Statistics of Mixed Data Traffic on a Local Area Network

W. T. MARSHALL and S. P. MORGAN

AT&T Bell Laboratories
Murray Hill, New Jersey, U.S.A.

ABSTRACT
We have analyzed a week's worth of data traffic on a DATAKIT* Virtual Circuit Switch network at AT&T Bell Laboratories. The network includes 5 nodes connecting 22 host computers and 226 terminals, with trunks to nodes elsewhere at Bell Laboratories. Users are predominantly researchers using the UNIX† operating system via teletypewriter terminals and diskless work stations at 9.6 kbs. Comparable fractions of the traffic are generated by terminal-to-host calls, by indirect logins, by interactive remote command executions, and by host-to-host file transfers. We display histograms representing the distributions of interarrival times and call lengths associated with the various types of calls, and the distributions of transmission bursts in individual calls. We characterize typical distributions by their means and coefficients of variation, and propose a model for time-sharing traffic which depends on a relatively small number of parameters and statistical distributions.

1. INTRODUCTION
Few detailed measurements of data traffic on local area networks have been published. Little appears to be known about the statistics of such traffic, beyond the conventional wisdom that data traffic is bursty and that the capabilities of networks, terminals, and hosts are evolving so fast that every case is different. However, even a snapshot of a particular system, if carefully interpreted, can give some guidance to network designers, and can suggest a model for traffic measurements on other systems.

We report measurements of a week's worth of traffic on a DATAKIT* Virtual Circuit Switch network at AT&T Bell Laboratories. The host computers (DEC VAX-11/780's and 750's) run the UNIX† time-shared operating system and are used by researchers for tasks such as prototype software development, graphics, text editing, and numerical computation, as well as data storage. Terminals run at an access speed of 9.6 kbs and include screen teletypewriters as well as diskless work stations such as the TELETYPES® 5620 Dot-Mapped Display. A UNIX/DATAKIT user can connect to a host directly from a terminal, or indirectly through one or more other hosts. He or she can execute a single interactive command on a remote host and return automatically to the original host. In addition, users can routinely cause files to be transferred from one host to another.

Data traffic measurements can be either user-oriented or network-oriented. Examples of user-oriented measurements would be timestamped records of successive characters sent from and received by a particular terminal, or the character counts and transmission times of host-to-host file transfers. Examples of network-oriented measurements would be counts of the numbers and lengths of packets passing various points in the network during given intervals of time. The relationship between user-level traffic and network-level traffic depends on the network architecture and protocols.

The easiest traffic data to obtain are mean values, such as number of characters transmitted per hour. However, for predictions of detailed network behavior much more complete load statistics are needed. If the interarrival times of messages are independent and exponentially distributed, and if the message lengths are also independent and exponentially distributed, then a large number of theoretical results are available. If one or both of the distributions are not exponential, the queueing problem must be treated by more complicated analysis, approximation, or simulation. Measurements of real traffic are therefore of substantial importance.

User-oriented traffic measurements including statistical distributions as well as mean values began with Fuchs and Jackson's measurements [1] of low-speed, half-duplex terminal traffic in the late 1960's. In a series of papers, summarized in 1981, Pawlita [2] has reported the statistics of terminal traffic generated by a number of different user populations using half-duplex terminals at speeds from 200 b/s to 4800 b/s. However, there are no published measurements for full-duplex terminals, for work stations, or for speeds of 9600 b/s.

Network-oriented traffic measurements include Shoch and Hupp's measurements [3] on the original Ethernet at Xerox PARC. These authors give histograms of packet length and interpacket arrival time, and they estimate the ratio of total overhead bits to user data bits on their network. However they do not break their traffic down in as much detail as we propose to do here.

Section 2 of this paper describes the configuration of our network and the nature of the per-call data collected by software monitors in each switching node. Sections 3 and 4 show qualitative features of the switching and packet traffic as a function of hour of day. Section 5 includes holding-time and packet-count histograms for different types of calls, and also samples of detailed character counts and timings for some individual calls.

