PERFORMANCE MODELS AND MEASUREMENTS OF THE ARPA COMPUTER NETWORK*

by

Leonard Kleinrock
Computer Science Department
University of California, Los Angeles

ABSTRACT

The ARPA computer communications network has been in existence for almost three years, and in this paper we summarize some of the performance evaluation methods which have been found useful in the design of that network. Analytical, simulation, and measurement results are described which have enabled us to predict as well as to evaluate system performance. Some of the remaining problems are also described.

*This research was supported by the Advanced Research Projects Agency of the Department of Defense under Contract No. DAHC-15-69-C-0285.

1. INTRODUCTION

The purpose of this paper is to describe some of the analytical modeling procedures which have been found useful in predicting the performance of the ARPA Computer Network and to relate these predictions to simulation and measurement results for that network. This material is presented mainly as a survey with ample reference to the published literature for further details.

In 1967 Roberts [1] proposed an experimental computer network which was later to become the ARPA Net. In September 1969 the network came to life when the first Interface Message Processor (IMP)* was connected to the UCLA Sigma-7 computer. Thus began the interconnection of many main processors (referred to as HOSTs) at various university, industrial and government research centers across the United States. The message switching service itself consists of the IMPs and high-speed (50 kilobit per second) lines linking them. In 1970, a series of five papers was presented at the Spring Joint Computer Conference in Atlantic City which summarized what was known about the network at that time [2-6]. The evolution of that network from a small four-node net in 1969 to the proposed 34-node net later this year is shown in Fig. 1. At this year's Spring Joint Computer Conference an additional five papers were given which summarized our experiences with this network [7-11].

2. NETWORK CONSIDERATIONS

Let us discuss a few of the salient features of the ARPA Network and also list some of its design goals. The HOST computers generate messages (whose maximum size is approximately 8000 bits) which are delivered to their local IMPs; the IMPs further partition these messages into packets (whose maximum size is approximately 1000 bits) and these packets make their way through the network independently in a store-and-forward fashion. Upon arrival at the "destination IMP," these packets are then reassembled into the original messages and delivered to the destination HOSTs. The message service generated by this collection of IMPs and 50 kilobit lines was originally specified to have the following properties:

- A communications cost of less than 30 cents per thousand packets (approximately a megabit).
- Average packet delays under 0.2 seconds through the net.
- Capacity for expansion to 64 IMPs without major hardware or software redesign.

---

*The message switching computer used in the ARPA Network.

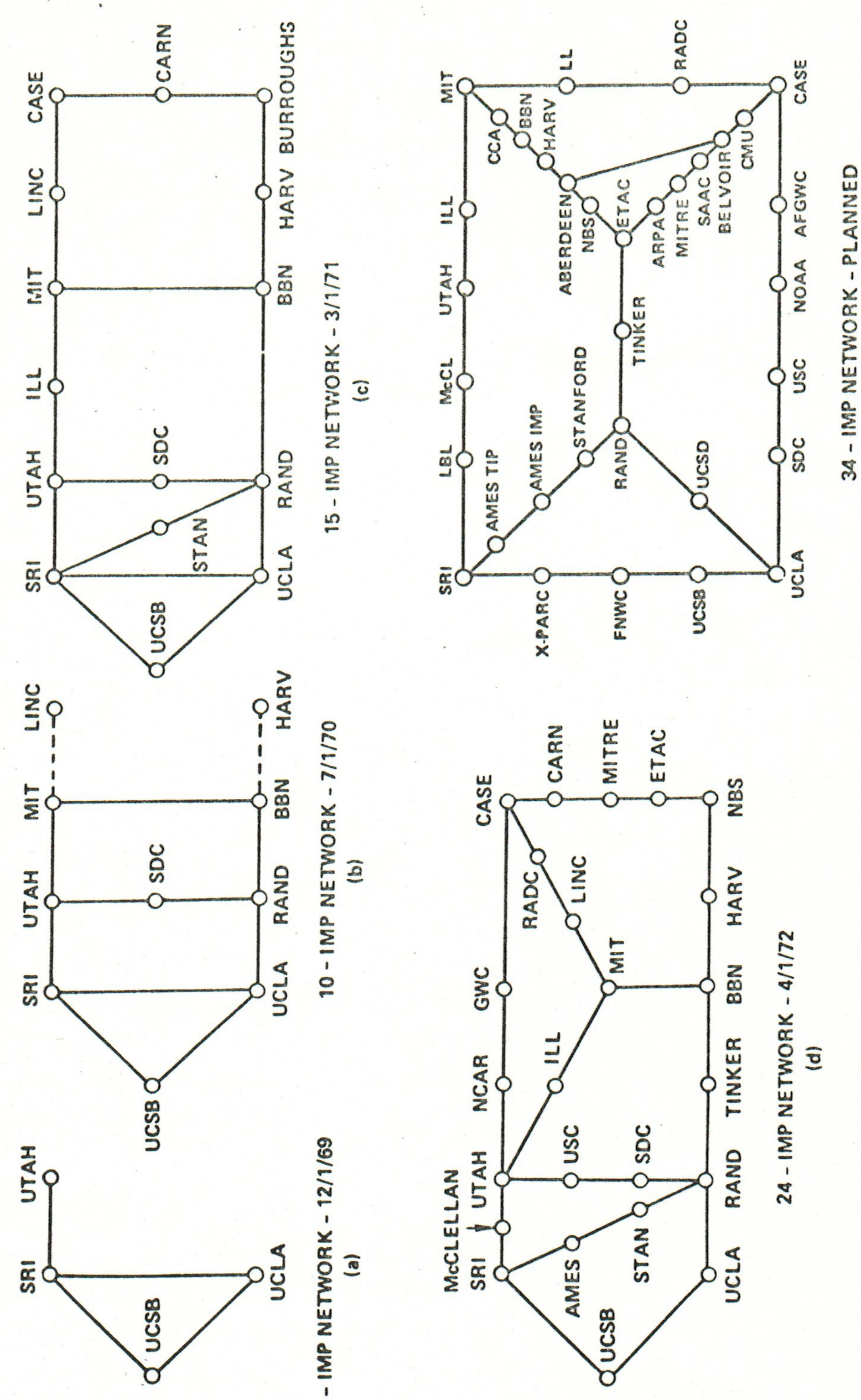

Fig. 1. Evolution of the ARPA Network

- Average total throughput capability of 10-15 kilobits/second for all HOSTs at an IMP.
- Peak throughput capability of 85 kilobits/second per pair of IMPs in an otherwise unloaded network.
- Transparent communications with error rates of one bit in 10^{12} or less.
- Approximately 98% availability of any IMP and close to 100% availability of all operating IMPs from any operable IMP.

