EUROCON 2003 Ljubljana, Slovenia

Measurement and Modelling of WWW Traffic in a LAN Environment

L. Liang, Z. Sun, and M. Howarth
CCSR, University of Surrey, Guildford, Surrey, GU2 7XH UK.
Email: L.Liang@surrey.ac.uk
Z.Sun@surrey.ac.uk
M.Howarth@surrey.ac.uk

Abstract--**The Internet is growing to play a core role in people's daily lives. Network traffic control and QoS guarantees are highly demanded in multiservice networks, but need to be based on sufficient knowledge of the network's performance; this can only be achieved by measuring and analyzing network traffic. This paper describes how software based on the Winpcap library has been developed and used to measure the most popular Internet application, World Wide Web (WWW) traffic, on a typical Ethernet LAN. We demonstrate how to measure and analyze the interarrival time and size of HTTP packets and derive statistical parameters from them. Analytical traffic models are developed based on the measurement results**

Index Terms-- **Impulse function, Inverse Gaussian, modeling, packet interarrival time, packet length.**

I. Introduction

TELEPHONY traffic modeling has historically been the basis for many network models. Some techniques and methodologies, such as monitoring packets, have been used successfully in network modeling for many years. Many theories and methodologies developed for voice networks can also be appropriately applied to the modern Internet. It is important that traffic engineering of IP networks should be based on realistic traffic measurement and modeling, since measurements will be used to predict real traffic characteristics. These measurements can be implemented by monitoring the traffic on the IP networks and collecting relevant data for analysis. The general statistical characteristics of the IP traffic then can be represented using mathematical models based on the analysis of these collected data. These statistical models of the IP traffic are needed by engineers to plan, design, configure and maintain their networks. Simulations also need these statistical models to generate traffic and evaluate network performance. However, given the intricate fast-growing nature of networks such as the Internet, the wide range of applications and network topologies makes traffic modeling difficult.

The traditional way to monitor IP traffic is to capture the interested packets and analyze their headers. The parameters in these headers and the arrival times of the packets can give enough information to statistically model the traffic if sufficient packets are captured over long periods.

Many people have tried to study the IP traffic by monitoring the packets on wires. Shoch [9]and Boggs [7] did early Ethernet measurement studies. Their work was performed at low resolution and high loss rates. In 1989, Leland [5] began taking high-resolution traffic traces at Bellcore Morristown Research and Engineering Center. The measurement lasted over three year and over 100 million packets were collected in traces. There have been a number of papers describing the self-similarity character of Ethernet traffic at a wide range of time scales ([5] [6]). This self-similarity is different from the generally accepted "Poisson-like" nature traditionally used in voice network traffic engineering.

This paper attempts to define the characteristics of packet length and interarrival time for World Wide Web (WWW) traffic based on measurements of an end user.

In section 2 this paper describes the design of a network traffic monitoring application based on the WinPcap library [2]. HTTP traces have been measured using this software and are presented in section 3. Section 4 analyses the captured traces in terms of packet length and interarrival time, and presents analytical models of these parameters. Finally, conclusions are drawn in section 5.

II. Software Design

A Windows NT workstation was used for all our measurements described in this paper. A Win32 console application was developed based on the WinPcap library [2] to filter and capture specified packets from the IP network. WinPcap is a tool for packet capturing and network analysis developed by the NetGroup of Politecnico di Torino. It can copy every packet received on the monitored network adapter to a buffer where a WinPcap application can handle it. A filter can be set to enable the adapter to copy only packets of interested while ignoring others.

The WinPcap capture driver does not capture the Cyclic Redundancy Check (CRC) (4 bytes) of Ethernet packets. Furthermore, it only captures the received raw packets with padding but does not capture padding for transmitted packets [3]. Consequently, the packet lengths in our traces were pre-processed, by increasing all packet length less than 60 bytes to 60 bytes and then increasing all packet lengths by 4 bytes. The result of this is that all packets were treated as fitting the standard IEEE 802.3 frame definition, i.e. a minimum packet length of 64 bytes and a maximum length of 1518 bytes. The packet length here means the length of the Ethernet frame excluding the MAC preamble bits.

