On measured behavior of the ARPA network*

by LEONARD KLEINROCK and WILLIAM E. NAYLOR

University of California
Los Angeles, California

INTRODUCTION

The purpose of this paper is to present and evaluate the results of recent measurements of the ARPA network. We first discuss the tools available for performing these measurements. We then describe the results of a particular experiment, which consisted of data collection over a continuous seven day period. The measured quantities included input traffic, line traffic, and message delays. This data is discussed in terms of network behavior and compared to analytic models. Lastly, we consider some implications and tradeoffs derived from these measurements which provide insight regarding the performance of computer networks.

The ARPANET is now more than four years old.[1–8] However, the network did not become generally useable until the middle of 1971 when the HOST-to-HOST protocol[5] was finally implemented at most of the sites connected to the network at that time. Currently, the network consists of approximately 40 switching computers (the IMPs and TIPs) and approximately 50 HOST machines attached to these switching computers as shown in Figure 1 (this map corresponds to the network configuration as of 1 August 1973; we use this particular map since it gives the network topology which existed at the initiation of our experiment; a 39th site had just been installed in the network by BBN for test purposes and thus does not appear in Figure 1). We notice that the ARPANET spans the United States, crossing over to Hawaii by means of a 50 KBPS (kilobit per second) satellite channel and extends to Europe by means of a trans-Atlantic 7.2 KBPS satellite channel. From October of 1971, the traffic and use of the network has been growing exponentially at a phenomenal rate, slowing down a bit toward the end of 1973; this traffic growth is shown in Figure 2 on a log-linear scale.[14] In this paper we examine the details of that traffic flow.

The ARPANET began as an *experimental* network and has since grown into a powerful tool for resource sharing. The essence of an experiment is measurement, and it is this aspect of the ARPANET which we wish to discuss herein. Can we, in fact, determine what is going on within the network? The answer is an emphatic yes, if we restrict ourselves to the behavior of the communication subnetwork which provides the message service for the user-HOST systems. Early on, during the days when the ARPANET was still a concept rather than a reality, we were careful to include in every specification of the network design the ability to monitor network behavior with the use of specific measurement tools. This paper deals with a description of those tools and how they have been used in a particular experiment designed to elucidate the behavior of traffic in the ARPANET.

Among the various centers in the network are two which are deeply concerned with measurements; the Network Control Center (NCC), at Bolt, Beranek and Newman, Inc. (BBN), and the Network Measurement Center (NMC) at the University of California, Los Angeles (UCLA). The experiment we describe below was designed, conducted and interpreted by the UCLA-NMC research staff.

At this point, it is perhaps helpful to review a few of the network parameters which affect traffic flow in the ARPANET.[9] All traffic entering the network is segmented into messages whose maximum length is 8063 bits. These, in turn, are partitioned into smaller pieces called packets which are at most 1008 bits long (a maximum length message, therefore, will be partitioned into eight packets, the last of which has a maximum length of 1007 bits). As messages enter the network from the HOSTs they carry with them a 32 bit "leader" which contains the addressing information necessary for delivery to the destination. Incoming messages also carry a small number of "padding" bits for word boundary adjustment between the IMP word size of 16 bits and various HOST word sizes. Packets are transmitted through the network with some addressing and control information which adds 168 bits to their transmitted length, while the packet overhead for storage within an IMP is 176 bits. The packets make their way through the network individually and are passed from IMP to IMP according to an adaptive routing procedure; in each IMP-to-IMP transmission an acknowledgment is returned if the packet was accepted; when possible, these acknowledgments are piggybacked on return traffic. The packets of a multipacket message are reassembled at the

* This research was supported by the Advanced Research Projects Agency of the Department of Defense under Contract No. DAHC-15-73-C-0368.

On measured behavior of the ARPA network*

by LEONARD KLEINROCK and WILLIAM E. NAYLOR

University of California
Los Angeles, California

INTRODUCTION

The purpose of this paper is to present and evaluate the results of recent measurements of the ARPA network. We first discuss the tools available for performing these measurements. We then describe the results of a particular experiment, which consisted of data collection over a continuous seven day period. The measured quantities included input traffic, line traffic, and message delays. This data is discussed in terms of network behavior and compared to analytic models. Lastly, we consider some implications and tradeoffs derived from these measurements which provide insight regarding the performance of computer networks.

The ARPANET is now more than four years old.[1-8] However, the network did not become generally useable until the middle of 1971 when the HOST-to-HOST protocol[5] was finally implemented at most of the sites connected to the network at that time. Currently, the network consists of approximately 40 switching computers (the IMPs and TIPs) and approximately 50 HOST machines attached to these switching computers as shown in Figure 1 (this map corresponds to the network configuration as of 1 August 1973; we use this particular map since it gives the network topology which existed at the initiation of our experiment; a 39th site had just been installed in the network by BBN for test purposes and thus does not appear in Figure 1). We notice that the ARPANET spans the United States, crossing over to Hawaii by means of a 50 KBPS (kilobit per second) satellite channel and extends to Europe by means of a trans-Atlantic 7.2 KBPS satellite channel. From October of 1971, the traffic and use of the network has been growing exponentially at a phenomenal rate, slowing down a bit toward the end of 1973; this traffic growth is shown in Figure 2 on a log-linear scale.[14] In this paper we examine the details of that traffic flow.

The ARPANET began as an *experimental* network and has since grown into a powerful tool for resource sharing. The essence of an experiment is measurement, and it is this aspect of the ARPANET which we wish to discuss herein. Can we, in fact, determine what is going on within the network? The answer is an emphatic yes, if we restrict ourselves to the behavior of the communication subnetwork which provides the message service for the user-HOST systems. Early on, during the days when the ARPANET was still a concept rather than a reality, we were careful to include in every specification of the network design the ability to monitor network behavior with the use of specific measurement tools. This paper deals with a description of those tools and how they have been used in a particular experiment designed to elucidate the behavior of traffic in the ARPANET.

Among the various centers in the network are two which are deeply concerned with measurements; the Network Control Center (NCC), at Bolt, Beranek and Newman, Inc. (BBN), and the Network Measurement Center (NMC) at the University of California, Los Angeles (UCLA). The experiment we describe below was designed, conducted and interpreted by the UCLA-NMC research staff.

At this point, it is perhaps helpful to review a few of the network parameters which affect traffic flow in the ARPANET.[9] All traffic entering the network is segmented into messages whose maximum length is 8063 bits. These, in turn, are partitioned into smaller pieces called packets which are at most 1008 bits long (a maximum length message, therefore, will be partitioned into eight packets, the last of which has a maximum length of 1007 bits). As messages enter the network from the HOSTs they carry with them a 32 bit "leader" which contains the addressing information necessary for delivery to the destination. Incoming messages also carry a small number of "padding" bits for word boundary adjustment between the IMP word size of 16 bits and various HOST word sizes. Packets are transmitted through the network with some addressing and control information which adds 168 bits to their transmitted length, while the packet overhead for storage within an IMP is 176 bits. The packets make their way through the network individually and are passed from IMP to IMP according to an adaptive routing procedure; in each IMP-to-IMP transmission an acknowledgment is returned if the packet was accepted; when possible, these acknowledgments are piggybacked on return traffic. The packets of a multipacket message are reassembled at the

* This research was supported by the Advanced Research Projects Agency of the Department of Defense under Contract No. DAHC-15-73-C-0368.

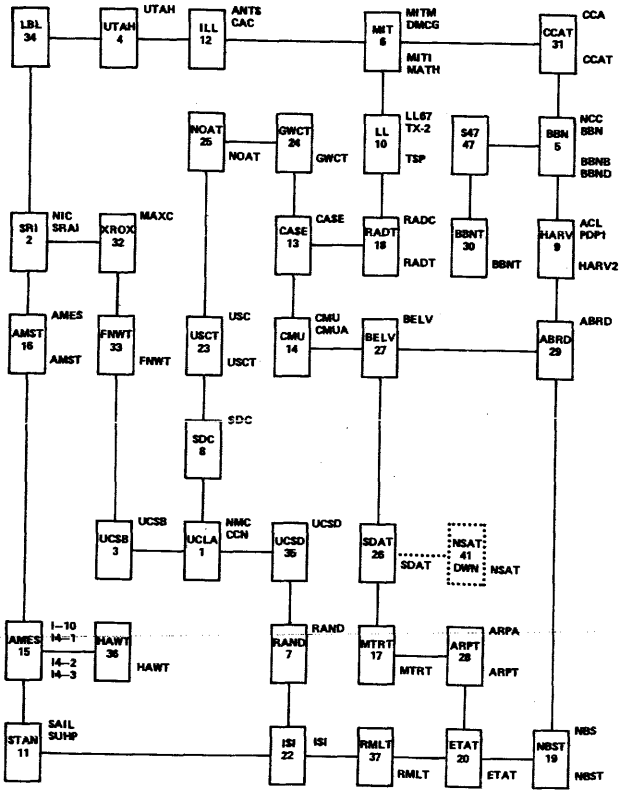

Figure 1—Logical map of the ARPANET (August 1, 1973 0834 PDT)

destination IMP before they are delivered to the destination HOST. When a message proceeds in its transmission to the destination HOST, a special control message (known as a Request For Next Message—RFNM) which acts as an end-to-end acknowledgment is returned from the destination IMP to the source HOST. The IMP itself buffers packets as they pass through the network and has the ability to store approximately 77 packets at most. Except for the channel connecting AMES to AMST (which is 230.4 KBPS) and the Atlantic satellite link (which is 7.2 KBPS) all lines in the network are 50 KBPS, full-duplex channels (as of August 1973).

In the following section, we describe the network measurement tools. Following that, we give details of a recently performed experiment and present its results in graphic form. We also include a section in which a mathematical model for delay is developed and the results of that model prediction are compared with measured network delays.

