M.I.T. Laboratory for Computer Science
Computer Systems Research Division

November 12, 1976
Request for Comments No. 130
Local Network Note No. 2

PLAN FOR A LOCAL, HIGH-SPEED COMPUTER NETWORK FOR THE LABORATORY FOR COMPUTER SCIENCE

by Kenneth T. Pogran

Abstract

In May of 1975, a subcommittee of the (then) Project MAC Computing Resources Committee commissioned a study to develop plans for a high-speed local data network to interconnect computers, both present and future, within what is now the Laboratory for Computer Science. This document constitutes a report of this study, and presents a specific proposal for the implementation of a high-speed local data network. In this report, various alternative technologies are described and compared, and the hardware of the technology selected -- a "buffered Ethernet" -- is described in detail. Protocols for the network are discussed, and topics for future exploration are presented.

This note is an informal working paper of the M.I.T. Laboratory for Computer Science, Computer Systems Research Division. It should not be reproduced without the author's permission, and it should not be referenced in other publications.

1. INTRODUCTION

In the spring of 1975, the Project MAC Computing Resources Committee established a subcommittee to examine the desirability of constructing a local, high-performance data network to interconnect present and future computers at Project MAC. This subcommittee, chaired by Prof. Steve Ward, concluded that construction of such a network was desirable and appropriate, and in May 1975, at the subcommittee's recommendation, a detailed study was commissioned to develop plans for the network.

The intended uses of the local network, as seen in 1975, were several: to provide for intercommunication among existing and future computers at the Laboratory for Computer Science, to provide a "gateway" through which all LCS computers could access the ARPANET, to provide a "central file system" computer whose sole function is to provide rapidly accessible file storage for computers which had none of their own, and to provide communication between a proposed terminal concentrator system and the various LCS computers, allowing for efficient, varied use of terminals within the Laboratory. These varied uses for the network posed differing traffic requirements which must be met by a common hardware base and set of protocols. The network must provide for efficient transfer of large blocks of data (high throughput, for "file transfer" applications), while also providing speedy transmission of small messages (low delay, for "TELNET"-like applications).

As the study progressed, it was realized that, with prudent planning, the network developed for the Laboratory could serve as the basis for a high-performance data network for the entire M.I.T. campus. This has been kept in mind throughout the evolution of the network design. The concept of the Laboratory's network as a "sub-network" of a future campus-wide network is a subject for further study and another paper, and is not discussed extensively in this document. However, it is mentioned, on occasion, in the discussion of design decisions on which it has had an impact.

This document constitutes a report of the study commissioned in 1975, and presents a specific proposal for the implementation of a high-performance local network for the Laboratory for Computer Science (*). It contains a detailed description of the hardware base chosen for the Network, along with a description of and comparison with the major alternative hardware base considered. Protocols for the network are also discussed.

(*) Referred to in this document as "the LCS Network", or, simply, "the Network".

2. CHOOSING A NETWORK TECHNOLOGY

One of the first tasks of the study was to select a hardware base for the LCS Network. An investigation was made of existing networks whose performance and uses were similar to those envisioned for ours. Three candidates emerged: a modified version of the Ethernet developed by Robert Metcalfe at the XEROX Palo Alto Research Center, the ring network built for the Distributed Computer System by Professor David Farber at the University of California at Irvine, and the SPIDER network developed by Sandy Fraser at Bell Laboratories. The general architecture of both the Ethernet and the ring network is depicted in Figure 1.

2.1 The Available Technologies

The Ethernet was fairly well known to us in the spring of 1975, and was recommended to the subcommittee by Richard Greenblatt of the M.I.T. Artificial Intelligence Laboratory. Greenblatt, at that time, was already planning to build a small, modified version of the Ethernet, which he termed the "Chaos-Net". In essence, the Ethernet uses coaxial cable as a transmission medium on which messages are broadcast in bit serial form by the sending host, and are picked up by the appropriate receiving host; the coaxial cable transmission medium is thus shared by all hosts. If two hosts begin to transmit at the same time, a "collision" of messages occurs, and each host must transmit its message again. The effective data rate on the PARC Ethernet is just under 3 Mb/s.

The ring network developed by Farber at UC-Irvine was somewhat less well known to us. Essentially, it uses point-to-point differential transmission over twisted pair between adjacent nodes connected in a ring. Each node's "ring interface" regenerates the incoming signal and passes it on to the next node, introducing several bit-times of delay. To transmit, a host's ring interface awaits the passage of a "control token" bit pattern, and then breaks the repeater connection across the interface, gating its message, bit serially, onto the ring. The intended receiver copies the message as it passes through its ring interface, and the sender's ring interface removes the message from the ring and passes on the control token bit pattern. The data rate of the original ring was on the order of 1 Mb/s; Farber now has a contract from ARPA to construct a new version of the ring network for ARPA's Intelligent Terminal Project. The new ring interface is to use specially-developed LSI circuits and programmed logic arrays in the ring host interface to help achieve greater reliability.

