In *Journal of High-Speed Networks*, vol. 2, no. 3, pp. 305-323, December 1993. A preliminary version appears in *Proc. ACM Sigcomm '93*, San Fransisco, CA, pp. 289-298, September 1993.

Characterizing End-to-End Packet Delay and Loss in the Internet

Jean-Chrysostome Bolot
INRIA
B. P. 93
06902 Sophia-Antipolis Cedex
France
bolot@sophia.inria.fr

Abstract

We use the measured round trip delays of small UDP probe packets sent at regular time intervals to characterize the end-to-end packet delay and loss behavior in the Internet. By varying the interval between probe packets, it is possible to study the structure of the Internet load over different time scales. In this paper, the time scales of interest range from a few milliseconds to a few minutes. Our observations agree with results obtained by others using simulation and experimental approaches. For example, our estimates of Internet workload are consistent with the hypothesis of a mix of bulk traffic with larger packet size, and interactive traffic with smaller packet size. The interarrival time distribution for Internet packets is consistent with an exponential distribution. We also observe a phenomenon of compression (or clustering) of the probe packets similar to the acknowledgement compression phenomenon recently observed in TCP. Our results also show interesting and less expected behavior. For example, we find that the losses of probe packets are essentially random when the probe traffic uses a small fraction of the available bandwidth.

1 Introduction

Current data networks typically use packet switching as a means of dynamically allocating network resources on a demand basis. Packet-switching has been widely used because it facilitates the interconnection of networks with different architectures, and it provides flexible resource allocation and good reliability against node and link failure [7]. However, packet switching provides little control over the packet delay at the switches.

Therefore, one fundamental characteristic of a packet-switched network is the delay required to deliver a packet from a source to a destination[1]. Each packet generated by a source is routed to the destination via a sequence of intermediate nodes. The end-to-end delay is thus the sum of the delays experienced at each hop on the way to the destination. Each such delay in turn consists of two components, a fixed component which includes the transmission delay at a node and the propagation delay on the link to the next node, and a variable component which includes the processing and queueing delays at the node. Packets may be rejected at the intermediate nodes because of buffer overflow. Hence, another important characteristic of a packet-switched network is its packet loss rate.

Our objective is to understand the packet delay and loss behavior in the Internet. Understanding this behavior is important for the proper design of network algorithms such as routing and flow control algorithms (e.g. [12]), for the dimensioning of buffers and link capacity (e.g. [14]), and for choosing parameters in simulation and analytic studies (e.g. [3]). It is also essential for designing the emerging audio and video applications [8, 24, 27]. For example, the shape of the delay distribution is crucial for the proper sizing of playback buffers [24].

Many studies of packet delay and loss in various network environments have been reported in the literature. Next we describe analytic, simulation, and experimental approaches.

[1] Throughout the paper, we refer to this delay as the end-to-end delay or simply the packet delay.

The obvious analytic approach is to use queueing network models to analyze packet delay in computer networks. The analysis of queueing models is relatively simple if certain independence assumptions hold. In this case, the analysis gives rise to so-called product-form solutions, i.e. the queue size distribution for an entire network of queues is equal to the product of the queue size distribution of the individual queues [14]. Using this result, a number of parameters including the queue size and delay distributions can be easily calculated. However, product-form networks cannot incorporate features of real-life networks such as the correlations introduced when traffic streams merge and split, the regulation of traffic by the routing and flow control mechanisms, or the packet losses due to buffer overflow. Although progress in these areas has been recently reported (e.g. [4]), it should be pointed out that due to the lack of analytic solutions, many studies of packet delay and loss behavior have been conducted with simulation and experimental approaches.

Regarding simulation approaches, recent work has examined the impact of routing and flow control mechanisms on end-to-end delay. For example, reference [25] concludes that both link state and distance vector routing yield similar average packet delay statistics in a NSFNET-like network. References [28, 29] investigate the dynamic behavior of TCP connections. In realistic situations (i.e. for connections with so-called two-way traffic), it is found that the interactions between data and acknowledgement packets generate a clustering of the acknowledgement packets which in turn gives rise to rapid fluctuations in queue lengths. These results emphasize the importance of studying the dynamics, i.e. the time-dependent behavior, of computer networks.

Regarding experimental approaches, systematic measurements of packet delay and loss were carried out on the ARPANET as early as 1971 [14, ch. 6]. They examined the variations of packet delay for different paths, different times of day and days of the week, etc. Other measurements were taken to determine how delays across the ARPANET were influenced by packet length. The results were used to assess whether TCP performance could be improved by including a dependence on packet length in the retransmission timeout algorithm [15]. Several other studies have addressed timeout adjustment in TCP, and they have proposed improvements to take into account packet losses, packet retransmissions, and the variance of packet round trip delays [12, 13].

The NSFNET replaced the ARPANET in 1990. Recent studies have measured the delay and loss behavior in the NSFNET, and more generally in the Internet. They have examined this behavior over different time scales.

Merit Network Inc. publishes monthly statistics of packet delay between the nodes of the NSFNET. These statistics are obtained from measurements performed at 15 minute intervals. They are used in [5] to examine the distribution of median delay between nodes of the NSFNET. Unfortunately, the Merit statistics are based on measurements performed between the exterior interfaces of the backbone nodes. Thus, they might not accurately characterize the end-to-end delay over paths which span a combination of backbone and regional or international networks.

The behavior of end-to-end round trip delays over somewhat shorter time scales is examined in [19]. There, groups of 10 ICMP echo packets [26] are sent periodically from a source node to a destination node and echoed back to the source node, with a 1 minute interval between successive groups. Packets within a group are sent at regular 1 second intervals. Round trip delays are measured for each packet, and then averaged over a group. Various paths, i.e. source destination pairs, are considered. The results indicate that the delay distribution for all paths is best modeled by a constant plus gamma distribution, where the parameters of the gamma distribution depend on the path (e.g. a path over a regional network vs. a path over the NSFNET backbone) and the time of the day. A spectral analysis of the average delays shows a clear diurnal cycle, suggesting the presence of a base congestion level which changes slowly with time. Furthermore, packet losses and reorderings are positively correlated with various statistics of delay.