MEASUREMENT TOOLS

In this section, we describe the means by which this and other measurements are performed. In order to evaluate the performance of the network, several measurement tools (as originally specified by the UCLA-NMC) were implemented as part of the first IMP program (and have been slightly modified throughout the developmental stages of the ARPANET). These tools, which execute in each IMP's "background" mode, may be used selectively at the various network nodes under program control. Upon request, they collect data regarding their node, summarize these data in special measurement messages, and then send these messages to a collection HOST (normally UCLA-NMC). We have, therefore, developed at UCLA-NMC the capability for control, collection, and analysis of the data messages. Below, we describe these network measurement tools.

Trace

Trace is a mechanism whereby messages may be "traced" as they pass through a sequence of IMPS. Those IMPs whose trace parameter has been set will generate one trace block for each marked packet (i.e., a packet with its trace bit set) which passes through that particular IMP. (An "auto-trace" facility exists by which every nth message entering the network at any node may be marked for tracing.) A trace block contains four time stamps which occur when: (1) the last bit of the packet arrives; (2) the packet is put on a queue; (3) the packet starts transmission; and (4) the acknowledgment is received (for store and forward packets sent to a neighboring IMP), or transmission is completed (for reassembly packets sent to a HOST). (Time (1) corresponds to the time at which storage is actually allocated to the packet

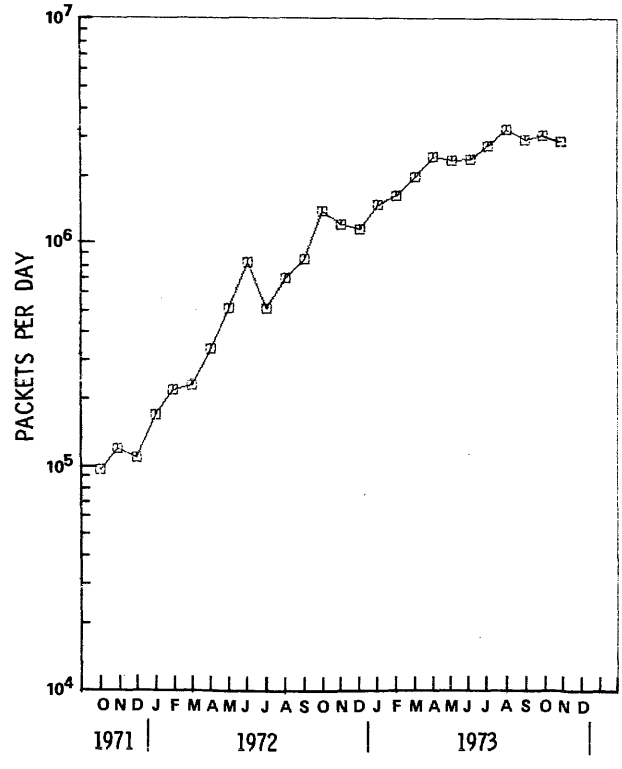

Figure 2—Long term traffic growth

rather than to the input source. Time (4) corresponds to the time at which the storage for the packet is returned to the free pool after successful transmission.) Also contained in the trace block are the length of the packet, an address indicating where the packet was sent, and the IMP header (which consists of the source and destination addresses and several other pieces of control information).

Accumulated statistics

The accumulated statistics message consists of several tables of data summarizing activity at a network node over an interval of time (ranging from 25.6 msec to some 14 minutes) which is under program control. Included in the accumulated statistics is a summary of the sizes of messages entering and exiting the network at the set of real (as opposed to fake, i.e. IMP-simulated) HOSTs connected to that IMP. The message size statistics include a histogram of message lengths (in packets) for multipacket messages and a log (base 2) histogram of packet lengths (in words) for all last packets (i.e., a count is recorded of those packets whose length, in data bits, is from 0 to 1, 2 to 3, 4 to 7, 8 to 15, 16 to 31, or 32 to 63 IMP words in length). Also included is the total number of IMP words in all the last packets, and the total number of messages from each HOST (real and fake), and the total number of control messages (RFNM, etc.) to each HOST.

A row of the global traffic matrix is contained in each IMP's round-trip statistics. These contain the number of round-trips (message sent and RFNM returned) sent from the probed site to each site, and the total time recorded for those round-trips. These statistics are listed for each possible destination from the probed site.

For those channels connected to the probed site, we have the channel statistics. These consist of: (1) the number of hellos sent per channel (channel test signals); (2) the number of data words sent per channel; (3) the number of inputs received per channel (all inputs: data packets, control packets, acknowledgments, etc.); (4) the number of errors detected per channel; (5) the number of "I-heard-you" packets received per channel (response to hello); (6) the number of times the free buffer list was empty per channel; and (7) log (2) histograms of packet length, in data words (one histogram per channel).

Snapshots

Snapshots give an instantaneous peek at an IMP. The snapshot records several queue lengths as well as the IMP's routing table. The HOST (real or fake) queue (normal and priority) lengths appear in each snapshot message. Also included is information about storage allocation: the length of the free storage list, the number of buffers in use for reassembly of messages, and the number of buffers allocated to reassembly (but not yet in use). Snapshots also include the IMP routing table and delay table. Entry i in the routing table contains the channel address indicating where to send a packet destined for site i. A delay table entry consists of the minimum number of hops to a site, and the delay estimate to reach a site.

Artificial message generation

In addition to the above instrumentation package built into each IMP, we have the capability to generate artificial messages. This message generator in any IMP can send fixed length messages to one destination at a fixed or RFNM driven interdeparture time. Together with the generation facility there exists a discard capability in each IMP. Several message generator/acceptor pairs have been implemented for a subset of the HOSTs on the network as well. These are extremely useful for experimentation, but we will not attempt to discuss them in this paper.

Control, collection, and analysis

The above-mentioned measurement and message generation facilities are controlled by sending messages to the "parameter change" background program in the IMPs. We have constructed a set of programs which, after an experiment is specified, automatically format and send the correct parameter change messages to initiate that experiment. In order to be able to send these messages, it was necessary to modify the system code in the NCP to bypass the normal HOST-to-HOST protocol.[5] The bypass was then used as the means of collecting the measurement messages as well, since these too do not adhere to HOST-to-HOST protocol. After a message is received over this mechanism, it is stored in the file system at UCLA-NMC. Reduction and analysis of the data is accomplished by supplying specific subroutines for a general driver program; the data analysis is currently done on the UCLA 360/91.

Status reports

In addition to the above tools, which are mainly for experimental use, the NCC has built into the IMPs a monitoring function called "status reports."[10] Each IMP sends a status report to the NCC HOST once a minute. Contained in the status report are the following: (1) The up/down status of the real HOSTs and channels; (2) for each channel, a count of the number of hello messages which failed to arrive (during the last minute); (3) for each channel, a count of the number of packets (transmitted in the last minute) for which acknowledgments were received; and (4) a count of the number of packets entering the IMP from each real HOST. These status reports are continually received at the NCC and are processed by a minicomputer which advises the operator of failures in the network and creates summary statistics.

Let us now address ourselves to the experiment itself.

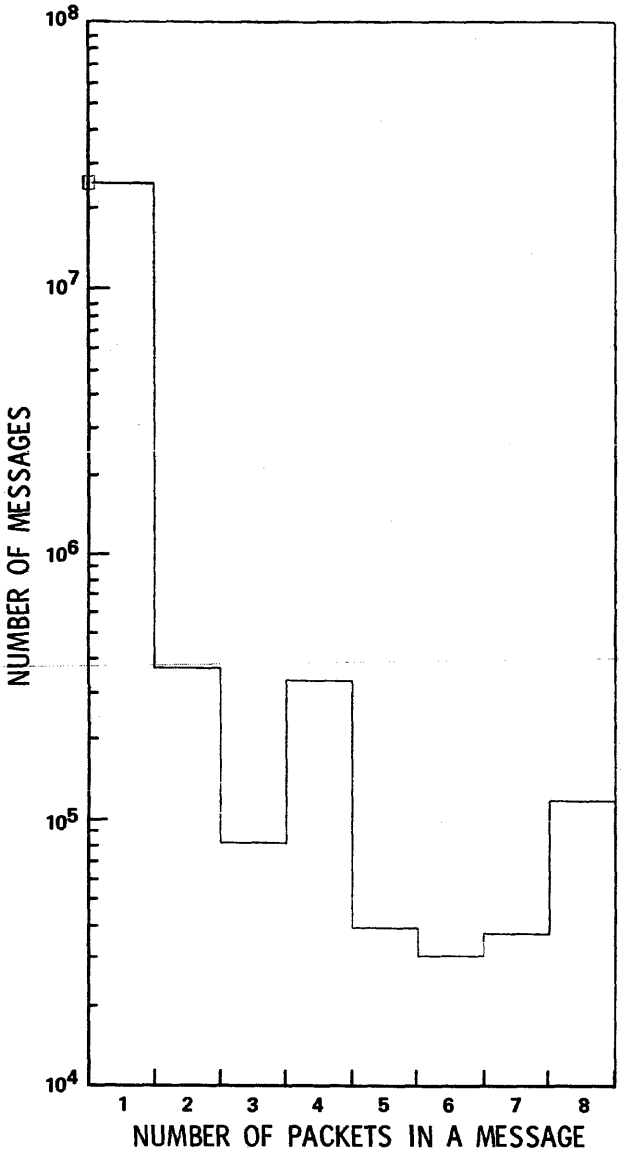

Figure 3—Histogram of HOST message length in packets

THE EXPERIMENT DESCRIPTION AND RESULTS

Experiment description

The purpose of this experiment was to observe the traffic characteristics of the operating network. These characteristics include: (1) message and packet size distributions; (2) mean round-trip delay; (3) mean traffic-weighted path length; (4) incest (the flow of traffic to and from HOSTs at the same local site); (5) most popular sites and channels; (6) favoritism (that property which a site demonstrates by sending many of its messages to one or a small number of sites); and (7) channel utilization. We consider this data to have more than just historical significance. In particular, there are several network parameters whose values were chosen prior to the actual network implementation and which deserve to be reevaluated as a result of the measurements reported here. Among these parameters are: packet (and therefore buffer) size, number of buffers, channel capacity, single/multiple packet message philosophy, etc.