Both the Ethernet and the ring network are totally decentralized; no single host, or special host, controls the operation of the network. This is an extremely desirable feature for a network to be used in an environment such as exists at the Laboratory, where independent groups of people work at all hours of the day. Reliable operation of a decentralized network is <u>not</u> dependent upon the reliable operation of a particular computer system; any number of hosts on the network may fail, and the network as a whole will continue to operate (*).

Because of the desirability of complete decentralization, the Bell Labs SPIDER network was of less interest to us. Although it is a ring network, one host on the ring is in complete charge of all communication on the ring. The SPIDER network, therefore, was not considered further in the study.

2.2 Comparison of the Ring Network and the Ethernet

The architecture and hardware of the ring network and the Ethernet, in their original form, are very different. At first glance, the functional capabilities of the two seem quite different as well. However, discussions with Metcalfe and Farber, and with others in our Laboratory, led to the conclusion that there are few inherent differences in the functional capabilities of the basic Ethernet and ring network communication schemes. In addition, as will be seen, a modification can be made to the design of the host interface for both of these networks which renders them nearly identical in functional capability. In this section, we begin by comparing the hardware architecture and functionality of the two networks as originally built. We then indicate ways in which the two can be made more similar.

(*) The ring network requires that either the repeater portion of all ring interfaces be powered, or that provisions be made to automatically bridge (e.g., with relay contacts) a repeater whose power has failed.

Figure 2 depicts, in block diagram form, the hardware architecture of the two networks as originally built. The host interface for each connects to the host computer on one side, and to the network's transmission medium on the other. The transmission medium of the Ethernet is coaxial cable, shared "party-line" fashion by all of the host interfaces. Each host interface is connected to the cable via a receiver/transmitter which provides appropriate cable driving and isolation circuitry. The ring network uses twisted pair as its transmission medium; signals transmitted around the ring are regenerated by each receiver/transmitter. The host interface directs the receiver/transmitter either to repeat the incoming data, bit for bit, or, when it desires to transmit, to gate data from the host interface onto the ring.

When an Ethernet host wishes to transmit a message, its host interface listens for other signals on the cable; when none are present, it begins to transmit its message. The Ethernet is thus a "contention" system; messages destroyed by collisions must be retransmitted. The ring network, on the other hand, is a completely ordered system; a "control token" passed around the ring determines when each host may transmit. In addition, the Ethernet provides no built-in message acknowledgement scheme, while the ring network does provide such a scheme; acknowledgement bits at the end of a message are turned on by the receiving host's ring interface so that a message, having traveled around the ring and returned to its originator, contains its own acknowledgement.

In addition to these architectural differences, the ring network interface, as originally built, contained a "name table" which can be viewed as providing either an array of process names (for use by Farber's Distributed Computing System) or, more generally, a set of software-settable network addresses for the host. The inclusion of this name table is incidental to the basic ring network communication scheme; it could just as easily have been part of the Ethernet, and thus its presence in the original ring network does not constitute a real "difference" between the communication schemes of the two networks.

We now consider how to bring the two networks closer together, both in functionality and in the interface they present to their host computers. While we do not wish to change the basic principles of either the Ethernet or the ring network, in designing the LCS Network neither do we wish to restrict ourselves to copying either of them exactly as they have previously been built. Greenblatt suggested a modification to the Ethernet host interface which can be applied to the ring network as well: he proposed that packet buffers be incorporated into the interface, so that data transfers from the host to the host interface, and from the host interface onto the network transmission medium itself, can take place independently and at different rates.

It is this change, applied to both the Ethernet and the ring network, which nearly unifies the appearance of both networks to a host. When the host has a message to transmit, it loads it into the interface's transmit buffer; when the interface decides the message can be sent -- in an Ethernet, when no other transmissions are in progress, or, in the ring network, when the "control token" comes around -- the interface begins its transmission. In the case of an Ethernet, the message is retained in the transmit buffer until it is successfully transmitted. In the event of a collision, the host interface can re-transmit the message itself; the re-transmission process can thus be completely hidden from the transmitting host. With a ring network, the meaning of the acknowledgement bits change somewhat; they become an indication that the message was delivered to the receive packet buffer of the destination host interface. Receipt of messages, too, becomes the same for both networks. When the host interface determines that a message on the network -- either ring or Ether -- is for its host, it gates the message into its receive packet buffer. When the entire message has been received, it can be transmitted from the host interface receive packet buffer to the host at the host's convenience.