We do not discuss the IMP design in this paper, nor do we discuss the HOST-HOST protocol for establishing connections and transmissions at that level. Rather, we are interested in the delay messages experience, in the routing procedures that handle the messages as they flow through the network, and in the throughput that the network delivers. The topology of the network has been sub-optimized by means of specially designed heuristic programs which achieve low cost and reasonably high throughput. The 50 kilobit lines provide rapid transmission for the short messages, as well as high throughput for the long ones. Network reliability is enhanced by providing two independent paths between each pair of IMPs. The basic optimization method was described in Ref. [5]. The choice of the particular heuristic used for exchanging connections within the network appears not to be critical; the important aspect of the algorithm for the topological design is how rapidly it can evaluate a proposed configuration. The topologies shown in Fig. 1 have been generated by this procedure.

The major modeling effort thus far has been the study of the behavior of networks of queues [12]. This emphasis is logical since in message switched systems, messages experience queueing delays as they pass from node to node and thus a significant performance measure is the speed at which messages can be delivered. The models, which at all times were recognized to be idealized statements about the real network, were nonetheless crucial to the ARPA Net topological design effort since they afforded the only known way to quantitatively predict the properties of different routing schemes and topological structures. The models have been subsequently demonstrated to be very accurate predictors of network throughput (see below) and indispensible in providing analytical insight into the network's behavior.

The key to the successful development of tractable models has been to factor the problem into a set of simpler queueing problems; if one specializes the problem and removes some of the real constraints, then theory and analysis become useful in providing understanding, intuition and design guidelines for the original constrained problem. This approach uncovers global properties of network behavior, which provide keys to good heuristic design procedures and ideal performance bounds.

3. ANALYSIS, SIMULATION AND MEASUREMENT

The effort to determine analytic models of system performance has been involved with predicting the average time delay encountered by a message as it passes through the network, and in using these queueing models to calculate optimum channel capacity assignments for minimum possible delay. The model used as a standard for the average message delay was first described in [12] where it served to predict delays in stochastic communication networks. In [13], this model was modified to describe the behavior of ARPA-like computer networks, while in [4,15] it was refined further to apply directly to the ARPA Net.

3.1 The Single Server Queueing Model

Queueing theory [16] provides an effective set of analytical tools for studying packet delay. Much of this theory considers systems in which messages place demands for transmission (service) upon a single communication channel (the single server). These systems are characterized by $A(\tau)$, the distribution of interarrival times between demands and $B(t)$, the distribution of service times. When the average demand for service is less than the capacity of the channel, the system is said to be stable.

When $A(\tau)$ is exponential (i.e., Poisson arrivals), and packets are transmitted on a first-come-first-served basis, the average time T in the stable system is

$$T = \frac{\lambda \overline{t^2}}{2(1 - \rho)} + \overline{t} \qquad (1)$$

where λ is the average arrival rate of messages, $\overline{t}$ and $\overline{t^2}$ are the first and second moments of $B(t)$, respectively, and $\rho = \lambda \overline{t} < 1$. If the service time is also exponential, then

$$T = \frac{\overline{t}}{1 - \rho} \qquad (2)$$

When $A(\tau)$ and $B(t)$ are arbitrary distributions, the situation becomes complex and only weak results are available. For example, no expression is available for T; however, the following upper bound yields an excellent approximation [17] as $\rho \to 1$.

$$T \le \frac{\lambda(\sigma_a^2 + \sigma_b^2)}{2(1-\rho)} + \bar{t} \tag{3}$$

where σ_a^2 and σ_b^2 are the variance of the interarrival time and service time distribution, respectively.*

3.2 Networks of Queues

Multiple channels in a network environment give rise to queueing problems that are far more difficult to solve than single server systems. For example, the variability in the choice of source and destination for a message is a network phenomenon which contributes to delay. A principal analytical difficulty results from the fact that flows throughout the network are correlated. The basic approach to solving these stochastic network problems is to decompose them into analyzable single-server problems which reflect the original network structure and traffic flow.

Early studies of queueing networks indicated that such a decomposition was possible [18,19]; however, those results do not carry over to message switched computer networks due to the correlation of traffic flows. In [12] it was shown for a wide variety of communication nets that this correlation could be removed by considering the length of a given packet to be an independent random variable as it passes from node to node. Although this "independence" assumption is not physically realistic, it results in a mathematically tractable model which does not seem to affect the accuracy of the predicted time delays. As the size and connectivity of the network increases, the assumption becomes increasingly more realistic. With this assumption, a successful decomposition is possible which permits the following channel-by-channel analysis.

The packet delay is defined as the time which a packet spends in the network from its entry until it reaches its destination. The average packet delay is denoted as T. Let Z_{jk} be the average delay for those packets whose origin is IMP j and whose destination is IMP k.

---

*Approximation techniques are currently being developed which permit one to estimate system behavior in the overloaded (unstable) case ($\rho \ge 1$).

We assume a Poisson arrival process for such packets with an average of γ_{jk} packets per second and an exponential distribution of packet lengths with an average of $1/\mu$ bits per packet. With these definitions, if γ is the sum of the quantities γ_{jk}, then

$$T = \sum_{j,k} \frac{\gamma_{jk}}{\gamma} Z_{jk} \tag{4}$$

Let us now reformulate Eq. (4) in terms of single channel delays. We first define the following quantities for the i^{th} channel: C_i as its capacity (bits/second); λ_i as the average packet traffic it carries (packets/second); and T_i as the average time a packet spends waiting for and using the i^{th} channel. By relating the $\{\lambda_i\}$ to the $\{\gamma_{jk}\}$ via the paths selected by the routing algorithm, it is easy to see that [12]

$$T = \sum_{i} \frac{\lambda_i}{\gamma} T_i \tag{5}$$

With the assumption of Poisson traffic and exponential service times, the quantities T_i are given by Eq. (2). For an average packet length of $1/\mu$ bits, we have $\bar{\tau} = 1/\mu C_i$ seconds, and so

$$T_i = \frac{1}{\mu C_i - \lambda_i} \tag{6}$$

Thus we have successfully decomposed the analysis problem into a set of simple single-channel problems as given by these last two equations.

A refinement of the decomposition permits a non-exponential packet length distribution and uses Eq. (1) rather than Eq. (2) to calculate T_i; as an approximation, the Markovian character of the traffic is assumed to be preserved. Furthermore, for computer networks we include the effect of propagation time and overhead traffic to obtain the following equation for average packet delay [4,13].

$$T = K + \sum_i \frac{\lambda_i}{\gamma}\left[\frac{1}{\mu' C_i} + \frac{\lambda_i/\mu C_i}{\mu C_i - \lambda_i} + P_i + K\right] \tag{7}$$

Here, $1/\mu'$ represents the average length of a HOST packet, and $1/\mu$ represents the average length of all packets (including acknowledgments, headers, requests for next messages, parity checks, etc.) within the network. The expression $1/\mu' C_i + [(\lambda_i/\mu C_i)/(\mu C_i - \lambda_i)] + P_i$ represents the average packet delay on the i^{th} channel. The term $(\lambda_i/\mu C_i)/(\mu C_i - \lambda_i)$ is the average time a packet spends waiting at the IMP for the i^{th} channel to become available. Since the packet must compete with acknowledgments and other overhead traffic, the overall average packet length $1/\mu$ appears in the expression. The term $1/\mu' C_i$ is the time required to transmit a packet of average length μ'. Finally: K is the nodal processing time, assumed constant, and for the ARPA IMP approximately equal to 0.35 ms; P_i is the propagation time on the i^{th} channel (about 20 ms for a 3000 mile channel).