In Section 6 we propose a traffic model for the Bell Laboratories DATAKIT population. We argue that our traffic can be represented as consisting of terminal-like calls and file transfer calls. We characterize the statistical distributions that are relevant to our model by their means $T$ and coefficients of variation $C_v$, where $C_v$ is the ratio of the standard deviation to the mean. These parameters can go directly into two-moment approximations for the behavior of queueing networks with non-Poisson arrival processes and nonexponential service time distributions [4]. Furthermore, if one does not wish to use the means and standard deviations measured for the Bell Laboratories population, it is apparent what measurements would be required to get the corresponding parameters for a different population.
In this paper we do not fit specific functions to the empirical statistical distributions. Most of the empirical distributions have $C_1$ substantially greater than unity; that is, they have longer tails than an exponential distribution. Some of the histograms, especially those corresponding to message interarrival times, look as if they could be well fitted by a lognormal distribution or a mixture of two lognormals; others, especially those having to do with work-station traffic, are irregular or represent samples too small to be definitive. More sophisticated statistical analyses of data traffic may be desirable in the future, but for the present it appears most important to publish the existing measurements, and to encourage the measurement of other systems.

2. NETWORK CONFIGURATION AND MEASUREMENTS

A DATAKIT network [5] consists of terminals and hosts connected to one or more virtual circuit switches or nodes. The network that we studied is shown in Fig. 1. The nodes are connected to each other and to other nodes at the same location by 1.7 Mb/s trunks. They are connected to nodes at other Bell Laboratories locations by 56 kb/s trunks. During the reference week, the network of Fig. 1 supported 22 hosts, 226 terminals, 10 dial-in lines, and 3 dial-out lines.

Fig. 1 Network configuration.

Raw traffic data are collected as follows. A program running in each node controller collects all call setup and teardown records, as well as cumulative packet counts for each call in progress approximately every 10 minutes. (A DATAKIT packet is 16 bytes long. For slow input devices, a packet will contain fewer than 16 bytes of data and will be padded with null bytes.) A program running on one of the hosts calls up each controller and receives the data collected.

The output includes three types of records, each time-stamped to the nearest second. Call-setup records identify the circuit endpoint of the originator and display the “dialstring”. The dialstring contains the destination name and service requested, and the source name, including the user identification of the caller. Packet-count records identify two circuit endpoints and give the cumulative packet count in each direction. Call takedown records identify two circuit endpoints and give the total packet count in each direction. A complete record of a call consists of one call setup record, zero or more packet-count records, and one call takedown record, all of which can be associated with each other by the circuit endpoints.

The present study is based on the week of November 7-13, 1983. During this time no special precautions were taken except to hold the hardware configuration constant, so that we could associate each circuit endpoint with a unique terminal, host, or trunk. In addition, because the dialstring for host-to-host calls includes the command name, it was possible to distinguish the characteristics of traffic generated by different types of commands. We inspected every call placed during the week and established the following classification.

1. Terminal calls. These calls connect a terminal directly to a host computer. Teletypewriter and work-station calls were separated into two subcategories.

2. Remote login calls. These calls between hosts logically attach a terminal to a destination, and are expected to have characteristics similar to terminal calls.

3. Interactive remote command executions. These host-to-host calls are characterized by short holding times and the transfer of small numbers of packets.

4. Host-to-host file transfers. These include not only obvious file-transfer commands but also commands that send a file to a host that serves an output device such as a printer.

3. SWITCHING TRAFFIC

Fig. 2 shows 10-minute average switching rates during a typical day on one of the controllers. A “switch” is either a setup or a takedown; that is, 100 completed calls would correspond to 200 switches. The figure, although quite spiky, shows the familiar double-humped shape with a lunch dip. Fig. 3 shows hourly averages of the number of virtual circuits connected through the controller of Fig. 2.

Fig. 2 10-minute average switching rates.

Fig. 3 Average circuit occupancy.
The network controllers are very lightly loaded. During the busiest 10 minutes of the week, the busiest controller is utilizing less than 4% of its processing capacity, and a similarly small fraction of its memory.

A further observation, which results from a more detailed breakdown of our data, is that the ratio of host-to-host to terminal-to-host calls is much larger than the ratio of host-to-host to terminal-to-host circuits. The reason is that terminal-to-host calls typically hold a circuit for a much longer time than host-to-host interactions.

Fig. 4 is a log-histogram of interarrival times of switching requests at a typical controller. The plot represents merged data for the 5 busiest hours of the week. All of the log-histograms for switching arrivals, including total calls and terminal-to-host calls separately, tend to be bimodal. There is no numerical difficulty in fitting a mixture of two lognormals to the empirical distributions. However, we have not yet made any attempt at interpretation.