The behavior of end-to-end round trip delays over even shorter time scales is examined in [21, 22]. There, small UDP packets are sent every 39.06 ms from a source node to a destination node, and echoed back to the source node. The authors show how their measurements can be used to detect problems in the Internet. For example, they observed in May 1992 that round trip delays would increase dramatically every 90 seconds. They identified the problem as being caused by a 'debug' option in some gateway software. They identified other problems caused by synchronized routing updates, by faulty Ethernet interfaces, etc. [22]. Their measurements were also used to observe the dynamics of the Internet, e.g. the changes in round trip delays caused by route changes [21].

Despite all the efforts and results described above, the end-to-end performance of the Internet remains an area which deserves more research attention. For example, there is no clear consensus yet on how "well" the Internet performs, or on how to characterize its performance.

In this paper, we use measurements of end-to-end delay and loss to characterize the behavior of the Internet. We obtain these measurements with the UDP echo tool used in [21, 22], which provides the round trip delays of UDP packets at regular time intervals. By varying the interval between successive packets, we can examine the delay and loss behavior of the Internet over different time scales.

Our observations agree with results obtained by others using simulation and experimental approaches. For example, our estimates of Internet workload are consistent with the hypothesis of a mix of bulk traffic with larger packet size, and interactive traffic with smaller packet size. The interarrival time distribution for Internet packets is consistent with an exponential distribution. We also observe a phenomenon of compression (or clustering) of the probe packets similar to the acknowledgement compression phenomenon recently observed in TCP. Our results also show interesting and less expected behavior. For example, we find that the losses of probe packets are essentially random when the probe traffic uses a small fraction of the available bandwidth.

The rest of the paper is organized as follows. In Section 2, we describe the data collection process, i.e. how the measurements of packet delay and loss are obtained. In Section 3, we outline our strategy for analyzing the measurements. In Section 4, we analyze the characteristics of the packet delays. In Section 5, we analyze the characteristics of the packet losses. Section 6 concludes the paper.

2 Data collection

Recent measurements indicate that the number of hosts in the Internet is in excess of 1.7 million. Clearly, it is impossible to study the delay and loss characteristics for all possible connections, i.e. for all source-destination pairs. In this paper, we examine one specific connection in detail. This connection links INRIA in France to the University of Maryland (UMd) in the United States. The routes taken by the packets sent over the connection can be obtained either with the route record option of ping, or with traceroute [26]. Table 1 shows the route between INRIA and UMd as obtained with traceroute in July 1992. Nodes 5 and 6 are distinct nodes in the Ithaca Nodal Switching System. Nodes 4 and 5 are the endpoints of the transatlantic link between France and the United States. At the time the experiments were carried out (July 1992), the transatlantic link was the the bottleneck link with a bandwidth equal to 128kb/s.

1	tom.inria.fr
2	t8-gw.inria.fr
3	sophia-gw.atlantic.fr
4	icm-sophia.icp.net
5	Ithaca.NY.NSS.NSF.NET
6	Ithaca1.NY.NSS.NSF.NET
7	nss-SURA-eth.sura.net
8	sura8-umd-c1.sura.net
9	csc2hub-gw.umd.edu
10	avwhub-gw.umd.edu

Table 1: Route between INRIA and the University of Maryland in July 1992

Packet delays and losses on the INRIA-UMd connection are obtained using *NetDyn*, a measurement tool developed by Dheeraj Sanghi [22]. This tool sends UDP packets at regular intervals from a source host to a destination host via an intermediate host. Throughout the rest of the paper, we refer to these packets as probe packets, or simply probes. Upon receipt of a probe packet from the source, the intermediate host immediately echoes the packet to the destination host. The user can specify the number of probe packets to be sent, the size of the packets, and the interval between successive packets. In our experiments, the size of the probe packets is equal to 32 bytes. The interval between successive packets ranges over the following values: 8, 20, 50, 100, 200, and 500 ms. Each experiment lasts 10 minutes.

A packet includes three 6-byte timestamp fields. The source timestamp is written when the packet is sent by the source host. The echo timestamp is written when the packet is received by the intermediate host. The destination timestamp is written when the packet is received by the destination host. Furthermore, each packet has a unique packet number in order to detect packet losses.

If the source, intermediate, and destination hosts are geographically distant, then their local clocks may not be synchronized and hence the timestamps in the UDP probe packets would be difficult to interpret. To avoid this problem, we let the source host be the same as the destination host. Furthermore, we measure only the difference between the source timestamp and the destination timestamp, i.e. we measure only roundtrip delays. In our experiments, we use a DECstation 5000 as a source host. Its clock resolution is 3.906 ms.

We have taken measurements of end-to-end packet delay and loss on connections other than the INRIA-UMd connection, e.g. connections between UMd and MIT, between UMd and the University of Pittsburgh, between INRIA and universities in Europe, etc. Even though the physical characteristics of these connections are very different, we have found that the observations made on the basis of the measurements taken on the INRIA-UMd connection essentially hold for the other connections.

3 Data analysis strategy

In this section, we present our approach to analyzing the measurements obtained with the tool described in Section 2. Recall that in each experiment, the source sends probe packets at regular intervals. We denote by s_n the time at which packet n is sent by the source, by r_n the time at which it is received by the source from the echo host, and by rtt_n the packet round trip delay. If the packet is lost, then r_n and hence rtt_n are undefined. In this paper, we let $rtt_n = 0$ if packet n is lost. Otherwise, we have $rtt_n = r_n - s_n$. We denote by δ the interval between the send times of two successive packets, i.e. $\delta = s_{n+1} - s_n$ for all n.

Figure 1 shows the evolutions of rtt_n as a function of n in the range $0 \leq n \leq 800$ when $\delta = 50$ ms. Notice the large number of packet losses (the loss probability for this experiment turns out to be equal to 9%).

