Characteristics of Wide-Area TCP/IP Conversations

Ramón Cáceres† Peter B. Danzig* Sugih Jamin* Danny J. Mitzel*

*Computer Science Department, University of Southern California,
Los Angeles, California 90089-0782

†Computer Science Division, University of California,
Berkeley, California 94720

traffic@excalibur.usc.edu

Abstract

In this paper, we characterize wide-area network applications that use the TCP transport protocol. We also describe a new way to model the wide-area traffic generated by a stub network. We believe the traffic model presented here will be useful in studying congestion control, routing algorithms, and other resource management schemes for existing and future networks.

Our model is based on trace analysis of TCP/IP wide-area internetwork traffic. We collected the TCP/IP packet headers of USC, UCB, and Bellcore networks at the point they connect with their respective regional access networks. We then wrote a handful of programs to analyze the traces. Our model characterizes individual TCP conversations by the distributions of: number of bytes transferred, duration, number of packets transferred, packet size, and packet interarrival time.

Our trace analysis shows that both interactive and bulk transfer traffic from all sites reflect a large number of short conversations. Similarly, it shows that a very large percentage of traffic is bidirectional, even for bulk transfer. We observed that interactive applications send significantly different amounts of data in each direction of a conversation, and that interarrival times for interactive applications closely follow a constant plus exponential model. Half of the conversations are directed to a handful of networks, but the other half are directed to hundreds of networks. Many of these observations contradict commonly held beliefs regarding wide-area traffic.

This research was supported by an equipment grant from the Charles Lee Powell Foundation. Ramón Cáceres was supported by the NSF and DARPA under Cooperative Agreement NCR-8919038 with CNRI, by AT&T Bell Laboratories, Hitachi, a University of California MICRO grant, and ICSI.

1. Introduction

"The key issue in the design or selection of a congestion management scheme is the traffic pattern, and traffic patterns are dependent upon the application [Jain90]." This paper presents conversation level analysis of wide-area TCP traces collected on two campus networks—University of Southern California (USC) and University of California, Berkeley (UCB), and one industrial research site—Bellcore. Most of the analysis was done as part of term projects for graduate courses in performance evaluation and distributed systems at the University of Southern California. Our goal was to collect information that would be useful in evaluating future network designs. Since TCP packets make up roughly 80% of all wide-area network traffic,[1] a model based on TCP traffic is necessary to study network behavior. We restrict our discussion to TCP in this paper. Table 1 summarizes our most important results.

When simulating new congestion, flow control, and routing algorithms one needs to model the overall pattern of traffic flowing through the network, from distribution of packet sizes and interarrival times to characteristics such as distribution of host reference patterns and direction of traffic flow. Current practice is to use FTP and TELNET sources, where FTP sources send huge quantities of data in one direction and TELNET sources send a Poisson stream of small packets in one or both directions [Demers89] [Rama90]. Current practice ignores the distribution of number of bytes transmitted, the bidirectionality of bulk traffic sources, and the duration of interactive connections.

Future broadband wide-area networks will probably transfer large amounts of data and carry a mix of traffic currently not found on the Internet. We believe this does not trivialize our present study, for several reasons. First, it will be several years before the current traffic mix changes appreciably. Second, as it changes, it will not obviate the

[1]For the UCB data, UDP packets make up 16% of all network traffic, while ICMP packets account for only 1% of all traffic. Of all UDP packets, 63.63% belongs to DNS, 15.82% to ROUTE, and 10.51% to NTP.

existence of traditional traffic. Third, we believe this paper illustrates a general technique for workload measurement of wide-area internetworks.

75-90% of the conversations belonging to bulk transfer applications send less than 10 kilobytes of data. Bulk transfer is request-response in nature.
Over 90% of interactive conversations send fewer than 1,000 packets and 50% of interactive conversations last less than a minute and a half. Packets belonging to interactive applications are mostly smaller than 512 bytes.
A constant plus exponential distribution best models interarrival times of packets belonging to interactive applications.
A large portion of bulk transfer applications, which are responsible for more than 50% of observed network traffic, show bidirectional traffic flow.
Interactive applications can generate 10 times more data in one direction than the other, using packet sizes ranging from 1 byte to 512 bytes.

Table 1: Selected results.

Previous traffic studies of TCP/IP have examined the statistics of the aggregated packet arrival process on local area networks [Jain86] [Gusella90] [Leland91], at border routers [Cáceres89], and inside a wide-area backbone [Heimlich89]. These studies have shown that packet interarrival times are not Poisson, but rather follow a *packet-train* model. The *packet-train* model has proven valuable in the design of packet routers [Feldmeier88] [Jain89].

The study presented in this paper is different from all the studies mentioned above. Instead of confining ourselves to the network and transport layers, we studied the characteristics of several applications. We believe these applications are representative of applications currently running on wide-area networks.

The decision to characterize application traffic was supported by the following observations. Measured interarrival times alone are not adequate to characterize conversations for the purpose of driving flow and congestion control algorithm simulations, because interarrival times are themselves a function of existing flow control mechanisms—interarrival times do characterize interactive traffic, which is unlikely to be constrained by flow control. In contrast, bulk traffic must be characterized by the amount of data transferred—the observed duration of bulk transfers mostly reflects network link speed and flow control algorithm. Furthermore, although interactive conversations are bidirectional, they send much more data in one direction than in the other; an accurate model must take this into account. Finally, some applications converse with more networks than do others (see Figure 6).

From these observations, we concluded that researchers would benefit from more realistic traffic models, particularly in studying switching and control mechanisms through simulation. This paper makes the first step towards an internetwork source model. It outlines the necessary steps to describe and simulate a new conversation between two networks. However, it does not seriously investigate the question of when to establish a conversation between two networks; we are currently addressing this problem.

The next section describes the data collection and analysis methods. Section 3 analyzes the characteristics of the TCP conversations observed. Section 4 discusses a network traffic model based on our findings. Section 5 discusses possible uses of our model and future work. Section 6 concludes the paper by discussing the relationship of our results to commonly held assumptions of wide-area network traffic.

2. Measurement and Analysis Methodology

Below we describe the data collection methods, loss rates, and our definition of a conversation.

2.1. Data Collection Sites

Wide-area traffic data was collected at two university campuses and one industrial research laboratory. The data collected at UCB traced all traffic between the campus and the Bay Area Regional Research Network (BARRnet); data collected at USC traced all traffic between the campus and Los Nettos; and data collected at Bellcore traced all traffic between their Morristown laboratory and the John von Neumann Center Network (JVNCnet).

2.2 Trace Contents

A total of 5,891,622 TCP packets were collected at UCB, 5,221,036 at USC, and 1,703,269 at Bellcore. Traces from UCB and USC were collected over a period of one day, traces from Bellcore were collected over a period of three days. The collection started at 10:20 on Tuesday, October 31, 1989 at UCB, 14:24 on Tuesday, January 22, 1991 at USC, and 14:37 on Tuesday, October 10, 1989 at Bellcore.

