Endpoint study of Internet paths and web pages transfers

B. V. Ghita, S. M. Furnell, B. M. Lines, E. C. Ifeachor

University of Plymouth, Plymouth, United Kingdom
e-mail: bghita@jack.see.plymouth.ac.uk

Abstract

This paper presents the findings of a pilot study to provide information about the characteristics of current networks and data transfers. The main aim of the study was to infer the properties of a large number of network paths. In addition, the study produced statistics relating to the average size of a typical web page and both under the restriction of a single-point connection. The study was performed in two steps: trace collection followed by TCP per-flow analysis. The trace collection used the functionality of a random link generator, combined with an automatic HTTP retrieval tool. The TCP analysis was applied to the collected traces and it involved an offline TCP per-flow method developed in previous research.

Keywords

TCP connection analysis, Internet characteristics, web page size, web transfer features.

1 Introduction

The current status of the Internet is one of the issues being researched intensively. The first concerted initiative to evaluate the properties of the Internet belongs to Paxson. He deployed his Network Probe Daemon (NPD), established a measurement mesh to evaluate the characteristics of network paths, and generated and analysed the transfers running through this mesh (Paxson, 1999). Several bodies, such as the Active Measurement Project (AMP, 2002) and the National Internet Measurement Infrastructure, built on the concept of NPD) (NIMI, 2002) projects aim to describe the Internet from a holistic perspective by employing complex measurement infrastructures. A different view is embraced by passive traffic surveys, which capture and analyse data from backbone segments / endpoint networks (Thompson and Miller, 1997). The need for such information comes from both the research and commercial domains. The rationale is similar for the two cases: the marketing directions, as well as the improvements of current Internet-related technologies, have to be based on actual information rather than assumptions or previous studies.

All of the aforementioned measurement initiatives are very successful in their place, and they aim to answer the question ‘How does the overall Internet behave?’. The study presented in this paper seeks to discuss the Internet characteristics from a different perspective: how the Internet is seen from an endpoint network and what are the characteristics of the data that may be retrieved from the Internet by other hosts connected to that respective endpoint network. Concluding, the question that this study aims to answer is ‘How does Internet behave for *my* Internet traffic?’, as would be asked by an endpoint network user / administrator.

2 Traffic collection

The study discussed by this paper presents analysis results based on two data sources: real traffic and artificially generated traffic. In both cases HTTP was used as the focused application for reasons of availability and convenience. It was observed before starting the experiments that most of the network TCP traffic is web browsing, which confirms the results of previous studies (Paxson, 1999). Also, as will be discussed later, artificial and random HTTP traffic was convenient to produce.

For the first option, i.e. capture real traffic, the hosts (approximately 15) within the Network Research Group (NRG) at University of Plymouth were used. The connectivity of the machines within the NRG is convenient for traffic capture, as they are all connected to a switch, and the capture machine was attached to the uplink of the switch through a hub. The traffic collection was performed continuously during spring 2002 for a period of two weeks and included only web traffic between the hosts in the NRG and hosts outside the UoP network. The second option, i.e. generate artificial traffic, allowed a more controlled approach to the data collection. The traffic was produced using the Random Yahoo Link page (RYL, 2002) from the Yahoo website, a CGI script that redirects a request to a random WWW page, taken from the Yahoo search engine database. The HTTP client used to perform the requests was wget (wget, 2002), a command-line HTTP retrieval tool, and the requests were controlled through a Linux shell script. The experiments in this case were also performed in two stages, but separately from the network segment traffic capture discussed above. It is known that at least one major event, in terms of network infrastructure changes, happened between the two experiments: an upgrade of the UoP network from a 100MB backbone / 10MB access speed to 1000MB backbone / 100MB access speed. As will be seen in the results section, all the network parameters (bandwidth, loss, delay) are improved for the second set of results. For both experiments, the traffic was captured using tcpdump (tcpdump, 2002), which was set to keep only the HTTP connections (using a *tcp and port 80* filter expression). The level of the monitored traffic was low in all cases and tcpdump did not report any dropped packets throughout the experiments. In all experiments, the traces were filtered offline in order to remove the unfinished or reseted connections, which could not be used for consistent analysis; the resulting figures are presented in Table 1.