A packet filter condition is inputted as an argument of the console software. Every packet that fits the filter condition is copied, and the first 64 bytes are recorded. The first 14 of these 64 bytes are the Ethernet MAC header. The following

20 bytes are the IP header, including source and destination IP address, protocol type and total length (this parameter is used later to derive the packet length distribution models). The following 20 bytes are the TCP header, and the final 10 bytes captured are data information that can provide detail about the packet. This is used to identify uncommon phenomena during the analysis of packets. The software architecture is shown in Figure 1.

The capture driver adds an arrival timestamp to every incoming packet and the software reads these timestamps to calculate the interarrival time, given by:

$$ArrivalInterval = ArrivalTime(i) - ArrivalTime(i-1) \quad (1)$$

Where ArrivalTime(i) and ArrivalTime(i-1) are the arrival times of packets i and i-1 respectively.

The software extracts and displays some information on the screen for each qualified packet captured on the network adapter. This information includes source and destination IP address, arrival time, packet length, protocol type, flow direction and the first 64 bytes of each packet. The capture beginning time and stop time are displayed respectively. All of these were redirected to a text file for analysis later. The software stops capturing upon its response to any keyboard button pressed by the user.

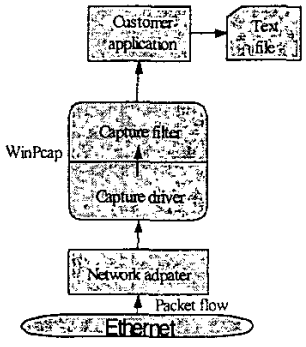

Figure 1 Packet monitor software Architecture

It is important to note that the test workstation was directly attached to a switch, so only the packets sent from and to the user could be monitored. The traffic generated by other terminals in the LAN is invisible to WinPcap applications.

III. WWW Traffic Measurement

This paper focuses on processing IP packets reaching the network adapter. Several WWW packet traces were captured at different times of day to ensure representative results. Each trace consists of five data groups: uplink packet length, downlink packet length, uplink packet interarrival time, downlink packet interarrival time and the summary information described in section 2.

The application used on the workstation to generate WWW traffic for the packets capture software was Microsoft Internet Explorer® 5.0. The WinPcap software was set to copy all packets with a TCP source or destination port number of 80. All other traffic, e.g. LAN management traffic and DNS traffic, was ignored. For each trace, the traffic is divided into uplink and downlink packets. Analysis is done based on each class of packets because the performance of these two classes was found to be different.

The Internet surfing includes email checking, news reading, online form filling and small java applets downloading. Various Internet web sites were visited in each measurement.

IV. Traces Analysis and Modeling

The trace data shows that opening a web page results in a set of packets. These packets represented several objects contained in the web page. Reading is the main operation in the long term so that the total packet number captured is not very big in the measurement.

In the following paragraphs, the WWW traffic is studied in terms of packet length and arrival interval.

A. HTTP Packet Length

The captured traces are in agreements with the common observation that the uplink generates less traffic than the downlink. For the reason of space limitation, the Probability Density Function (PDF) plot of only one trace is given in Figure 2 and is differentiated in each direction.

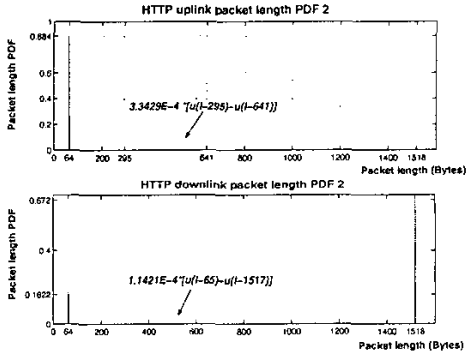

Figure 2 Packet length PDF of HTTP 2

1). Uplink packet length

Approximately 80-90% of the uplink packets have the same size of 64 bytes, which is the minimum Ethernet packet size. This is because most uploaded packets are simple page requests or acknowledgements of received packets.

