

On Communications and Networks

LEONARD KLEINROCK, FELLOW, IEEE

Abstract—Data communications has come of age. In this paper, we highlight some of the principal events that led up to the revolution in communications among information processing systems. Perhaps the most important event was the technological development of packet switching in the form of the ARPANET. We devote most of this presentation to a brief summary of the ARPANET experience, emphasizing the description, functions, analysis, design and performance measurement of packet-switching networks. We also discuss some recent advances in radio packet switching for long-haul (i.e., satellite) and terminal-access communications.

Index Terms—ARPANET, communication networks, computer networks, ground radio packet switching, networks, packet switching, satellite packet switching.

I. INTRODUCTION

TO most computer systems designers, communication problems are a nuisance. If only data communication were instantaneous, perfectly reliable, and free, then we could get on with the serious business of building computer systems. However, as Bertold Brecht said in *The Three Penny Opera*, "It seems that circumstance won't have it so." In fact it seems that circumstance will have it less and less as distributed processing, distributed data bases, distributed resources, and distributed demands grow in size. Even within a single computer system, the data bus problem and its associated protocol has been an important design consideration.

Remote data processing and network access to information and information processing has become possible due to the wedding of two large and dissimilar industries, namely, the communication and computing industries. (Perhaps we should think of theirs as a "shotgun" wedding, with at least one of the marriage partners acting under coercion.) The courtship began in the late fifties and early sixties when time sharing and remote data processing was in its infancy. At the old SJCC and FJCC meetings, many were the complaints by those in the data processing industry that the communication offerings were quite inadequate for their needs and that, in addition, the communication industry was not responding to those needs. We recall in 1959 when the FCC first permitted private networks to be constructed by those large industries that could afford them; this was the first step away from ATT-supplied communication services. In 1969 Microwave Communications, Inc. was authorized to construct a common carrier private line service between St. Louis and

Chicago. In 1971, finally, the FCC agreed that the specialized common carrier industry was in the best interest of the public and this opened the way to new networks and other specialized communication services in direct competition with ATT and Western Union. DATRAN was the first entry to propose the construction of a nation-wide switched all-digital network. Along with these developments, the established carriers had begun widespread use of PCM transmission and the introduction of the T-carrier service (e.g., T1 at 1.544 Mbits/s). In fact, ATT's Dataphone Digital Service (DDS) has come about in response to some of the specialized communications offerings by others. The use of microwave transmission has greatly reduced communications costs as has the introduction of the satellite technology. In 1972 the FCC authorized domestic communications satellite systems; this further expanded the specialized communications services. Furthermore, the use of modulated light beams travelling down very thin optical fibers (i.e., fiber optics) holds much promise for further reductions in communication costs.

While all these developments were taking place, an experimental data communications network had been constructed under the support of the U.S. Department of Defense Advanced Research Projects Agency (ARPA); this was the celebrated ARPANET. The credit for this network goes to Lawrence G. Roberts, who guided its development while Director of the Information Processing Techniques Office within ARPA. Indeed, the communications hurdles that he experienced [1] in attempting a computer-to-computer link between Lincoln Laboratory in Lexington, MA, and System Development Corporation in Santa Monica, CA, motivated him to develop the ARPANET when he later came to ARPA. This network ushered in the technology of packet switching and has captured the attention of all those interested in data communications. Indeed, by using an economic argument, Roberts showed [2] that a packet-switching network was due to evolve sometime in the very late sixties; the ARPANET came to life in September 1969. The great success of the ARPANET in providing cost effective data communications among geographically distributed terminals and computers has spawned a number of other government, private, and commercial packet-switching networks around the world; for example there is TELENET's Intelligent Network in the United States, DATAPAC in Canada, TRANSPAC in France, CYCLADES also in France, EPSS in the U.K., the European Integrated Network across Europe, TIDAS in Sweden, and a number of others currently being considered (e.g., the U.S. Autodin II net).

Due to the enormous impact of packet switching, we feel

Manuscript received March 22, 1976; revised June 7, 1976. This research was supported by the Advanced Research Projects Agency of the Department of Defense under Contract DAHC 15-73-C-0368.

The author is with the Department of Computer Science, University of California, Los Angeles, CA 90024.

justified in devoting the rest of this paper to the technology which has emerged from the ARPANET experience. In the following sections we discuss the concept of packet switching, describe the ARPANET, summarize the packet-switching network analysis and design technology, comment on the ARPANET performance, and then finally introduce some of the advanced packet-switching technologies which are currently emerging. In the final section we say a few words about things to come as seen by this author. To discuss less than this would be inappropriate for an anniversary issue, and yet to discuss all of this in so few pages is folly. In spite of this, we permit ourselves to proceed and urge the reader to consult [3] for a more complete treatment of these issues.

II. PACKET SWITCHING

Information processing devices tend to generate data at a very low duty cycle with occasional heavy data bursts; this is a nasty combination and produces many of the problems we face in data communications. If the data flow were smooth, then we might consider providing point-to-point communications to handle this traffic, but such is not ordinarily the case. Consequently, we must provide a computer-communications service which allows dynamic sharing of the communications plant in an efficient fashion. A computer-communication network employing packet switching is one way to accomplish this.

A computer communication network is a collection of nodes at which reside the computing resources (which themselves are connected into the network through nodal switching computers, i.e., "fancy" switches) which communicate with each other via a set of links (the data communication channels) [4], [5]. Messages in the form of commands, inquiries, file transmissions, and the like, travel through this network over the data transmission lines. At the nodal switching computers, the communications-oriented tasks of relaying messages (with appropriate routing, acknowledging, error and flow controlling, queuing, and so on) and of inserting and removing messages that originate and terminate at the terminals and main processors at that node must be carried out; these tasks are separated from the main computing functions required of the node and are relegated to a switching computer which is dedicated to these tasks (e.g., the interface message processors (IMP's) in the ARPANET—see below).