2.3 Choosing Between Two Interchangeable Networks: Practical Criteria

Because the augmented host interface for both networks presents the same appearance to a host, the electrical interface to the host can be identical for both networks. Also, the same protocol can be utilized for both networks. Operationally, from a host's point of view, the networks can be the same, with the exception that the ring network provides low-level acknowledgement information which the Ethernet does not. The ring network scheme may thus have a slight advantage over the Ethernet in an application where this information can be used. With this single exception, the functionality of the two networks is the same.

What criteria, then, should be used to choose between the two network technologies? We now consider practical criteria, such as ease of implementation, ease of reconfiguration, speed (data rate), and cost.

Shielded twisted pair, as used in Farber's ring network, is an excellent point-to-point transmission medium. It can easily handle the 1 Mb/s data rate of Farber's original ring over moderate distances (up to 1000 feet between repeaters), and is expected to work well at the 2-4 Mb/s data rate which is the target for Farber's new ring. Coaxial cable, especially the variety used for CATV applications, can handle the 4-10 Mb/s data rate anticipated for the AI Lab Chaos-Net, but is physically somewhat more difficult to work with. Also, attenuation of signals on the coax, as well as an increased frequency of collisions due to greater transmission delay, make the Ethernet scheme less suitable for application where the overall length of the network is long -- greater than a half mile or so.

The receiver/transmitter for the Ethernet is, of necessity, more complex than that for the ring network, and thus will be somewhat more expensive. Part of the complexity results from the fact that the Ethernet receiver/transmitter must not disturb the transmission medium when it is not transmitting; in particular, it must not fail in such a way as to "jam" the cable. On the other hand, the ring network's receiver/transmitter must contain some means by which it can automatically be removed from the ring, or "bridged" in the event of failure, so that the operation of the entire ring is not disrupted by the failure of a single host interface.

We now consider once again the single remaining functional advantage of the ring network: its built-in acknowledgement mechanism. In a ring network which is totally self-contained -- *i.e.*, not connected to any other network -- the acknowledgement bits do serve to indicate that a message was, in fact, delivered to the appropriate destination host. However, in any sort of internetworking situation, the bits merely indicate that a message destined for some other network was handed over to the appropriate "gateway" host for dispatch to that other network. Any acknowledgement of receipt of the message at its final destination must be done by protocol. Since the LCS Network will, in fact, be in an internetworking environment, it will not be able to take advantage of the ring network's acknowledgement bits.

Which network technology, then, shall we choose for the LCS Network? The LCS Network is designed to be the first subnetwork of a potential M.I.T. campus-wide network. Various sub-networks can be built with varying network technologies, with appropriate "bridges" between sub-networks. Given this flexibility, we propose to adopt the Ethernet technology for the initial LCS Network. We leave open the possibility of building a second sub-net, using ring network technology, soon after the first, so as to give some practical experience operating functionally equivalent networks using different technologies.

The remainder of this paper (except for Section 6) deals only with the initial, Ethernet-technology LCS Network.

3. GENERAL NETWORK ARCHITECTURE

This section considers the hosts to be attached to the LCS Network, and discusses the nature and arrangement of the coaxial cable transmission medium. The structure of the host-to-network interface, discussed in very general terms in Sections 2.1 and 2.2, is discussed here in somewhat greater detail. Also discussed here are the packet size and data rate of the network, the re-transmission algorithm to be used after collisions, and the modulation or encoding scheme to be used on the cable.

3.1 Hosts for the Local Network

Having decided to implement a buffered Ethernet, we must now develop a general architecture, or layout, for the Laboratory's network, based upon the location and type of host computers we wish to have in the Network. We begin by enumerating the hosts at the Laboratory for Computer Science:

Quan.	Type	"Owner"	Location
1	PDP-10	Mathlab	9th floor
1	PDP-10	Programming Technology	9th floor
1	KL-10	MACSYMA Consortium	9th floor
1	ARPANET Gateway	(projected)	9th floor (near IMP)
1	PDP-11/70	Delphi	5th floor

Two hosts distant from the Laboratory, but yet extremely important to the Laboratory, are:

1	Honeywell 6180	IPC-Multics	Bldg. 39
1	Honeywell 68/80	Multics Development	Bldg. NE45

In truth, the list of LCS hosts existing today is rather short. Two important questions which have an impact on the number of hosts we can expect to have on the Network are: to what extent does the AI Laboratory wish to participate in the LCS Network? To what extent would we like the AI Laboratory to participate? Full AI Laboratory participation would add:

1 PDP-10	AI	9th floor
? PDP-11's	AI, assorted	9th floor
? PDP-11's	Logo, assorted	3rd floor

The near future holds possibilities for other local hosts as well:

- Professor Jack Dennis' "data flow" machine
- A terminal concentrator (possibly the PDP-11 associated with the MACSYMA Consortium KL-10)
- A "central file system" machine.
- A second KL-10, for the Programming Technology Division

In addition, the future may bring:

- Ten to forty "personal computers".