An interesting point is that 99% of the uplink packets are either 64 bytes long or in the range from 295 bytes to 641 bytes. The traces show that most of those packets with a length between 295 bytes and 641 bytes were generated by some interaction operation between users and servers, such as the user logging in to the server, the servers retrieving information from cookies and the user sending out messages, including emails. To simplify the model, we ignored the probability $P_{ignore} = p(65 \le l \le 294,\ 642 \le l \le 1518)$, which in each

trace is around 1%. To make the overall probability of the new model equal to 1, we scale up $p_1 = p(l = 64)$ and $p_2 = p(295 \le l \le 641)$ so that $p_1 + p_2 = 1$.

The packet length distribution in the range of 295 bytes to 641 bytes is white noise like (**Figure 3**). Also to simplify the model, we use an average over this range, i.e. $p_2'/(641 - 295 + 1) = p_2'/347$. The corresponding plot can be found in **Figure 4**.

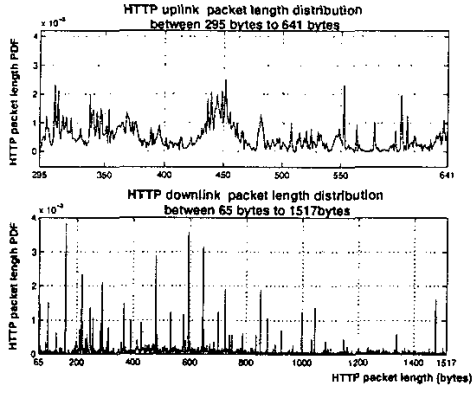

Figure 3 Noise like distribution of HTTP 1 packet length in specified ranges

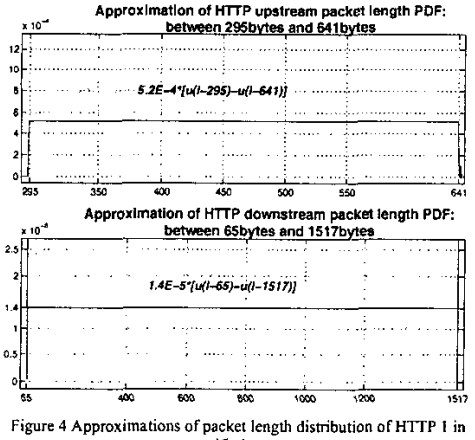

Figure 4 Approximations of packet length distribution of HTTP 1 in specified ranges

Hence, we can develop a model for the uplink WWW packet length distribution:

$$p_{uplink\_len}(l) = p_1'\delta(l - 64) + (p_2'/347)[u(l - 295) - u(l - 641)] \quad (2)$$
$$(64 \le l \le 1518)$$

where p_1' is the value of the probability of packet length equal to 64 bytes and p_2' is the probability of packet length falling the range of 295 bytes to 641 bytes. The δ function in (2) is the standard impulse function and the u function is the standard unit step function.

In our measurement, $p_1' = 0.88 \pm 0.06$ and $p_2' = 0.12 \pm 0.06$. The variations of the parameters were caused by different packet traces.

2). Downlink packet length

The main difference between uplink and downlink packet length distributions is that the biggest packet length spike in the downlink traffic happens at 1518 bytes while it is 64 bytes on the uplink. Clearly, the downlink has bigger data blocks than the uplink traffic.