4. PACKET TRAFFIC

Fig. 5 is a bar plot of the total one-way packet traffic on a particular trunk during a typical day (10-minute averages). No two such plots are alike in detail; spikes corresponding to file transmissions can and do occur at any time of the day or night. We observed, however, that sustained average rates in excess of 400 packets per second for as long as 10 minutes were extremely rare. Such a rate is only about 4% of the capacity of a 1.7 Mbit/s trunk.

The traffic due to terminal calls alone, which might be of interest in an environment where users interact with a single host and rarely transfer files, is much less spiky than traffic including file transfers. Terminal packet traffic shows the characteristic two-humped profile with a lunch dip, and the number of packets sent from terminal to host is roughly half the number of packets sent from host to terminal.

5. CHARACTERISTICS OF DIFFERENT TYPES OF CALLS

As described in Section 2, setup and teardown times are recorded for each call, and cumulative packet counts for each continuing call are recorded by a polling program at approximately 10-minute intervals. For terminal calls and remote logins, frequently there are polling intervals during which no packets are transmitted. We can join together all the contiguous polling intervals in which any packets are transmitted and call these the active segments of a given call. If polls were taken more frequently, one could vary the definition of active segment, but it is not clear how much additional insight would be gained.

From the present data, for each call we know the lengths of the active segments and the total holding time, as well as the packet counts in each direction for each active segment. From these data we can calculate the mean number of packets transmitted in each direction per active second, as well as the ratio of mean active seconds to mean holding seconds per call. The results are shown approximately in Table 1.

![Log-histogram of interarrival times of switching requests](image1)

**Fig. 4 Interswitch arrival distribution.**

**TABLE 1. MEAN VALUES OF CALL PARAMETERS**

<table>
<thead>
<tr>
<th></th>
<th>T'type-writer</th>
<th>Work station</th>
<th>Remote login</th>
<th>Remote exec</th>
<th>File transfer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Holding time</td>
<td>150 min</td>
<td>150 min</td>
<td>45 min</td>
<td>120 sec</td>
<td>50 sec</td>
</tr>
<tr>
<td>Active segment</td>
<td>30 min</td>
<td>45 min</td>
<td>20 min</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Active/holding ratio</td>
<td>0.4</td>
<td>0.6</td>
<td>0.6</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Terminal packets/active second</td>
<td>0.8</td>
<td>1.1</td>
<td>1.5</td>
<td>1.3</td>
<td>12^</td>
</tr>
<tr>
<td>Host packets/active second</td>
<td>1.7</td>
<td>1.9</td>
<td>2.4</td>
<td>2.5</td>
<td>45^</td>
</tr>
</tbody>
</table>

* Forward direction
+ Backward direction

A. Terminal traffic

Holding times average about 2.5 hours, while the average active segment is about 30 minutes for teletypewriters and 45 minutes for work stations. (The difference in mean active segments may be related to the fact that some work-station users run a program that continually displays the load on the host. Such a work station would never show an inactive interval by our definition.) Typical log-histograms of holding times and active segments are shown in Figs. 6 and 7. The log-histogram for holding time falls off sharply above $2^{15}$ seconds ($\approx 9.1$ hours), suggesting that many people turn their terminals off only at the end of the day. The ratio of total active time to holding time is between 40 and 60%. Perhaps in a more structured environment, the ratio of active time to holding time would be higher.

The average packet rate from host to terminal during an active segment is a little less than 2 packets per second, while the average packet rate from terminal to host is about 1 packet per second. Interactions via remote login are similar to but more concentrated than interactions between a terminal and its immediate host (shorter active and holding times, but more packets per second). Interactive remote command executions...
differ from terminal-to-host calls in that the mean holding times are much shorter, being only a couple of minutes, and the mean packet rates per second are a little larger than for terminal-to-host calls.

![Graph 6: Work-station holding times.](image)

Fig. 6 Work-station holding times.

![Graph 7: Work-station active segments.](image)

Fig. 7 Work-station active segments.

To understand the relationship between character and packet traffic in terminal calls, we used a hardware monitor to observe the character traffic produced by users during individual terminal sessions. The monitor records every character transmitted in either direction between the terminal and the network, timestamped to the nearest millisecond. Altogether, 12 different users were monitored for a total of just under 20 hours. This sample is not really large enough for statistical analysis. Nevertheless the data do indicate some differences among the different types of terminals and users.