Traffic type	Number of connections collected			
	2001		2002	
	Raw	Filtered	Raw	Filtered
Wget generated	15106	12469	16844	13674
Real	-	-	14288	11322

Table 1 – Capture statistics for the traffic collection experiments performed

3 Analysis

One of the aims of this paper is to advance the traffic analysis from an overall study, currently preferred for convenience and simplicity, to per-flow examination, in order to get an insight of the network conditions that are behind the traffic. The overall analysis studies present only the total figures of traffic (overall throughput in bytes, packets, or flows per second), the distribution of traffic per application (based on the port numbers), or the distribution of the packet lengths. The only concerted efforts in the area of TCP per-flow analysis were the ones

made by Paxson in (Paxson, 1997a), (Paxson, 1999), which are quoted by most articles when discussing current characteristics of the Internet.

Two types of analysis were applied to the collected traces: network performance-related, to reveal the end-to-end network paths characteristics, and connection-related, to classify the web pages in terms of size and content. The network performance analysis was performed using a previously developed tool, described in (Ghita et al, 2001); the method employed is similar to other TCP flow analysers, like tcpanaly (Paxson, 1997b) and tcptrace (Osterman, 2002), with improvements for single point monitoring and network parameters inference. The connection analysis investigated the size and the content (object-wise) of the web pages for two reasons: to determine the average size of a page (together with the containing objects, e.g. images) and to establish the efficiency in practice of the HTTP 1.1 pipelining capabilities.

4 The Random Yahoo Link experiments - Results

4.1 Network topology

The UoP network is connected to the Internet, as mentioned before, via the UK academic network, JANET. As a result, the first 8 hops of all paths are part of the JANET infrastructure, and, implicitly, were common for all connections but the routes diverged at the exit from JANET, depending on the destination host. A separate experiment was carried out to estimate the number of individual paths explored within the performed experiments. A traceroute was run on a random subset of the sites (350 out of the 2744 unique servers which were used during the spring 2002 round of experiments) to see the number of different individual paths. The results are shown in Figure 1.

The number of routes differing by at least one hop was found to be as high as 180, figure that is approximately half of the number of hosts probed (the number of unique hops decreases towards the end of the graph due to path size, with an average hop count of 22.2 hops). Concluding, although the study was performed from a single point, this additional measurement indicates that the survey analysed a fairly large number of different Internet paths.

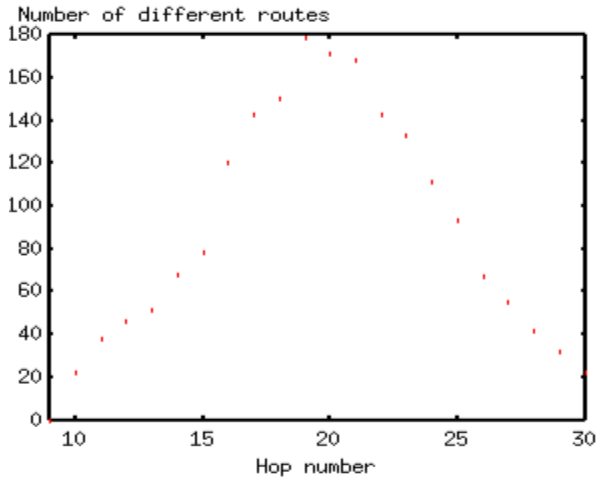

Figure 1 - Routing distribution, spring 2002 experiments

4.2 Round Trip Time results

The distribution of the RTT for the two sets of experiments is presented in Figure 2 (left). As can be observed, in both cases the average RTT values are very low for most of the connections, with an overall average of 200.5 ms for the first round of experiments and 136.5 ms for the second round. The difference between the figures may be associated with the network upgrade mentioned previously (unfortunately, there was no path information collected during the autumn 2001 experiments), as the shape of the distribution remained the same for the two sets of results.