Assuming a relatively homogeneous set of C_i and P_i, no individual term in the expression for delay will dominate the summation until the flow λ_i/μ in one channel (say channel i_0) approaches the capacity C_{i_0}. At that point, the term T_{i_0}, and hence T will grow rapidly. The expression for delay is then dominated by one term and exhibits a threshold behavior. Prior to this threshold, T remains relatively constant.

This time delay model was demonstrated on a 19-node network [15] originally proposed (but never implemented) for ARPA and produced the results shown in Figs. 2 and 3. In Fig. 2 we see the single-packet message delay as a function of the load on the network (RE)*; Eq. (7) is plotted as a solid line and the simulation data is shown by dots. The agreement is remarkable. Moreover, we begin to see the threshold effect in which the packet delay rises much more rapidly in the network than

---

*RE = 1.0 corresponds to a total throughput (γ) of 225 kilobits/second using a non-uniform traffic matrix (see [13]).

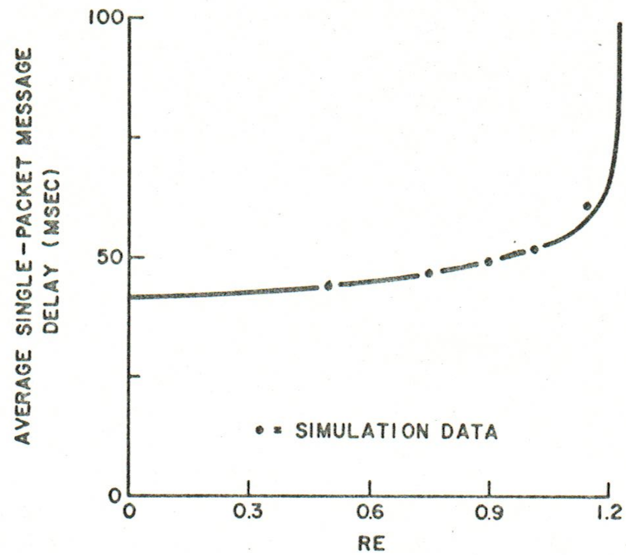

Fig. 2 Single-packet message delay as a function of network load.

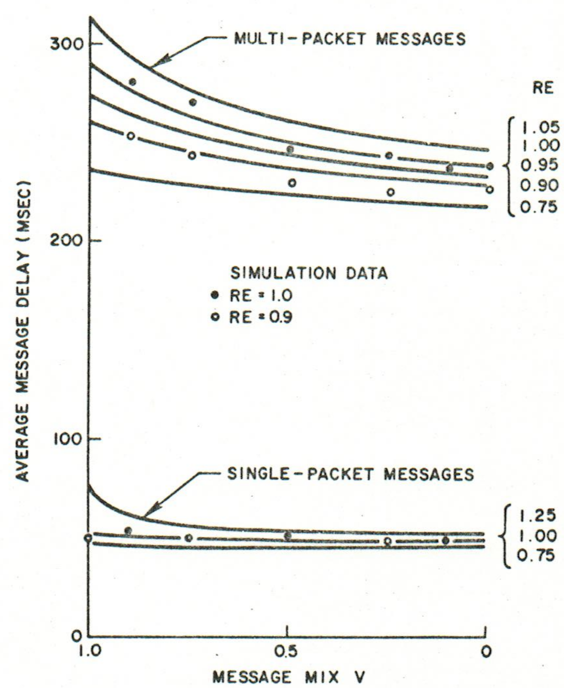

Fig. 3. Message delay for the 19-node net versus message mix.

would be predicted for a single node (as in Eq. (1)). Fig. 3 is concerned with multi-packets as well as single packets; recall that messages are decomposed into packets as they pass through the network, and a message consisting of more than one packet is referred to as a multi-packet message. For this simulation, it was assumed that single-packet messages had priority over multi-packet messages when competing for a channel. In Fig. 3 we plot message delay as a function of the percentage of traffic which is single-packet while maintaining the constant average bit rate into the network; when $V = 1.0$, we have all single-packets, whereas $V = 0$ corresponds to all multi-packets. A multi-packet model similar to Eq. (7) [20] provides the solid theoretical curves in this figure and again the dots represent simulation results. From these two figures it is clear that we have generated a suitable model for predicting average message delay.

For the ten-node ARPA Net derived from Fig. 1(c) by deleting the five rightmost IMPs, and using equal traffic between all node pairs, the channel flows λ_i were found using a simple routing algorithm, and the delay curves shown in Fig. 4 were calculated. Curve A was obtained with fixed 1000 bit packets,* while curve B was generated for exponentially distributed variable length packets with average size of 500 bits. Note that the delay remains small until a total throughput slightly greater

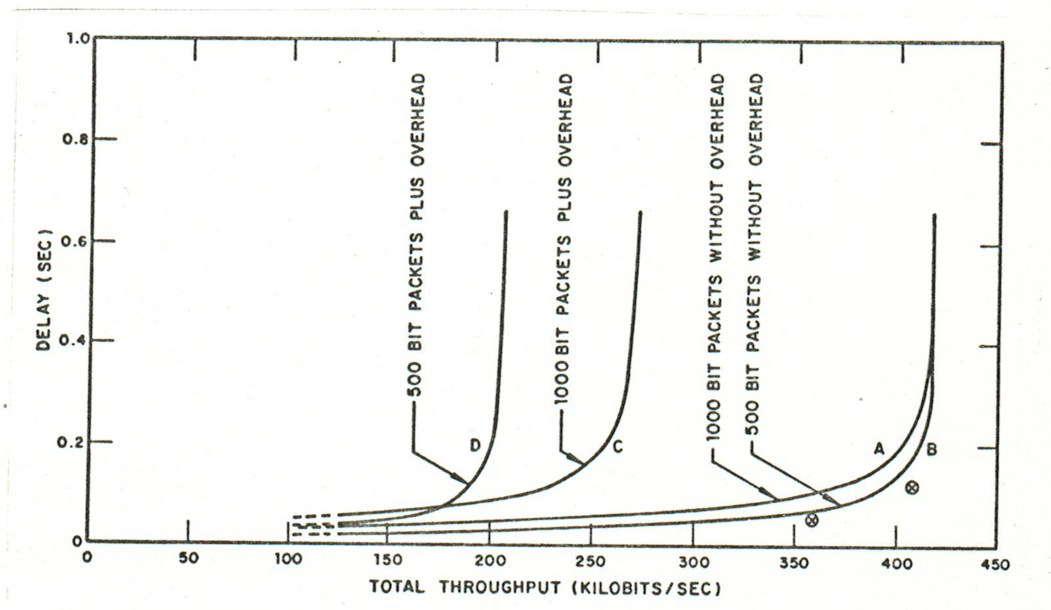

Fig. 4. Delay as a function of throughput (network load).