During the study of these traces, we found there was one spike in the downlink direction at the point of packet length equal to 594 bytes in all traces. The spike were studied rather than simply ignored considering their probability values, which is 0.036 in HTTP1, 0.0732 in HTTP2, and 0.0206 in HTTP3. After having a closer look at the associated layout data, we found almost all of packets with a length equal to 594 bytes are caused by a few particular web sites. The packet data shows that these web sites divided their pages into blocks with the same size of 594 bytes. This is due to the Maximum Segment Size (MSS) announcement that is used to initiate a TCP connection in its SYNchronization (SYN) segment [4]. When a TCP connection is established, each end sends out the MSS it wants to receive in the TCP header Options field. Considering the IP and TCP overhead, the bigger the data segment, the more efficient use of the bandwidth. The Ethernet can allow a maximum data segment size 1460 bytes. With the 14 bytes MAC header, 20 bytes IP 7header, 20 bytes header and 4 bytes CRC, we then have the maximum Ethernet frame length of 1518 bytes. Thus, we can understand why there is a spike at 1518 bytes in our measured packet length distribution. However, if one end of a TCP connection can't handle 1460 bytes segment size, the default segment size is 536 bytes, which means 594 bytes long Ethernet frames. This situation happens when one end of a TCP connection doesn't receive MSS announcement from the other end or the destination address is "nonlocal". The measurement results show that most web sites are configured to support a maximum data segment size of 1460 bytes. Thus the spike at 1518 bytes is far more significant than the one at 594 bytes. To generate a simple model for WWW traffic, we decide to firstly eliminate the effect of the spike caused by the 536 bytes MSS before proceeding with further analysis.

For the downlink traffic, we found the packet length almost evenly distributed in the range between 65 bytes and 1517 bytes, unlike our limited range (295 to 641 bytes) for the uplink traffic. We therefore decided to model the entire probability of packet length falling to the range of 65 bytes to 1517 bytes, in addition to two spikes at 64 bytes and 1518 bytes.

Considering that most of the downloaded packets, around 80%, have a size either at the minimum value (64 bytes) or maximum (1518 bytes), we ignored the 594 bytes spike discussed above by substituting the mean values of $p(l = 593)$ and $p(l = 595)$ for $p(l = 594)$, and then scaling all probabilities $p(l = 64)$, $p(65 \le l \le 1517)$ and $p(l = 1518)$.

The packet length distribution in the range of 65 bytes to 1517 bytes is very complicated (Figure 3). However, to simplify our model, following our approach in the uplink case, we treat it as a white noise like distribution and use a mean probability in this range, that is $p_2'/(1517-65+1) = p_2'/1453$.

Thus, the model for the downlink WWW packet can be presented as:

$$p_{downlink\_len}(l) = p_1'\delta(l-64) + p_3'\delta(l-1518) + (p_2'/1453)[u(l-65)-u(l-1517)] \quad (64 \le l \le 1518) \tag{3}$$

where p_1' is the value of the probability of packet length equal to 64 bytes and p_2' is the probability of packet length falling the range of 65 bytes to 1517 bytes.

In our measurement, $p_1' = 0.15 \pm 0.02$, $p_2' = 0.17 \pm 0.06$ and $p_3' = 0.68 \pm 0.1$. The variations of the parameters were caused by different packet traces.

B. HTTP Packet interarrival time

The packet interarrival time is the difference of the arrival times of the ith packet and the (i-1)th packet as given by (1). We can see in Figure 5, Figure 6 and Figure 7, the measured traces have long-tailed distributions in both uplink and downlink traffic, which are indicative of the self-similar nature of the Ethernet traffic [5], [6], [8].

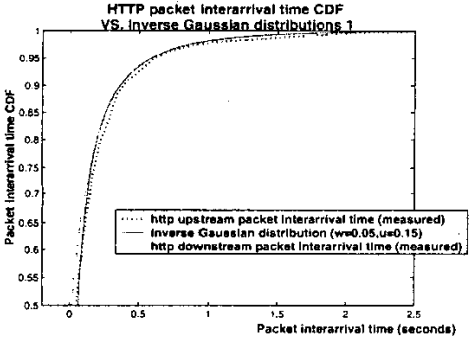

Figure 5 Packet interarrival time CDF of HTTP1

VS. Inverse Gaussian CDF

The HTTP requests and responses for one web page that contain several objects are typically captured with very short interarrival times. However between each web page loading, a user needs reading time or thinking time, which may last up to tens of minutes and cause the tail of the distribution to be significant. Furthermore, the round-trip time and the interarrival time between each object in one page also contribute to the shape of the curve.