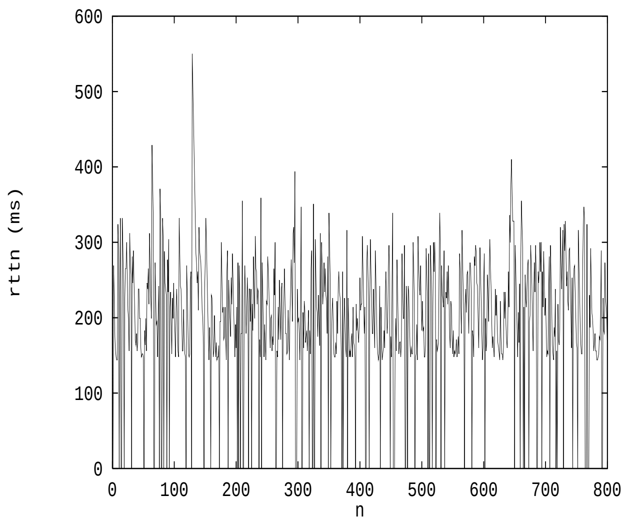

Figure 1: Evolutions of rtt_n vs. n when $\delta = 50$ ms

The evolutions of rtt_n as shown in Figure 1 are often referred to as time series plots, by which is meant a finite set of observations of a random process in which the parameter is time [2]. A large body of principles and techniques referred to as time series analysis has been developed to analyze time series, the goal being to infer as much as possible about the whole process from the observations [2]. Two important problems considered in time series analysis are the model fitting problem and the prediction problem. In the first problem, the goal is to obtain a model which fits the observations. In the second problem, the goal is to predict a future value of a process given a record of past observations.

One problem we examine in this paper is the model fitting problem. However, we do not use the standard procedures from time series analysis, for the following reason. It turns out that much of the work on this problem has been devoted to fitting observations to a variety of general types of models, the most well-known being the so-called AR (autoregressive), MA (moving average), and ARMA (auto regressive and moving average) models. The standard model fitting procedures do not require any assumption about the underlying structure of the observed system, and hence they are suitable even for systems where no background information is available. In our case, however, much is known about the system under study, namely the connection over which the probe packets are sent. Examples of available information include the number of hops, the mix of traffic expected at the intermediate nodes [3], etc. Therefore, in this paper, our approach is to use this information to interpret our observations and to suggest a specific model for the system behavior.

In parallel with this work, we are examining the model fitting problem and the prediction problem using standard procedures from time series analysis. Specifically, we examine whether ARMA models are adequate to model queueing delays in communication networks. This has consequences for the performance of predictive control mechanisms, since most such mechanisms described so far use these or related models (e.g. [16]).

4 Analysis of packet delay

Figure 1 above shows a time series plot of measured round trip delays, i.e. the evolutions of rtt_n as a function of n. In this section, we find it convenient to examine instead a so-called *phase plot* of the round trip delays. In a phase plot, a marker is printed at coordinates (x, y) if there exist a value n such that $x = rtt_n$ and $y = rtt_{n+1}$. The (x, y) plane is referred to as the phase plane. Figure 2 shows a phase plot of rtt_n in the range $0 \leq n \leq 800$ when $\delta = 50$ ms (the diagonal lines will be useful shortly). The structure of the phase plot is clearly very different from the structure of the time series plot. To understand this structure, it is convenient to introduce a simple model which captures the essential features of our experiments. Refer to Figure 3. In our model, we use a constant delay D to model the fixed component of the round trip delay of the probe packets, and a single server queue with finite buffer and FIFO service discipline to model the variable compo-

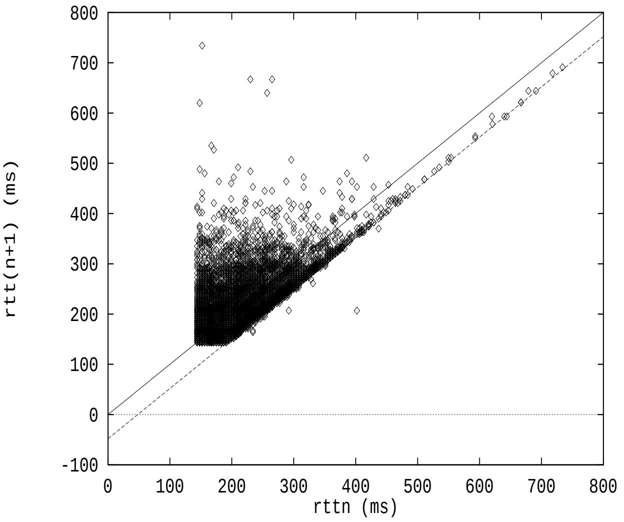

Figure 2: Phase plot of rtt_n in the range $0 \leq n \leq 800$ when $\delta = 50$ ms

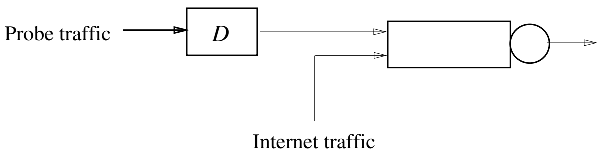

Figure 3: A model for our experiments

nent of this delay. We denote by μ the service rate, expressed in bit/s, of the server. We denote by K the buffer size. The arrival stream at the queue is the superposition of two streams. The probe stream is modeled as a periodic stream of fixed-size packets. We denote by P the length, expressed in bits, of the probe packets. The Internet stream is in turn the superposition of many streams which share common Internet resources with the probe connection. Throughout the rest of the paper, we refer to packets from the Internet stream as Internet packets.

We now explain the structure of the phase plot in Figure 2. Consider the situation where the workload seen by consecutive probe packets is constant. This situation occurs if the Internet traffic is light, i.e. if the number of Internet packets in the buffer is small, and if these packets are small, e.g. Telnet packets. In this case, the waiting time of consecutive probe packets is approximately constant. Let w_n denote the waiting time (not including the service time) of the probe packet n. Thus we have

$$w_{n+1} = w_n + \epsilon_n$$

where ϵ_n is a random process with mean 0 and low variance. Since $rtt_{n+1} = D + w_{n+1} + P/\mu$ and $rtt_n = D + w_n + P/\mu$, we obtain

$$rtt_{n+1} - rtt_n = \epsilon_n \tag{1}$$

Thus, in this case, the points on the phase plane are centered around the diagonal $rtt_{n+1} = rtt_n$ (this is the plain diagonal line in Figure 2), and they stay close to the minimum delay point with coordinates (D, D). In Figure 2, we find $D \approx 140$ ms.