*In case A, the application of Eq. (1) allows for constant packet lengths (i.e., zero variance).

than 400 kilobits/second is reached. The delay then increases rapidly. Curves C and D, respectively, represent the same situations when the overhead of 136 bits per packet and per RFNM (see below) and 152 bits per acknowledgment are included. Notice that the total throughput per IMP is reduced to 250 kilobits/second in case C and to approximately 200 kilobits/second in case D. In the same figure, we have illustrated with ⓧ's the results of a simulation performed with a realistic routing and metering strategy. The simulation assumed fixed lengths of 1000 bits for all packets. It is notable that the delay estimates from the simulation (which used a dynamic routing strategy) and the computation (which used a static routing strategy and the message delay equation) are in close agreement. In particular, they both accurately determined the vertical rise of the delay curve in the range just above 400 kilobits/second, the formula by predicting infinite delay and the simulation by rejecting the further input of traffic.

In practice and from the analytic and simulation studies of the ARPA Net, the average queueing delay is observed to remain small (almost that of an unloaded net) and well within the design constraint of 0.2 seconds until the traffic within the network approaches the capacity of a cutset. The delay then increases rapidly. Thus, as long as traffic is low enough and the routing adaptive enough to avoid the premature saturation of cutsets by guiding traffic along paths with excess capacity, queueing delays are not significant.

Another quantity of interest is the time separation at the destination IMP between packets of the same message. This inter-packet gap time has been studied in [21] and shows under certain simplifying assumptions that the average gap size increases with the message path length, but reaches a limiting value as a function of path length.

3.3 Routing Procedures

An efficient message routing procedure is an essential ingredient for the successful operation of a computer network. The function of a routing procedure is to direct the message traffic along paths within the network in a fashion which avoids congestion. In conjunction with a routing procedure one must provide methods for controlling the flow of traffic entering the network which would otherwise congest the system. It is not difficult to conceive of "good" routing procedures, and even of "fair" flow control procedures; however, at this time there are virtually no optimum procedures known. In what follows we describe some simulation and measurement experiments which have been performed for the ARPA Network as it currently functions. Some alternative algorithms have been described in [22] and are to be implemented shortly as the operating procedure for the network; we have little experience with these new procedures and therefore defer commenting on their effectiveness until the appropriate measurements have been made.

The basic requirements for a good routing procedure are as follows:

- It should insure rapid delivery of messages.
- It should adapt to changes in the network topology resulting from nodal and channel failures.
- It should adapt to varying source-destination traffic loads.
- It should route packets away from temporarily congested nodes within the network.

In [15,20] a number of routing algorithms have been defined and classified. It has been possible to generate approximate lower bounds on the message delay as a function of load for various of these algorithms, and in Fig. 5 this lower envelope is shown. This figure corresponds to the 19-node network mentioned earlier. Shown also is the performance of a routing procedure similar to the one currently used in the ARPA Network, namely a periodic update algorithm (PUA1). We see that its performance deteriorates as the network load grows. Procedures using an asynchronous update algorithm (AUA) or the shortest queue plus bias algorithm (SQ+BA) are seen to be superior, although they are more costly in terms of the processing load they place upon the IMP. A good candidate for a routing algorithm is the shortest queue plus bias plus periodic update algorithm (SQ+B+PUA) which is reasonably simple to implement and not too costly in processing load.

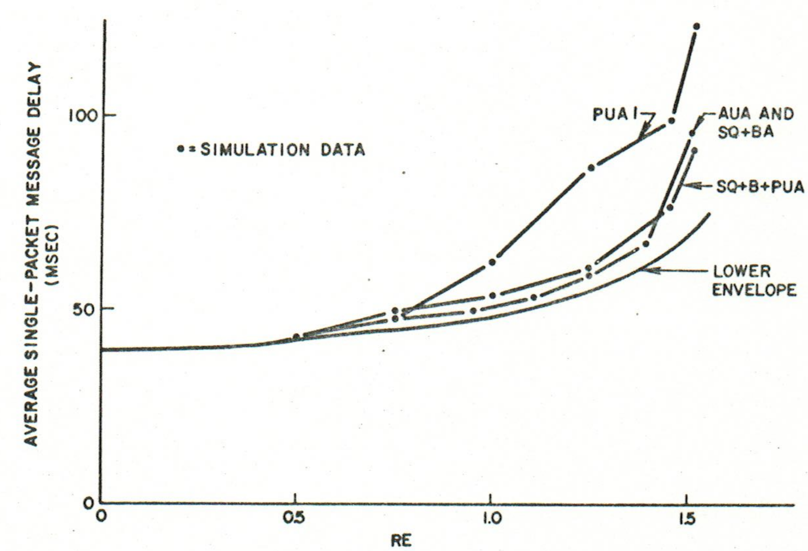

Fig. 5. Comparison of routing algorithm performance as a function of network load.

3.4 Network Measurements

Measurement of network activity is of major importance in the ARPA Net. These measurements provide information regarding the traffic on the network, provide means for testing the limits of its traffic-carrying capacity, provide a method for validating and improving the theoretical models, as discussed herein, and lastly, may eventually provide an on-line mechanism for controlling network traffic and routing. Measurements of round trip message delay and throughput have been conducted at UCLA, and we wish now to compare these measurement results with some of the theoretical and simulation results we have so far discussed. In [21] it is shown that a simple calculation for the expected round trip delay between two nodes in the network (SRI-Utah) may be calculated to give 20 + 0.48L milliseconds where L is the length of the message (in IMP words) travelling this round trip; this equation assumes no other traffic exists in the network and therefore provides a strict lower bound on such delays. In Fig. 6 we give the measured results for this round trip delay, along with the theoretical equation just described; the calculations are shown to be accurate, and the delays above the minimum are due to interfering traffic within the network. A throughput measurement made for the network of Fig. 1(c) is shown in Fig. 7. Here we see the flow of traffic from UCLA to UCSB. This test was made using an artificial traffic generator at UCLA which has the