To observe the traffic characteristics, we gathered data over a continuous seven-day period from 8:36 on 1 August 1973 through 17:06 on 7 August 1973. The network configuration during this period is shown in Figure 1. (A teletype-compatible network map containing similar information may be generated from an updatable NMC survey of the network.) The experiment consisted of sending accumulated statistics messages to UCLA-NMC from each site in the network at intervals of approximately seven minutes. The

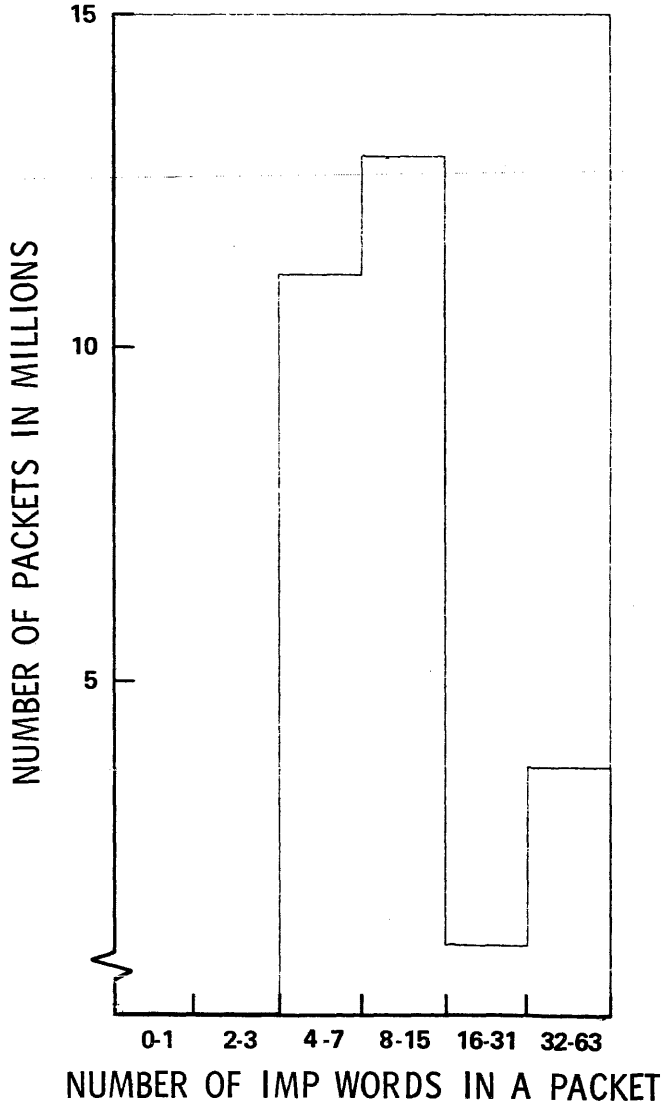

Figure 4—Histogram of packet length in words

data were subsequently processed, and the general results appear below.

Measured results

During the seven days a total of some 6.3 billion bits were carried through the network by some 26 million messages. This means that on the average the entire network was accepting some 47 messages per second and carrying roughly 11500 bits per second among HOST computers. The HOST messages were distributed in length as shown in Figure 3, and from these data, we observe a mean of 1.12 packets per message! Moreover, the mean length of a message is 243 bits of data! These facts indicate not only are there very few multipacket messages, but also that most single packet messages are quite short. This latter fact is borne out in the log (2) histogram of packet length for packets entering the network from the HOSTs as shown in Figure 4; the mean packet length is 218 bits of data.

The small message size has an impact on the efficiency of storage utilization. This may be seen by defining the buffer utilization efficiency as follows:

$$\eta = \frac{\bar{l}_p}{L+H}$$

where

$\bar{l}_p$ = the mean packet length,
L = the maximum length of data in a packet, and
H = the length of the packet storage overhead.

Using the measured value of $\bar{l}_p = 218$ bits, and the constants $L = 1008$ bits and $H = 176$ bits, we have a measured buffer utilization efficiency of .184!

There exists a buffer length which yields an optimal buffer efficiency for a given message length distribution, as shown by Cole;[11] this calculation assumes an exponential message length distribution (which we shall adopt). In the packeting of messages into L bit pieces we have truncated the exponential message length distribution at the point L, thus giving a mean packet size of

$$\bar{l}_p = \bar{l}[1 - e^{-L/\bar{l}}] \qquad (1)$$

where $\bar{l}$ = the mean message size (exponential). This gives $\bar{l}_p = 239$ bits when the value of $\bar{l} = 243$* is used in Eq. (1) and which in turn yields an efficiency of .202. (The fact that $\bar{l}_p = 239$ is greater than the measured $\bar{l}_p$ means that the actual distribution weights shorter messages more heavily than the exponential distribution.) Since $\bar{l}_p$ is significantly less than L, the truncation at L does not cause a large accumulation of packets whose length is L bits; we see this from the moderate number (12.9 percent) of maximum length packets in Figure 4.

The optimal value for buffer size L_0 is obtained by solving the following equation:

$$e^{-L_0/\bar{l}}[L_0 + H] - \bar{l}[1 - e^{-L_0/\bar{l}}] = 0$$

Using $\bar{l} = 243$, and $H = 176$ we obtain the optimal buffer size of $L_0 = 244$ bits which yields a maximum efficiency of .366 for this overhead. Thus, based upon this particular week's measured data, (which is supported by previous and later measurements), we find that the maximum efficiency can be increased significantly by reducing the packet buffer size to roughly one-quarter of its current size.

The measured mean round-trip* message delay for the seven-day period was approximately 93 milliseconds. Indeed, the network is meeting its design goal of less than 200 milliseconds for single packet messages. Thus, as desired, the communication subnet is essentially transparent to the user, so far as delay is concerned. The principal source of delay seen during a user interaction comes both from his local HOST and from the destination HOST on which he is being served. Major contributors to the small network message delay are the small message size and the fact that a significant number of messages traverse very short paths in network.

We shall return to a discussion of delay in the next section. For now, let us study the traffic distribution and the source of short paths, incest, favoritism, etc. From Reference 12 we know that the mean path length (in hops—i.e., number of channels traversed) may be calculated by forming the ratio of the total channel traffic to the externally applied traffic. This gives a value of 3.31 hops. Moreover, we may form a lower bound on the average path length by assuming all traffic flows along shortest paths; this gives a value of 3.24 hops, showing that indeed most of the traffic follows shortest paths. The (uniformly weighted) path length (average distance) between node-pairs is 5.32 as can be calculated directly from the topology shown in Figure 1. The difference between these measures of path length suggests that network users tend to communicate with sites which are nearby. This is surprising since distance in the network should be invisible to the users! This phenomenon may be explained by examining how much traffic travels over paths of a given length (in hops) as shown in Figure 5. Observe that a surprisingly large fraction (22 percent) of the traffic travels a distance of zero hops and is due to (incestuous) traffic between two HOSTs connected to the same IMP; after all, the IMP is a very convenient interface between local machines as well. Also note that 16 percent of the network traffic travels a hop distance of one; the major portion of this (13 percent of the total) is due to communication between AMST and AMES (this too is incestuous in spirit). This curve fails to account for the number of site-pairs at a given distance. For the topology

* A truncation effect occurs before messages enter the ARPA network as well. Hence the measured mean message length is actually the mean taken from the actual distribution truncated at 8063 bits (8 packets). Assuming that messages are exponentially distributed we may solve an equation similar to Eq. (1) to obtain the untruncated mean message length; this computation yields 243 bits, the same as the truncated mean message length.

* Round-trip delay is measured by the IMPS and is the time from when a message enters the network until the network's end-to-end acknowledgment in the form of a RFNM is returned.