Each record in all of the traces consists of a time stamp and the first 56 bytes of raw network data. The time stamp records the arrival time of the packet at the tracing apparatus. The 56 bytes of data hold the packet headers from the datalink layer (Ethernet), the network layer (e.g. IP), and the transport layer (e.g. TCP and UDP).[2]

[2]We did not encounter any packets with IP or TCP protocol options. In the UCB trace, we found 0.02% of the IP packets carrying TCP data to be IP fragments. For USC, the number was 0.05%, and for Bellcore, the number was 0.02%. We ignored these fragments.

Traffic Type	% Packets			% Bytes			% Conversations		
	UCB	USC	BELL	UCB	USC	BELL	UCB	USC	BELL
ftp (ctrl+data)	**12.0**	**5.0**	**18.7**	**36.2**	**10.6**	**54.9**	**2.2**	**1.8**	**4.7**
shell (rcp)	0.2	3.6	1.4	0.4	12.5	4.3	0.2	0.1	0.6
smtp	**11.6**	**3.1**	**12.6**	**11.0**	**1.9**	**10.6**	**54.0**	**29.3**	**65.2**
dc_10	—	3.5	—	—	0.8	—	—	0.8	—
vmnet (bitnet)	10.0	9.1	—	25.4	20.7	—	0.1	1.8	—
uucp	0.2	0.1	0.8	0.4	0.1	1.3	0.3	0.6	2.1
nntp	**11.6**	**36.3**	**9.2**	**15.8**	**44.5**	**15.6**	**22.5**	**44.8**	**4.7**
telnet	**28.0**	**16.6**	**36.3**	**5.5**	**2.3**	**6.5**	**3.2**	**4.9**	**8.4**
rlogin	**15.5**	**5.8**	**18.5**	**2.8**	**0.7**	**3.1**	**1.6**	**1.5**	**4.1**
x11	0.2	5.0	0.4	0.2	2.5	0.1	—	0.3	0.4
ircd	4.6	—	—	1.3	—	—	0.5	0.1	—
finger	1.1	0.4	0.5	0.6	0.2	0.2	14.2	10.0	7.3
domain	0.1	0.1	—	—	0.2	—	0.1	1.8	0.1
other	4.9	11.3	1.6	0.4	3.1	3.1	1.1	2.2	2.4

Table 2: Breakdown of unidirectional TCP traffic, by packets, by bytes, and by conversations.

2.3. Tracing Instrumentation and Packet Loss Rate

The UCB data was collected with a Sun 3 workstation equipped with a microsecond timer [Danzig90]. The resulting time stamp resolution was 10 microseconds. The workstation ran a modified Unix kernel with a circular buffer big enough to hold 128 full-size Ethernet packets. A dedicated user program transferred trace records from this buffer to tape. No packet losses due to buffer overflows were detected during the UCB measurements. The packet loss rate induced by separate stress testing was less than 5% in the worst case.

The USC data was collected using the NNStat program suite [Braden89] on a Sun SparcServer 4/490. The NNStat program uses the Sun *gettimeofday()* system call which has a 20-millisecond resolution. During similar measurements, we estimated the loss rate by sending a Poisson stream of *ping* packets. We observed that 0.6% of these packets were missing from the tape.

The Bellcore data was collected using a Sun 3 workstation augmented with a microsecond interval timer and a single board computer dedicated to collecting and timestamping trace packets. The timestamps have a 10 microsecond resolution. A hierarchical system of double buffering carried the trace records from the single-board computer to tape. No packet loss was detected anywhere in the monitoring system during the Bellcore measurements [Leland91].

2.4. Are the Traces Representative?

Both USC and UCB campuses use mostly UNIX and IBM computing systems. Bellcore uses mostly UNIX systems.[3] We believe that the systems traced are representative of sites currently attached to the Internet, and that our analysis also applies to other sites. However, we recognize that traces collected at other sites might show a different application breakdown than the ones reported here, and we are currently negotiating with other sites for additional data collection to further validate our results.

The breakdown of traffic varies greatly from site to site (see Table 2). However, the *characteristics* of conversations are essentially identical between the three sites, even though the USC trace was collected one year and three months after the others. Furthermore, these characteristics are also shared by two different days of UCB traces, and by a one-day trace and a three-day trace of Bellcore traffic. That is, the distributions of number of bytes transferred, conversation durations, total packets per conversation, and packet sizes are indistinguishable. For legibility, we present only UCB data in the body of the paper. Appendix 1 contains representative figures comparing data from the three sites. Additional data can be found in [Danzig91]. One does need to account for the differences in traffic breakdown when generating the actual sequence of conversations to simulate.

2.5. Traffic Pattern Analyzer

We wrote a traffic pattern analyzer to reduce the raw packet trace data and produce the statistics presented in this paper. One of the first decisions we had to make was how to break up the trace into meaningful units. Should we adopt the *packet-train* model or should we maintain a state machine per TCP connection? We look at these alternatives below.

The *packet-train* model has replaced earlier Markov models of network traffic [Jain86] [Heimlich88] [Gusella90]. In the *packet-train* model, a stream of packets is broken up into *trains*. Two consecutive trains are delimited by a MAIG (maximum allowable inter-car gap). The MAIG is usually chosen to encompass 90 percent of all interarrival gaps. Different researchers have used different

[3] A short glossary of Internet protocols and applications is provided in Appendix 2.

MAIGs, ranging from 500 milliseconds to 50 seconds, depending on the network measured.

In contrast, we divided up the traffic into application-level *conversations*. We define a *conversation* to be a stream of packets travelling between the end points of an association, delimited by a twenty-minute silence; an association is in turn defined as a <protocol, source address, source socket, destination address, destination socket> tuple. A twenty-minute silence is longer than FTP's idle connection timeout value of fifteen minutes. Early on we experimented with a five-minute silence rule. The difference in results was minimal. We could have detected the TCP connection establishment handshakes between a source and destination pair and used them to determine the beginning and end of a conversation. This required maintaining a state machine and associated timers for every live connection. Lack of memory space prevented us from doing so.