We find that a technical user of a teletypewriter terminal generates about 1 character per active second from the keyboard, whereas a professional typist engaged in word processing sends bursts of varying length to the host, and the host responds with bursts of varying length, but not on an individual keystroke basis. It turns out, however, that the traffic between a work station and a host is dominated by downloads of programs from the host into the terminal, and by transfers of bitmaps from the terminal screen to hard-copy devices such as laser printers. The end result is that overall character rates between a work station and a host are an order of magnitude higher than between a teletypewriter and a host. We have observed average rates as high as nearly 15 characters per active second from work station to host and 3 times as much from host to work station, again depending strongly on what the user is doing.

The observation that a teletypewriter generates about 1 character per active second agrees with measurements by Fuchs and Jackson [1] and by Pawlita [2]. The average number of characters returned to the teletypewriter by the host depended for them, as it does for us, on what the users were doing. No previous results have been published for work station terminals.

The flow of packets during a terminal session was not recorded directly but was calculated from the flow of characters by applying the DATAKIT packetizing algorithm, as follows: A "packetizing clock" ticks every 16 2/3 ms (60 times per second). A packet is begun when the first character arrives, and is closed after the 16th character or the third clock tick, whichever arrives first. If the packet does not contain 16 characters by the third tick, the remaining bytes are padded with nulls. For terminal calls, the ratio of packets-from-host to packets-from-terminal is smaller than the ratio of characters, since the host sends to the terminal at line speed or nearly so, and thus there are more characters on average in a packet from the host than in a packet from the terminal. For the same reason, the markedly higher character rates of work stations are reflected in only modestly higher packet rates.

As might have been expected, packet rates from the monitored sessions are comparable to but somewhat higher than the much-quoted averages. This is understandable because the monitored sessions were relatively short and users were aware of the monitoring. The network data are probably more representative of long-term averages.

Detailed analyses of user-computer interactions have generally followed the model introduced by Fuchs and Jackson [1] for low-speed, half-duplex teletypewriter terminals. In this model, each time-sharing session is described as a sequence of contiguous, nonoverlapping dialog segments identified as user think time, user input time, compute time, and computer output time. When characters are being transmitted, the variability of sending rate is described by breaking up the transmission into "user bursts" or "computer bursts" separated by interburst intervals. In full-duplex transmission, characters can be sent synchronously from terminal to host and from host to terminal, and automatic separation of a session into a sequence of disjoint intervals defined as for the half-duplex model is difficult. We have accordingly considered the streams of characters from terminal to host and from host to terminal to be separate processes.

It is still useful to treat each process as a series of bursts separated by interburst intervals. The distribution of interburst intervals between characters in any one stream contains relatively few intervals in the neighborhood of 25 ms. A human user does not strike successive keys as close together as 25 ms, while if a work station sends several bytes as a result of one keystroke, they go at access line speed (1 character per millisecond). Similarly, host responses go at line speed, with pauses due to timesharing which generally exceed 25 ms. We have accordingly defined a burst from either the terminal or the host as a sequence of characters separated by intervals of not more than 25 ms. This definition is obviously somewhat arbitrary.
User bursts from asynchronous teletypewriters are invariably single characters. By far the largest number of host bursts are one or two characters, representing single-character echoes or carriage-return-line-feed combinations. Accordingly we separate host bursts into one-or two-character echoes and longer responses.

For the TELETYPE 5620 workstation, user bursts can be as short as 4 bytes, although there is a strong preference for multiples of 6. Host responses to keyboard input can also be as short as 4 bytes, although the greater number contain 9 or 10 bytes. We accordingly define host "echoes" as bursts of 10 bytes or less, and distinguish them from host responses of more than 10 bytes; "echoes" are not paired with user bursts on a one-for-one basis. Histograms of burst lengths and interburst intervals for a particular work-station session are shown in Fig. 8. In this figure, the spikes in the interarrival distributions at 2000 ms represent a program which reports the load on the system via a 9-byte burst every 2 seconds whether anything else is going on at the terminal or not.

In all of the user sessions that we recorded, user interarrival times look roughly lognormal, as do echo interarrival times for teletypewriter terminals. Lognormal interarrival distributions appear in the observations of Fuchs and Jackson [1]; Pawlita [2] fits mixtures of exponentials to his empirical distributions. "Echo" times for work stations are more irregularly distributed. Host response lengths and interresponse times are also somewhat irregular, but seem frequently to be bimodal on a logarithmic scale. Substantially more data would be required in order to determine whether there are significant regularities in the statistics of computer response bursts.