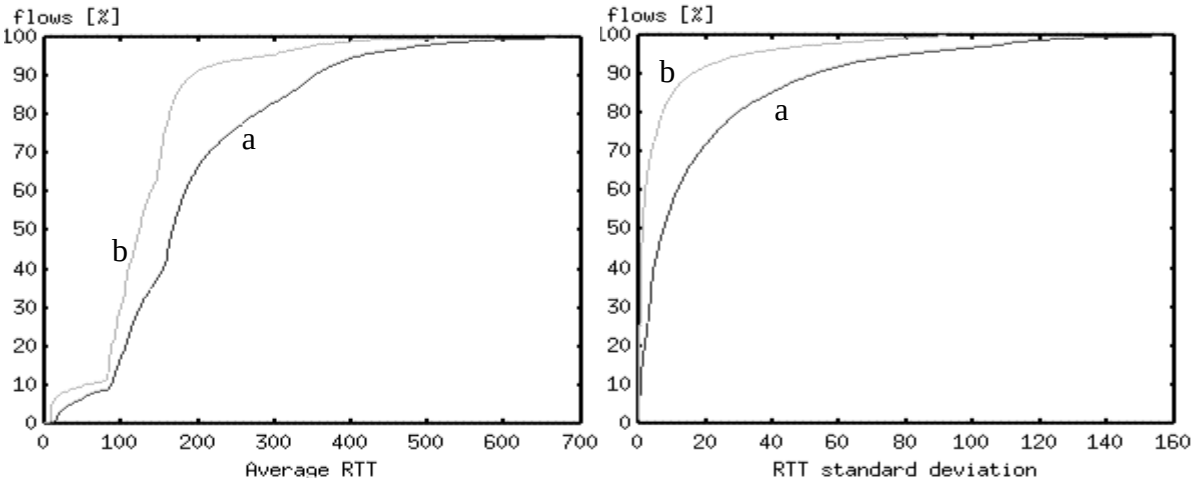

Figure 2 – left - RTT average [ms] cumulative distribution for: a) autumn 2001, b) spring 2002; right - RTT standard deviation [ms] cumulative distribution for: a) autumn 2001, b) spring 2002

Aside from the actual value of the RTT, the standard deviation of the RTT throughout a connection was calculated; the result is shown in Figure 2 (right). The average value of the standard deviation was 22.3 ms (10.4 % of the RTT averages) for the autumn 2001 round and 7.8 ms (4.7 % of the RTT averages) for spring 2002. The data results from spring 2002 indicate that, for 87% of the flows, the RTT standard deviation was 10 ms. This value is relevant as, at the moment, it is the default resolution for timers, at least for Linux based systems. Future work, aims to analyse the implications of these low figures for RTT estimation within the TCP clients, since the RTT variation plays an important role in the TCP retransmission mechanism (Jacobson and Karels, 1988). Since successive connections were made to different sites, conclusions could not be drawn with regard to the long / short term autocorrelation of either RTT average or RTT standard deviation.

4.3 Loss

Due to its self-adjusting behaviour (Jacobson and Karels, 1988), TCP performance is critically affected by loss. Nevertheless, previous studies (Paxson, 1997a) have shown that packet loss is low, at least for the analysed mesh of Internet paths. One of the purposes of this paper is to produce a similar study, but based only on traces collected from a single point and with no control over the senders. It may be argued that the survey carried out as part of this study was somehow limited, as the wget client does not support HTTP1.1. As a result, the objects from a page are downloaded in separate connections, which leads to smaller

congestion windows. Further, the resulting figures for loss may be lower than the ones obtained for a long-lived connection, with larger congestion windows. The losses were split into visible retransmissions and inferred retransmissions. The first category, visible retransmissions, is represented by losses which are indicated by anomalies in the TCP segments sequence. The second category, inferred retransmissions, includes the losses that are not apparent from the sequence of succesive TCP segments (more details on this subject are given in (Ghita et al, 2001)). The second category was named *inferred* because the process of identifying a loss is not based on sequencing, but only on packet spacing. The technique is reliable for a simple HTTP 1.0 retrieval, where the reply is a single object. Additional problems arise if HTTP 1.1 is used, due to spacing introduced between retrievals of successive objects within the same connection. In this case, the method requires comprehensive information from the application layer; since this is currently under analysis, it is reserved for future work.