Consider now the situation where δ is small and a large workload of Internet packets (e.g. one or more FTP packets) is received by the queue between the arrival instants of two consecutive probes. Let B denote the size of this Internet workload (expressed in bits) ahead of probe packet $n+1$. Then the queueing delay of probe packet $n+1$ is

$$w_{n+1} = w_n + B/\mu$$

and hence

$$rtt_{n+1} - rtt_n = B/\mu \tag{2}$$

If $w_{n+1} > \delta$, then one or more probe packets will accumulate behind probe packet $n+1$ waiting for the Internet workload to be processed by the server. Let k denote the number of such packets. Assume that no Internet packet is received at the queue between the arrival times of probe packets $n+1$ and $n+k$. Then probe packets $n+1$ through $n+k$ will leave the queue at regular intervals, specifically P/μ seconds apart. Therefore, we have

$$\begin{aligned} rtt_{n+2} - rtt_{n+1} &= (r_{n+2} - s_{n+2}) - (r_{n+1} - s_{n+1}) \\ &= (r_{n+2} - r_{n+1}) - (s_{n+2} - s_{n+1}) \\ &= P/\mu - \delta \end{aligned} \tag{3}$$

Similarly,

$$rtt_{n+3} - rtt_{n+2} = ... = rtt_{n+k} - rtt_{n+k-1} = P/\mu - \delta$$

and hence the points on the phase plane corresponding to packets $n+1$ to $n+k$ will be located on the straight line $rtt_{n+1} = rtt_n + P/\mu - \delta$. This straight line is the thin dashed line in Figure 2. If $\delta < P/\mu$, then the probe traffic alone completely saturates the queue. Therefore, it is reasonable to keep $\delta \geq P/\mu$ in all experiments.

The existence of points on the line $rtt_{n+1} = rtt_n + P/\mu - \delta$ in the phase plane indicates that probe packets accumulate behind large Internet packets. We refer to this phenomenon as probe compression because of its similarity with the phenomenon of ACK compression which has been observed in simulations [29] and in measurements on the NSFNET [18]. We note that the line $y = x + P/\mu - \delta$ intersects the x-axis for $x = \delta - P/\mu$. In Figure 2, we find this point to be about 48 ms. Since $\delta = 50$ ms and $P = 32$ bytes, we obtain $\mu \approx 130$ kb/s.

This confirms that the transatlantic link between France and the United States with a bandwidth equal to 128 kb/s is the bottleneck link on the path from INRIA to UMd.

We now examine the packet delay when δ is very large, i.e. when probe packets are sent infrequently. Figure 4 shows the phase plot of rtt_n in the range $0 \leq n \leq 800$ when $\delta = 500$ ms. In this case, we have $\delta - P/\mu = 490$ ms. We observe that only two points on the phase plot are located on the line $rtt_{n+1} = rtt_n - 490$, indicating that consecutive probes almost never accumulate behind one another. This is expected since the maximum queueing delay measurement in this experiment is 620 ms (corresponding to a round trip delay of 760 ms and a propagation delay of 140 ms), i.e. barely larger than the interarrival time between successive probe packets. Equation (1) indicates that for large values of δ, the points in the phase plane should be scattered around the diagonal. This is indeed what we observe in Figure 4.

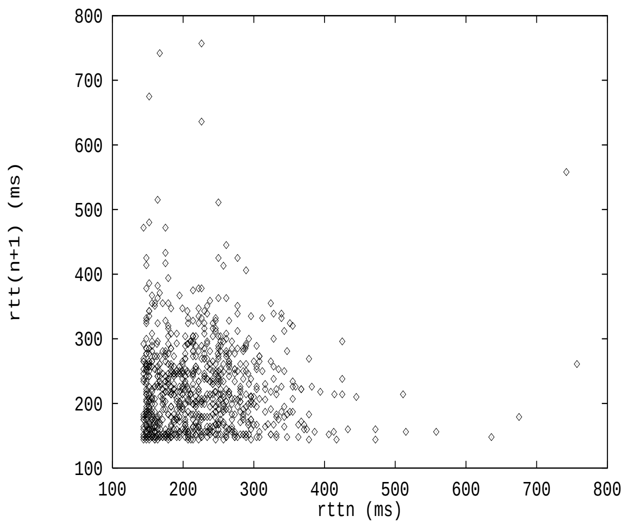

Figure 4: Phase plot of rtt_n in the range $0 \leq n \leq 800$ when $\delta = 500$ ms

We have examined the round trip delays of probe packets over connections other than the INRIA-UMd connection. In all cases, we have found that the structure of the phase plots of rtt_n is similar to that described above. To illustrate this, we next present the phase plots of round trip delays measured in May 1993 between UMd and the University of Pittsburgh. Table 2 shows the path taken by the probe packets at the time of the experiments. It is not clear which link in the path is the bottleneck link. However, it is very likely that the bottleneck bandwidth is much higher than the bottleneck bandwidth between INRIA and UMd in June 1992, namely 128 kb/s. Figure 5 shows the phase plot

1	lena.cs.umd.edu
2	avw1hub-gw.umd.edu
3	csc2hub-gw.umd.edu
4	192.221.38.5
5	en-0.enss136.t3.nsf.net
6	t3-1.Washington-DC-cnss58.t3.ans.net
7	t3-3.Washington-DC-cnss56.t3.ans.net
8	t3-0.New-York-cnss32.t3.ans.net
9	t3-1.Cleveland-cnss40.t3.ans.net
10	t3-0.Cleveland-cnss41.t3.ans.net
11	t3-0.enss132.t3.ans.net
12	externals.gw.pitt.edu
13	136.142.2.54
14	hub-eh.gw.pitt.edu

Table 2: Route between the University of Maryland and the University of Pittsburgh in May 1993

of rtt_n in the range $0 \leq n \leq 800$ when $\delta = 8$ ms. The figure also shows the straight lines $rtt_{n+1} = rtt_n$ and $rtt_{n+1} = rtt_n - 8$. Figure 6 shows the phase plot of rtt_n in the range $0 \leq n \leq 800$ when $\delta = 50$ ms. The somewhat regular spacing between the points in the phase plane is caused by the 3 ms clock resolution of the source host at UMd. When δ is small, we observe