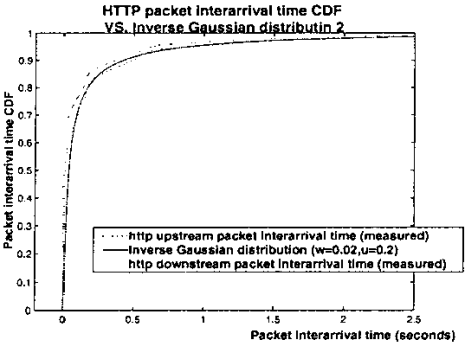

Figure 6 Packet interarrival time CDF of HTTP2

VS. Inverse Gaussian CDF

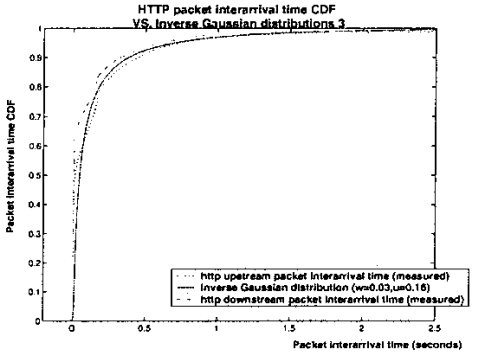

Figure 7 Packet interarrival time CDF of HTTP3

VS. Inverse Gaussian CDF

To establish the most suitable statistic distribution for modelling the measured curves, many heavy-tailed distributions were tested by varying relevant parameters. These distributions include Chi-squared distribution, exponential distribution, inverse Gaussian (Wald) distribution, lognormal distribution, Pareto distribution and Rayleigh distribution [1]. Both Probability Density Functions (PDFs) and Cumulative Distribution Functions (CDFs) of these distributions were used to compare with the measurement plots of each of the traces. However, we found the CDF is a better way to present all of the theoretical curves and the measurement curves as well as being easier to use.

In all of these distributions, it was found that the inverse Gaussian distribution is the best fit to the measurements (Figure 5, Figure 6 and Figure 7). Other distributions are much worse (Figure 8). Thus, the Inverse Gaussian distribution was chosen to model both uplink and downlink WWW packet interarrival time. The mathematical representation of probability density function (PDF) [1] of the curve is:

$$f_T(t) = \left[\frac{\lambda}{2\pi t^3}\right]^{\frac{1}{2}} \exp\left\{\frac{-\lambda(t-\mu)^2}{2\mu^2 t}\right\} \tag{4}$$

However, the CDF of the Inverse Gaussian distribution is not given explicitly, and we used the observation that the CDF is the integral of the PDF to approximate it:

$$F_T(t) = \int_{\min}^{\max} f_T(t)dt \approx \sum_{\min}^{\max} f_T(\Delta t)\Delta t \qquad (5)$$

In the above formula, Δt should be a very small value to reach a reasonable accuracy. In this particular case, Δt was 10-4.

The parameters $\lambda > 0$ and $\mu > 0$ are the scale parameter and location parameter respectively for the inverse Gaussian distribution. μ is the mean of the distribution and the variance of the distribution is μ^3 / λ. For our different traces, we found slightly different values of λ and μ. These will cause the variations of both parameters (see legends of **Figure 5, Figure 6** and **Figure 7**). The characters w and u in the legend of each above figure correspond to the variables λ and μ in equation (4).

However, we also noted that for each trace, the uplink and downlink have very similar packet interarrival time distributions. This is reasonable because the arrival of packets in each direction is interdependent, and the time at which a packet should be sent in one direction (e.g. an acknowledgement) tends to follow the arrival of a packet from the other direction. Thus, in situations that don't require very high accuracy, both uplink and downlink packet interarrival time distributions can use the inverse Gaussian distribution, modelled with the same values of λ and μ (**Figure 5, Figure 6** and **Figure 7**).