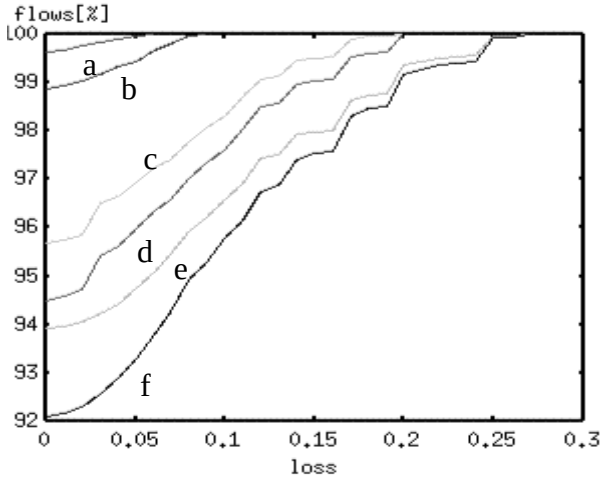

Figure 3 - Packet loss distribution for a) / b) visible retransmissions 2002 / 2001, c) / d) inferred retransmissions 2002 / 2001, e) / f) all losses 2002 / 2001

The distributions for both types of losses, as well as their sum, are displayed in Figure 3. The average figures for loss were: 0.18 / 0.83 / 1.1 % (visible / inferred / total) for the 2001 experiments and 0.16 / 0.47 / 0.63 % (visible / inferred / total) for the 2002 trace. It is noticeable that the inferred losses accounted for the vast majority of the total loss, which may be caused by the short-lived character of most connections. This may be due to the small number of packets / connection which, again, leads to low values for congestion window and, implicitly, too few acknowledgements returned for triggering a retransmission when a loss happens.

The short-lived connections have an additional undesired effect: the accuracy of the measurement cannot go beyond the granularity of the download due to the low number of packets exchanged. For example, having a transfer consisting of 10 packets, the minimum detectable loss is 0.1, a situation also described in (Paxson, 1997a). To reduce this error granularity, we calculated the loss based on the total number of packets. The year 2001 tests had a total of 137297 packets, with 295 visible and 1033 inferred retransmissions, producing the overall packet loss figures 0.21 / 0.75 / 0.96 % (visible / inferred / total). For the 2002 tests, a number of 297 packets were visible retransmissions and 604 packets were inferred retransmissions; comparing this with the total of 129404 packets, results in an overall packet loss of 0.22 / 0.46 / 0.68 % (visible / inferred / total).

4.4 Bandwidth

An estimate of the total bandwidth was produced for each connection. The estimate used delay between pairs of consecutive packets inferred to be sent in a back-to-back manner, and it was based on the method proposed by Keshav in (Keshav, 1991). The problems that may occur due to clock granularity were avoided by using a microsecond kernel timer, the Kansas University Real-Time Linux (KURT) (Niehaus, 2002). The obtained figure might be affected by the problems associated with packet-pair bandwidth inference but, due to the unknown behaviour of the senders, it was not possible to apply the Receiver Based Packet Pair as described in (Paxson, 1997a) to avoid these problems.

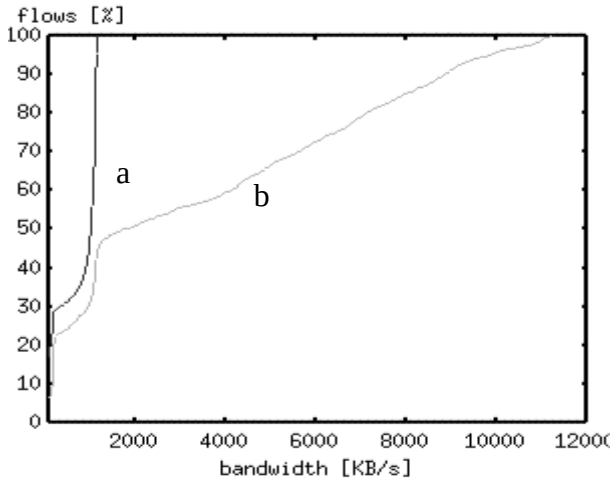

Figure 4 - Bandwidth cumulative distribution for a) autumn 2001 and b) spring 2002