In the case of FTP, conversations can subsume multiple TCP connections. We clump several TCP connections into one conversation because each FTP session initiates one FTP-control and zero or more FTP-data connections. We also clumped back-to-back and concurrent FTP sessions between the same source-destination IP-address pair into one conversation.[4]

Since we want to model the characteristics of transport layer traffic in general, independent of TCP itself, we further decided to drop all TCP-specific traffic. We dropped TCP connection establishment packets and all zero-byte packets assuming that these were acknowledgement packets. We also filtered out all retransmitted packets.[5] Retransmitted packets were detected by matching their sequence numbers against those of the last 128 packets from the same conversation. Most retransmitted packets match one recently transmitted within the previous 64 packets. The oldest retransmitted packet detected in the analysis of the traces was at position 104 into the buffer. Since we are throwing away retransmissions, we are also throwing away most of the keep-alive packets, which share a single sequence number. This also means that every now and then we would see a lonesome keep-alive as a conversation transferring a single 1-data-byte packet. We filter out all such false conversations in our analysis. For the Bellcore trace, we further notice that 50% of all NNTP conversations between Bellcore and Rutgers consist of a single 6-data-byte packet. After closer examination, we attribute those conversations to an implementation fault at either Bellcore or Rutgers. Our traffic pattern analyzer filters out all such conversations.

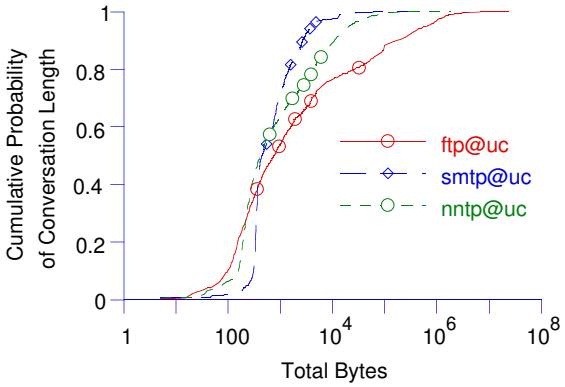
[4]For UCB data, each FTP conversation averages about 4.2 connections consisting of ftp-ctrl and ftp-data connections. For USC data, the average is about 3.5 connections per conversation. For both sets of data, a little over 60% of all FTP conversations consist of only one connection; this is due to the client's side making only the ftp-ctrl connection.

[5]Retransmitted packets accounted for between 0.3% to a little below 3% of all packets belonging to an application.

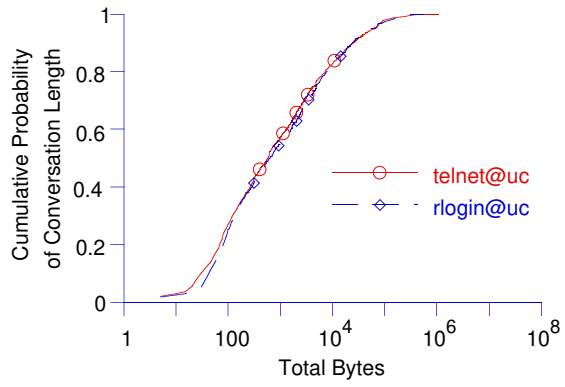
3. Characterization of Application Conversations

Our trace study is divided into two parts. The first part measures applications running on TCP/IP internetworks. The results are presented in this section under five general categories: traffic breakdown, bulk data transfer applications, interactive applications, traffic flow, and wide-area network locality. We are interested in such questions as:

- How does TCP traffic break down into interactive and bulk traffic?
- How "bulky" is the data transferred by bulk applications?
- What are the characteristics of interactive applications in terms of bytes transferred, burstiness, duration, and interarrival time?
- Is traffic flow unidirectional or bidirectional?
- Is there network-pair locality on wide-area networks and how many concurrent conversations are there between such network pairs?



(a) Bulk transfer



(b) Interactive applications

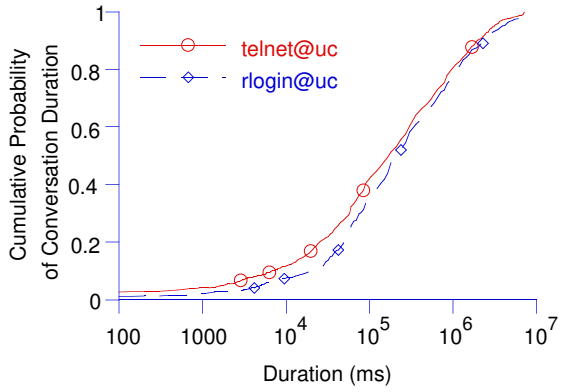
Fig. 1: Total bytes transferred per unidirectional conversation.

The second part of the study, constructing a traffic source model, is presented in Section 4.

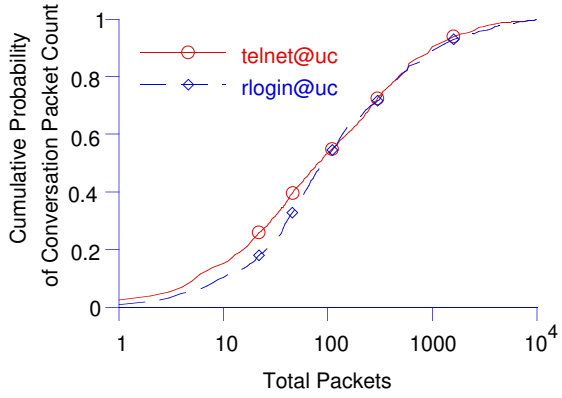
3.1. Traffic Breakdown

For lack of a more accurate model, previous studies that simulate flow control, congestion control, multiple access protocols, and traffic dynamics in general have been forced to assume a rather simple traffic model [Demers89], [Floyd91], [Wilder91], [Will91], [Zhang90], [Zhang91]. These studies either use a continuous bulk transfer or an arbitrary mix of bulk and interactive traffic.

Table 2[6] shows that while TCP traffic does consist of bulk and interactive traffic as commonly assumed, the distributions of number of bytes, packets, and conversations attributed to each application could be more representative. Even though bulk applications send more data than interactive ones, interactive conversations still send 5-10% of network bytes and 25-45% of network packets.



(a) Conversation duration



(b) Packets transferred

Fig. 2: Duration and packets transferred per conversation for interactive applications.

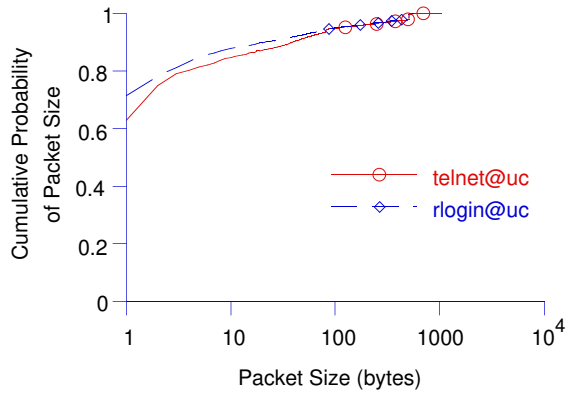
[6]The applications which appear in boldface are the ones we concentrate our study on.