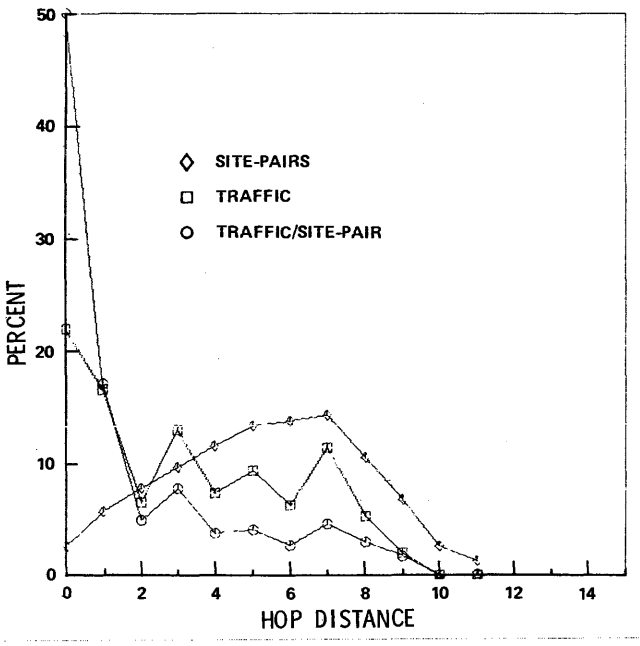

Figure 5—Distance dependence of traffic

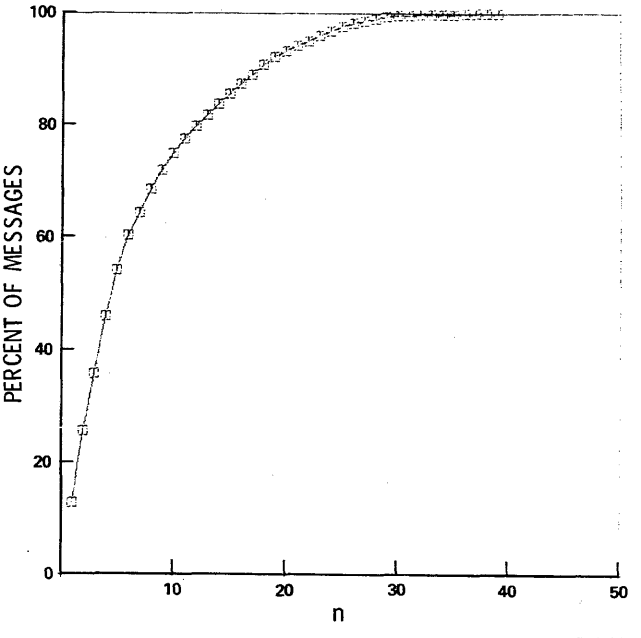

Figure 7—Busy source distribution

existing during this experiment, it can be seen that the following list of ordered pairs (x, y) provides the distribution of site-pair minimum distances (where x=hop distance and y=number* of site-pairs at this distance): (0,39), (1,86), (2,118), (3,148), (4,176), (5,204), (6,210), (7,218), (8,160), (9,102), (10,40), and (11,20). No sites are more than 11 hops apart. This data is also plotted in Figure 5. Note that more sites are at a distance of 7 than any other distance (with the average distance equal to 5.32 as mentioned above). (In a network with N nodes and M full-duplex channels, the first two entries on the list must always be (0, N), (1, $2M$).) With this information, we may "correct" our curve by plotting the ratio of the number of messages sent between site-pairs at a given distance to the number of site-pairs at that given distance; see Figure 5 again. The ratios are normalized to sum to one. If the traffic were uniformly distributed in the network, then the resulting curve would be a horizontal line at the value 8.3 percent. We note that an even larger fraction

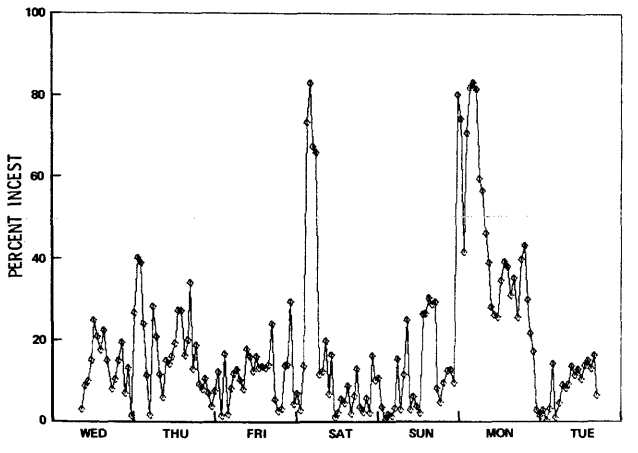

Figure 6—Incest

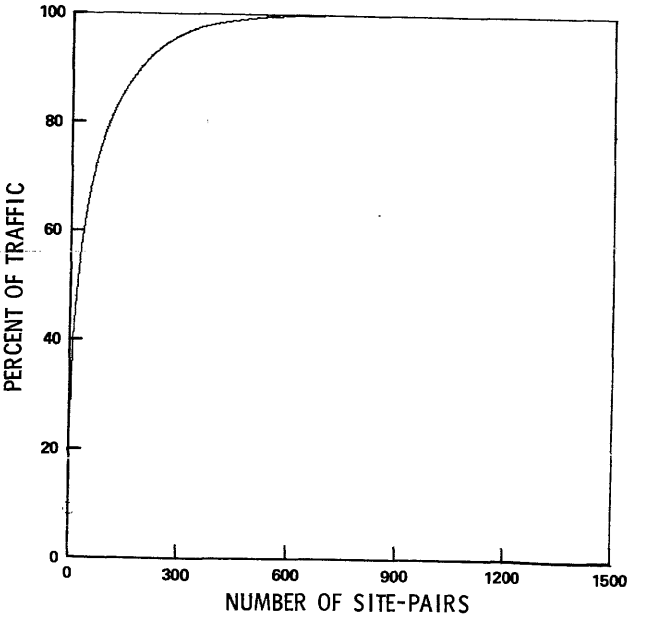

Figure 8—Busy site-pair distribution

* We consider site pairs as ordered pairs; thus, the pair (MIT, UCLA) is distinct from (UCLA, MIT). This is natural since the traffic flow is not necessarily symmetrical. The (important) special case of (SITE i, SITE i) counts as one "pair".

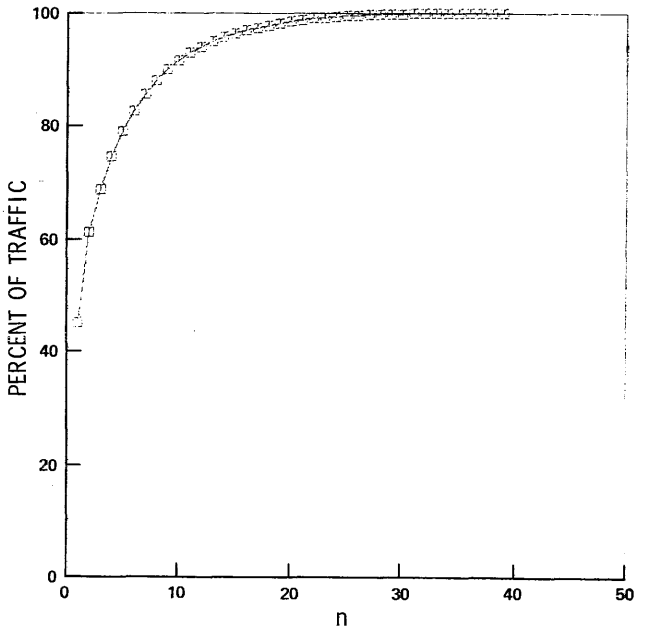

Figure 9—Distribution of traffic to favorite destinations

of the traffic is now identified with distance zero. At distances 2, 3, . . ., 9, we now see a better uniformity than earlier. The last effect which contributes to the remaining non-uniformity is the location of the large traffic users (e.g., ILL) and large servers (e.g., ISI). In Figure 6, we display the percent of incest in the network during each hour* of the experiment. Note that incest accounts for over 80 percent of the traffic during certain hours (the weekly average is 22 percent), peaking in the wee hours of the morning.

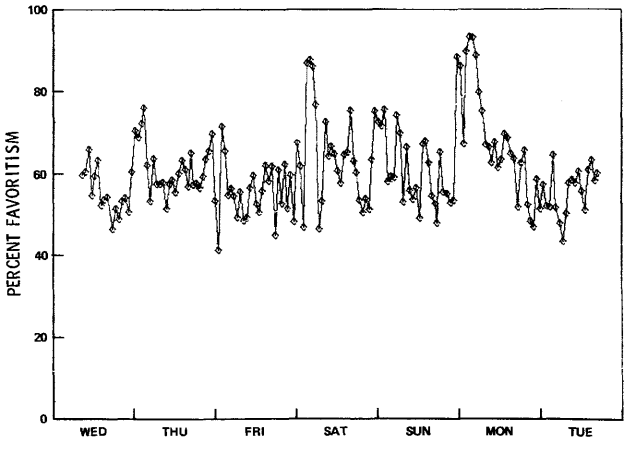

Figure 10—Percent of traffic to most favored destinations

* This, and the other "hourly" plots show points separated by approximately 56 minutes (an integral multiple of the accumulated statistics interval of roughly 7 minutes). The separation between the days on the horizontal axes occurs at midnight.

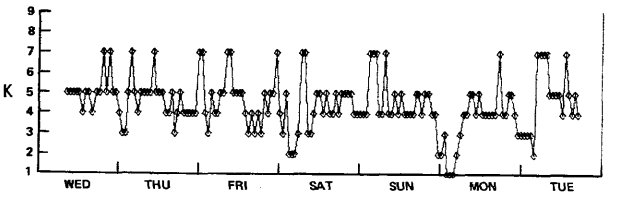

Figure 11—Number of favored destinations required to achieve 90 percent traffic

A further illustration of the non-uniformity of the traffic is seen in Figure 7. Here, we have plotted the cumulative percent of messages sent from the n busiest sources. Notice that over 80 percent of the traffic is generated by the busiest one third of the sites. A similar effect is true for the busiest (most popular) destinations. Even more striking is Figure 8, in which we have plotted the cumulative percent of traffic between site-pairs. Notice that 90 percent of the total traffic is between 192 (12.6 percent) of the site-pairs.

The interesting property of favoritism is shown in Figure 9. For each source, the destinations may be ordered by the frequency of messages to those destinations. In Figure 9, we show (summed over all sources) the percent of traffic to a source's n most favored destinations. If these orderings and percentages remained invariant over time (i.e., a stationary traffic matrix), then one could use this information in the topological design; however, it can be shown[4,13] that both the network design and performance are relatively insensitive to changes in the traffic matrix (and so, a uniform requirement is usually assumed). Note that 44 percent of the network traffic goes to the most favored sites! (A uniform traffic matrix would give a percentage of only $1/N = 2.56$ percent). Also,

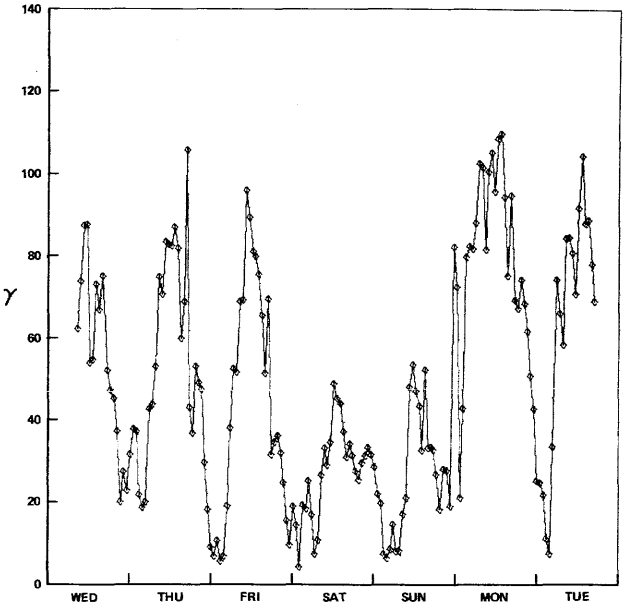

Figure 12—Arrival rate of HOST messages per second (γ)