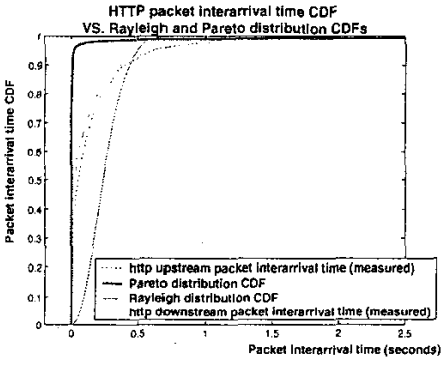

Figure 8 Packet interarrival time CDF of HTTP 1 VS. CDFs of various standard long-tailed distributions

V. CONCLUSION

To understand the WWW traffic of a single PC in a LAN, packet capture software was developed using a packet capture library WinPcap. A series of HTTP packet traces has been captured and studied. The study focused on the packet length and interarrival time in both uplink and downlink WWW traffic.

Most of the WWW uplink Ethernet packets have the minimum length, 64bytes, required in IEEE 802.3 while the downlink packet length is concentrated at the maximum value of 1518 bytes. Although the distributions of packet length in each direction are not the same, similar analysis methods can be used on them with or without special phenomena caused by specific web sites. An approximate algorithm was used for the packet length model in order to employ the standard unit step function and impulse function to describe the very complex measurement data. Using the models presented in the paper, a set of WWW traffic for a single PC can be described in both uplink and downlink by employing different parameters p_1, p_2 and p_3 (see (2) and (3)).

The packet interarrival times of WWW traffic were treated as continuous data and the inverse Gaussian distribution (see (4)) was determined to be the best fitting model compared with other standard long-tailed distributions. The CDF is more convenient than the PDF when studying the packet interarrival time. For those distributions without an explicit CDF formula, an approximation formula can be used with reasonable accuracy based on the definition of the integration. The uplink and downlink packet interarrival times have very similar distributions that reflect their dependent relationships with each other. Hence, with the same values of parameters of λ and μ an inverse Gaussian distribution can be used to model the packet interarrival time for a pair of uplinks and downlinks.

The results showed that all of the models fit the experimental results very well in this study. It may be noted that experimental the results obtained here may be constrained by personal habit, and that user behaviours can be an important aspect to be considered in future modelling work. Further measurements might be based on different traces from different people and more traces could be collected to make the model more general. However, the methodology used and described here has worked very well and is expected to be suitable for other studies on WWW traffic.

REFERENCE

[1] Merran Evans, Nicholas Hastings and Brian Peacock, "Statistical Distributions," John Wiley & Sons, Inc. 1993, pp. 45-136

[2] "WinPcap: the Free Packet Capture Architecture for Windows" http://netgroup-serv.polito.it/winpcap/.

[3] Ethereal, "How can I capture packets with CRC errors?" http://www.ethereal.com/faq.html - q4.22, April 14 2002.

[4] W. Richard Stevens, "TCP/IP Illustrated, Volume 1, The Protocols", Addison Wesley Longman, Inc., 12th Printing, Sep 1998, pp. 229-238.

[5] W. Leland, M. Taqqu, W. Willinger, D. Wilson, "On the Self-Similar Nature of Ethernet Traffic", IEEE/ACM Transactions on Networking, Vol. 2, n 1, Feb. 1994, pp. 1-15.

[6] M. Crovella, A. Bestavros, "Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes," IEEE/ACM Transactions on Networking, Vol. 5, no 6, Dec. 1997, pp. 835-846.

[7] D. R. Boggs, J. C. Mogul, C. A. Kent, "Measured Capacity of an Ethernet: Myths and Reality", Proceedings of SIGCOMM'88, Computer Communications Review 18, 1988, pp. 222-234.

[8] V. Paxson, S. Floyd, "Wide-Area Traffic: The Failure of Poisson Modeling", IEEE/ACM Transactions on Networking, Vol. 3, June 1995, pp. 226-244.

[9] J. F. Shoch, J. A. Hupp, "Measured Performance of an Ethernet Local Network", *Communications of the ACM* 23, 1980, pp. 711-721.