We think it important to realize that interactive applications are responsible for 25-45% of all Internet packets. Simulations that model internetwork traffic as mostly large bulk transfers may overestimate the benefit of mechanisms proposed to improve bulk transfer performance. Most existing studies evaluate the robustness of designs and algorithms under worst case loads, but fail to contrast their performance to that of equally robust designs or algorithms when running under average loads.

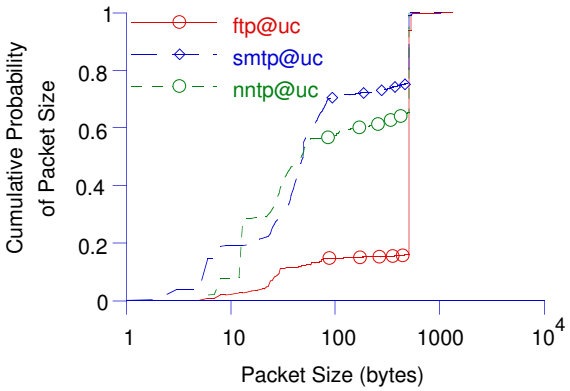
3.2. Bulk Data Transfer

Many simulation studies commonly overestimate the amount of data sent by bulk data transfer applications such as FTP. Transfer sizes usually range from 80K to 2M bytes, or simply continue to the end of the simulation run [Demers89] [Floyd91] [Wilder91] [Zhang90] [Zhang91]. Figure 1a shows that about 75-90% of bulk transfer conversations transfer less than 10K bytes. We think this observation is correlated with the observation made in [Ouster85] that most files are small.

If this is true of Internet source traffic in general, then it should be taken into account in future internetwork



(a) Interactive applications



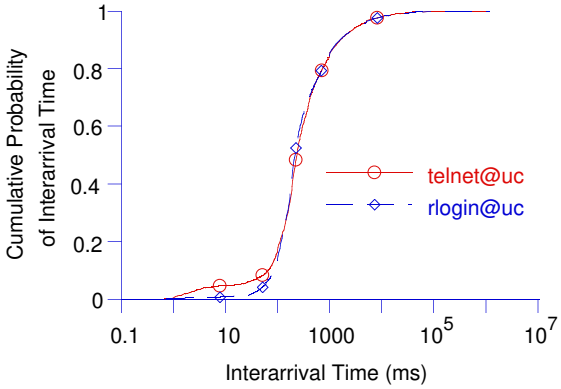
(b) Bulk transfer

Fig. 3: Distribution of packet size by application. Packet sizes reflect only user data without protocol headers.

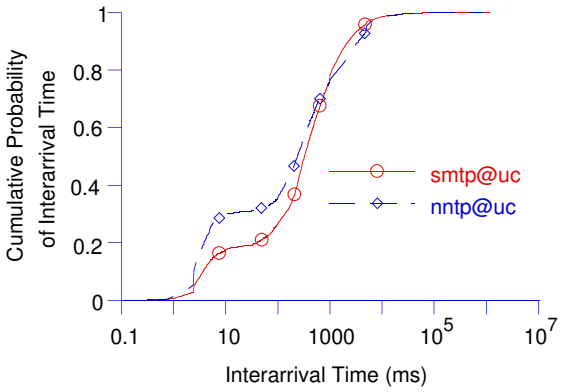
simulations. To the extent that simulated algorithms employ feedback mechanisms (such as congestion or flow control) [Rama90], it is important to know that in most sessions data transfer will complete before any such feedback is received. We believe this observation is important because the emergence of voluminous multimedia traffic will not make existing traffic disappear.

3.3. Interactive Applications

Network flow control and the Maximum Transferrable Unit (MTU)[7] determine, to a great extent, the measured statistics of bulk internetwork traffic. In contrast, Figure 1b and 2a show that about 90% of TELNET and RLOGIN conversations send less than 10K bytes over a duration of 1.5 to 50 minutes. Figure 3a shows that about 90% of TELNET and RLOGIN packets carry less than 10 bytes of user data, which is much smaller than the MTU. Thus interactive applications are more or less unaffected by flow control and MTU size.



(a) Interactive applications



(b) Bulk transfer

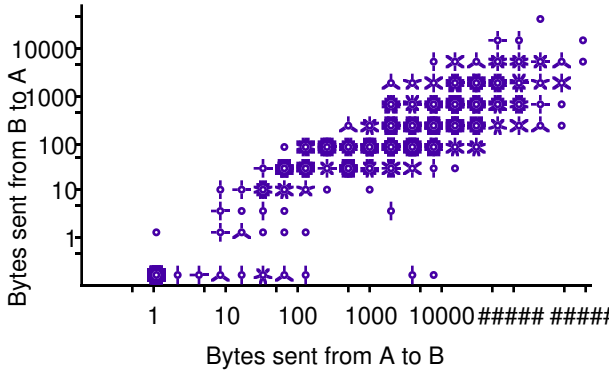
Fig. 4: Distribution of packet interarrival by application.

[7]For historical reasons, wide-area TCP connections still use an MTU of 512 data bytes despite the fact that the NSFNet backbone supports 1500-byte packet.

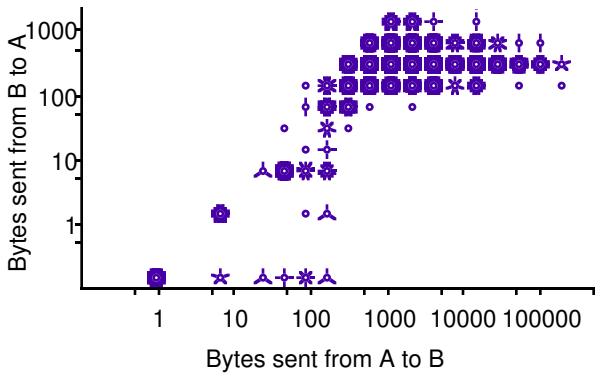
If interactive applications are not affected by network flow control and MTU, then the observed characteristics reflect the true nature of such applications. However, we should not assume that interactive traffic carries less data—Figure 1b shows that 80% of all interactive conversations send as much data as the average bulk transfer conversation—rather, it means that bulk transfer applications send a smaller amount of data than is often assumed.

In most traffic models used in existing simulations or testbed studies, conversations are assumed to last anywhere from 500 seconds, 600 seconds, to "keep on forever" [Demers89] [Mankin90] [Floyd91] [Wilder91] [Zhang91]. Figure 2a shows that the duration of interactive conversations is highly variable. This fact, along with the small number of packets per conversation (see Figure 2b), might influence steady state feedback assumptions, as well as per packet processing time with respect to gateway algorithms.

Finally, our data shows that while interarrival times for bulk data transfers exhibit the packet-train phenomenon, interarrival times for interactive applications should be modeled by a constant plus exponential random time (see Figure 4a). Section 4 describes this phenomenon in more detail.



(a) Telnet



(b) SMTP

Fig. 5: Bidirectionality of traffic flow.