Computer-communication networks may therefore be conveniently partitioned into two separate subnetworks: the communication subnetwork providing the message service and the collection of computer and terminal resources that forms the "user-resource" subnetwork (see Fig. 1). In this figure, we show a fairly general structural model of a computer-communication network. First, note the computer facilities (C) denoted as square boxes that carry out the useful processing and storage tasks (as far as the user is concerned). These are connected together by means of the communication subnetwork (which consists of the switching computers and high-speed data commu-

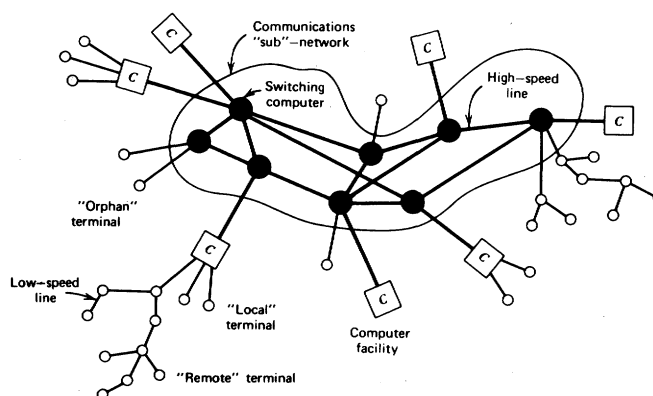


Fig. 1. The structure of a computer-communication network.

nication channels). As far as the communication subnetwork is concerned, all entry and exit for the network passes through the switching computers. More than one computer facility may be connected through a given switching computer. The terminals may either be local to a computer facility (in which case they may access the network through this facility), or they may be remote (in which case some "remote terminal network" must be provided to connect them to the facility, and then into the high-speed net), or they may not be associated with any computer facility at all, in which case these "orphan" terminals (or even networks of orphan terminals) may gain access to the switching computer directly (e.g., the terminal IMP's (TIP's) in the ARPANET). Most of our attention in this paper is directed to the communication subnetwork forming the message service. The function of this message service is to accept messages from any message source (such as a computer or a data terminal), route these messages through the communication network, and then deliver them to their destination in a rapid and reliable fashion.

Communication networks may conveniently be divided into three types: circuit (or line) switching; message switching; and packet switching. A *circuit-switching* network provides service by setting up a total path of connected lines from the origin to the destination of the "call" or demand; this complete circuit is set up by a special signaling message that threads its way through the network, seizing channels in the path as it proceeds. After the path is established, a return signal informs the source that data transmission may proceed, at which time all channels in the path are used simultaneously. The entire path remains allocated to the transmission (whether or not it is in use), and only when the source releases the circuit, will all these channels be returned to the available pool for use in other paths. Circuit switching is the common method for telephone systems [6]. In *message switching*, only one channel is used at a time for a given transmission. The message first travels from its source node to the next node in its path, and when the entire message is received at this node, then the next step in its journey is selected; if this selected channel is busy, the message waits in a queue, and finally, when the message reaches the head of the queue,

transmission begins. Thus the message "hops" from node to node through the network using only one channel at a time, possibly queueing at busy channels, as it is successively stored and forwarded through the network [5]. *Packet switching* is basically the same as message switching except that the messages are decomposed into smaller pieces called packets, each of which has a maximum length. These make their way through the net in a packet-switched (store-and-forward) fashion. Thus many packets of the same message may be in transmission simultaneously, thereby giving one of the main advantages of packet switching, namely, the "pipelining" effect; the transmission delay may be considerably reduced (over message switching) as a result (the reduction may be as large as a *factor* proportional to the number of packets into which the message is broken) [3], [7]–[9]. The decision as to which form of switching to use is a difficult one. Some studies have been conducted in an attempt to compare these switching techniques [10]–[15], but to date, no satisfactory comprehensive treatment has been given. One thing is clear, namely, if there is need for transmitting a long, continuous stream of data, then a leased line (or a circuit-switched connection) makes good sense. On the other hand, if the data flow is bursty (as is typical of computer and terminal data [16]) then some form of resource sharing can be used to great advantage; packet switching is an effective choice here.

There are numerous properties and issues with regard to packet switching that are important to discuss. For example, since packets are stored as they pass through switching nodes, it is possible to conduct speed, format, and code conversion during the switching process (this is true of any store-and-forward system such as message switching as well); this is not possible with circuit switching, which therefore requires complete end-to-end compatibility in this regard. Furthermore, in a moderately busy circuit-switched network, a set-up signal may find it difficult to locate a complete path of available channels from source to destination, and may return a "busy" signal to the source, i.e., the network is blocked. With packet switching, only the next channel in the path need be available, subject, however, to the ability of the message to initially acquire other kinds of network resources in response to the flow control procedure. Another key feature of packet switching is its ability to adaptively select good paths for packet journeys as a function of the network congestion. Besides providing small network delays, packet switching has the desirable feature of rapidly handling small messages in spite of the presence of long messages that may be in transport at the same time; this is because of the decomposition of (long) messages into packets. Another useful property of this decomposition into packets is that the nodal storage requirement is reduced (imagine if we used message switching with message lengths of 10^6 bits!).

We now have built up a picture of a packet-switching network as one that pipelines addressed messages along a single path as well as among alternate paths, partitions

messages into (pipelined) packets, places headers on packets and messages, and ships them through the network in a store-and-forward fashion. The packets and messages may encounter some unforeseen adventures as they journey through the network, and so for many possible reasons (e.g., errors, blocked storage, timeouts, etc.) the packets and messages may arrive at the destination out of order, or duplicated, or perhaps they may even get lost! The network must be prepared to handle these eventualities in an acceptable way (after all, how many users would pay for service on a network that had the habit of accepting a message, losing it, and never informing the user of this loss?). We may summarize the properties of packet-switching networks as follows [17]:

- 1) random delay;
- 2) random throughput;
- 3) out-of-order packets and messages;
- 4) lost and duplicate packets and messages; and
- 5) speed matching between the net and attached systems.

In order to respond to these properties, the network must provide many of the following functions:

- 1) packetizing;
- 2) buffering;
- 3) pipelining;
- 4) routing procedures;
- 5) sequencing and numbering;
- 6) error control (noise, duplicate, and lost message detection);
- 7) storage (resource) allocation, and
- 8) flow control.

The ARPANET provides these functions, as we discuss in the next section.

III. PACKET SWITCHING IN THE ARPANET

In September 1969, the embryonic one-node (!) ARPANET came to life when the first packet-switching computer was connected to the Sigma 7 computer at UCLA. Shortly thereafter began the interconnection of many main processors (referred to as HOST's) at various university, industrial, and government research centers across the United States. In 1970, a series of five papers was presented at the SJCC that summarized what we knew about the network at that time [9], [18]–[21]. Five additional papers summarized our experiences with this network up to 1972 [22]–[26]. Improvements in the operating procedure were reported in [27] and [28]. A five year re-evaluation was given in [29] and a capsule history was reported in [30].