From all network characteristics, the network upgrade mentioned earlier affected bandwidth the most. It can be noticed in the distribution from Figure 4 that bandwidth reached a maximum of approximately 1.2MB/s for the autumn 2001 round of experiments. This matches, in fact, the configuration of the network: at the time of the experiment, the connectivity of the desktops was 10Mb LAN. For the spring 2002, the maximum figure is 12MB/s, which reflects the tenfold increase in desktop bandwidth.

4.5 Congestion window analysis

The congestion window inference includes a high level of assumption in terms of TCP connection analysis. In our case, the task had an increased level of difficulty due to the characteristics of the monitored transfers: unknown senders, receiver-based capture, and no control over the endpoints / transfer. The fact that the senders use an unknown TCP implementation does not allow any inference in regards to profiling of the congestion window evolution. The intention was to produce a rough estimate of the congestion window, not to compete with *tcpanaly* (Paxson, 1997b), which includes more complex analysis but also requires traffic capture at / near the endpoints. The receiver-based capture brings with it uncertainty in regards to if, when, and as a response to what acknowledgement, the sender transmitted a data segment. Due to the variety of window increase policies and the uncertainty of which acknowledgements reached the server, the congestion window inference was based exclusively on timing between different trains of packets rather than acknowledgement dialogue. The actual method focused on isolating groups of packets that appear to be

transmitted as part of the same round, based on the distance between successive in-sequence packets. The third problem, no control over the endpoints, differentiates the study from Internet measurement efforts (Paxson, 1999), (NIMI, 2002). Within measurement infrastructures, endpoints running dedicated clients transfer large files between them at regular intervals in order to determine the network characteristics. Within this study, all the senders were remote sites on the Internet and the objects transferred were various web pages residing on the servers; as a result, there was no control over the size / timing of the connections.

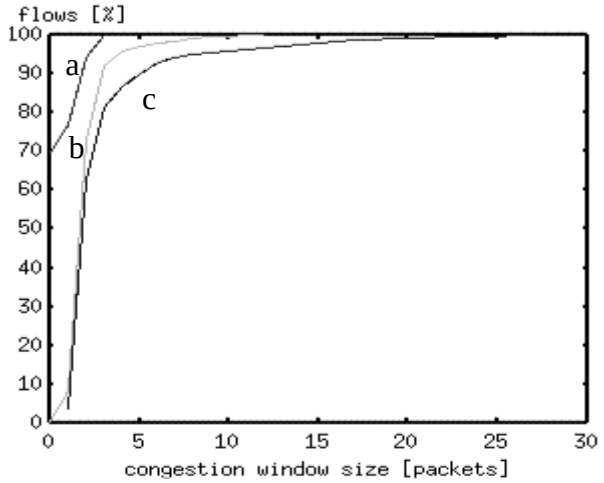

Figure 5 - Cumulative distribution of the a) initial, b) average, and c) maximum congestion window size

The resulting distribution is displayed in Figure 5. The average figures for the three variables (initial / average / maximum congestion window) were 1.91 / 3.47 / 4.95 for 2001 experiments and 1.77 / 3.16 / 4.52 for the 2002 experiments round. The difference between the figures can be attributed, again, to the network upgrade that reduced the packet loss and delay figures, as mentioned before.

5 The NRG network traces

5.6 Page content analysis

When the first round of experiments was run, the latest version at the time did not allow for a full download of the web pages (e.g. for a page with 4 images, only the HTML file was retrieved). At the second round of experiments, the newer version of wget had the facility to parse web pages and download the objects hosted on the same server with the page), which allowed a rough estimation of the actual content of the page. In the case of a HTTP1.1 client, these objects would be downloaded in a single connection. This gives an approximate indication of the actual length, in terms of size, of a connection.