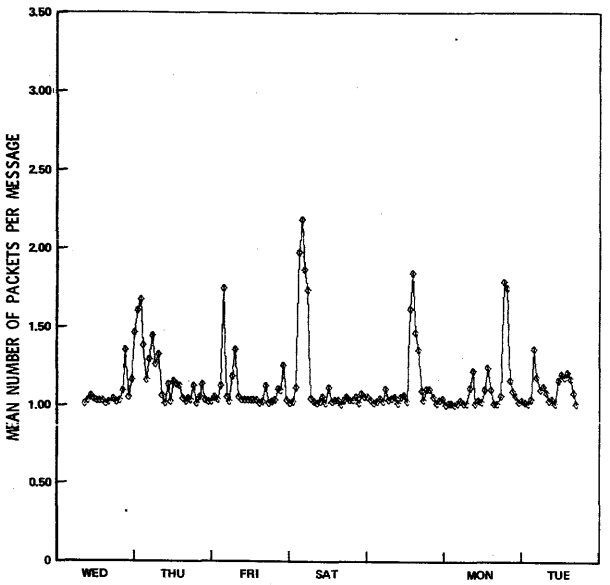

Figure 13—Mean number of packets per message

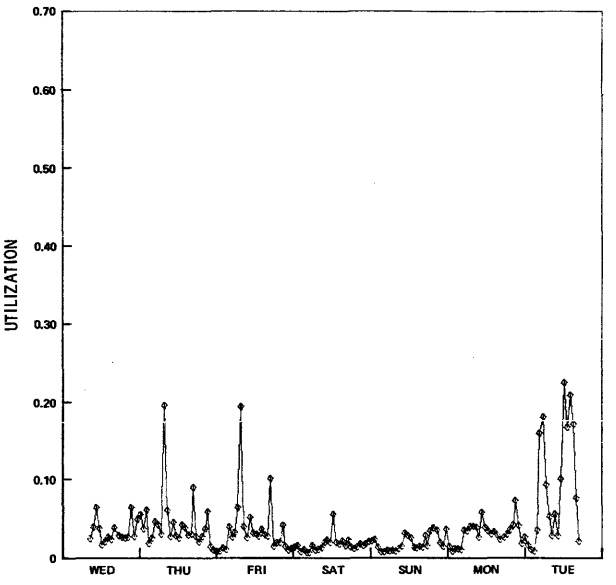

Figure 15A—Utilization of most heavily used channel in each hour (without overhead)

90 percent of the traffic goes to the nine most favorite sites; however, it is important to realize that this involves more than nine sites (in fact, 33 unique destinations are involved), since each source need not have the same set of nine most favorites. This favorite site effect is more dramatically displayed in Figure 10, which shows the percent of traffic to the most favored destination of all sources on an hourly basis. Most of the traffic (a minimum of 40 percent and an average of 61 percent) was caused by conversations between the N sources and their favorites. There are N^2 pairs in total; thus, on a weekly basis, the N favorites account for $.44N$ times the traffic they would have generated if the traffix matrix had been uniform (on an hourly basis it is $.61N$). Note that the favorite site effect must increase as we shrink the time interval over which "favorite" is defined;* in fact, if we choose an interval comparable to a message transmission time, then the

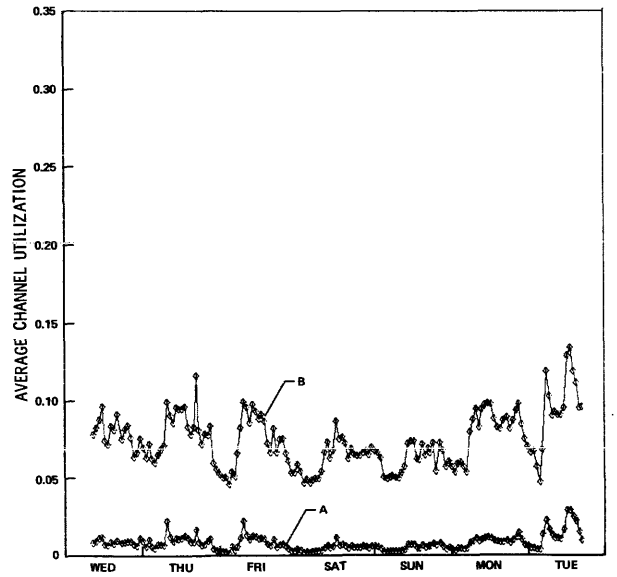

Figure 14—Network-wide mean channel utilization: (A) without overhead; (B) with overhead

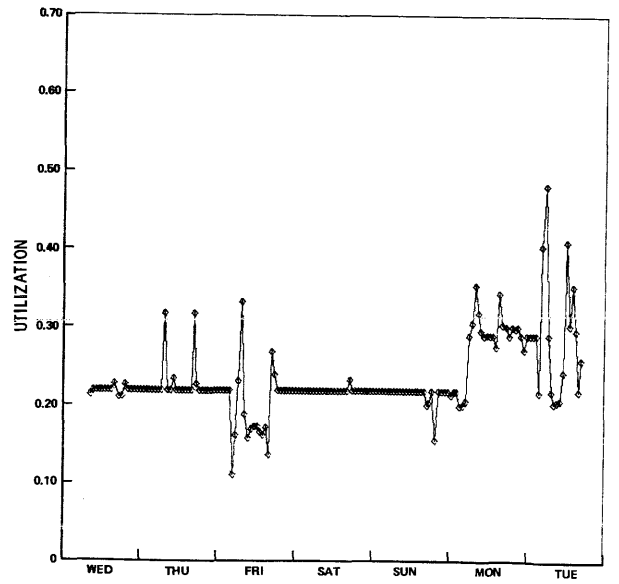

Figure 15B—Utilization of the most heavily used channel in each hour (with overhead)

* We are pleased to acknowledge the assistance of Stanley Lieberson in explaining this effect.

most favorite sites will account for almost 100 percent of the traffic, since the name of each source's favorite site will change dynamically to equal the name of the destination site for this source's traffic of the moment. Thus, the amount of traffic due to favorite sites has an interpretation which changes as the time interval changes. The weekly value of 44 percent has two possible interpretations. The first is that there exists a true phenomenon of favoritism due, perhaps, to the existence of a few useful "server" systems. The second interpretation is that network users are lazy; once a user becomes familiar with some destination HOST, he continues to favor (and possibly encourages others to favor) that HOST in the future rather than experimenting with other systems, too. A further explanation for this phenomen is that it is not especially easy to use a foreign HOST at this stage of network development; this trend should diminish as network use becomes more user oriented.

Related to Figure 10 is Figure 11 in which we have plotted the number, K, of favored destinations necessary to sum to 90 percent of the overall traffic on an hourly basis. This means that in any hour, 90 percent of the messages were sent between at most NK of the total $N^2 = 1521$ pairs in the network. Notice that K has a maximum hourly value of 7 (this is less than the weekly average of $K = 9$ due to the smaller averaging interval as discussed above). Therefore, for any hour, it requires at most 18 percent of the site-pairs to send 90 percent of the messages (in the most extreme case, $K = 1$ and so for those hours at most $1/N$ or 2.56 percent of the site-pairs sent 90 percent of the messages).

Let us now discuss other global measures of the network behavior. In Figure 12, we show the average rate at which HOST messages were generated (per second) on an hourly basis; this gives us an indication as to when the work was

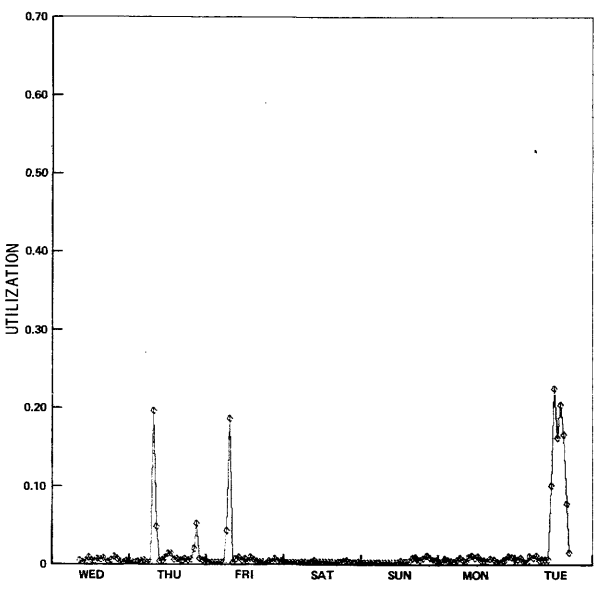

Figure 16A—Utilization of the channel (GWCT to CASE) with the highest hourly average (without overhead)

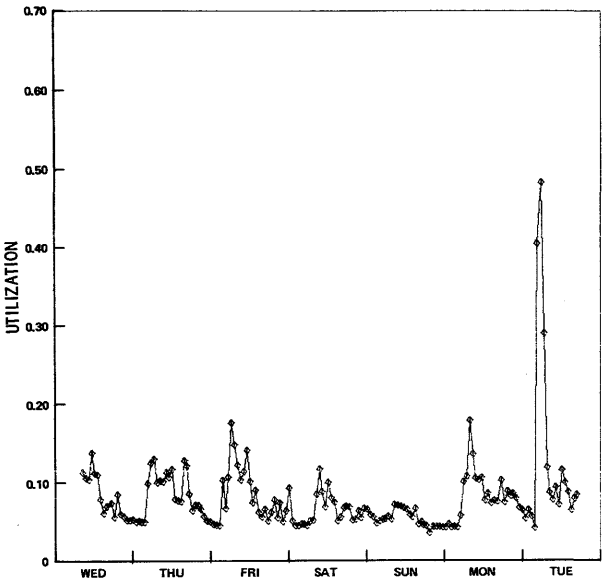

Figure 16B—Utilization of the channel (HARV to ARBD) with the highest hourly average (with overhead)

done on the network. There are no real surprises here: the curve shows a predominance of traffic during daylight hours and on weekdays. It is interesting that Monday had noticeably heavier traffic than the other weekdays (were the users manifesting feelings of guilt or anxiety for having slowed down during the weekend?). Observe that a truly worldwide network with its time zones could perhaps take advantage of these hourly and daily slow periods.