The network currently provides a message service for almost 100 computers geographically distributed across the continental United States and extending by satellite to Hawaii and to a few nodes in Europe. The (HOST) computers are in many ways incompatible with each other, coming from different manufacturers and containing specialized software, data bases, and so on; this in fact presented the challenge of the original network experi-

ment, namely, to provide effective communication among, and utilization of, this collection of incompatible machines.

The topological connection is in the form of a distributed network which provides protection against total line and node failures by providing at least two physically separate paths between each pair of nodes. Each HOST is connected (through an asynchronous serial 100-kbit/s (KBPS) channel) to the small (local) nodal switching computer called an interface message processor (IMP); the latter are themselves interconnected mainly by leased 50 KBPS full duplex synchronous channels. The IMP's act as network doorways for the HOST computers. The IMP was introduced to relieve the HOST from many of the message-handling tasks of the communication network. In each HOST, a program referred to as the Network Control Program (NCP) must be implemented and inserted into the operating system (this is major surgery!); the NCP allows HOST computers to communicate with each other according to a HOST-HOST protocol, which is a network-wide standard. In addition, a program known as TELNET acts as a convenient interface between the user and the NCP, allowing him to converse with the network in a more natural way.

In order for a byte stream from (say) a terminal user to be sent to a remote computing system, the user's HOST must package the byte stream into a message stream. This (originating) HOST then delivers each message, including a destination HOST address, to its local IMP. The network IMP's then determine the route, provide error control, handle all message buffering and transmission functions, and finally notify the sender of the eventual receipt of the message at its destination HOST. The collection of HOST's, IMP's, and channels forms the packet-switched resource-sharing computer-communication network; the IMP's and channels form the message service (i.e., the communication subnet) for the HOST computers.

A dedicated path is not set up between HOST computers that are in communication but rather this communication involves a sequence of message transmissions sharing the communication lines with other messages in transit. The maximum message size is 8063 bits (plus 32 bits used as a HOST-HOST header). The IMP program partitions each message into one or more packets, each containing at most 1008 bits. Each packet of a message is transmitted independently to the destination IMP which reassembles the message before shipment to that destination HOST. Following this, an end-to-end acknowledgment is sent back to the source HOST. In its journey through the net, a packet will "hop" from IMP to neighboring IMP; if the neighbor accepts the packet (i.e., the transmission is found to be valid and there is storage space available), then an IMP-to-IMP packet acknowledgment is returned. If no acknowledgment is received after a time out period (125 ms), then the packet is retransmitted. This is the ARPANET implementation of a packet-switching network.

Routing strategies for this distributed network use a distributed control over the routing decisions that are

made in each IMP [31]. These routing computations are made using information received from neighboring IMP's and local information such as the status of its channels. In practice this approach has worked quite effectively with moderate levels of traffic so far experienced in the network; this has been one of the successes of the ARPANET experiment. The flow control procedure has presented some important challenges that continue to intrigue us; see, for example [3].

As so far described, access to the network comes only from a terminal through a HOST to an IMP. The introduction of a device known as a terminal IMP provides direct terminal access to the network [24]. The TIP performs the dual task of acting as an IMP and as a HOST. That which distinguishes a TIP from an IMP (aside from the additional 12K of storage) is a device known as a multiline controller which allows direct connection of up to 63 terminals to the net.

In October 1972, the first public demonstration of the ARPANET was conducted in conjunction with the first International Conference on Computer Communications (ICCC) in Washington, DC. Approximately 30 terminals from various manufacturers were connected to a TIP at the conference site. Instruction booklets were made available to the conference attendees to describe methods for accessing various resources on the ARPANET through these terminals. The procedure (demonstrated there and still in use) that a user goes through in reaching a remote computer facility is as follows. First he sits down, powers up the terminal, and then initiates a simple (login) dialogue with the TIP (or his own HOST if he is IMP-connected). Then he requests the TIP to make a connection to a remote HOST, and when this is accomplished he ignores both the TIP and the net and proceeds to login to the remote HOST. Following this, as has always been the case, the user then ignores the operating system of that HOST and communicates directly with the user process with which he has now been put in contact. During that ICCC demonstration, the true power of the ARPANET became apparent not only to the uninitiated users of the network, but also to the sophisticated and experienced users as they observed peak traffic rates of 60 000 packets per hour passing through the TIP and out into the network. The network traffic has been climbing at a phenomenal rate since 1971 and now sustains a fairly substantial load of roughly 9 million packets per day.

So much for the description of the ARPANET. Let us now proceed with the broad analytical and synthesis questions associated with a general communication subnet.

IV. NETWORK ANALYSIS AND DESIGN

The original analytical model of network delay is basically still in use today [5]. It consists of an M -channel N -node network. The M -communication channels are assumed to be noiseless, perfectly reliable, and to have a capacity denoted by C_i (bits per second) for the i th channel, whereas the N -nodes refer to the message- (or packet-)

switching centers which are also considered to be perfectly reliable and in which all of the message-switching functions take place. It is assumed that the nodal processing times are constant (usually assumed to be negligible). In addition, there are, of course, the channel queueing and transmission delays. Traffic entering the network from external sources forms a Poisson process with a mean γ_{jk} (messages per second) for those messages originating at node j and destined for node k . We further define the total external traffic entering (and therefore leaving) the network by

$$\gamma = \sum_{j=1}^N \sum_{k=1}^N \gamma_{jk}. \quad (1)$$

All messages are assumed to have lengths that are drawn independently from an exponential distribution with mean $1/\mu$ (bits). In order to accommodate these messages we assume that all nodes in the network have unlimited storage capacity. For many of the analytical results obtained we assume that messages are directed through the network according to a fixed routing procedure; this therefore implies that a unique path exists through the network for a given origin-destination pair. Each channel in the network is considered to be a separate server, and we adopt the notation λ_i as the average number of messages per second which travel over the i th channel. As with the external traffic we define the total traffic within the network by

$$\lambda = \sum_{i=1}^M \lambda_i. \quad (2)$$

In addition one can account for the propagation time on a given channel which is the time required for the energy represented in a single bit to propagate down the length of that channel.

One of the main objectives of network analysis is to solve for T , the average message delay (i.e., the time a message spends in the network). Fortunately, it is possible to express this overall delay in terms of the individual channel delays T_i , where T_i is the average time spent waiting for and using the i th channel. The decomposition of the overall network delay into its channel-by-channel components leads us to the key result [5]

$$T = \sum_{i=1}^M \frac{\lambda_i}{\gamma} T_i. \quad (3)$$

This result is perfectly general and our analysis problem has now reduced to the calculation of T_i .