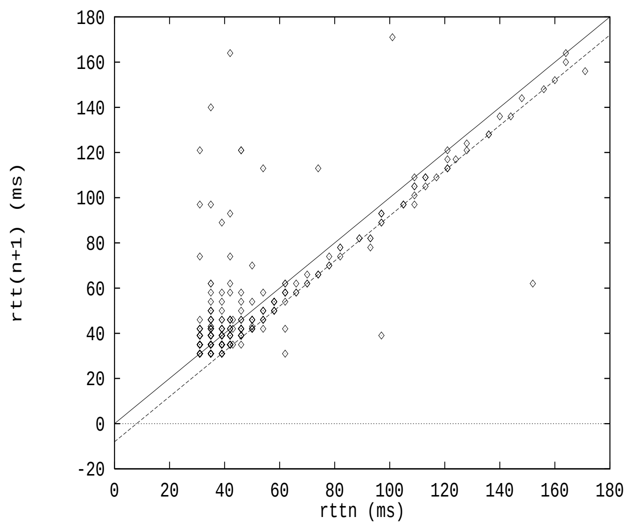

Figure 5: Phase plot of rtt_n in the range $0 \leq n \leq 800$ when $\delta = 8$ ms

the same probe compression phenomenon described earlier. When δ is large, we observe that the points in the phase plane are scattered around the line $rtt_{n+1} = rtt_n$.

Throughout the rest of the paper, we analyze only those measurements taken on the INRIA-UMd connection. Next, we present an exact analysis of the queueing

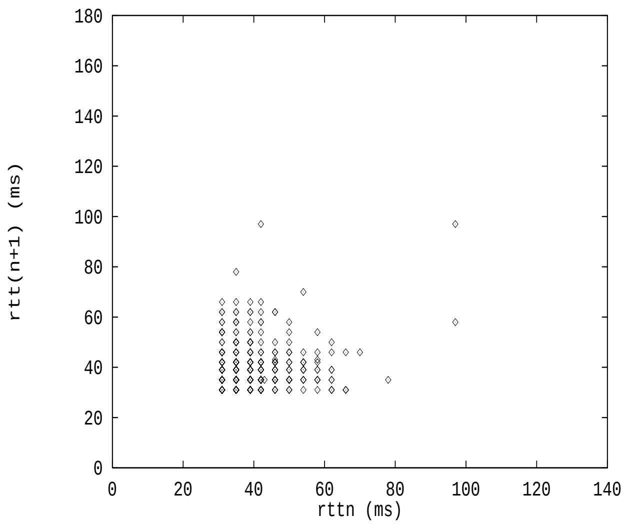

Figure 6: Phase plot of rtt_n in the range $0 \leq n \leq 800$ when $\delta = 50$ ms

model shown in Figure 3. The analysis is based on two successive applications of Lindley's recurrence equation [14]. Lindley's recurrence equation expresses the relationship between the waiting times of two successive customers in a single channel queue. Specifically, let w_n denote the waiting time of packet n, y_n denote the service time of packet n, and x_n denote the interarrival time between packets n and $n+1$. Then

$$w_{n+1} = \begin{cases} w_n + y_n - x_n & \text{if } w_n + y_n - x_n > 0 \\ 0 & \text{otherwise} \end{cases}$$

A "graphical proof" of this equation is shown in Figure 7. Throughout the rest of the paper, we use x^+ to denote $\max(x, 0)$. With this notation, the above equation can be rewritten as

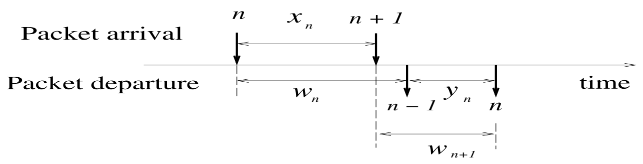

Figure 7: A proof of Lindley's recurrence equation

$$w_{n+1} = (w_n + y_n - x_n)^+$$

We now go back to our queueing model shown in Figure 3. The model can be thought of as a slotted-time model in which slot boundaries are defined by probe arrival times. We assume that the first probe packet arrives at the queue at time δ, and hence that probe packet n arrives at time $n\delta$. We refer to the interval $[n\delta, (n+1)\delta]$ as 'slot n'. We assume that the Internet stream contributes b_n bits during slot n. b_n is a random variable which characterizes the traffic pattern of the Internet stream. We further assume that all b_n bits arrive at the queue at the same time t_n (clearly $n\delta \leq t_n \leq (n+1)\delta$). Let wb_n denote the waiting time of the Internet packet. Applying Lindley's recurrence equation to wb_n and w_n, we obtain

$$wb_n = (w_n + P/\mu - t_n)^+ \tag{4}$$

Applying Lindley's recurrence equation to w_{n+1} and wb_n, we obtain

$$w_{n+1} = (wb_n + b_n/\mu - (\delta - t_n))^+ \tag{5}$$

Substituting equation (4) into equation (5), we obtain

$$w_{n+1} = ((w_n + P/\mu - t_n)^+ + b_n/\mu - (\delta - t_n))^+$$

The term $w_n + P/\mu - t_n$ is positive if the buffer does not empty during the interval $[n\delta, t_n + n\delta]$. Then the above equation becomes

$$\begin{aligned} w_{n+1} &= (w_n + P/\mu - t_n + +b_n/\mu - (\delta - t_n))^+ \\ &= (w_n + (P + b_n)/\mu - \delta)^+ \end{aligned}$$

The term $w_n + (P + b_n)/\mu - \delta$ is positive if the buffer does not empty during the interval $[n\delta, t_n + n\delta]$. Then the above equation becomes

$$w_{n+1} = w_n + (P + b_n)/\mu - \delta$$

and hence

$$b_n = \mu(w_{n+1} - w_n + \delta) - P \tag{6}$$

Thus the probability distribution of b_n, i.e. the probability distribution of the Internet workload over an interval of length δ, can be estimated from the distribution of $w_{n+1} - w_n$. Recall, however, that the above equality holds only if the buffer does not empty during the interval $[n\delta, (n+1)\delta]$. If the buffer does empty during this interval, then equation (6) might not hold. Therefore, it is reasonable to estimate b_n using equation (6) if δ is sufficiently small, typically if the product $\delta\mu$ is smaller than some average value of b_n.