Figure 13 illustrates the change in network use as a function of time by showing the time behavior of the mean number of packets per message. The peaks are associated with those hours during which file transfers dominated the interactive traffic. These peaks in general occur during off-shift hours (as with incest). Perhaps users feel that they get better data rates, reliability, or HOST service late at night; or, perhaps the background of file transfers is continually present, but is noticed only when the interactive users are alseep.

The internal traffic on channels is one measure of the effectiveness of the network design and use. In Figure 14, we show the channel utilization averaged over the entire network on an hourly basis, both with and without overhead. The utilization (whose weekly average was .071 if overhead is included or .0077 neglecting overhead) is rather low and suggests that the lines in the network have a great deal of excess capacity on the average (this excess capacity is desirable for peak loads). The maximum hourly line load (including overhead, and averaged over all channels) was approximately 13.4 percent (occurring five hours before the end of the measurement) and corresponded to an internal network flow of roughly 600 KBPS; without overhead the maximum hourly average utilization was approximately 2.9 percent (129 KBPS internal traffic). It is interesting to obesrve the *heaviest loaded line* during each hour; this we plot in Figure 15

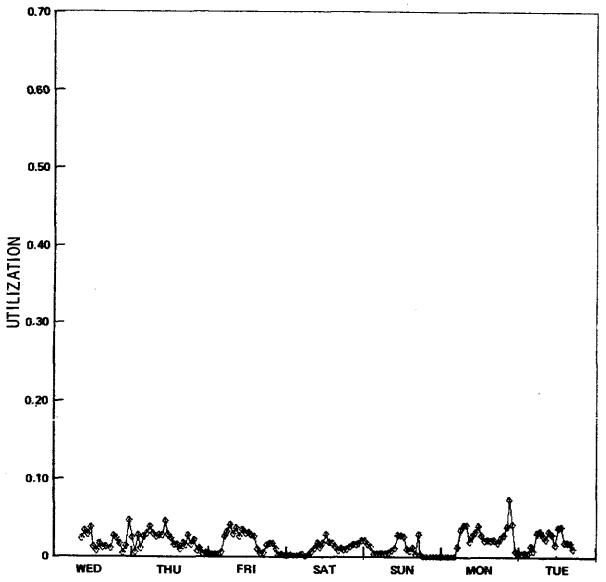

Figure 17A—Utilization of the channel (ISI to RMLT) with the highest weekly average (without overhead)

both without (part A) and with (part B) overhead. Note from part B that the *busiest line of any hour* (HARV to ABRD) had a utilization of 0.48 for that hour; without overhead the busiest line (GWCT to CASE) had a utilization of 0.225 for its busiest hour. Over the seven days, these channels had hourly load histories as shown in Figure 16. Note how bursty the traffic was on these lines (even averaged over an hour). Another interesting line is that one which had the maximum load averaged over the week. Neglecting routing updates and all other overhead the channel from ISI to RMLT had the largest weekly load (0.017), and its hourly behavior is shown in Figure 17A; again we see bursty behavior. If we include overhead then the satellite channel to Norway (SDAT to NSAT) had the *largest utilization averaged over the week* since it is only a 7.2 KBPS channel and therefore, all traffic placed almost seven (50/7.2) times the load on it (in this case, roughly 2KBPS, or 28 percent of the line, is used for routing updates alone). The hourly history for this channel is shown in Figure 17B. Also on this figure we have shown the UP/DOWN status of this line (in both directions).* Note that the channel was operational in both directions for a small fraction of the measurement (mainly on Monday) and only during this time was it carrying its own routing updates as well as responses to the NSAT to SDAT channel's routing updates in the form of "I heard you's"; this gives the 28 percent overhead mentioned above. This channel was down for a large part of Friday during which time it carried no traffic. For the rest (most) of the week the NSAT to SDAT channel was down and so no "I heard you" traffic was recorded on the SDAT to NSAT channel as can be seen in Figure 17B.

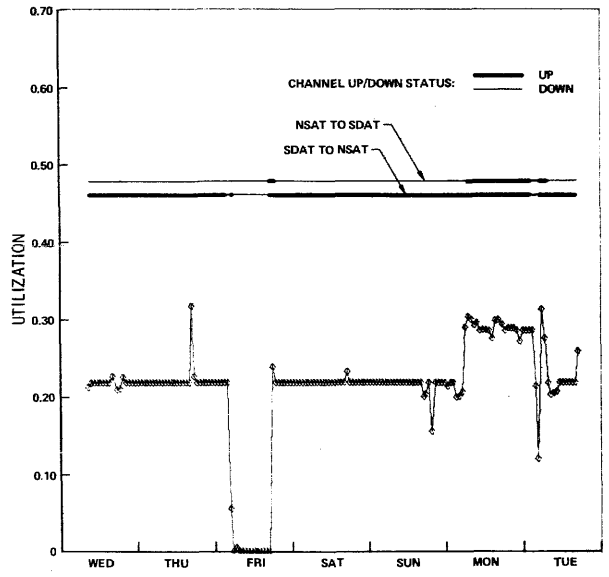

Figure 17B—Utilization of the channel (SDAT to NSAT) with the highest weekly average (with overhead)

With few exceptions the channels in the network are fairly reliable. Over half of the channels reported packet error rates less than one in 100,000. The average packet error rate was one error in 12,880 packets transmitted. Of the 86 channels in the network 14 reported no errors during the seven days,

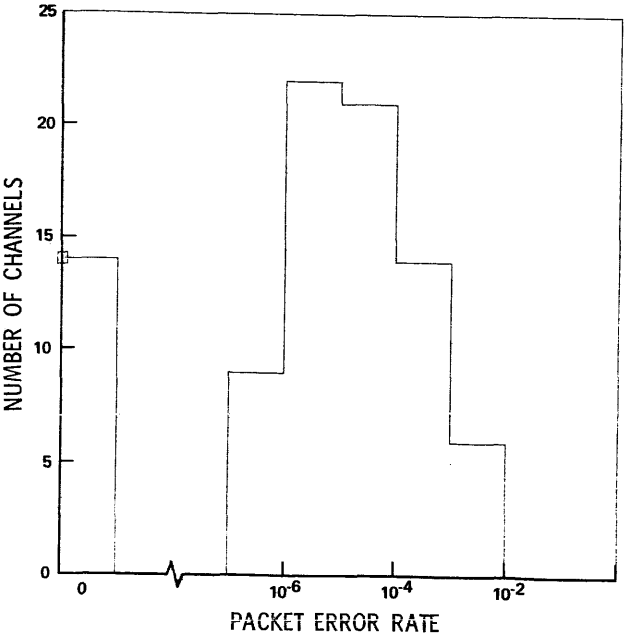

Figure 18—Channel packet error behavior

* Our measurements actually give the UP/DOWN status of the IMPs as seen by the NMC. When NSAT is declared down, we have displayed the NSAT to SDAT channel as being down in Figure 17B, and similarly, when SDAT is declared down we have shown the SDAT to NSAT (and the NSAT to SDAT) channel down.

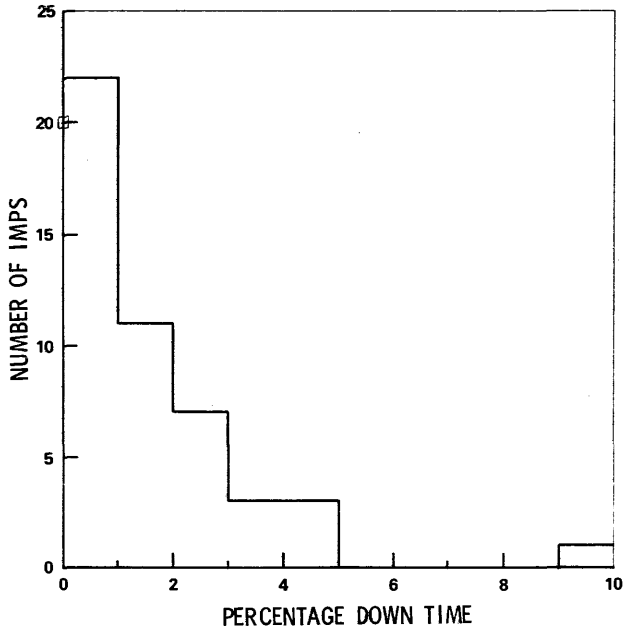

Figure 19—IMP failure behavior

while six channels had packet error rates worse than one in 1000. The worst case was one in 340 packets for the channel from RADT to LL. While these error rates are large enough to warrant the inclusion of error detection hardware and software, they are small enough so that traffic flow through the network is not impaired. In Figure 18, we show the error behavior of these lines during the seven day measurement. The failure rate of the IMPs should be included here, but clearly the seven day measurement is insufficient for this purpose. For completeness, therefore, in Figure 19 we show the performance characteristics of the IMPs over a 19 month interval (June 1972 through December 1973).[14] The average IMP down rate was 1.64 percent, with the worst case being 9.13 percent.

MODEL FOR NETWORK DELAY

In this section, we present a network delay model originally introduced by Kleinrock[12] and which was extended by Fultz[15] and Cole.[11] We then further extend this model to fit the specific implementation of the ARPA network. Following the model formulation, we present a comparison between the predicted and measured delay.