At first glance, the network we have so far described is similar to Jackson's open networks [32]. However, in the Jackson networks the service time (i.e., transmission time on a channel) at each server is an independent random variable whereas in these computer-communication network models we see that the service time for a given message at different channels is directly related to the message length and the fixed parameters of the channels; this is far from independent. This dependency causes great analytic difficulty and it is fortunate that an independence as-

sumption, which permits us to remove this dependency, is a good approximation to real networks [5]. We then find ourselves in the situation of Jackson-type networks and may therefore use his results to evaluate T_i as follows:

$$T_i = \frac{1}{\mu C_i - \lambda_i}. \quad (4)$$

The expression for T_i may be refined to include the effects of nodal processing time, control traffic, and channel propagation time.

From this delay analysis we may predict quantitative as well as phenomenological behavior of the average message delay in networks. It can be shown that T will exhibit a rather sharp threshold behavior; prior to this threshold, T remains relatively constant and as we approach the threshold, T will suddenly grow in a fashion far sharper than one would expect within an isolated queueing system.

Let us now consider some of the salient aspects of network design [3]. In forming our objective function we assume that the cost (say in dollars) of constructing the i th channel with capacity C_i is given by $d_i(C_i)$ an arbitrary function of the capacity and of the channel. We let D (dollars) represent the cost of the entire network, which we assume to consist only of the cost for channel construction, and so we have

$$D = \sum_{i=1}^M d_i(C_i). \quad (5)$$

We may now define four optimization problems that differ only in the set of permissible design variables. In each of these problems it is assumed that we are given the node locations, the external traffic flow requirements γ_{jk} , the channel costs $d_i(C_i)$ and the constants D and μ . We also assume that the flow $\{\lambda_i\}$ we use is feasible (i.e., it satisfies capacity, conservation, and external traffic requirement constraints). First, we have the *capacity assignment* (CA) problem:

CA Problem:

Given: Flows $\{\lambda_i\}$ and network topology.

Minimize: T .

With respect to: $\{C_i\}$.

Under constraint: $D = \sum_{i=1}^M d_i(C_i)$.

Second, we have the *flow assignment* (FA) problem:

FA Problem:

Given: Capacities $\{C_i\}$ and network topology.

Minimize: T .

With respect to: $\{\lambda_i\}$.

Third, we have the *capacity and flow assignment* (CFA) problem:

CFA Problem:

Given: Network topology.

Minimize: T .

With respect to: $\{C_i\}$ and $\{\lambda_i\}$.

Under constraint: $D = \sum_{i=1}^M d_i(C_i)$.

Last, we have the *topology, capacity, and flow assignment* (TCFA) problem:

TCFA Problem:

Minimize: T .

With respect to: Topological design, $\{C_i\}$ and $\{\lambda_i\}$.

Under constraint: $D = \sum_{i=1}^M d_i(C_i)$.

These four problems are presently solved in various degrees of completeness; that completeness depends very strongly upon the form of the cost functions $d_i(C_i)$. The CA problem has been solved exactly in the case of linear capacity costs [5], namely, $d_i(C_i) = d_i C_i$. A special case of great importance is when $d_i = d$; this case appears when one considers (stationary) satellite communication channels in which the distance between any two points on earth within the shadow of the satellite is essentially the same regardless of the terrestrial distance between these two points. Other cases which have been solved but which require numerical evaluation include the logarithmic and the power-law cost function [21]. With a more realistic discrete capacity cost function, an exact solution involves a dynamic programming approach [33], but may also be approximated using some of the simpler cost functions [34].

The FA problem turns out to be a relatively straightforward problem to solve and a fairly efficient procedure known as the flow deviation method may be found in [35]. The CFA problem turns out to be quite difficult and a suboptimal heuristic solution may be found in [34]. Finally, the true design problem for networks, namely the TCFA problem, is extremely difficult and a number of heuristic suboptimal solutions are available such as the concave branch elimination method and the cut saturation method; a review of some such techniques is given in [36]. These analysis and design procedures are presented in [3]. The issue of routing and design for very large networks is discussed in [37] and [38].

The field of network analysis and design is rich with interesting problems. However, the creation of the ARPANET has led us to evolve an effective design technology much of which is contained in the aforementioned problems and their solutions. This part of the technology is reasonably well in hand and will certainly be advanced with further innovative work.

IV. NETWORK PERFORMANCE

Early in the design of the ARPANET, it was understood that the ability to make direct measurements was of paramount importance and would form an essential part of the experiment of creating the network itself. Consequently, a fairly sophisticated measurement capability has been built into the network both within the IMP's and within certain of the HOST computers [especially at the UCLA Network Measurement Center (NMC)].

A number of measurement experiments have been made

by the NMC and many of these are summarized in [3]. These measurements identified and quantified network throughput, network congestion and delays, network deadlocks and degradations, message and packet size distributions, mean traffic-weighted path length, incest (the flow of traffic to and from HOST's at the same local site), "most popular" sites and channels, "favoritism" (that property which a site demonstrates by sending many of its messages to one or a small number of sites), and channel utilization. This measurement data verified that the ARPANET was, in large part, meeting its design goals.

In the last paragraph we casually referred to deadlocks and degradations which were observed in the ARPANET. These problems arise largely due to the subtleties of flow control procedures as discussed in [3]. Indeed one of the significant contributions of the ARPANET experience has been to alert us to the kinds of problems which the flow control procedure may cause. Among the deadlocks we have observed are reassembly deadlock (messages were launched into the network for which no destination buffer space was provided; they formed an impenetrable barrier around the destination and prevented complete reassembly of messages); store-and-forward lockup (a standoff occurred when two adjacent IMP's unsuccessfully tried to send packets into each other's fully occupied buffer pool); the Christmas lockup (late in December 1973, we ran an experiment whose measurement messages caused buffer assignments which could never be unassigned due to a lack of pointers to these buffers); and piggyback lockup (an attempt to handle network control messages in an efficient piggyback fashion led to a buffer assignment which could never be released). Among the degradations we have observed are those due to looping and hold-down in the routing procedure, gaps in message transmission, single packet turbulence (a single out-of-order packet created a fourfold and persistent throughput degradation), and phasing (it was difficult for the system to collect the correct mix of system resources simultaneously at the needed location). The purpose of a flow control procedure is to throttle the input traffic and to allocate network resources in a way which expedites the flow of traffic through the network. Among the natural functions required of a flow control procedure, we often include the desire that the order in which data enter a network at some origin be the same order in which that data leave the network at its destination. In addition we often require that no messages be lost, that duplicate messages be detected and eliminated, and that messages which are delivered are error-free. Certain other constraints may also be required of the network. There's the rub! Whenever one imposes constraints on the flow of traffic, it is possible in certain (perhaps unusual) circumstances that these constraints may be impossible to meet, in which case the result will be a deadlock, whereas if the constraint is slow in being met then this may result in degradation to throughput. We conclude that the design of effective control procedures is still in some ways a black art and serious thought must be given to its implementation. In spite of this, we do have

packet-switching networks which work exceedingly well most of the time and are more than capable of operating within their design goals.