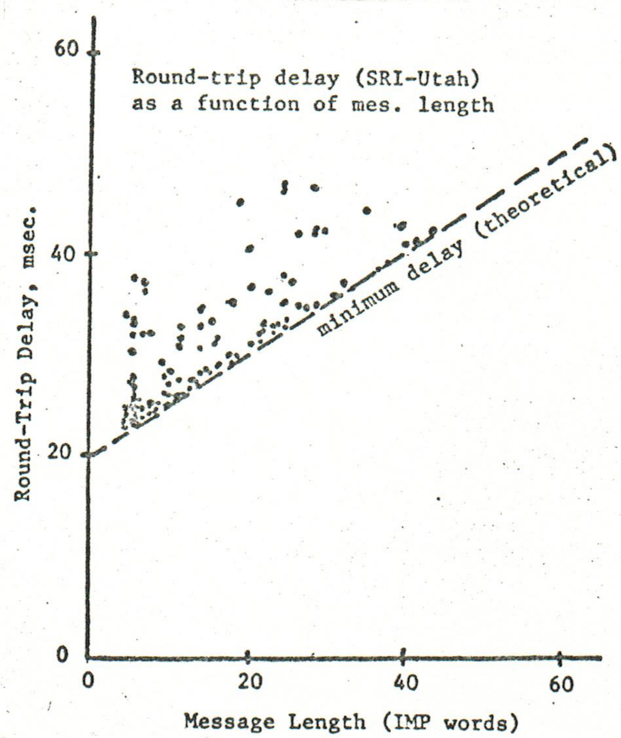

Fig. 6. Round-trip delay measurements of the SRI-Utah traffic.

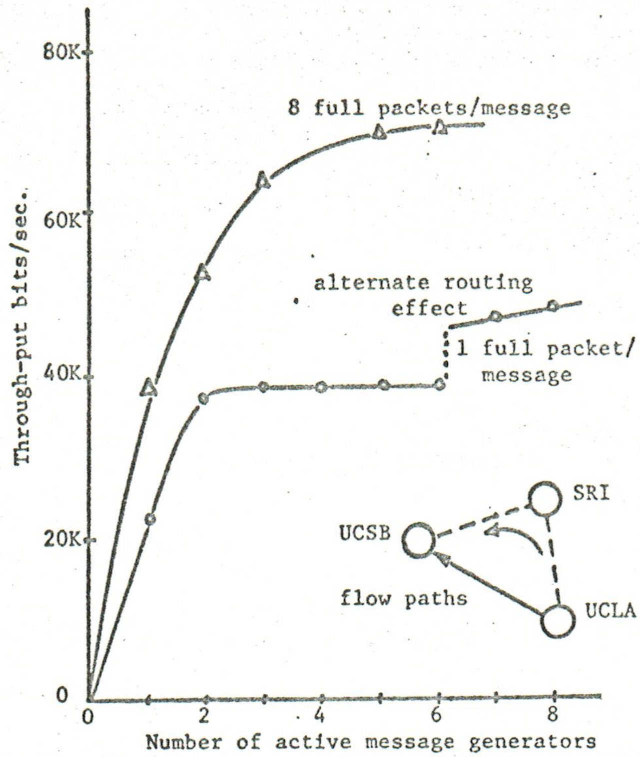

Fig. 7. Throughput measurements between UCLA and the neighboring UCSB IMP.

capability of creating more than one "conversation" between a pair of sites. The current network flow control procedure constrains each conversation to have at most one message in the network at any time; only when that message is delivered may the next message be generated in this conversation. That is, when a message reaches its destination, the flow control procedure sends a message (known as a RFNM, a "request for next message") from the destination IMP back to the source IMP which signals its readiness to accept a new message into the network. In Fig. 7 we see the effect of this control procedure. The lower curve corresponds to single packet messages, and we see the growth in throughput as the number of generators (conversations) increases, reaching a plateau when two or more generators are in operation; this plateau occurs at approximately 40 kilobits/second, which corresponds to the effective data rate permissible on a single 50 kilobit line when overhead considerations are accounted for. This level of traffic is not obtainable with a single message generator due to delays created by the RFNM. Note that the routing procedure permits alternate routing to take place when more than six generators are active; at this point a large enough queue is formed at UCLA to trigger the introduction of alternate routing. The upper curve in Fig. 7 corresponds to full length messages composed of eight packets each. Here, we do not expect the RFNM to constrain the throughput in a significant way since it comes only once in every eight packets now, and we note that a single message generator immediately achieves the "plateau" value of throughput seen earlier. In this multi-packet mode we observe that its plateau is reached at about 70 kilobits/second; this is below the expected 80 kilobit/second for the two paths in use since the finite storage capacity of the IMP was saturated at this point.

This last throughput measurement was carried out in the presence of no interfering traffic from other sources. A more interesting experiment is to observe the mutual interference among competing conversations within the network. This was done for a network similar* to that in Fig. 1(d) in which UCLA was attempting to send message traffic to RAND. Measurements were taken at UCLA (which is the Network Measurement Center) as, one by one, additional nodes were instructed to send interfering traffic to RAND (each IMP has the capability for generating artificial traffic of a simple type for such purposes). The order in which new IMPs entered this sequence is as follows:

Number of Senders		Next Additional IMP
1	→	UCLA
2	→	BBN
3	→	SDC
4	→	Stanford
5	→	Harvard
6	→	Utah
7	→	Ames
8	→	MIT
9	→	Illinois
10	→	BBNTIP
11	→	Case
12	→	Linc
13	→	Carnegie

In Fig. 8 we show the way in which the average round trip delay (from UCLA to RAND and back) varies with the number of interfering users. Three curves are shown, each for different message lengths; these are the lengths used for all messages generated within the network for this experiment. We note for the single packet messages that essentially no interference occurred as the number of users increased;** however, with

*The measured network differed from that in Fig. 1(d) as follows: NCAR and GWC were not yet connected, and so there was a direct link from Utah to Case; furthermore, in series with the MIT-BBN link there was placed an additional node (BBNTIP).

**This result is conditional on the fact that UCLA is directly connected to RAND. Experiments with longer paths and with paths through "congested" nodes are currently being conducted.