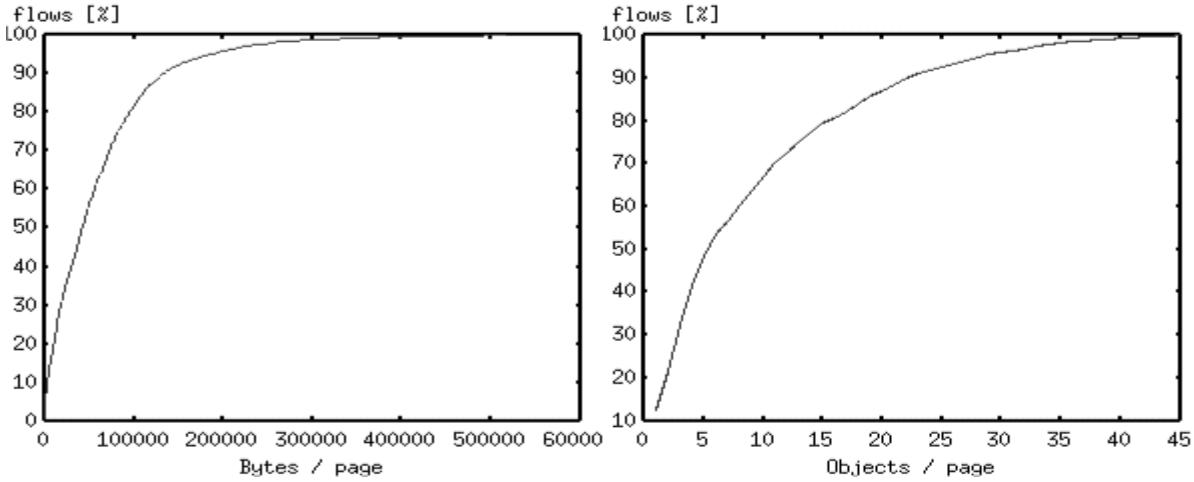

Figure 6 - Distribution of page content in bytes / page and objects / page

Figure 6 shows that most web pages have relatively large size (for some of his experiments, Paxson considered 100 KB files to be satisfactory for evaluating the properties of Internet). Also, from the distribution of objects per page, it may be concluded that full usage of HTTP 1.1 request pipelining would considerably reduce the overall time to retrieve the web page. The average figures for Figure 6 are 72607 bytes / page and 10.5 objects / page.

5.7 Connection analysis

Although a convenient and comprehensive tool, even the latest version of wget does not include some major functionality such as supporting frames and request pipelining (according to the author, there are no plans to expand it in the future in these areas). The set of traces captured from the traffic produced by the NRG members was therefore used for the connection analysis. The machines in the NRG were running either on a flavour of Linux (RedHat or SuSE) or Windows (NT4 or 2000 Professional), with Netscape Navigator 4.76-4.77 or Internet Explorer 5.0-5.5 as correspondent web clients. All the mentioned versions have HTTP1.1 enabled as standard, therefore they all should pipeline the requests whenever possible. The analysis of the captured traffic focused on the connection length, in order to determine the average length of an HTTP retrieval for the real traffic case. It may be argued that the amount of users involved in the study was relatively small; however, for future, it is aimed to compare these figures with results obtained from bigger, backbone collected traces. The result of the connection size analysis is displayed in Figure 7. It was observed that approximately three quarters of the flows had a download size of under 5KB with average numbers of 6220 bytes / connection and 7.12 packets / connection. These are very low figures, considering the previously estimated average of 72607 bytes / page obtained from the Random Yahoo Link experiments, and indicate that, in spite of the rich content of the Internet, the HTTP pipelining capabilities are not efficiently used.