3.4. Traffic Flow

Most simulations on gateway queueing such as [Demers89], [Floyd91], and [Zhang90] have assumed unidirectional data flow. Figure 5 shows that a large percentage of traffic, both interactive and bulk, is bidirectional. Simulations should generate traffic in both directions. Furthermore, Figure 3b and 4b together affirm that many bulk transfer applications contain a request-response phase, which causes a synchronization point where no data is flowing in either direction. In turn, this synchronization point causes classic packet train behavior: a handshake followed by a big burst. For example, NNTP sends a query, waits for a response, and then does a bulk transfer. This behavior may influence congestion and transport mechanisms and should be included in simulation studies of these mechanisms. Small packets, short conversations, and birectional flow all contribute to the traffic dynamics of the internetwork. These characteristics of current internetwork traffic could affect traffic segregation and oscillation findings [Floyd91] [Wilder91] [Zhang91].

3.5. Wide-Area Network Locality

Mogul reports strong locality of reference between pairs of hosts on a local area network [Mogul91]. This locality of reference means that certain hosts communicate more with one another than with other hosts. Does such locality of reference exists between host pairs or network pairs in wide-area internetworks? Figure 6 shows that it indeed occurs. For example, half of UCB telnet conversations are directed to just 10 sites.

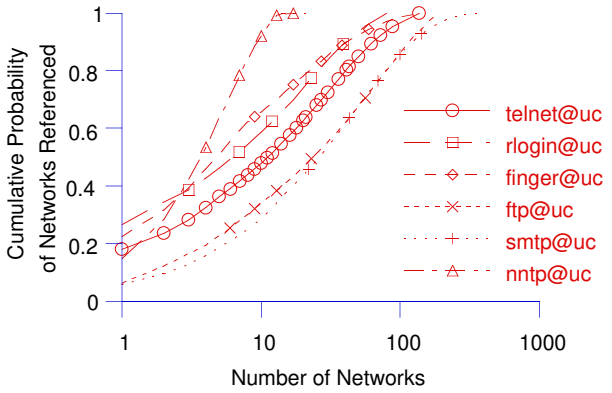
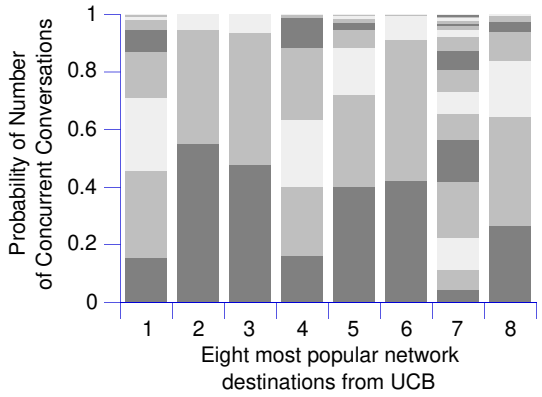
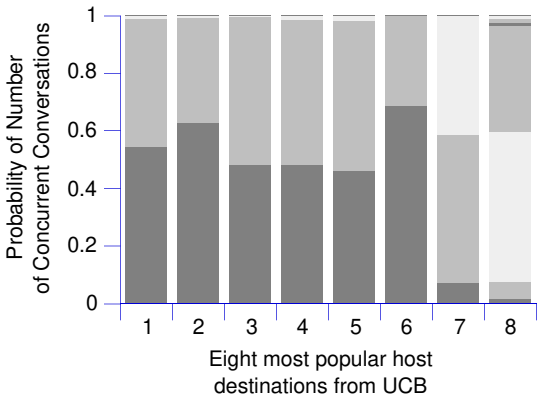


Fig. 6: Number of networks referenced by UCB.

Given network-pair locality on wide-area networks, we want to know how many concurrent conversations run between popular network-pairs. Figure 7a shows the concurrent conversations to UCB's eight most popular destination networks. In Figure 7, each band represents a number of concurrently running conversations. The band at the bottom of a bar represents the probability of finding zero on-going conversations. The next band up represents the probability of finding one on-going conversation. The third represents the probability of finding two simultaneous conversations, and so on. The seventh bar in Figure 7a shows that it is very probable to find more than two concurrently running conversations between these two networks. However, this particular bar represents the traffic between UCB and Lawrence Berkeley Laboratory which are located several hundred yards from each other. The other network pairs do not show as many concurrent conversations as do this pair.



(a) Network pair



(b) Host pair

Fig. 7: Number of concurrent conversations for the eight most popular network and host pairs coming out of UCB (see section 3.5).

Given that we frequently find concurrent conversations between popular network pairs, how often do we find concurrent conversations between host pairs on wide area networks? Figure 7b shows that it is unlikely with the present Internet traffic, but this may change in the future. Eight of the ten most frequently referenced host pairs correspond to NNTP exchanges. The eighth host-pair in Figure 7b frequently exhibits two or three concurrent conversations. This host pair connects an UCB host to an Andrew host at CMU; we suspect that we captured traces of an experiment with the Andrew File System. The seventh host-pair shows a site that frequently has two simultaneous conversations. Nearly all of these are simultaneous NNTP conversations, and reflects that NNTP transfers news

messages in only one direction per TCP connection.[8] From this measurement of current traffic, we can say that there are not many concurrent conversations between host pairs.

4. Source and Internet Traffic Model

This section describes a source model for generating a random but realistic sequence of traditional internetwork conversations. Because 6 of the 35 applications we identified in our traces account for more than 96% of the bytes transmitted, we model only these applications. They are FTP, SMTP, NNTP, VMNET, TELNET, and RLOGIN.

We must first solve a difficult problem: how to specify the matrix of sites between which application traffic flows. We call this the traffic matrix. This is hard because certain applications reference more sites than do others (see Figure 6). For example, we see that half of UCB TELNET conversations are directed to just 11 sites, with the other half referencing over 100 sites. NNTP references just 11 sites for the whole trace. Half of SMTP conversations reference over 50 sites, and the other half reference 300 other sites. Overall, half of UCB's conversations are directed to just 17 sites. Specifying the traffic matrix is made more difficult because the application mix changes from site to site. We are pursuing an algorithm to generate internetworks with representative traffic patterns. The rest of this section assumes this has been done.

Assuming we are given the traffic matrix, there are four steps to generating a sequence of realistic internetwork conversations for a set of sites. First we must determine when to establish the next conversation for a given application. Second, depending upon whether the application is bulk or interactive, we must either select the amount of data exchanged in each direction or the conversation's duration. Third, we must choose an appropriate destination host for this conversation, and fourth, we must choose the rule that determines the sequence of packets that this conversation sends. Below, we describe these steps.