Figure 8 shows the distribution of $w_{n+1} - w_n + \delta$ for $\delta = 20$ ms. From equation (6), we see that

$$w_{n+1} - w_n + \delta = (b_n + P)/\mu$$

i.e. $w_{n+1} - w_n + \delta$ is the Internet workload, expressed in ms, received by the server during slot n. We note that $w_{n+1} - w_n + \delta$ is also the interarrival time between the probe packets n and $n+1$ when they return to the source after their journey in the Internet.

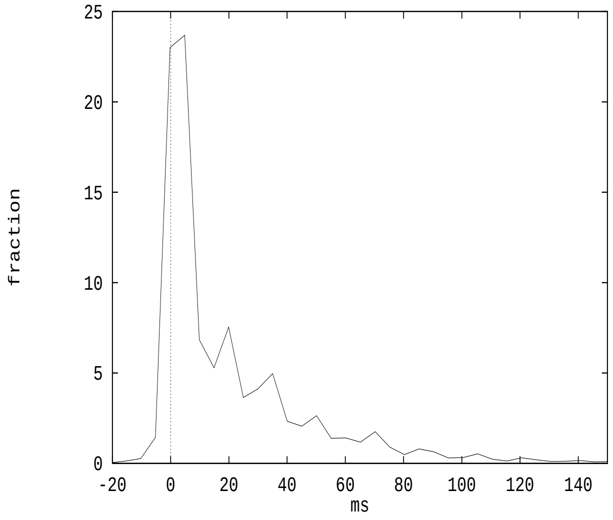

Figure 8: Distribution of $w_{n+1} - w_n + \delta$ for $\delta = 20$ ms

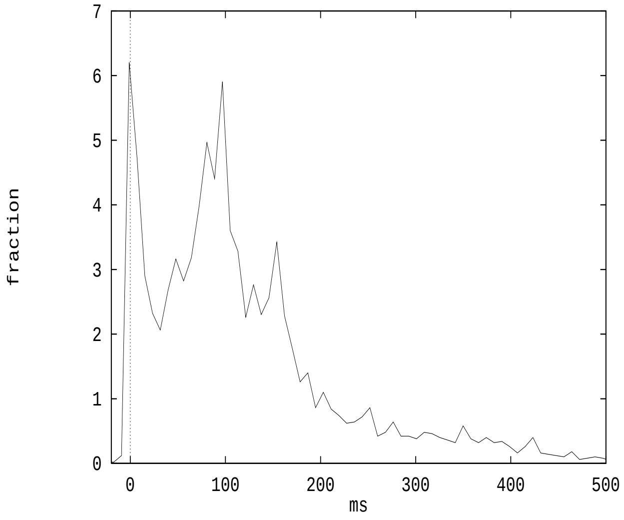

Figure 9: Distribution of $w_{n+1} - w_n + \delta$ for $\delta = 100$ ms

The same arguments used earlier to explain the structure of the phase plot in Figure 2 can be used to explain the structure of the distribution in Figure 8. Consider for example the leftmost peak in the distribution. This peak is centered around the value P/μ, i.e. it corresponds to those packets for which $w_{n+1} - w_n = P/\mu - \delta$. These are the packets that accumulated in the buffer behind a large Internet packet (cf. Equation 3). The second leftmost peak is centered around the value δ. Thus, it corresponds to those packets for which $w_{n+1} = w_n$, i.e. to those packets which experience very small queueing delays in the queue (cf. Equation 1). The third leftmost peak corresponds to those packets which were the first packets in a series of probe packets to accumulate behind a large Internet packet. Using Equation (6), we can find the size b_n of this packet. We obtain

$$\begin{aligned} b_n &= \mu(w_{n+1} - w_n + \delta) - P \\ &= 128 * 35 - 32 * 8 \\ &= 4224 \text{ bits} = 528 \text{ bytes} \end{aligned}$$

i.e. approximately the size of one FTP packet. Similarly, we find that the fourth leftmost peak corresponds to those packets which accumulated behind 2 FTP packets, etc.

Figure 9 shows the distribution of $w_{n+1} - w_n + \delta$ for $\delta = 100$ ms. The structure of the distribution is similar to that found when $\delta = 20$ ms. However, we note that the height of the leftmost peak relative to that of the other peaks is much smaller than in Figure 8. This is expected, since the number of consecutive probe packets that accumulate behind one or more FTP packets decreases as δ increases (i.e. probe compression becomes less frequent as δ increases).

We now examine the arrival process of the Internet stream, i.e. the interarrival times between consecutive Internet packets. This process can be estimated from measures of $w_{n+1} - w_n$ as follows. From Equation (6), we obtain

$$w_{n+1} - w_n = (P + b_n)/\mu - \delta$$

If the Internet workload received by the queue during slot n is zero then $b_n = 0$ and hence

$$w_{n+1} - w_n = P/\mu - \delta$$

Thus, if $w_{n+1} - w_n > P/\mu - \delta$, then one or more Internet packets arrived at the queue during slot n. However, it is not possible to determine how many such packets arrived during the slot. Clearly, the probability that more than one packet arrived increases as δ increases. Therefore, it is necessary to choose a small value of δ to accurately estimate the interarrival time distribution of Internet packets. Bursty periods with many packets of small interarrival times would be missed if δ is large. In this case, the interarrival distribution would be skewed toward the larger values. Below, we use $\delta = 8$ ms. Since $P = 32$ bytes and $\mu = 128$ kb/s, it follows that $P/\mu - \delta \approx -6$ ms. Thus, an Internet arrival occurs during slot n if $w_{n+1} - w_n > -6$ ms. However, recall that in our experiments the roundtrip delays (and hence the waiting times w_n and w_{n+1}) are measured with a clock accuracy of 3 ms (refer to Section 2). To avoid the spurious detection of Internet arrivals, we say that an Internet arrival occurs during slot n if $w_{n+1} - w_n \geq 0$.

The plain line is Figure 10 shows the distribution of the interarrival times of Internet packets, i.e. the time expressed in slots which separates two Internet arrivals.