B. File traffic

Host-to-host file transfers differ from terminal-like calls in that file transfer calls hold for shorter times (less than a minute on the average), and the mean packet rates are more than an order of magnitude greater than the rates for terminal-like calls.
In addition, file transfers involve a window flow control mechanism that leads to a substantial reverse flow of packets during each transfer. In Table 1, the reverse flow is about a quarter of the forward flow.

The effective limiting speed of file transfers on a local area network is almost never set by the raw transmission speed of the network, but rather by the host's operating system and the hardware interface between the host and the network. At the time these traffic data were taken, almost all of the hosts were connected to the network via DEC DR11-C interfaces. At the present writing, almost all of the interfaces are DEC KMC11-B microcomputers. Measurements of maximum disk-to-disk file transfer rates between otherwise idle hosts on our network were made for both interfaces. We obtained about 720 packets/second (7700 characters/second) for DR11 interfaces, and 5900 packets/second (75,000 characters/second) for KMC11 interfaces, with slightly higher rates for program-to-program transfers that bypass the UNIX file system overhead. The character-to-packet ratios would be lower for short files because of startup overhead.

Log-histograms of the distribution of packet counts of file-transfer calls and the distribution of busy-hour interarrival times are shown in Fig. 9. It would be convenient, if one were modeling a network with a different set of file transfer protocols, to know something about the distribution of file-transfer requests from users. Unfortunately one cannot get information about individual user requests from DATAKIT call records, because a single file-transfer call may involve the transfer of several files. Existing UNIX system software captures user-level data for certain classes of file transfers, but not for all file transfers.

![Fig. 9a Packets per file-transfer call.](image)

![Fig. 9b File-call interarrival times.](image)

6. TRAFFIC MODEL

The traffic generated by a typical user consists of the traffic between the user's own terminal and the host into which that terminal is connected, plus all the host-to-host traffic that the primary call generates. The approximate percentages of traffic are shown in Table 2. On the average one terminal call generates approximately 8 host-to-host calls, and one terminal packet generates somewhat more than 1.5 host-to-host packets.

<table>
<thead>
<tr>
<th>TABLE 2. TRAFFIC PERCENTAGES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calls</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>Terminal calls</td>
</tr>
<tr>
<td>Remote logins</td>
</tr>
<tr>
<td>Remote executions</td>
</tr>
<tr>
<td>File transfers</td>
</tr>
<tr>
<td>Total</td>
</tr>
</tbody>
</table>

Table 2 does not show the relative numbers of user characters transmitted by the different types of calls, because the network measurements only recorded packets. However, we estimate that the average packet contains 8 bytes of user data (perhaps a little more than this for terminal traffic, and a little less for files). Since one trunk frame encapsulates 3 packets in 530 bits, this suggests that only \((3 \times 8 \times 8)/530 \approx 36\%\) of the bits on the local trunk represent user data. This estimate may be compared with Shoch and Hupp's estimate [3] that about 69\% of the bits on their Ethernet represent user data, while about 31\% encompass all forms of overhead. It should be noted, however, that null bytes are stripped from DATAKIT packets before being transmitted over long-distance trunks (Fig. 1), where bandwidth may be at more of a premium.

We now construct a per-user description of the traffic on our network. Suppose that \(N\) terminals are available for connection to a given node. According to our observations, the maximum number that will be simultaneously connected to a host at any time during the week is about \(0.5N\). Furthermore, the maximum number of simultaneously existing host-to-host circuits is about one-half the maximum number of simultaneously existing terminal-to-host circuits. Since the instantaneous number of circuits does not fluctuate much about its local mean, one would feel safe if the circuit capacity of the switch were, say, twice the maximum number of simultaneous terminal connections expected during the week.

So far as the switching load is concerned, we see by looking at peak 10-minute averages that each simultaneously connected port generates terminal-to-host switching transitions at an average rate from 2 to 3 per hour. If we add in the switching rate for host-to-host calls, the total rate in the busiest 10 minutes corresponds to 10 to 20 switches per hour per connected terminal. This suggests that the controllers need to be able to handle a statistical distribution of 20 switches per hour per simultaneously connected terminal.