We choose the application type of a site's next conversation from the site's traffic breakdown. This is not as obvious as it seems, because conversations depend on one another. For example, one is more likely to send mail to a site shortly after fingering it than if one had never referenced it before. However this effect is not particularly pronounced in the data. We found that the types of successive conversations are independent, although we did not investigate correlations on the sequence of conversation types between a specific network pair or host pair. Hence, we model arrivals of new application conversations as time-varying Poisson processes with site and time-of-day dependent rates. For example, Figure 8 plots measured arrival rates of UCB conversations for several applications. By making the rate depend on time of day, it is possible to model site-specific configurations. For example, at UCB, VMNET runs just four times per day at specified times, while at USC VMNET runs on demand.

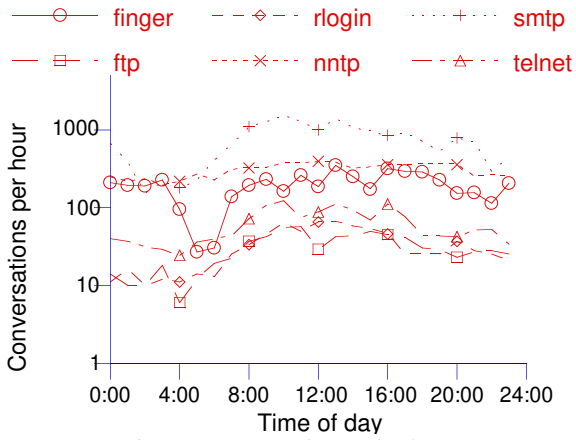


Fig. 8: Conversation arrival rate.

The next step depends on whether the conversation is interactive or bulk. If it is bulk, we choose the number of bytes transmitted in each direction from the joint distribution of bidirectional bytes transmitted. We illustrate one such distribution in Figure 5b for SMTP. This figure plots the larger side of a conversation on the x-axis and the smaller side on the y-axis. Bigger, darker marks indicate higher likelihood. If the conversation is interactive, we choose its duration from the distribution of duration. We illustrate one such distribution in Figure 2a. We indicate a distribution of duration for bulk protocols in Figure 2b, but do not employ it in the model because the duration of a bulk transfer depends on network bandwidth and flow control, rather than the traffic sources.

If required, one can spend a bit more effort and model the number of items transferred by bulk applications, such as the number of news articles exchanged during an NNTP conversation. For example, given the distribution of the number of items transferred and the distribution of the number of bytes in an item, we can model the synchronous interactive phase inherent in all four bulk applications, during which file names, commands, and article numbers are exchanged. These interactive phases act as synchronization points. At the start of one of these phases, no outstanding packets exist between end points. Hence, there is at least one round trip time between bulk exchanges.

The third step is to choose the destination site for this conversation. This is done from the traffic matrix discussed in the second paragraph of this section.

The fourth and final step, specifying packet arrival times and sizes, depends on the application. For bulk transfers, packet sizes and interarrival times depend on physical characteristics of the network, the bidirectional distribution of bytes transferred, and the distribution of items transferred (if the synchronous nature of bulk transfer is being modelled). While their packet interarrival times

[8] In the future, we may decide to clump several TCP connections from one NNTP session, as we have done for FTP. So doing will accentuate our observations on traffic bidirectionality and the number of concurrent conversations between host pairs.

depend on the network, their packet sizes depend on the application. During bulk transfer, packet sizes are a network MTU followed, if necessary, by a final smaller fragment. During control exchanges, packet sizes are smaller, corresponding to file names and commands; it is necessary to draw their packet sizes from the measured distributions (see Figure 3b).

In contrast to bulk traffic, packet interarrival times of interactive traffic depends on the user. Users' keystrokes generate "byte-sized" packets with a constant plus exponential interarrival time. The destination process sends a response for every packet that it receives; occasionally it returns a large response (see Figure 3a). A close inspection of the interarrival time of TELNET and RLOGIN traffic presented in Figure 4a reveals that 10% of the time, interarrival times are less than 100 milliseconds. These short interarrival times occur for two reasons. First, when the destination sends a response greater than a network MTU, its packets arrive in rapid succession. These back to back MTUs account for roughly a quarter of the interarrival times less than 100 milliseconds. Second, network queueing and operating system unresponsiveness can deliver single key strokes to the destination in rapid succession. Back to back single data byte packets constitute roughly three quarters of these short interarrival times.

The network traffic matrix and this sequence of four steps can be used to create a realistic source of internetwork conversations. We are in the process of creating a tool to automatically generate internetworks and sequences of conversations to drive internetwork traffic simulations. We are also investigating techniques to simulate much larger internetworks than is currently possible. In the next section we discuss one possible application of such a tool.

5. Applying the Traffic Characterizations

Since we are not suggesting that algorithm robustness testing should use our workload model in place of worst-case scenarios, just what good is a tool for generating realistic internetwork traffic? This section describes one problem that needs a realistic internetwork traffic model.

The problem of multiplexing application datagram traffic over wide-area virtual circuits reappears with the advent of high-speed Asynchronous Transfer Mode (ATM) networks. Assuming the existence of a reservation scheme for handling the requirements of multimedia traffic [Ferrari90], we still have to accommodate the dynamics and requirements of traditional datagram traffic. When a datagram arrives at an ATM gateway, it needs to be routed onto an appropriate virtual circuit. If such a circuit doesn't exist, data transmission must wait until one is established. On the other hand, idle virtual circuits consume resources inside the ATM network. We want to find ways to multiplex TCP conversations over ATM virtual circuits that provide adequate performance while making efficient use of network resources.

We need to trade the performance costs of establishing new virtual circuits with the resource utilization advantages of closing idle circuits. Evaluating this tradeoff requires a good, average case internetwork traffic source model. With such a model we could decide how to map a set of TCP conversations onto a possibly smaller set of ATM virtual circuits, choose the queueing discipline for multiplexing datagrams onto these virtual circuits, and arrive at a timeout algorithm for reclaiming idle virtual circuits.

No previous model of wide-area traffic is appropriate for this study. To evaluate the performance of different mapping schemes, we need a realistic internetwork traffic matrix. Without accurate knowledge of application mix and behavior, we cannot predict the effect of multiplexing several different TCP conversations through a single ATM virtual circuit. To evaluate timeout schemes, we need the distribution of conversation durations and conversation interarrival times.

We believe there are other cases where a detailed characterization of applications as presented in this paper will be required. Even for studies that aim to prove only the robustness of new designs or algorithms, using our model can show how new designs or algorithms perform on the common case.

6. Implications and Conclusions

The application characteristics we identify contradict the following commonly held beliefs regarding current wide-area traffic:

- Bulk sources transfer large amounts of data per conversation.
- Bulk sources send large packets in only one direction.
- Interactive sources send small packets in one direction, and receive echoes of comparable size in the opposite direction.
- Internetwork traffic can be modeled by either a Poisson interarrival process or a packet-train model alone.