With the assumption of negligible nodal processing delays and channel propagation delays, the average message delay T (the time to traverse the network from source to destination) originally appeared as[12]

$$T=\sum_{i=1}^{M}\left[\frac{\lambda_i}{\gamma}T_i\right]$$

where

λ_i = the mean arrival rate of messages to the ith channel,
γ = the mean arrival rate of messages entering the network,
T_i = the mean time spent waiting for and using the ith channel, and
M = the number of channels in the network.

This very general result is easily extended to include nodal and propagation delays as follows:

$$T=K+\sum_{i=1}^{M}\left[\frac{\lambda_i}{\gamma}\left(\frac{1}{\mu C_i}+P_i+K+W_i\right)\right]$$

where

$1/\mu$ = mean message size,
C_i = capacity of the ith channel,
P_i = propagation delay on the ith channel,
K = nodal processing delay, and
$W_i=T_i-1/\mu C_i$ = waiting time in queue for channel i.

The delay analysis now simply requires that we solve for W_i. Perhaps the simplest (Markovian) assumption is[16]

$$W_i=\frac{\lambda_i/(\mu C_i)}{\mu C_i-\lambda_i}$$

When the queueing delay due to control traffic is also considered, we have

$$W_i'=\frac{\lambda_i'/(\mu' C_i)}{\mu' C_i-\lambda_i'}$$

where

λ_i' = arrival rate of data messages and control messages to the ith channel, and
$1/\mu'$ = mean message size including control messages.

Removing the assumption that nodal processing delay is constant and including the destination HOST transmission time we obtain the following expression for the average delay experienced by single packet messages.

$$T_{SP}=\sum_{i=1}^{M}\left[\frac{\lambda_i}{\gamma}\left(\frac{1}{\mu C_i}+P_i+K_l+\frac{\lambda_i'/(\mu' C_i)}{\mu' C_i-\lambda_i'}\right)\right]+\sum_{i=1}^{N}\left[\frac{\gamma_{\cdot j}}{\gamma}\left(K_j+\frac{1}{\mu_H C_{Hj}}\right)\right]$$

where

K_l is the packet processing time at node l (l is the origin node of channel i),
$\gamma_{\cdot j}$ = the mean departure rate of messages from the network to the HOSTs at site j,
$1/\mu_H C_{Hj}$ = the mean transmission time of messages to a HOST at site j, and
N = the number of nodes in the network.

The above formulae assume unpacketed message traffic, while in the ARPANET, messages are divided into from 1 to

8 packets. Fultz[15] and Cole,[11] therefore, extended the model to obtain the mean delay experienced by multi-packet messages

$$T_{MP}=K+\sum_{i=1}^{M}\left[\frac{\lambda_i}{\gamma}\left(\frac{1}{\mu C}+P_i+K+W_i'\right)\right]$$

$$+(\bar{m}-1)\left(\frac{1}{\mu C}+\sum_{jk}\left[\frac{\gamma_{jk}}{\gamma}\bar{\tau}_{jk}\right]\right)$$

where

C = line capacity (temporarily assumed constant)
$\bar{m}$ = mean number of packets in a multipacket message,
γ_{jk} = the arrival rate of messages from j to k, and
$\bar{\tau}_{jk}$ = mean inter-packet gap time for messages from source j to destination k.

It is difficult to measure $\bar{\tau}_{jk}$ for each j,k pair in the network. We, therefore, introduce an approximation due to Cole,[11] which yields

$$E[\tau(n \text{ hops})]=\frac{\rho(1-\rho^{(n-1)})}{1-\rho}\frac{1}{\mu C}$$

The above expression gives the expected value of τ_{jk} for nodes j and k which are n hops apart. It assumes that the channel utilizations ρ_i for the channels in the path from j to k are constant and equal to ρ. The path is assumed unique and the channel capacities are constant with value C. We will use the first two assumptions to obtain an approximation to the network-wide mean interpacket gap. Note that the average path length traveled by a message is given by

$$\bar{n}=\frac{\lambda}{\gamma}$$

where

$$\lambda=\sum_{i=1}^{M}\lambda_i$$

The average line utilization is

$$\bar{\rho}=\frac{\sum_{i=1}^{M}\frac{\lambda_i'}{\mu' C_i}}{M}$$

Where once again we let C_i = capacity of the ith channel. The time it takes to transmit a full packet averaged over all channels in the network is

$$\bar{S}_F=\sum_{i=1}^{M}\left[\frac{\lambda_i}{\lambda}\frac{1}{\mu_F C_i}\right]$$

where $1/\mu_F$ = the length of a full packet.
Thus, we will use the following approximation for $\bar{\tau}$:

$$\bar{\tau}=\frac{\bar{\rho}(1-\bar{\rho}^{(\bar{n}-1)})}{1-\bar{\rho}}\bar{S}_F$$

Removing the assumptions of constant K and C, adding the HOST transmission time, and assuming that the last packets of multipacket messages have the same mean length as the single packet messages, we have the average message delay for multipacket messages:

$$T_{MP}=\sum_{i=1}^{M}\left[\frac{\lambda_i}{\gamma}\left(\frac{1}{\mu_F C_i}+P_i+K_l+\frac{\lambda_i'/(\mu' C_i)}{\mu' C_i-\lambda_i'}\right)\right]$$

$$+\sum_{i=1}^{M}\left[\frac{\lambda_i}{\lambda}\left((\bar{m}-2)\frac{1}{\mu_F C_i}+\frac{1}{\mu C_i}\right)\right]$$

$$+\sum_{j=1}^{N}\left[\frac{\gamma_{\cdot j}}{\gamma}\left(K_j+\frac{1}{\mu_{FH}C_{Hj}}\right)\right]$$

$$+(\bar{m}-1)\bar{\tau}$$

where $1/\mu_{FH}C_{Hj}$ = the transmission time of a full packet to a HOST at site j.

Let β be the fraction of the total number of messages which are single packet messages. We obtain the final expression for average message delay (from source to destination) in the network.

$$T=\beta T_{SP}+(1-\beta)T_{MP}$$

The measure of delay which is supplied by the IMPs is round-trip delay. Therefore, in order to compare the model with the measurements we need an expression for round-trip delay (i.e., we need to include the average RFNM delay T_{RFNM} in the model). A RFNM is simply another single packet message traveling from destination to source. Thus, it experiences the single packet message delay T_{SP} with an appropriate value for μ and λ without the HOST transmission term as follows:

$$T_{RFNM}=\sum_{i=1}^{M}\left[\frac{\lambda_{Ri}}{\gamma}\left(\frac{1}{\mu_R C_i}+P_i+K_l+\frac{\lambda_i'/(\mu' C_i)}{\mu' C_i-\lambda_i'}\right)\right]$$

$$+\sum_{j=1}^{N}\left[\frac{\gamma_{j\cdot}}{\gamma}K_j\right]$$

where

λ_{Ri} = the mean arrival rate of RFNMs to channel i,
$1/\mu_R$ = the length of a RFNM, and
$\gamma_{j\cdot}$ = the mean departure rate of RFNMs from the network to the HOSTs at site j (= the mean arrival rate of messages from the HOSTs at site j to the network)

The expression for mean round-trip delay T_R is therefore,

$$T_R=T+T_{RFNM}$$

For the week-long measurement we calculated the zero-load value of T_R and obtained $T_R=69$ msec; the hourly variation of this quantity is shown in Figure 20. The source of the variation is the shift in the origin-destination traffic mix. This zero-load case corresponds to forcing λ_i and γ to zero, (keeping the same ratio as before for each i). The zero load value must be less than the measured value, and compares with the measurements displayed in Figure 21. This emphasizes the fact that the network is introducing very small congestion

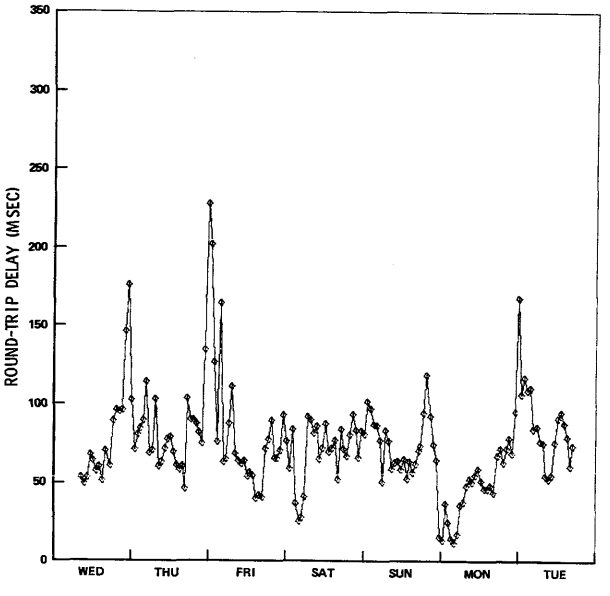

Figure 20—Computed (zero load) average message delay

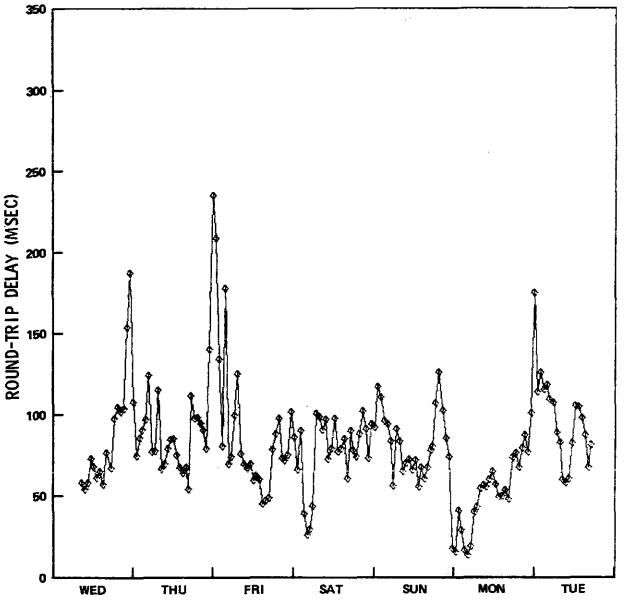

Figure 22—Computed (measured load) average message delay

effects. Furthermore, in Figure 22 we show the hourly variation of T_R (whose weekly average was $T_R=73$ msec) calculated for the actual load value as measured.

