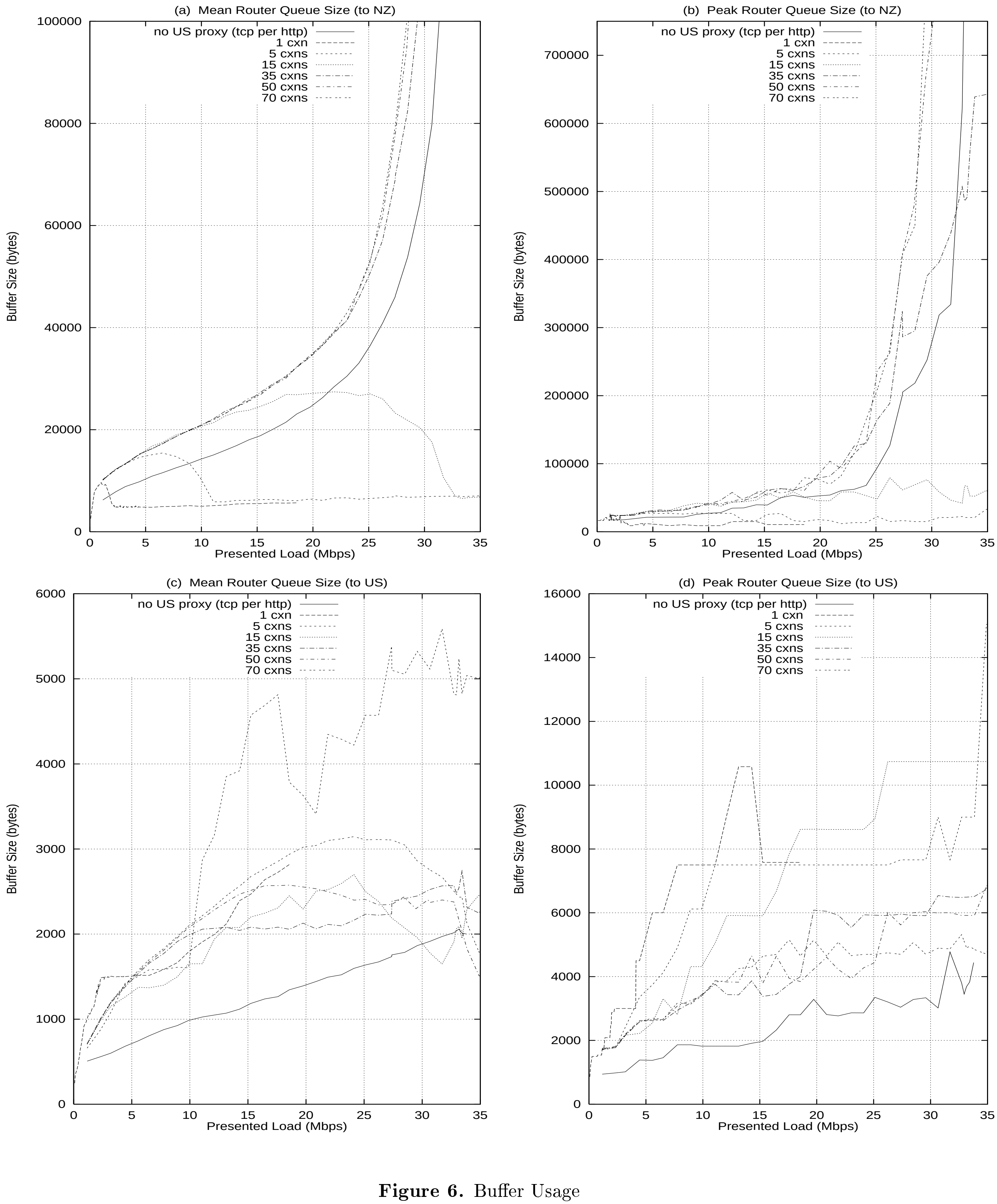

Figure 6. Buffer Usage

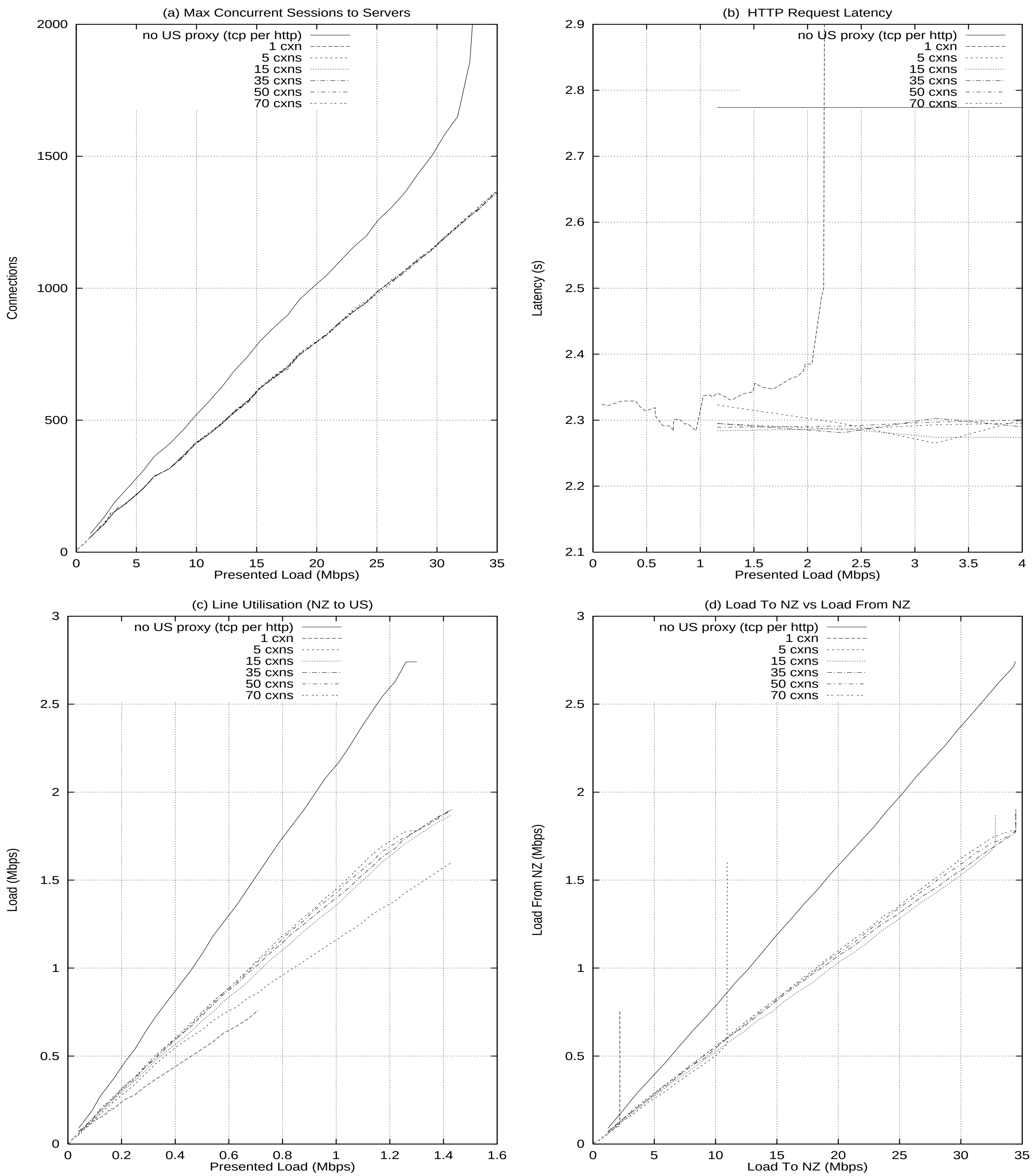
(a) Max Concurrent Sessions to Servers
no US proxy (tcp per http)
1 cxn
5 cxns
15 cxns
35 cxns
50 cxns
70 cxns
Connections
Presented Load (Mbps)
(b) HTTP Request Latency
Latency (s)
Presented Load (Mbps)
(c) Line Utilisation (NZ to US)
Load (Mbps)
Presented Load (Mbps)
(d) Load To NZ vs Load From NZ
Load From NZ (Mbps)
Load To NZ (Mbps)

Figure 7.