VI. ADVANCED PACKET SWITCHING

In this section we briefly discuss the use of radio as a communication medium for data networks. The reader is again referred to [3] for further details. The development of these packet radio techniques is due in large part to the encouragement, guidance, and foresight of Robert E. Kahn of ARPA.

Packet-switching networks, such as the ARPANET, came into existence in response to the need for rapid, efficient, and economical data communications. As such networks grow in size and coverage, the need to provide inexpensive, long-haul, high-capacity communication channels becomes more pressing. There is also the problem of providing inexpensive communications from the users' terminals into the high-level network itself. In this section we wish to discuss the use of packet switching over a broad-band satellite channel as a solution to the long-haul problem and also the use of ground radio packet switching for terminal access.

There are a number of characteristics of a (stationary) earth satellite that are of importance in its use for packet switching. First, and perhaps most striking, is the *long-propagation delay* in a roundtrip transmission (up and down) to a satellite transponder that is in a synchronous orbit roughly 36 000 km above the earth; this delay is approximately 0.25 s. Second, a single digitized voice channel provides a *broad-band* data communications capability. Third, the satellite transponder can retransmit back down to earth in a *broadcast* mode to all earth stations in its broadbeam "shadow." Fourth, each transmitter can listen to his own transmission since he, too, is in the broadcast shadow; if errors due to random noise can be neglected (as we assume), then we have "perfect feedback" that gives us *automatic acknowledgments*.

The existence of the inherent quarter-second propagation delay suggests that we introduce access schemes that differ radically from land-based communications. In particular, packet switching permits us to take advantage of the long delay, the broad-band, the broadcast, and the automatic acknowledgment of these channels. (The case of ground radio packet switching discussed below takes advantage of the fact that the propagation delays of interest are *small* compared to a packet transmission time.)

There are many ways to use a given satellite channel for data communications. For example, one could make permanent subchannel assignments to private users; this can be very wasteful in a bursty user environment. Also one could permit a dial-up procedure for sharing a set of subchannels; this too can be problematical. Alternatively, one could provide the entire channel capacity to users on a demand basis (with some form of polling control); this requires buffering at the source. As another alternative,

one could permit "random" access to the full capacity of the channel in a packet-switching mode; this too is in the spirit of the ARPANET philosophy, and here we wish to discuss some random access schemes in this category.

Of interest, then, is the consideration of satellite channels for packet switching. As earlier, a packet is defined merely as an addressed package of data that has been prepared by one user for transmission to some other user in the system. The satellite is characterized as a high-capacity channel with a fixed propagation delay that is large compared to the packet transmission time. We consider a transmission scheme wherein a given transmitter forms his packet and then *bursts* it out rapidly on the channel at full capacity. Many users operating in this fashion automatically multiplex their transmissions on a demand basis. The (stationary) satellite acts as a pure transponder repeating whatever it receives and beaming this transmission back down to earth; this broadcasted transmission can be heard by every user of the system and in particular a user can listen to his own transmission on its way back down (this is a crucial property!). Since the satellite is merely transponding, then whenever a portion of one user's transmission reaches the satellite while another user's transmission is being transponded, the two collide and "destroy" each other. The problem we are then faced with is how to control the allocation of time at the satellite in a fashion that produces an acceptable level of performance.

The ideal situation would be for the users to agree collectively when each could transmit. The difficulty is that the means for communication available to these geographically distributed users is the satellite channel itself, and we are faced with attempting to control a channel that must carry its own control information. There are essentially three decentralized approaches to the solution of this packet-switching problem. The first has come to be known as a "pure ALOHA" system in which users transmit any time they desire. If, after one propagation delay, they hear their successful transmission, then they assume that no conflict occurred (i.e., they have a positive acknowledgment); otherwise, they know a collision (or perhaps some other source of noise) did occur and they must retransmit (i.e., they assume a negative acknowledgment). If all users retransmit immediately upon hearing a conflict, then they are sure to conflict again, and so some scheme must be devised for introducing a retransmission delay to spread these conflicting packets over time.

The second method for using the satellite channel is to "slot" time into segments whose duration is exactly equal to the transmission time of a single packet (we assume constant length packets). If we now require all packets to begin their transmission only at the beginning of a slot (where time is referenced to the satellite), then we enjoy a gain in efficiency, since collisions are now restricted to a single slot duration; such a scheme is referred to as a "slotted ALOHA" system.

The third method for using these channels is to attempt to *schedule* their use in some direct fashion; this introduces

the notion of a reservation system in which time slots are reserved on a fixed or demand basis for specific users' transmissions.

Thus, we are faced with a finite-capacity communication channel subject to unpredictable and conflicting demands. When these demands collide we "lose" some of the effective capacity of the channel. Note that it is possible to use the channel up to its full rated capacity when only a single user is demanding service; this is true since a user will never conflict with himself (he has the capability to schedule his own use).

Let S denote the satellite channel throughput (average number of successful transmissions per transmission period) and let G denote the average channel traffic (in attempted packet transmissions per transmission period). If we assume that the channel traffic is Poisson then the pure ALOHA channel has a throughput given by

$$S = Ge^{-2G}. \quad (6)$$

This was first obtained by Abramson [39] and one sees that the maximum throughput is simply $1/2e \cong 0.184$. Roberts introduced the notion of a slotted ALOHA channel [40] and found the throughput to be

$$S = Ge^{-G}. \quad (7)$$

Here the maximum throughput is $1/e \cong 0.368$ (twice that of a pure ALOHA system). In [39] the maximum throughput contours were obtained for a finite population model. This last result ignores delay considerations which were analyzed for slotted ALOHA for an infinite population of users in [41]. It turns out that the ALOHA channels are inherently unstable and special measures must be taken to control them; appropriate measures of instability and control procedures were first discussed in [42].