The model presented above is rather complex due mainly to the fact that not all channels (or IMPs) need have the same speed. In addition, the waiting time terms complicate the expressions as well, and represent the part of the model which is most subject to question (i.e., the Markovian assumptions). However, from Figures 20 and 22, we see that the zero-load and measured load calculations are nearly the same. This shows that the effect of W_i' is quite negligible and so any improvement over Markovian assumptions will yield negligible changes to T_R. This suggests a far simpler no-load model for estimating T_R as follows.[17] The expressions for T_{SP} (and T_{RFNM} which is similar in form), may be simplified by dropping the W_i' terms, and setting all $K_i=K$ (a constant), all $C_i=C$ (a constant at 50 KBPS), and $C_{Hj}=C_H$ (a constant at 100 KBPS). The result is

$$\hat{T}_{SP}=\bar{n}\left(\frac{1}{\mu C}+K\right)+K+\frac{1}{\mu_H C_H}+\sum_{i=1}^{M}\left[\frac{\lambda_i}{\gamma}P_i\right]$$

(and a similar expression for $\hat{T}_{RFNM}$). Except for the last summation, these parameters are easily computed. For the sum, one must estimate (or measure) the channel traffic λ_i and the network throughput γ. The propagation delays P_i are known constants. With these simplifications (and assuming $\beta=1$, since the measured value of $\beta=0.96$ was observed), we then have the approximation

$$\hat{T}_R=\hat{T}_{SP}+\hat{T}_{RFNM}$$

Our calculation gives $\hat{T}_R=70$ msec* which is an excellent approximation to the earlier stated value of $T_R=69$ msec (at zero-load) and $T_R=73$ (at measured load)!

On the other hand, the measured value of $T_R=93$ msec is significantly larger than measured load estimate of the model of $T_R=73$ msec. This difference is due to the fact the model does not include: any delay by the destination HOST in accepting the message; any delay due to the request for storage at the destination IMP; exact data on P_i; time variations in

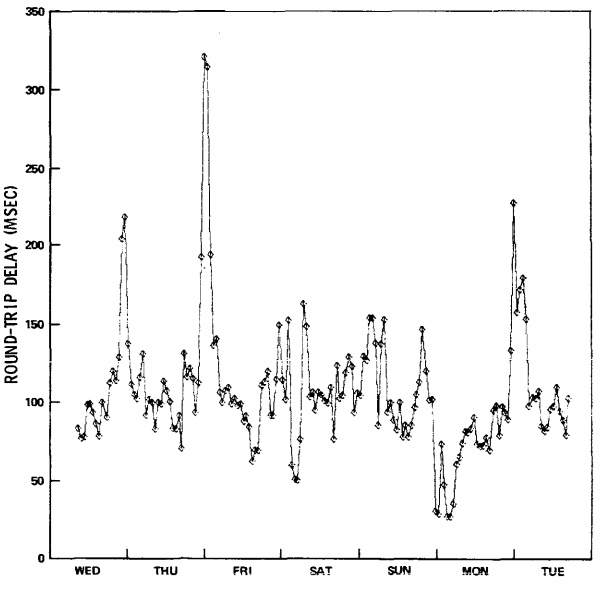

Figure 21—Measured average message delay

* The components for $\hat{T}_R$ are: $\bar{n}=3.31$, $1/\mu C=8.2$ msec, $K=.75$ msec, $1/\mu_H C_H=2.75$ msec, the propagation sum $=11.4$ msec, and $1/\mu_R C=3.36$ msec.

ρ finer than the hourly computations used; and non-Markovian assumptions. All the above omissions (except possibly the last) will increase the computed value of T_R.

CONCLUSIONS

The purpose of this paper was to present results of a week-long measurement of the ARPANET traffic behavior. In reporting upon the results of that experiment, we have observed a number of quantitative relationships which suggest that values assigned to certain of the network parameters should perhaps be reexamined. For example, we observed that the vast majority of messages are single-packet messages and one wonders at the wisdom of providing within the network the rather sophisticated mechanisms for handling multi-packet messages. Furthermore, we observed that the single-packet messages themselves are extremely small and it may be possible to improve the efficiency of the network if, in fact, the maximum packet lengths (and, therefore, the IMP buffer length) were reduced; one source of these small packet lengths is the preponderance of interactive traffic which typically creates packets containing one or a few characters. The mode of communication perhaps itself needs to be re-examined in an attempt to improve the network efficiency while maintaining a comfortable interactive feeling and response time. Incest is rampant in the network and it might be worthwhile to investigate other means for handling such traffic. Favoritism is (and perhaps will remain) even more dominant and how one would take advantage of this effect is not at this point clear. The non-uniformity of the traffic is striking and future network designs should attempt to capitalize upon this feature. The time variation of network use was discussed above and we see a fairly cyclic behavior both in traffic intensity and type of use. The lines themselves are not heavily utilized, and at the same time the network delays are so small as to render the network invisible to the typical user. We have described, in addition, a fairly extensive model for network delay and comparing it to our measured results it seems to be a fairly valid model both for single-packet and multi-packet messages. We also give a simplified model which appears adequate.

In this paper, our major purpose has been to report the measured results from our experiment. Secondarily, we have scratched the surface in attempting to evaluate and draw conclusions regarding the chosen values for some of the design parameters. In this effort, we have avoided the depth of discussion required to make a meaningful evaluation of these parameters, but rather have discussed their values only in terms of the measured data. For example, the choice of IMP buffer size depends upon many considerations beyond those we have measured (e.g., IMP processing speed, interrupt structure, line error rates, maximum network throughput, etc.); therefore, the presentation and commentary on the measured data given herein should certainly not be used alone in the selection of network parameters. The broad class of issues which must be included in decisions of this type are discussed, for example, in Kahn.[18]

The experiment described above is repeated every two months at the Network Measurement Center, and has so far produced results similar in flavor to those reported upon here. Numerous other experiments are currently being conducted and many more are in the planning stages. It is only through such experiments and through careful evaluation of the measured data that one can gain understanding of the network behavior, which in turn impacts the design and growth of the network.

ACKNOWLEDGMENT

We gratefully acknowledge the contributions of Holger Opderbeck, Stanley Lieberson, and the remainder of the staff at UCLA-NMC.

REFERENCES

1. Roberts, L. G., and B. D. Wessler, "Computer network development to achieve resource sharing," *AFIPS Conference Proceedings*, 36, pp. 543-549, SJCC, Atlantic City, N.J., 1970.
2. Heart, F. E., R. E. Kahn, S. M. Ornstein, W. R. Crowther, and D. C. Walden, "The interface message processor for the ARPA computer network," *AFIPS Conference Proceedings*, 36, pp. 551-567, SJCC, Atlantic City, N.J., 1970.
3. Kleinrock, L., "Analytic and simulation methods in computer network design," *AFIPS Conference Proceedings*, 36, pp. 569-579, SJCC, Atlantic City, N.J., 1970.
4. Frank, H., I. T. Frisch, and W. Chou, "Topological considerations in the design of the ARPA computer network," *AFIPS Conference Proceedings*, 36, pp. 581-587, SJCC, Atlantic City, N.J., 1970.
5. McKenzie, A.A., *HOST/HOST Protocol for the ARPA Network* ARPA Network Information Center # 8246, January 1972.
6. Ornstein, S. M., F. E. Heart, W. R. Crowther, H. K. Rising, S. B. Russell, and A. Michel, "The Terminal IMP for the ARPA network," *AFIPS Conference Proceedings*, 40, pp. 243-254, SJCC, Atlantic City, N.J., 1972.
7. Frank, H., R. E. Kahn, and L. Kleinrock, "Computer communication network design—Experience with theory and practice," *AFIPS Conference Proceedings*, 40, pp. 255-270, SJCC, Atlantic City, N.J., 1972.
8. Crocker, S. D., J. F. Heafner, R. M. Metcalfe, and J. B. Postel, "Function-oriented protocols for the ARPA Computer Network," *AFIPS Conference Proceedings*, 40, pp. 271-279, SJCC, Atlantic City, N.J., 1972.
9. *Specifications for the Interconnection of a HOST and an IMP*, Bolt, Beranek and Newman, Inc., Cambridge, Mass., Report No. 1822 May 1969.
10. McKenzie, A. A., B. P. Cosell, J. M. McQuillan, and M. J. Thrope, "The Network Control Center for the ARPA Network," *Proceedings of the First International Conference on Computer Communication*, 1, pp. 185-191, Washington, D. C., October, 1972.
11. Cole, G. D., *Computer Network Measurements: Techniques and Experiments*, Engineering Report No. UCLA-ENG-7165, University of California, Los Angeles, Calif., 1971.
12. Kleinrock, L., *Communication Nets: Stochastic Message Flow and Delay*, McGraw Hill, N.Y., 1964, reprinted by Dover, N.Y., 1972.
13. Gerla, M., *The Design of Store-and-Forward (S/F) Networks for Computer Communications*, Engineering Report No. UCLA-ENG-7319, University of California, Los Angeles, Calif., 1973.
14. McKenzie, A. A., Letter to S. D. Crocker, 16 January 1974.
15. Fultz, G. L., *Adaptive Routing Techniques for Message Switching Computer-Communication Networks*, Engineering Report No. UCLA-ENG-7252, University of California, Los Angeles, Calif., 1972.
16. Kleinrock, L., *Queueing Systems, Volume I: Theory*, Wiley, N.Y., 1974.
17. Kleinrock, L., *Queueing Systems, Volume II: Computer Applications*, Wiley, N.Y., 1974.
18. Kahn, R. E., "Resource-Sharing Computer Communication Networks," *Proceedings of the IEEE*, 60, pp. 1397-1407, November 1972.