Addressing these myths in order, we have shown that:

- Eighty percent of the time, classic bulk transfer application such as FTP transfer less than 10 kilobytes per conversation. Other applications commonly categorized as bulk traffic sources, such as SMTP and NNTP, transfer even smaller amounts of data (see Figure 1a).
- Traffic generated by FTP, SMTP, NNTP, and VMNET is strongly bidirectional. Furthermore, SMTP and NNTP send as many small packets as large packets (see Figures 5b and 3b).
- Interactive applications routinely generate 10 times more data in one direction than the other, using packet sizes ranging from 1 byte to 512 bytes (see Figures 5a and 3a).
- Interactive packet interarrivals closely match a constant plus exponential distribution (see Figure 4a).

We are continuing work on tools to create wide-area network traffic based upon our characterizations. We plan to study various algorithms' responses to average case data, especially flow control and congestion control algorithms

whose robustness, but not average case behavior, was evaluated in previous studies. We feel that there is more work to be done in understanding traffic reference patterns, and believe that a better understanding of these could impact the design of future networks.

Acknowledgements

We would like to especially thank Deborah Estrin for her contributions to the project. D. Ferrari, S. Keshav, S. Morgan, J. Mogul, M. Sullivan, and the reviewers provided useful suggestions. For the data collection at Berkeley, B. Prabhakar was instrumental in collecting the traces, and C. Frost provided access to the gateway Ethernet. At USC, J. Pepin, M. Brown, and R. Kaplan provided access to the campus network, and J. Conti assisted in collecting the traces. At Bellcore, D. V. Wilson gathered the traces with his Ethernet monitor and made them available to us.

References

[Braden89] Braden, B. and DeSchon, A.L., *NNStat: Internet Statistics Collection Package—Introduction and User Guide*, USC-ISI, Dec. '89.

[Cáceres89] Cáceres, R.. *Measurements of Wide-Area Internet Traffic*. UCB/CSD 89/550, University of California, Berkeley, Dec. '89.

[Danzig90] Danzig, P.B. and Melvin, S., "High Resolution Timing with Low Resolution Clocks and a Microsecond Timer for Sun Workstations," *ACM OS Review*, 24:1, Jan '90, pp. 23-26.

[Danzig91] Danzig, P.B., Jamin, S., Cáceres, R., Mitzel, D.J., and Estrin, D., *Characteristics of Wide-Area TCP/IP Conversations*, USC-TR-91-4, Jan '91.

[Demers89] Demers, A., Keshav, S., and Shenker, S., "Analysis and Simulation of a Fair Queueing Algorithm," *ACM SIGCOMM '89*, pp. 2-12.

[Floyd91] Floyd, S. and Jacobson, V., "Traffic Phase Effects in Packet-Switched Gateways," *Comp. Communication Review*, April 1991, pp. 26-42.

[Feldmeier88] Feldmeier, D., "Improving Gateway Performance with a Routing-Table Cache," *Proc. IEEE INFOCOM '88*, March '88.

[Ferrari90] Ferrari, D. and Verma, D.C., "A Scheme for Real-Time Channel Establishment in Wide-Area Networks," *IEEE JSAC*, 8:3, April '90.

[Gusella90] Riccardo Gusella. "A Measurement Study of Diskless Workstation Traffic on an Ethernet," *IEEE Transactions on Communications*, Sep. 1990.

[Heimlich89] Heimlich, H., "Traffic Characterization of the NSFNET National Backbone," *USENIX Conf. Proc.*, Winter '89.

[Jain86] Jain, R. and Routhier, S., "Packet Trains—Measurements and a New Model for Computer Network Traffic," *IEEE JSAC*, Sep. '86.

[Jain89] Jain, R., *Characteristics of Destination Address Locality in Computer Networks: A Comparison of Caching Schemes*, DEC-TR-592, Feb. '89.

[Jain90] Jain, R., *Myths about Congestion Management in High-Speed Networks,* DEC-TR-724, Oct '90.

[Mankin90] Mankin, A., "Random Drop Congestion Control," *Proc. ACM SIGCOMM '90*, pp. 1-7.

[Mogul91] Mogul, J.., "Network Locality at the Scale of Processes," *Proc. ACM SIGCOMM '91*.

[Leland91] Leland, W.E. and Wilson, D.V., "High Time-Resolution Measurement and Analysis of LAN Traffic: Implications for LAN Interconnection," *Proc. of INFOCOM '91*.

[Ouster85] Ousterhout, J.K. et al., "A Trace-Driven Analysis of the UNIX 4.2 BSD File System," *Proc. of the 10th ACM SOSP*, Dec 1-4, 1985, pp. 15-24.

[Rama90] Ramakrishnan, K.K. and Jain, R., "A Binary Feedback Scheme for Congestion Avoidance in Computer Networks," *ACM TOCS*, 8:2, May 1990, pp. 158-181.

[Wilder91] Wilder, R., Ramakrishnan, K.K., and Mankin, A., "Dynamics of Congestion Control and Avoidance of Two-Way Traffic in an OSI Testbed," *Comp. Comm Review*, 21:2, April 1991, pp. 43-58.

[Will91] Williamson, C.L. and Cheriton, D.R., "Load Curves: Support for Rate-Based Congestion Control in High-Speed Datagram Networks," *Proc. ACM SIGCOMM '91*.

[Zhang90] Zhang, L., "Virtual Clock: A New Traffic Control Algorithm for Packet Switching Networks," *Proc. of SIGCOMM '90*, pp. 19-29.

[Zhang91] Zhang, L., Shenker, S., and Clark, D.D., "Observations on the Dynamics of a Congestion Control Algorithm: The Effects of Two-Way Traffic," *Proc. ACM SIGCOMM '91*.

Appendix 1

Comparative Data from the Three Sites

In the following figures, curves labelled *uc* represent UCB data, ones labelled *bc* represent Bellcore data, and ones labelled *sc* represent USC data.

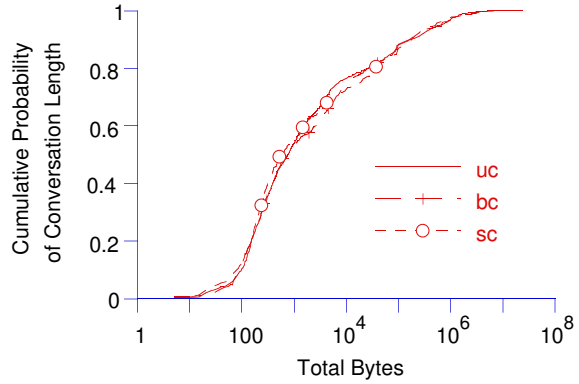


Fig. A: Total bytes transferred per unidirectional FTP conversation.