The dashed line shows a geometric distribution with parameter $p = 0.37$ (i.e. it shows the curve $p(1-p)^{k-1}$ versus k). The close fit indicates that the Internet inter-

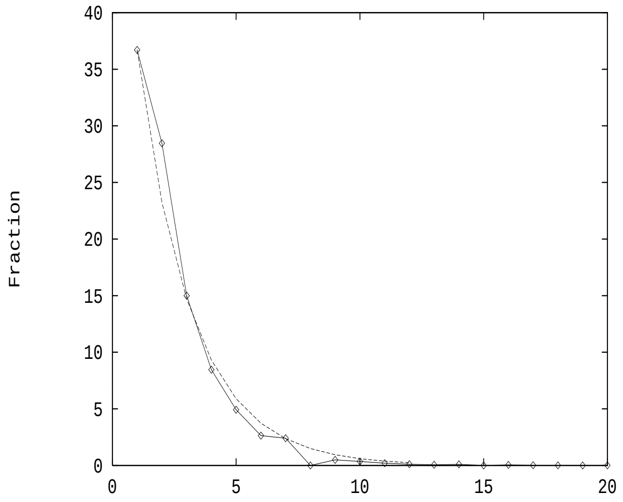

Figure 10: Distribution of Internet interarrival times for $\delta = 8$ ms

arrival time distribution of the Internet traffic is consistent with the hypothesis of an exponential distribution. This result ties in well with other measurements done at the Ann Arbor, MI, node of the NSFNET [11].

5 Analysis of packet loss

The structure of the loss process in a network is typically characterized by the packet loss probability. We let ulp denote the (unconditional) loss probability for the probe packets. Thus, ulp is defined by

$$ulp = P(rtt_n = 0)$$

Table 3 presents the measured values of ulp for different values of δ (ignore ulp and plg for the moment). The loss

$\delta(ms)$	8	20	50	100	200	500
ulp	0.23	0.16	0.12	0.10	0.11	0.09
clp	0.60	0.42	0.27	0.18	0.18	0.09
plg	2.5	1.7	1.3	1.2	1.2	1.1

Table 3: Variations of ulp, clp, and plg for different values of δ

probability increases when δ becomes very small because the contribution of the probe packets to the buffer queue length becomes non negligible. Furthermore, if a probe packet is lost at time t because of buffer overflow, then the next probe packet which arrives at time $t + \delta$ will also be lost if δ is less than the service time of the packet in service. Clearly, the probability of this occurrence increases as δ decreases.

This indicates that probe losses might be correlated and occur in bursts. It is well-known that the burstiness of the packet loss process has a significant impact on the performance of various network protocols. For example, correlated losses decrease the effectiveness of open-loop error control schemes such as forward error correction (FEC) schemes, but they increase the effectiveness of closed-loop control schemes such as the automatic request (ARQ) schemes [4]. In this paper, we characterize the burstiness of the probe packet loss by the conditional probability that a probe packet is lost given that the previous probe packet was lost. A related performance measure is the number of consecutively lost probe packets, which we refer to as the packet loss gap [9]. We denote by clp the conditional probe loss probability, i.e.

$$clp = P(rtt_{n+1} = 0 | rtt_n = 0)$$

We denote by plg the packet loss gap.[2]

Table 3 presents the measured values of clp and plg for different values of δ (refer to the beginning of the section). We observe that clp is greater than ulp for all values of δ. This can be explained as follows. The conditional loss probability is the probability that a given probe packet, say packet $n+1$, is lost given that packet n is lost, i.e. given that the buffer occupancy upon arrival of probe packet n is equal to K. The unconditional loss probability is the probability that packet $n+1$ is lost, irrespective of the buffer occupancy upon arrival of probe n. Clearly, the loss probability for probe $n+1$ increases with the buffer occupancy upon arrival of probe n. Therefore, $clp \geq ulp$. For large values of δ, clp and ulp are almost identical. This is expected since the states of the buffer seen by two successive probe packets become less and less correlated as δ increases. Our results show that the losses of probe packets are essentially random as long as the probe traffic uses less than 10% of the available capacity of the connection over which the probes are sent.

The occurrence of random losses in the Internet has already been documented in the literature. For example, it is reported in [17] that faulty Ethernet and FDDI interface cards randomly drop packets, with a loss rate reaching up to 3% in the Suranet mid-level network.

[2] Assuming that the sequence of rtt_n is stationary and ergodic, plg can be expressed in terms of clp by

$$plg = 1/(1 - clp)$$

A proof of this can be obtained using results from Palm probabilities (see e.g. [1, 9]).

Since the path of the connection between INRIA and the University of Maryland crosses over Suranet, it is likely that a fraction of the observed probe losses are caused by such faulty interface cards. However, it is not clear why *ulp* stays as high as 10% even when δ is very large.

It is important to note that the loss gap stays close to 1 even for relatively small values of δ. This result has important consequences for the design of audio and video applications over the Internet. Audio applications send audio packets at regular intervals. The interval length depends on the audio sampling frequency, the number of bits used to encode each sample, and the number of samples aggregated in each audio packet. Typical values for the interval length range between 22.5 ms [24] and 125 ms [27]. Our experiments suggest that an open loop error control mechanism based on FEC would be useful to compensate for lost audio packets (an example such mechanism is described in [23]).

Video applications do not send video packets at regular intervals. For example, the video codec of IVS [27], a software codec recently developed at INRIA for videoconference over the Internet, generates variable-size packets at intervals ranging from 15 to 120 ms. The interval length depends on the format of the video picture being encoded, the movement detected between two consecutive frames, etc. Thus, it is not clear whether the conclusions above still apply in this case. We are currently investigating this issue.

6 Conclusion

In [22], it was shown that the UDP echo tool was useful to study network problems such as faulty gateway hardware and software components, faulty network interface cards, etc. In this paper, we have shown that this tool is also useful to analyze the end-to-end characteristics of the Internet over different time scales.