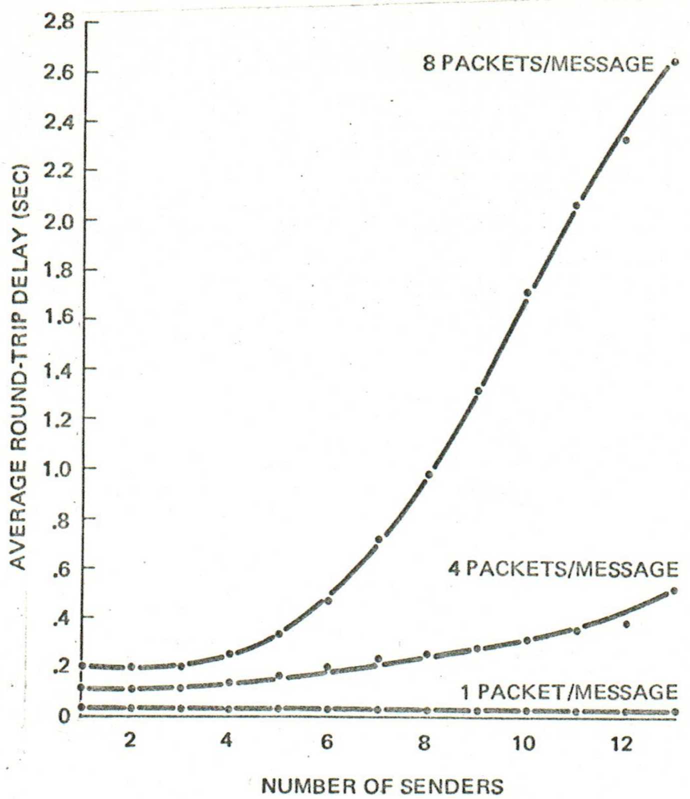

Fig. 8. Measured round-trip times.

four, and in particular with eight, packets per message, the interference caused significant round trip delays to the message traffic. A large part of this delay was due to the competition for reassembly storage space at the RAND IMP. In Fig. 9 we see a similar plot for the throughput per user. Again, the single-packet messages maintain an essentially constant data rate. When we go to four packets per message, and then to eight packets per message, we observe that the data rate increases so long as the number of senders is less than four. This increase in throughput is due to the large penalty paid by smaller messages for the flow control procedure and its use of RFNM's. The delay due to a RFNM comes once per packet for single-packet messages, but only once per eight packets for eight-packet messages, and this accounts for the increased throughput for a small number of interfering users. The reason that the multi-packet message throughput begins to deteriorate seriously when four or more senders are active is due to the finite storage capacity at each IMP which is set aside for the reassembly of multi-packet messages; at present, this storage is sufficient to hold approximately three eight-packet messages. From Figs. 8 and 9 we see the effect of this interference on the increase in delay and the decrease in throughput. However, one may argue that the total throughput in the network is increasing as the number of conversations increase. In Fig. 9 we displayed the throughput per user as a function of number of users. We may approximate the total throughput as a function of

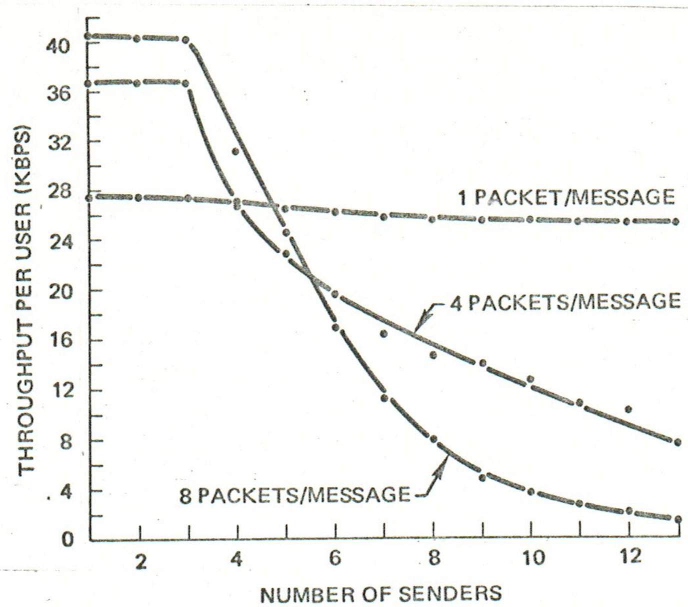

Fig. 9. Measured throughput per user.

number of users by taking each measured point in Fig. 9 and multiplying by the number of users to which that point corresponds; this, of course, is a fiction, since the measured points from Fig. 9 correspond to the throughput seen in the UCLA-RAND transmission only. However, this derived curve will provide some insight as to what is happening with regard to total throughput, and this we show in Fig. 10. We see that the single packet per message case yields a linearly increasing total throughput which is part of the fiction since, clearly, this function can rise only until it reaches the maximum throughput capability of the IMP itself. In the case of four packets per message, we see that the throughput saturates when three or more senders are active, and this saturation level occurs somewhat above 100 kilobits/second. The more interesting (albeit unfortunate) case is for eight packets per message in which the total throughput peaks at around four users and then begins to fall off in a disastrous fashion* as the network goes into what is known as reassembly lockup [22]. This again is the effect of the flow control procedure choking off the input traffic as messages take longer and longer to reassemble; in the limit, reassembly lockup occurs when partially reassembled messages cannot be completely reassembled since the congested network prevents their remaining packets from reaching their destination.

---

*This is similar to the fundamental diagram of road traffic [14] for which strong analogies may be drawn.

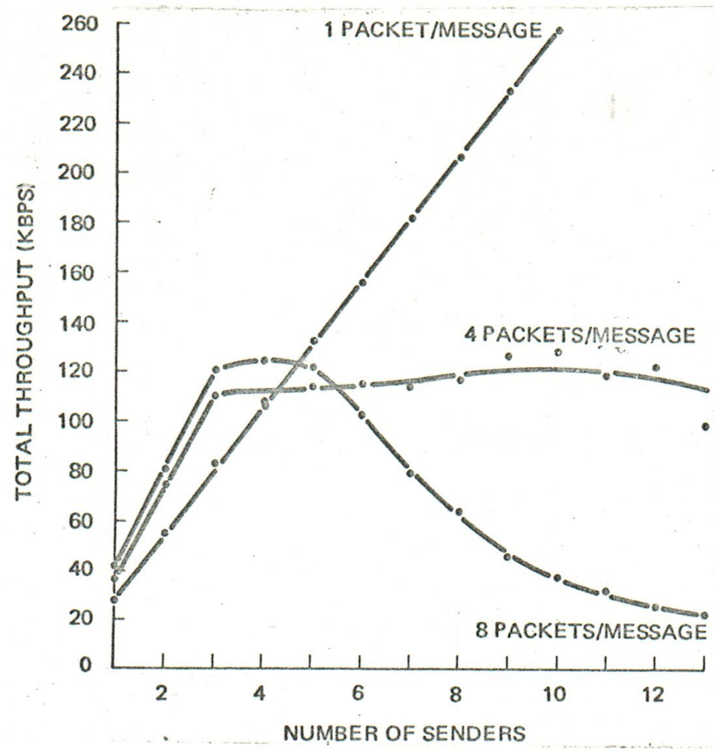

Fig. 10. Measured total throughput.