A number of clever schemes are presented in [43] and [44] for accessing a packet-switched satellite channel using dynamic reservations. These schemes can be quite effective when the channel traffic begins to get heavy. In fact, a static reservation scheme such as TDMA or FDMA leads to a maximum channel utilization in very heavy traffic; the disadvantage of such systems is that they have poor throughput-delay characteristics when the traffic level is low, but turn out to be optimum when the traffic level increases to the channel limit [45].

Let us now discuss the use of ground radio packet switching in a communications system for local (possibly mobile) terminal interconnection as well as for access to a high-level network [46]. We are concerned with the suitability of a highly multiplexed random-access packet-switching technology for this application. Of course the basic principles involved are quite similar to the satellite packet-switching studies. The fundamental difference is that with ground radio, the roundtrip propagation delay is small compared to a packet transmission time. This suggests a fourth method for using the packet-switched channel, namely, the carrier sense multiple access mode (CSMA). In CSMA, we permit the terminal to listen to ("sense") the channel; and, if the carrier signal is heard,

then the terminal realizes that the channel is in use by some other terminal and will politely postpone its transmission until the channel is sensed to be idle. (This information is useless with satellites since the sensed channel state provides information about the channel that is "ancient" history.)

There are many rules (i.e., protocols) for deciding when a terminal user may transmit and what action he must take if he collides with another transmission. The CSMA protocol has been analyzed in [46]–[50]. It is found that the proper use of these protocols can lead to channel capacities which approach unity (compared to the poor efficiency of the ALOHA channels). We hasten to point out that, in addition to questions of access protocols, there are many other important aspects to ground radio packet switching. For example, the topological organization of a packet radio network is plagued with fascinating problems. Also, radio propagation, with the usual multipath, barriers, and noise effects, gives rise to serious considerations. Add to this the mobility of the terminals, and one finds that the use of packet switching in such an environment leads to many new and interesting problems.

The performance described above was based on the (strong) assumption that all terminals were in line-of-sight and within range of each other. There are many instances where this is not the case, forcing us to relax that assumption. Terminals can be within range of a central station (computer center, gateway to a network, satellite IMP, etc.) but out-of-range of each other, or they can be separated by some physical obstacle opaque to UHF radio signals. This gives rise as to what is called the "hidden terminal" effect. It is evident that the existence of hidden terminals in an environment affects (degrades) the performance of CSMA. In [49] and [50] the effect of this degradation is quantified. A solution to this hidden terminal problem is the Busy-Tone Multiple Access (BTMA) mode. The operation of BTMA rests on the assumption that the station is, by definition, within range and line-of-sight of all terminals. With BTMA, the total available bandwidth is divided into two channels: a message channel and a busy tone (BT) channel. As long as the station senses a (terminal) carrier on the incoming message channel it transmits a (sine-wave) BT signal on the BT channel. It is by sensing a carrier on the BT channel that terminals determine when the message channel is busy. The performance of BTMA may be found in [49] and [50] and it turns out that the cost in delay and throughput is really quite small due to hidden terminals when we use the BTMA solution.

In Table I we summarize the throughput results for the many radio access protocols we have presented. In this table, we have assumed that the ratio a of propagation delay to transmission time is $a = 0.01$. Also in this table we have named certain of the access protocols for CSMA and BTMA; the reader should merely note the capacity (throughput) which is achievable. From the table, we see that most of the throughput which is lost with the ALOHA systems may be regained with various of the CSMA sys-

TABLE I
Capacity for Various Access Protocols

Protocol	Capacity C
Pure ALOHA	0.184
Slotted ALOHA	0.368
1-persistent CSMA	0.529
Slotted 1-persistent CSMA	0.531
Nonpersistent BTMA	
100-kHz bandwidth	0.680
1000-kHz bandwidth	0.720
0.1-persistent CSMA	0.791
Nonpersistent CSMA	0.815
0.03-persistent CSMA	0.827
Slotted nonpersistent CSMA	0.857
Perfect scheduling	1.000

tems. Yet newer schemes have recently been studied which offer improved performance [51].

The use of packet switching in a satellite radio and ground radio environment has progressed significantly in the last two or three years. Indeed ARPA has an experimental program for these two systems, and in each has already demonstrated the feasibility of multiple-access broadcast packet transmission.

VII. CONCLUSIONS AND DIRECTIONS

It is fairly clear that informaton processing has come to depend heavily upon data communications. Rather than ignore the communications problem, computer scientists are dealing with the issues involved and have already learned to take advantage of its properties. We can already foresee the day when there will exist an all-digital terrestrial network taking advantage of the advanced technologies of fiber optics and cable TV along with the use of satellites to help tie these systems together.

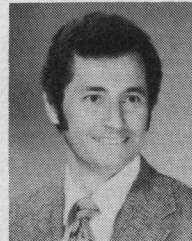
Along with these advanced data communication networks, we find a number of major issues which demand considerable thought and attention. For example, the issue of a distributed operating system and large distributed data bases involves problems which are hardly defined, much less solved. Security issues in the distributed environment provided by a network are of great concern in many communities. The proliferation of networks around the world introduces its own set of problems involving methods for interconnecting incompatible distributed systems and networks; the consideration of CCITT's X.25 international network access protocol [52] is an important step toward the solution of these problems.

With efficient communications in a network environment, one is then challenged to identify the meaningful applications for which such a system is suitable. We can foresee the transmission of digitized voice through such networks as well as the widespread use of intelligent programmable terminals. However, as our sophistication in processing and communication increases we must not forget that there is an enormous population of users out there who still prefer the good old days of simple-minded systems, people and terminals (Teletype Corporation enshrined its 500 000th teletype not so many years ago!)