Our results can be interpreted using a simple single-server queueing model with 2 input streams, where one stream represents the probe traffic and the other stream represents the Internet traffic. We are currently analyzing one such model in which the probe arrival process is deterministic and the Internet arrival process is batch deterministic with a general batch size distribution (this is a special case of the model described in [20]). We derive the batch size distribution from our measurements (see Figures 8 and 9). Preliminary investigations show that the analytical results tie in well with our experimental data. In particular, they bring out the probe compression phenomenon. They also indicate that probe losses appear to be random except when the Internet traffic intensity is very high. We are currently continuing the analysis of this model.

Acknowledgements

Special thanks to Dheeraj Sanghi who provided me with the UDP echo tool. Daniel Mossé, Ellen Siegel, and Karen Sollins provided machines on which to run the UDP echo process. Srinivasan Keshav, Ashok Erramilli, and the members of the networking research group at INRIA provided valuable feedback.

References

[1] F. Baccelli, P. Brémaud, *Palm Probabilities and Stationary Queues*, Lecture Notes in Statistics, vol. 41, Springer, Heidelberg, 1987.

[2] G. Box, G. M. Jenkins, *Time Series Analysis, Forecasting, and Control*, Holden-Day, San Fransisco, 1970.

[3] R. Caceres, P. Danzig, S. Jamin, D. Mitzel, "Characteristics of wide-area TCP/IP conversations", *Proc. ACM Sigcomm '91*, Zurich, Switzerland, pp. 101-112, Sept. 1991.

[4] I. Cidon, A. Khamisy, M. Sidi, "Analysis of packet loss processes in high-speed networks", *IEEE Trans. Info. Theory*, vol. 39, no. 1, pp. 98-108, Jan. 1993.

[5] K. Claffy, G. Polyzos, H-W. Braun, "Traffic characteristics of the T1 NSFNET backbone", *Proc. IEEE Infocom '93*, San Fransisco, CA, pp. 885-892, April 1993.

[6] K. Claffy, G. Polyzos, H-W. Braun, "Application of sampling methodologies to network traffic characterization", to appear in *Internetworking: Research and Experience*.

[7] D. D. Clark, "The design philosophy of the DARPA Internet protocols", *Proc. ACM Sigcomm '88*, Stanford, CA, pp. 106-114, Aug. 1988.

[8] D. D. Clark, S. Shenker, L. Zhang, "Supporting real-time applications in an integrated services packet network: Architecture and mechanism", *Proc. ACM Sigcomm '92*, Baltimore, MD, pp. 14-26, Aug. 1992.

[9] J. Ferrandiz, A. Lazar, "Monitoring the packet gap of real-time packet traffic", *Queueing Systems*, vol. 12, pp. 231-242, Dec. 1992.

[10] S. Floyd, V. Jacobson, "Traffic phase effects in packet-switched gateways", *Internetworking: Research and Experience*, vol. 3, no. 3, pp. 115-156, Sept. 1992.

[11] S. Heimlich, "Traffic characterization of the NSFNET national backbone", *Proc. Winter USENIX Conference '90*, Washington, DC, Jan. 1990.

[12] V. Jacobson, "Congestion avoidance and control", *Proc. ACM Sigcomm '88*, Stanford, CA, pp. 314-329, August 1988.

[13] P. Karn, C. Partridge, "Improving round-trip time estimates in reliable transport protocols", *ACM Trans. on Computer Systems*, vol. 9, no. 4, pp. 364-373, Nov. 1991.

[14] L. Kleinrock, *Queueing Systems. Volume 2: Computer Applications*, Wiley-Interscience, New York 1976.

[15] D. Mills, "Internet delay experiments", RFC 889, December 1983.

[16] P. P. Mishra, H. Kanakia, "A hop-by-hop rate-based congestion control scheme", *Proc. ACM Sigcomm '92*, Baltimore, MD, pp. , Aug. 1992.

[17] P. P. Mishra, D. Sanghi, "TCP flow control in lossy networks: Analysis and enhancement", *Proc. Networks '92*, Trivandrum, India, Oct. 1992.

[18] J. Mogul, "Observing TCP dynamics in real networks", *Proc. ACM Sigcomm '92*, Baltimore, MD, pp. 305-317, August 1992.

[19] A. Mukherjee, "On the dynamics and significance of low frequency components of Internet load", Technical report CIS-92-83, University of Pennsylvania, Philadelphia, PA, December 1992.

[20] M. Murata, Y. Oie, T. Suda, "Analysis of a discrete-time single-server queue with bursty inputs for traffic control in ATM networks", *IEEE JSAC*, vol. 8, no. 3, pp. 447-458, April 1990.

[21] D. Sanghi, A. K. Agrawala, B. Jain, "Experimental assessment of end-to-end behavior on Internet", *Proc. IEEE Infocom '93*, San Fransisco, CA, pp. 867-874, March 1993.

[22] D. Sanghi, O. Gudmundsson, A. K. Agrawala, "Study of network dynamics", *Proc. 4th Joint European Networking Conference*, Trondheim, Norway, pp. 241-249, May 1993.

[23] N. Shacham, P. McKenney, "Packet recovery in high-speed networks using coding and buffer management", *Proc. IEEE Infocom '90*, San Fransisco, CA, pp. 124-131, May 1990.

[24] H. Schulzrinne, "Voice communication across the Internet: A Network Voice Terminal", University of Massachussetts Technical Report 92-50, June 1992.

[25] A. U. Shankar et al., "Performance comparison of routing protocols using MaRS: Distance-vector versus link-state", *Proc. ACM Sigmetrics and Performance '92*, Newport, RI, pp. 181-192, June 1992.

[26] R. Stine, " FYI on a Network Management Tool Catalog: Tools for Monitoring and Debugging TCP/IP Internets and Interconnected Devices", RFC 1147, April 1990.

[27] T. Turletti, "A H.261 software codec for videoconferencing over the Internet", INRIA Research Report 1834, January 1993.

[28] L. Zhang, D. D. Clark, "Oscillating behavior of network traffic: A case study simulation", *Internetworking: Research and Experience*, vol. 1, no. 2, pp. 101-112, Dec. 1990.

[29] L. Zhang, S. Shenker, D. D. Clark, "Observations on the dynamics of a congestion control algorithm: the effects of 2-way traffic", *Proc. ACM Sigcomm '91*, Zurich, Switzerland, pp. 133-147, Sept. 1991.