3.5 Network Optimization

One of the more difficult design problems is the optimal selection of capacities from a finite set of options. Although there are many heuristic approaches to this problem, analytic results are relatively scarce. (For the specialized case of centralized networks, an algorithm yielding optimal results is available [23].) While it is possible to find an economical assignment of discrete capacities for up to, say, a 200 IMP network, very little is known about the relationships among such capacity assignments, the message delay, and cost.

To obtain theoretical properties of optimal capacity assignments, we first ignore the constraint that capacities are obtainable only in discrete sizes. In [12] the problem was posed where the network topology and average traffic flow were assumed to be known and fixed and an optimal match of capacities to traffic flow was found. Also, the traffic was assumed to be Markovian (Poisson arrivals and exponential packet lengths) and the independence assumption and decomposition method were applied. For each channel, the capacity C_i was found which minimized the average message delay T. Since λ_i/μ is the average bit rate on the i^{th} channel, the solution to any optimal assignment problem must provide more than this minimal capacity to each channel. This is clear since both Eqs. (6) and (7) indicate that T_i will become arbitrarily large with less than (or equal to) this amount of capacity. It is not critical exactly how the excess capacity is assigned, as long as $C_i > \lambda_i/\mu$. The optimization further assumed that a total of D dollars was available to provide the channel capacities and that the cost of the i^{th} channel was linear at a rate of d_i dollars per unit of channel capacity; that is, $D = \Sigma d_i C_i$. The simpler form for T_i in Eq. (6) is used in this formulation and T is as given in Eq. (5). The solution to this problem assigns a capacity to the i^{th} channel in an amount equal to λ_i/μ plus some excess capacity proportional to the square root of that traffic. With T evaluated for this assignment,

$$T = \frac{\bar{n}}{\mu D_e}\left(\sum_i \sqrt{d_i(\lambda_i/\lambda)}\right)^2 \tag{8}$$

Here $\lambda = \Sigma\lambda_i$ represents the total rate at which packets flow within the net and D_e is the difference between D and the amount which must be spent to provide each channel with capacity λ_i/μ, namely

$$D_e = D - \sum_i \frac{\lambda_i d_i}{\mu} \tag{9}$$

Moreover, $\bar{n} = \lambda/\gamma$ is easily shown to represent the average path length for a packet.

If $d_i = 1$ for all channels, $D = \Sigma C_i = C$ where C represents the total capacity within the network.* In this case,

$$T = \frac{\bar{n}}{\mu C(1 - \bar{n}\rho)}\left(\sum_i \sqrt{\lambda_i/\lambda}\right)^2 \tag{10}$$

---

*The assumption $d_i = 1$ is of practical importance in the case of satellite channels [24].

Here $\rho = \gamma/\mu C$ is the ratio of the rate γ/μ at which bits enter the network to the rate C at which the net can handle bits. The quantity ρ represents a dimensionless form of network "load." As the load ρ approaches $1/\bar{n}$, the delay T grows very quickly, and the point $\rho = 1/\bar{n}$ represents the maximum load which the network can support. If capacities are assigned optimally, all channels saturate simultaneously at this point. In this formulation $\bar{n}$ is a design parameter which depends upon the topology and the routing procedure, while ρ is a given parameter which depends upon the input rate and the total capacity of the network. Equation (10) provides insight into topological structure and routing procedures [12].

In a recent paper [25], it was observed that, in minimizing T, a wide variation was possible among the line delays T_i. As a result, the problem of finding the sets of channel capacities which minimize $T^{(k)}$ was considered, where

$$T^{(k)} = \left[\sum_i \frac{\lambda_i}{\gamma} (T_i)^k\right]^{1/k} \tag{11}$$

The solution for the optimal channel capacity assignment with a given value k, denoted by $C_i^{(k)}$ is

$$C_i^{(k)} = \frac{\lambda_i}{\mu} + \left(\frac{D_e}{d_i}\right) \frac{(\lambda_i d_i^{\,k})^{1/k+1}}{\sum_j (\lambda_j d_j^{\,k})^{1/k+1}} \tag{12}$$

With this capacity assignment,

$$T^{(k)} = \left(\frac{\bar{n}}{\mu D_e}\right) \left(\sum_i (\lambda_i d_i^{\,k}/\lambda)^{1/1+k}\right)^{1+k} \tag{13}$$

Note that the assignment $C_i^{(1)}$ is the previously mentioned assignment which minimizes T, and $T^{(1)}$ is the previous value for T in Eq. (8). As k increases, the variation in the T_i decreases and as $k \to \infty$

$$\lim_{k\to\infty} C_i^{(k)} = \frac{\lambda_i}{\mu} + \frac{D_e}{\sum_j d_j} \tag{14}$$

In this limit, the channel capacity is assigned to give each channel its minimum required amount λ_i/μ plus a <u>constant</u> additional amount; all the T_i are equal, and so

$$T = \frac{\bar{n}}{\mu D_e} \sum_j d_j \tag{15}$$

On the other hand, setting $k = 0$ yields

$$C_i^{(0)} = \frac{\lambda_i}{\mu} + \frac{\lambda_i D_e}{\bar{n}\gamma d_i} \tag{16}$$

For this assignment, the value of T is identical to the value it achieves in Eq. (15) for $k \to \infty$, although different channel capacity assignments typically occur at these extremes [26]. If all $d_i = 1$ a channel capacity is assigned in proportion to the traffic carried by that channel (commonly known as the proportional capacity assignment). Although the value of T is minimized for the capacity assignment which results when $k = 1$, T increases rather slowly as k varies from unity and, moreover, the variance of packet delay is minimized when $k = 2$.

In studying the ARPA Network [4] a closer representation of the actual tariffs for high speed telephone data channels used in that network was provided by setting $D = \Sigma d_i C_i^{\alpha}$ where $0 \le \alpha \le 1$.* This approach requires the solution of a non-linear equation by numerical techniques. On solving the equation, it can be shown that the packet delay T varies insignificantly with α for $.3 \le \alpha \le 1$. This indicates that the closed form solution discussed earlier with $\alpha = 1$ is a reasonable approximation to the more difficult non-linear problem.

---

*Of course the tariffs reflect the discrete nature of available channels. The use of the exponent α provides a continuous fit to the discrete cost function. For the ARPA Net, $\alpha \cong .8$.

In practice, the selection of channel capacities must be made from a small finite set and not from a continuum as assumed above. Although some theoretical work has been done in this case by approximating the discrete cost-capacity functions by continuous ones, much remains to be done [27-29]. Because of the discrete capacities and the time varying nature of network traffic, it is not generally possible to match channel capacities to the anticipated flows within the channels. If this were possible, all channels would saturate at the same externally applied load. Instead, capacities are assigned on the basis of reasonable estimates of average of peak traffic flows. It is the responsibility of the routing procedure to allow the traffic to adapt to the available capacity [15]. Often two IMP sites will engage in heavy communication and thus saturate one or more critical network cutsets. In such cases, the routing will not be able to send additional flow across these cuts. The network will therefore experience "premature" saturation in one or a small set of channels leading to the threshold behavior described earlier.