REFERENCES

- [1] T. Marill and L. G. Roberts, "Toward a cooperative network of time-shared computers," in *Fall Joint Comput. Conf., AFIPS Conf. Proc.*, vol. 29. Washington, DC: Spartan, pp. 425-431, 1970.
- [2] L. G. Roberts, "Data by the packet," *IEEE Spectrum*, vol. 11, pp. 46-51, Feb. 1974.
- [3] L. Kleinrock, *Queueing Systems, Vol II: Computer Applications*. New York: Wiley-Interscience, 1976.
- [4] D. W. Davies and D. Barber, *Communication Networks for Computers*. New York: Wiley, 1973.
- [5] L. Kleinrock, *Communication Nets; Stochastic Message Flow and Delay*. New York: McGraw-Hill, 1964, out of print, New York: Dover, reprinted 1972.
- [6] R. Syski, *Introduction to Congestion in Telephone Systems*. Edinburg and London: Oliver and Boyd, 1960.
- [7] P. Baran, "On distributed communications," RAND Corp., Santa Monica, CA, RAND Series Rep., Aug. 1964.
- [8] D. W. Davies, "The principles of a data communication network for computers and remote peripherals," in *Proc. IFIP Congress 68*, (Hardware), pp. 709-714, Edinburg, Scotland, 1968.
- [9] L. G. Roberts and B. D. Wessler, "Computer network development to achieve resource sharing," *Spring Joint Comput. Conf., AFIPS Conf. Proc.*, vol. 36. Montvale, NJ: AFIPS Press, pp. 543-549, 1970.
- [10] E. Port and F. Closs, "Cmparison of switched data networks on the basis of waiting times," IBM Zurich Res. Lab. Rep., Apr. 1971.
- [11] F. Closs, "Time delays and trunk capacity requirements in line-switched networks," *Int. Switching Symp. Rec.*, pp. 428-433, 1972.
- [12] G. J. Clowes and C. S. Jayasuriya, "Traffic considerations in switched data networks," in *Proc. 3rd IEEE Symp. on Data Networks Analysis and Design*, pp. 18-22, Nov. 1973.
- [13] K. Itoh and T. Kato, "An analysis of traffic handling capacity of packet switched and circuit switched networks," in *Proc. 3rd IEEE Symp. Data Networks Analysis and Design*, pp. 29-37, Nov. 1973.
- [14] H. Miyahara, T. Hasegawa, and Y. Teshigawara, "A comparative analysis of switching methods in computer communication networks," in *Proc. Int. Commun. Conf.*, pp. 6-6-6-10, June 1975.
- [15] R. D. Rosner and B. Springer, "Circuit and packet switching," *Comput. Networks*, vol. 1, pp. 7-26, June 1976.
- [16] E. Fuchs and P. E. Jackson, "Estimates of distributions of random variables for certain computer communications traffic models," *Commun. Ass. Comput. Mach.*, vol. 13, pp. 752-757, Dec. 1970.
- [17] W. R. Crowther, F. E. Heart, A. A. McKenzie, J. M. McQuillan, and D. C. Walden, "Issues in packet-switching network design," in *Nat. Comput. Conf., AFIPS Conf. Proc.*, vol. 44. Montvale, NJ: AFIPS Press, pp. 161-175, 1975.
- [18] C. S. Carr, S. D. Crocker, and V. G. Cerf, "HOST-HOST communication protocol in the ARPA network," in *Spring Joint Comput. Conf., AFIPS Conf. Proc.*, vol. 36. Montvale, NJ: AFIPS Press, pp. 589-597, 1970.
- [19] H. Frank, I. T. Frish, and W. Chou, "Topological considerations in the design of the Arpa network," in *Spring Joint Comput. Conf., AFIPS Conf. Proc.*, vol. 36. Montvale, NJ: AFIPS Press, pp. 581-587, 1970.
- [20] F. W. Heart, R. E. Kahn, S. M. Ornstein, W. R. Crowther, and D. C. Walden, "The interface message processor for the ARPA computer network," in *Spring Joint Comput. Conf., AFIPS Conf. Proc.*, vol. 36. Montvale, NJ: AFIPS Press, pp. 551-567, 1970.
- [21] L. Kleinrock, "Analytic and simulation methods in computer network design," in *Spring Joint Comput. Conf., AFIPS Conf. Proc.*, vol. 36. Montvale, NJ: AFIPS Press, pp. 569-579, 1970.
- [22] S. Crocker, J. Heafner, R. Metcalfe, and J. Postel, "Function-oriented protocols for the ARPA computer network," in *Spring Joint Comput. Conf., AFIPS Conf. Proc.*, vol. 40. Montvale, NJ: AFIPS Press, pp. 271-279, 1972.
- [23] H. Frank, R. Kahn, and L. Kleinrock, "Computer communication network design—Experience with theory and practice," in *Spring Joint Comput. Conf., AFIPS Conf. Proc.*, vol. 40. Montvale, NJ: AFIPS Press, pp. 255-270, 1972.
- [24] S. Ornstein, F. Heart, W. Crowther, H. Rising, S. Russell, and A. Michel, "The terminal IMP for the ARPA computer network," in *Spring Joint Comput. Conf., AFIPS Conf. Proc.*, vol. 40. Montvale, NJ: AFIPS Press, pp. 243-254, 1972.
- [25] L. G. Roerts, "Extensions of packet communication technology to