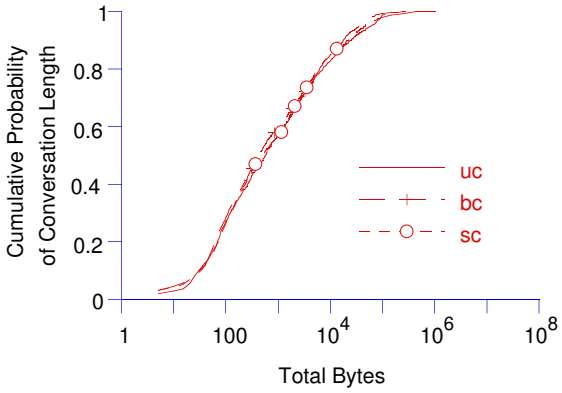


Fig. B: Total bytes transferred per unidirectional TELNET conversation.

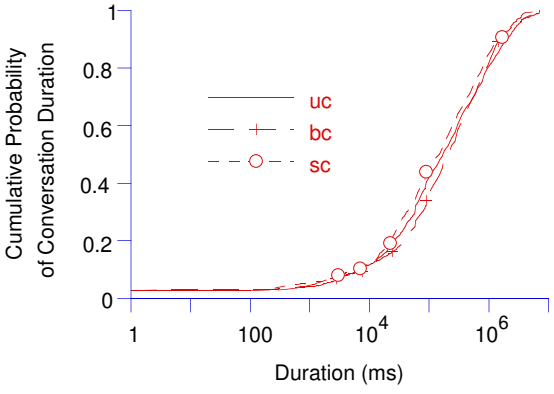


Fig. C: Duration of TELNET conversations.

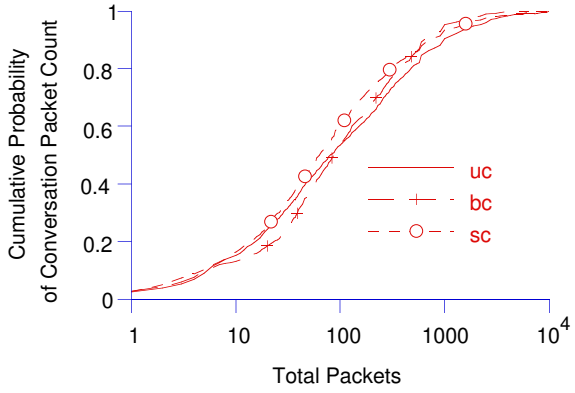


Fig. D: Packets transferred per TELNET conversation.

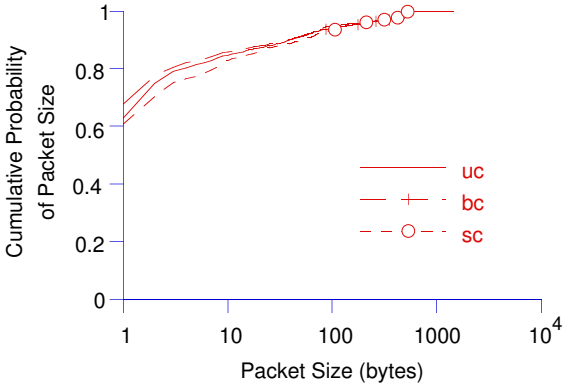


Fig. E: Distribution of TELNET packet sizes.

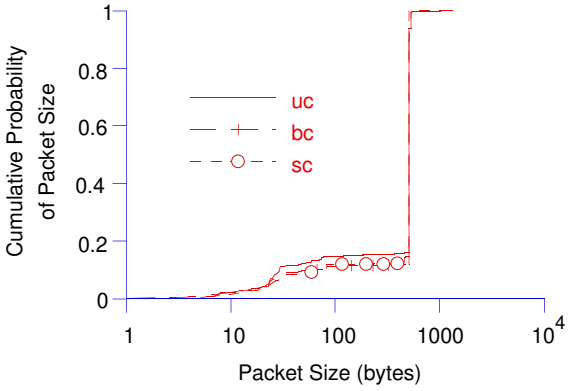


Fig. F: Distribution of FTP packet sizes.

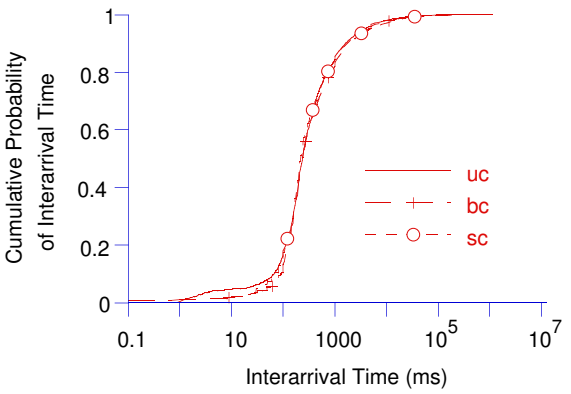


Fig. G: Distribution of TELNET packet interarrivals.

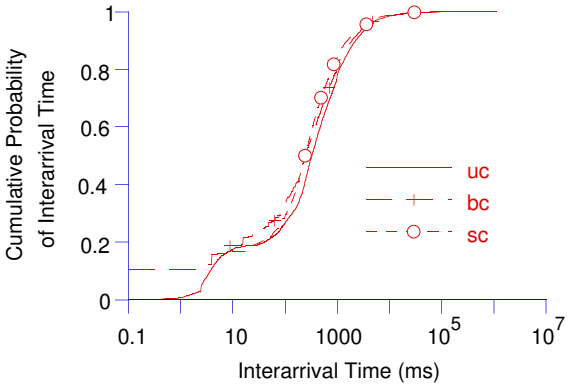


Fig. H: Distribution of FTP packet interarrivals.

Appendix 2

Glossary of Internet Protocols and Applications

DC_10	Cadre Teamwork Mailbox 10.
DNS	Domain Name Service, host name resolution protocol.
DOMAIN	Domain Name Service.
FINGER	User information query application.
FTP	File Transfer Protocol.
ICMP	Internet Control Message Protocol.
IP	Internet Protocol, a network layer datagram protocol.
IRCD	Internet Relay Chat Program Server, a tele-conferencing application.
NTP	Network Time Protocol.
NNTP	Network News Transfer Protocol.
RLOGIN	Remote login application.
ROUTE	Routing information exchange protocol.
SHELL	Remote shell application, often used for remote copy (rcp) operations.
SMTP	Simple Mail Transfer Protocol.
TCP	Transmission Control Protocol, a reliable transport layer protocol.
TELNET	Remote terminal application.
UDP	User Datagram Protocol, an unreliable transport layer protocol.
UUCP	Unix to Unix Copy Program, used for mail, news, and file transfer.
VMNET	A method of running the RSCS protocol (usually from IBM mainframes running VM) on top of TCP; it is used to handle a majority of the BITNET backbone traffic.
X11	X window system.