4. CONCLUSIONS

In this paper we have surveyed some of the modeling techniques which have been found useful in the study of the ARPA Network. We have also reviewed some of the simulation and measurement results which, by and large, lend validity to the mathematical models we have used. We see that message switching networks of the ARPA type may be implemented in a rather straightforward fashion and provide an economical message service compared to other current techniques. We observe the heavy reliance in this paper on approximate methods, simulation and actual network measurement, which together permit us to understand some aspects of network behavior. Heuristics are also necessary in this process for the selection of good topologies and traffic flow assignments. The study of really large networks (hundreds of nodes) will undoubtedly require the development of new tools; it is clear that some clever decomposition or partitioning of the network into supernodes and regions (perhaps in a hierarchical structure) will be necessary.

Among the more difficult design problems which remain are: the specification of routing and flow control procedures; the design of optimal topologies; the optimal assignment of capacities with non-linear discrete cost functions; the consideration of large message-switching nodes judiciously placed in the network; and many other related questions. Considerable work is being expended in these directions, and we hope that some of these questions will be answered in the near future.

Acknowledgments

The author takes great pleasure in acknowledging the assistance rendered by Gary L. Fultz in the simulation experiments and by Gerald D. Cole, William E. Naylor, and Vinton G. Cerf in the measurement experiments.

REFERENCES

1. Roberts, L. G., "Multiple Computer Networks and Inter-Computer Communications," Proc. of the ACM Symposium on Operating Systems, Gatlinburg, Tenn., 1967.

2. Roberts, L. G., and B. Wessler, "Computer Network Development to Achieve Resource Sharing," AFIPS Conference Proceedings, Spring Joint Computer Conference 1970, 36:543-549.

3. Heart, F., R. Kahn, S. Ornstein, W. Crowther and D. Walden, "The Interface Message Processor for the ARPA Computer Network," AFIPS Conference Proceedings, Spring Joint Computer Conference 1970, 36:551-567.

4. Kleinrock, L., "Analysis and Simulation Methods in Computer Network Design," AFIPS Conference Proceedings, Spring Joint Computer Conference 1970, 36:569-579.

5. Frank, H., I. T. Frisch, and W. Chou, "Topological Considerations in the Design of the ARPA Computer Network," AFIPS Conference Proceedings, Spring Joint Computer Conference 1970, 36:581-587.

6. Carr, S., S. Crocker, and V. Cerf, "Host-Host Communication Protocol in the ARPA Network," AFIPS Conference Proceedings, Spring Joint Computer Conference 1970, 36:589-597.

7. Crocker, S. D., J. F. Heafner, R. M. Metcalfe, and J. J. Postel, "Function-Oriented Protocols for the ARPA Computer Network," AFIPS Conference Proceedings, Spring Joint Computer Conference 1972, 40:271-279.

8. Frank, H., R. Kahn, and L. Kleinrock, "Computer-Communication Network Design -- Experience with Theory and Practice," AFIPS Conference Proceedings, Spring Joint Computer Conference 1972, 40:255-270.

9. Ornstein, S. M., F. E. Heart, W. R. Crowther, H. K. Rising, S. B. Russell, and S. Michel, "The Terminal IMP for the ARPA Computer Network," AFIPS Conference Proceedings, Spring Joint Computer Conference 1972, 40:243-254.

10. Thomas, R. H., and D. A. Henderson, "McRoss -- A Multi-Computer Programming System," AFIPS Conference Proceedings, Spring Joint Computer Conference 1972, 40:281-293.

11. Roberts, L. G., "Extensions of Packet Communication Technology to a Hand-Held Personal Terminal," AFIPS Conference Proceedings, Spring Joint Computer Conference 1972, 40:295-298.

12. Kleinrock, L., Communication Nets: Stochastic Message Flow and Delay, McGraw-Hill, 1964.

13. Kleinrock, L., "Models for Computer Networks," Proc. of the International Conference on Communications, 1969, pp. 21-9 to 21-16.

14. Haight, F. A., Mathematical Theories of Traffic Flow (New York: Academic Press), 1963.

15. Fultz, G., and L. Kleinrock, "Adaptive Routing Techniques for Store-and-Forward Computer-Communication Networks," Proc. of the International Conference on Communications, 1971, pp. 39-1 to 39-8.

16. Kleinrock, L. Queueing Systems: Theory and Application, to be published by Wiley Interscience, 1973.

17. Kingman, J. F. C., "Some Inequalities for the Queue GI/G/1," Biometrica, 1962, pp. 315-324.

18. Burke, P., "The Output of a Queueing System," Operations Research, 1956, 4: 699-704.

19. Jackson, J. R., "Networks of Waiting Lines," Operations Research, 1957, 5: 518-521.

20. Fultz, G., "Adaptive Routing Techniques for Message Switching Computer-Communication Networks," Ph.D. Dissertation, Computer Science Department, School of Engineering and Applied Science, University of California, Los Angeles, Calif., June 1972.

21. Cole, G. D., "Computer Network Measurements: Techniques and Experiments," Ph.D. Dissertation, Computer Science Department, School of Engineering and Applied Science, University of California, Los Angeles, 1971, UCLA Engineering Report UCLA-ENG-7165.

22. Kahn, R. E., and W. R. Crowther, "A Study of the ARPA Network Design and Performance," BBN Report No. 2161, August 1971.

23. Frank, H. I. T. Frisch, W. Chou, and R. Van Slyke, "Optimal Design of Centralized Computer Networks," Networks, John Wiley, 1971, 1:43-57.

24. Raymond, H. G., "A Queueing Theory Approach to Communication Satellite Network Design," Proc. of the International Conference on Communication, 1971, pp. 42-26 to 42-31.

25. Meister, B., H. Muller and H. Rudin, "New Optimization Criteria for Message-Switching Networks," IEEE Transactions on Communication Technology, Com-19, June 1971, pp. 256-260.

26. Kleinrock, L., "Scheduling, Queueing and Delays in Time-Sharing Systems and Computer Networks," in Computer Communication Networks, Abramson, N., and F. Kuo (eds.), Prentice-Hall, Englewood Cliffs, N.J., 1972.

27. Fratta, L., M. Gerla, and L. Kleinrock, "The Flow Deviation Method: An Approach to Store-and-Forward Network Design," to be published in Networks.

28. Meister, B., H. Muller and H. Rudin, "Optimization of a New Model for Message-Switching Networks," Proc. of the International Conference on Communications, 1971, pp. 39-16 to 39-21.

29. Cantor, D. G., and M. Gerla, "The Optimal Routing of Messages in a Computer Network via Mathematical Programming," to be presented at the IEEE Computer Science Conference, San Francisco, Calif., September 12-14, 1972.