- a hand held personal terminal," in *Spring Joint Comput. Conf., AFIPS Conf. Proc.*, vol. 40. Montvale, NJ: AFIPS Press, pp. 295-298, 1972.
- [26] R. H. Thomas and D. A. Henderson, "McROSS—A multi-computer programming system," in *Spring Joint Comput. Conf., AFIPS Conf. Proc.*, vol. 40. Montvale, NJ: AFIPS Press, pp. 281-293, 1972.
- [27] J. McQuillan, W. Crowther, P. Cossell, D. Walden, and F. Heart, "Improvements in the design and performance of the ARPA network," in *Fall Joint Comput. Conf., AFIPS Conf. Proc.*, vol. 41. Montvale, NJ: AFIPS Press, pp. 741-754, 1972.
- [28] N. Mimno, B. Cossell, D. Walden, S. Butterfield, and J. Leven, "Terminal access to the ARPA network: Experience and improvement," in *1973 CompCon Proc. 7th IEEE Comput. Soc. Int. Conf.*, pp. 39-42, Feb. 1973.
- [29] L. G. Roberts, "Network rationale: A 5-year re-evaluation," in *1973 CompCon Proc. 7th IEEE Comput. Soc. Int. Conf.*, pp. 3-5, Feb. 1973.
- [30] P. M. Karp, "Origin, development and current status of the ARPA network," in *1973 CompCon Proc. 7th IEEE Comput. Soc. Int. Conf.*, pp. 49-52, Feb. 1973.
- [31] G. L. Fultz, "Adaptive routing techniques for message switching computer-communication network," Sch. of Engr. and Appl. Sci., Univ. of California, Los Angeles, CA, Rep. UCLA-ENG-7252, July 1972.
- [32] J. R. Jackson, "Networks of waiting lines," *Oper. Res.*, vol. 5, pp. 518-521, 1957.
- [33] D. G. Cantor and M. Gerla, "Capacity allocation in distributed computer networks," in *Proc. 7th Hawaii Int. Conf. Syst. Sci.*, Jan. 1974, pp. 115-117.
- [34] M. Gerla, "The design of store-and-forward (S/F) networks for computer communications," Sch. of Engr. and Appl. Sci., Univ. of California, Los Angeles, CA, Rep. UCLA-ENG-7319, Jan. 1973.
- [35] L. Fratta, M. Gerla, and L. Kleinrock, "The flow deviation method—An approach to store-and-forward communication network design," *Networks*, vol. 3, pp. 97-133, 1973.
- [36] M. Gerla, H. Frank, W. Chou, and J. Eckl, "A cut-saturation algorithm for topological design of packet switched communications networks," in *Proc. Nat. Telecommun. Conf.*, San Diego, CA, Dec. 1974.
- [37] L. Kleinrock and F. Kamoun, "Data communication through large packet switching networks," in *Proc. 8th Int. Teletraffic Congr.*, Nov. 1976, to be published.
- [38] F. Kamoun, "Design considerations for large computer communication networks," Sch. of Engr. and Appl. Sci., Univ. of California, Los Angeles, CA, Rep. UCLA-ENG-7642, Apr. 1976.
- [39] N. Abramson, "Packet switching with satellites," in *Nat. Comput. Conf., AFIPS Conf. Proc.*, vol. 42. Montvale, NJ: AFIPS Press, pp. 695-702, 1973.
- [40] L. G. Roberts, "Dynamic allocation of satellite capacity through packet reservation," in *1973 Nat. Comput. Conf., AFIPS Conf. Proc.*, vol. 44. Montvale, NJ: AFIPS Press, 1973, pp. 711-716.
- [41] L. Kleinrock and S. Lam, "Packet switching in a slotted satellite channel," in *Nat. Comput. Conf., AFIPS Conf. Proc.*, vol. 42. Montvale, NJ: AFIPS Press, pp. 703-710, 1973.
- [42] S. Lam, "Packet switching in a multi-access broadcast channel with applications to satellite communication in a computer network," Sch. of Eng. and Appl. Sci., Univ. of California, Los Angeles, CA, Rep. UCLA-ENG-7429, Mar. 1974.
- [43] L. G. Roberts, "Dynamic allocation of satellite capacity through packet reservation," in *Nat. Comput. Conf., AFIPS Conf. Proc.*, vol. 42. Montvale, NJ: AFIPS Press, pp. 711-716, 1973.
- [44] R. Binder, "A dynamic packet-switching system for satellite broadcast channels," in *Proc. Int. Commun. Conf.*, pp. 41-1-41-5, June 1975.
- [45] L. Kleinrock and F. Tobagi, "Random access techniques for data transmission over packet switched radio channels," in *Nat. Comput. Conf., AFIPS Conf. Proc.*, vol. 44. Montvale, NJ: AFIPS Press, pp. 187-201, 1975.
- [46] R. E. Kahn, "The organization of computer resources into a packet radio network," in *1975 Nat. Comput. Conf., AFIPS Conf. Proc.*, vol. 44. Montvale, NJ: AFIPS Press, 1975, pp. 177-186.
- [47] L. Kleinrock and F. Tobagi, "Carrier sense multiple access for packet switched radio channels," in *Proc. Int. Commun. Conf.*, pp. 21B-1-21B-7, June 1974.
- [48] L. Kleinrock and F. Tobagi, "Packet switching in radio channels: Part I—Carrier sense multiple access modes and their throughput-delay characteristics," *IEEE Trans. Commun.*, vol. COM-23, pp. 1400-1416, Dec. 1975.
- [49] F. Tobagi, "Random access techniques for data transmission over packet switched radio networks," Sch. of Engr. and Appl. Sci., Univ. of California, Los Angeles, CA, Rep. UCLA-ENG-7499, Dec. 1974.
- [50] F. Tobagi and L. Kleinrock, "Packet switching in radio channels: Part II—The hidden terminal problem and the carrier sense multiple access mode with a busy tone," *IEEE Trans. Commun.*, vol. COM-23, pp. 1417-1433, Dec. 1975.
- [51] M. Scholl, "Multiplexing techniques for data transmission over packet switched radio systems," Ph.D. dissertation, Sch. of Engr. and Appl. Sci., Univ. of California, Los Angeles, 1976.
- [52] A. Rybczynski, B. Wessler, R. Despres, and J. Wedlake, "A new communication protocol for accessing data networks—The international packet-mode interface," in *1976 Nat. Comput. Conf., AFIPS Conf. Proc.*, vol. 45. Montvale, NJ: AFIPS Press, 1976, pp. 477-482.



Leonard Kleinrock (S'55-M'64-SM'71-F'73) received the B.E.E. degree from City College of New York, New York, NY, and the M.S.E.E. and Ph.D. E. E. degrees from the Massachusetts Institute of Technology, Cambridge, in 1957, 1959 and 1963, respectively.

From 1951 to 1957 he was an Assistant Engineer at the Photobell Company, New York. From 1957 to 1963 he was a Staff Associate at the Massachusetts Institute of Technology Lincoln Laboratory. In 1963 he joined the faculty of the School of Engineering and Applied Science at the University of California, Los Angeles, where he is currently Professor of Computer Science. He is the author of 3 major books in the field of computer networks: *Communication Nets: Stochastic Message Flow and Delay* (New York: McGraw-Hill, 1964; reprinted: New York: Dover, 1972), *Queueing Systems, Vol. I: Theory* (New York: Wiley-Interscience, 1975), and *Queueing Systems, Vol. II: Computer Applications* (New York: Wiley-Interscience, 1976). He has published over 70 articles and contributed to several books. He serves as consultant for many domestic and foreign corporations and governments and he is a referee for numerous scholarly publications and a book reviewer for several publishers. His research spans the fields of computer networks, computer systems modeling and analysis, queueing theory, and resource sharing, and allocation, in general. At UCLA, he directs a large group in advanced teleprocessing systems and computer networks.

Dr. Kleinrock was awarded a Guggenheim Fellowship in 1971-1972. He is the recipient of the 1975 Communications Society Leonard G. Abraham Price paper award for "Packet Switching in a Multiaccess Broadcast Channel: Performance Evaluation" (with Simon S. Lam).