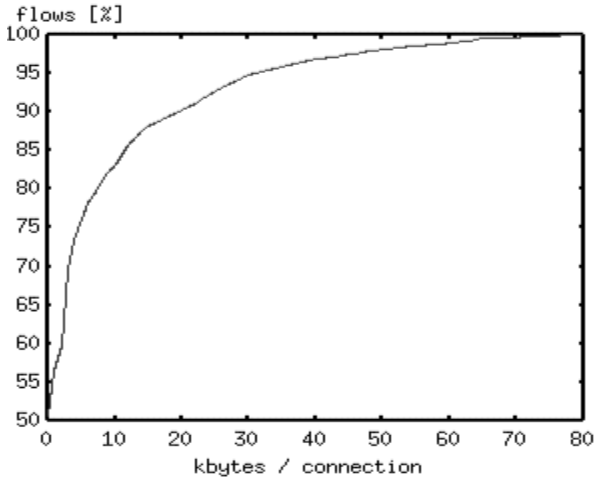

Figure 7 - Cumulative distribution of connection size

6 Conclusions

This paper presented the findings from an endpoint-based network per-flow trace analysis. In spite of its limited scope, the proposed analysis allowed characterisation of a fairly large number of network paths. The traffic was studied with a two-fold purpose: to evaluate the network conditions experienced by the flows and to determine the characteristics of a web page in terms of total content and number of elements per page. For the analysis, artificial traffic was mainly used. The experiment was carried out in two rounds: autumn 2001 and spring 2002. It used a random page generator combined with a command line HTTP retrieval tool, which was preferred instead of the real traffic due to the complexity in interpreting HTTP 1.1 pipelined transfers. Nevertheless, in order to evaluate the size characteristics of real transfers, a pilot traffic capture in a limited environment was performed.

The TCP analysis revealed a loss-free image of the Internet, with an average loss of 1.1% for the first round of experiments and 0.63% for the second round. The overall figures indicated even a smaller loss probability, of 0.96% and 0.96% respectively. The round trip delay values were also fairly low, with an average 200.5 ms for autumn 2001 and 196 ms for spring 2002 experiments and a standard deviation of 22.3 ms and 7.8 ms (4.7 % of the RTT averages) respectively. The page content analysis revealed that the average page size is approximately 70 KB, with an average of 10 objects / page that fully justify usage of HTTP 1.1 pipelining. However, the real network traffic showed much lower figures, of only 6220 bytes / connection and 7.12 packets / connection, which indicates that either the HTTP 1.1 pipelining mechanisms are inefficiently used or that current web pages are not suitable for pipelined requests.

For future work, it is primarily aimed to reduce the uncertainty of packet loss estimation and to expand the analysis towards connecting the TCP analysis with the HTTP retrieval, in order to be able to isolate individual retrievals of objects. Also, if possible, the per-flow analysis will be applied to larger traces to determine whether or not these findings are scalable and may be applied to traffic collected from core internet links.

7 References

AMP, The Active Measurement Project (AMP) homepage, http://moat.nlanr.net/AMP/, 2002

Ghita, B., Lines, B, Furnell, S., Ifeachor, E., "Non-intrusive IP Network Performance Monitoring for TCP flows", *Proceedings of IEEE ICT 2001*, 2001

Jacobson, V., Karels, M., 'Congestion Avoidance and Control', *Proceedings of SIGCOMM '88*, 1988

Keshav, S., "A Control-Theoretic Approach to Flow Control," *Proceedings of SIGCOMM '91*, pp. 3-15, 1991.

NIMI, The National Internet Measurement Infrastructure homepage, http://ncne.nlanr.net/nimi/, 2002

Niehaus, D., "KURT-Linux: Kansas University Real-Time Linux", http://www.ittc.ku.edu/kurt/, 2002

Osterman S., "tcptrace homepage", http://www.tcptrace.org, 2002

Paxson, V., "Measurements and Analysis of End-to-End Internet Dynamics", PhD thesis, 1997

Paxson, V., "Automated Packet Trace Analysis of TCP Implementations", *Proceedings of SIGCOMM '97*, 1997

Paxson V., "End-to-end internet packet dynamics", *IEEE/ACM Transactions on Networking*, vol 7, no 3, 1999

Random Yahoo Link (RYL), Random Yahoo Link Page, http://random.yahoo.com/bin/ryl, 2002

tcpdump, tcpdump public repository, http://www.tcpdump.org/, 2002

Thompson K., Miller G.J. 'Wide-Area Internet Traffic Patterns and Characteristics', *IEEE network*, nov 1997

Wget, GNU Wget home page, http://www.gnu.org/software/wget/, 2002