A calculation of congestion in the controller could make the following assumptions. A single setup or teardown operation on a controller takes about 0.1 s. A log-histogram of an interswitch arrival time distribution is shown in Fig. 4. If one is willing to use two-moment approximations for queueing delays, one can employ the following parameters for the interarrival and service time distributions:

\[
\bar{t} = 3600/n_s, \quad C_r = 1.25;
\]

\[
\bar{x} = 0.1 \text{ second,} \quad C_s = 0,
\]

where \(n_s\) is the average number of switches per hour.
Turning now to the modeling of packet flow, we can assume that on the average half the number of terminal circuits that are up are active. For each packet generated by a terminal circuit, roughly one-half of a terminal-circuit-like packet is generated on remote login and/or interactive remote command execution circuits, and about one file-transfer packet (Table 2).

We model a terminal-like circuit in the following way. An active asynchronous terminal sends a burst of x characters after t seconds to the host, where x and t are random variables with specified means and coefficients of variation. The host sends back a stream of "echoes" and a stream of longer responses. Our current best assessment of the parameters describing teletypewriter and work-station traffic on the Bell Laboratories DATAKIT network is shown in Table 3. Conversion of these character streams into equivalent packet streams depends, of course, on the packetizing algorithm associated with the given network.

### TABLE 3. CHARACTER-BURST PARAMETERS

<table>
<thead>
<tr>
<th></th>
<th>( t ) char</th>
<th>( C_t )</th>
<th>( t ) sec</th>
<th>( C_t )</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Teletypewriter</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>From terminal</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Host echo</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Host response</td>
<td>128</td>
<td>2</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td><strong>Work station</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>From terminal</td>
<td>8</td>
<td>0.5</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Host echo</td>
<td>8</td>
<td>0.3</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Host response</td>
<td>96</td>
<td>3</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>

Finally, we need a model for file transfer calls. On our network, the total number of packets transferred in file-transfer calls is approximately equal to the total number of packets transferred in terminal-to-host calls, but file transfer packets go in much larger bursts. A pessimistic estimate of the congestion due to file transfer calls would be obtained by assuming that the distribution of forward packet counts follows the log-histogram of Fig. 9a (unimodal with a long tail), with a mean length of 2000 user characters (2000 packets) per forward transfer, and a mean "delivery" rate from the sending host of, say, 4000 packets/sec. Each forward transmission would be paired with a backward transmission of one-quarter the length—that is, 500 packets including no user characters. The intervals between transmissions could be distributed according to the plot in Fig. 9b, with mean adjusted to achieve the desired overall average rate of file transfers. If there are \( n_f \) file transfers per hour on the average, then

\[
\bar{t} = \frac{3600}{n_f} \text{ seconds,} \quad C_t = 1,
\]

\[
\bar{x} = 20000 \text{ characters,} \quad C_x = 10.
\]

The file-transfer assumptions are conservative on two counts, so far as congestion is concerned. In the first place, a single file-transfer call may involve the transfer of a number of short files rather than a single long file; we had no way of recording individual user files during the packet-count measurements. Secondly, although a file transfer between KMC11 host interfaces can run at 6000 packets/second under ideal conditions, in practice file transfers run substantially slower because the hosts are being timeshared.

Finally, it should be emphasized that in none of the distributions of interarrival times have we investigated possible correlations between successive arrivals. If the arrival process is not a renewal process (successive intervals independently and identically distributed), the results of a conventional queueing-theory analysis may be seriously in error. This matter deserves further inquiry.

### 7. CONCLUSIONS

We have obtained a reasonably detailed statistical description of mixed data traffic, in terms of terminal-like calls and file transfer calls, by combining user-level and network-level measurements on a given network. The resulting model should be useful in predicting the response of networks having various different architectures to similar kinds of traffic.

Models of teletypewriter traffic are inadequate to describe the volume and statistical characteristics of traffic generated by work-station terminals. We have made some measurements of traffic due to diskless work stations. Measurements are also needed on networks of semiautonomous work stations, which have their own disk storage combined with substantial internal processing capability.

Statistical distributions associated with data traffic have coefficients of variation substantially greater than unity (up to 10 in the present study). It is important to learn how such large coefficients of variation affect the performance of queueing networks.

### ACKNOWLEDGMENTS

A. E. Kaplan made the file-transfer packet flow measurements, and E. J. Sitar did the measurements of terminal character traffic.

### REFERENCES