Thus, we can expect from five to twenty hosts, excluding the personal computers, to be attached to the LCS Network in the first several years of its existence. The personal computers, if they become a reality, could increase this estimate to a range of fifteen to sixty hosts. Further expansion outside our building, to other locations on the M.I.T. campus, is a possibility which is discussed later in this report.

One limitation on the number of hosts is imposed by the addressing scheme used in the LCS Network. If an eight-bit byte is used for a host address, the Network will be limited to 256 hosts; this should be adequate for a network serving only the Laboratory itself for the forseeable future. Jerry Saltzer has suggested that we take a broader view, however, and devise a "numbering plan" for the Network which can be expanded to meet the needs of the entire M.I.T. campus. The LCS Network will then be the initial sub-network of a campus-wide network. A second eight-bit byte could be used to specify the desired sub-network, bringing the number of bits used to address a network host to 16.

Another, "softer" limitation on the number of hosts in the LCS Network is imposed by traffic on the Network. As the number of hosts increases, the traffic on the net will increase, and, especially during peak traffic periods, transmission delays will increase. As we gain experience with the net, we will be able to determine the maximum acceptable delay and load imposed by individual hosts, and, from that, determine the maximum number of hosts which should be allowed on the Network.

3.2 Arrangement of the Cable

Two arrangements for the coaxial cable suggest themselves (Figure 3): first, a single segment of coax, meandering around all floors of the building on which we expect to have computers to be connected; second, a multi-segment layout consisting of a "spine" running vertically in the building and "spur" cables on each floor, connected to the spine with a simple repeater composed of a cross-coupled pair of receiver/transmitters (*).

The choice between these two is based upon simplicity and ease of installation, maintenance and modification, and not upon density of hosts in the Network. All signals are repeated on all spurs, so traffic is identical on each spur. The spine/spur scheme can have a major advantage over the single-segment scheme in a network composed of many hosts, widely scattered throughout a building: it reduces the maximum electrical "distance" between two hosts, since the same physical cable segment need not pass by all hosts. The distance between any pair of hosts, as shown in Figure 4, is the sum of the lengths of the spur cable from each host to the spine, plus the length of the spine between these two spurs. In effect, a spine/spur arrangement reduces the total logical length of the network.

The spine/spur scheme appears to be best for the Laboratory over the long term, especially as we obtain a number of personal computers, and other mini- and micro-computers, which we wish to connect to the Network, on an ever-increasing number of floors of the Laboratory. Prudence suggests, however, that we initially use a single cable segment to interconnect the limited number of hosts we currently have. The reasons for this are three-fold. First, it will be easier to bring up the network initially without concerning ourselves with the spine/spur repeaters. Second, the number of hosts which will initially be on the net is small and does not justify the more complex scheme. Third, the coaxial cable transmission medium

---

(*) This multi-segment technique has been used successfully at Xerox-PARC.

is one of the lowest-cost components of the network; the single-segment cable layout can easily be replaced with the spine/spur arrangement at a later date.

3.3 Overview of Network Host Attachment

The host-to-network interface consists of a host interface proper and a receiver/transmitter. The receiver/transmitter is a small, shielded package directly connected to the coaxial cable; it is connected to the host interface, by means of a multiple twisted pair cable. Present plans call for the receiver/transmitter to obtain its power from the host interface over this cable; however, another possibility would be to power the receiver/transmitter via the coax, as is done in CATV systems. Although there may be several types of host interface, one for each type of host computer, there will be only one type of receiver/transmitter, and one standard interconnecting cable. The receiver/transmitter power supply issue is an important one, and is discussed in Section 4.2.

The host interface itself, described in detail in Section 4.3, is mounted in, or near, the host computer, and can be powered from the host computer. It communicates with the host using an appropriate mechanism, e.g., Unibus for a PDP-11, DMA (Direct Memory Access channel) for some hosts, etc. This means, essentially, that there must be a separate type of host interface for each type of computer that is to be directly connected to the network. These various types of host interfaces differ only in their method of communicating with their respective host; most of the circuitry is the same for all types. In particular, there is but a single interface to the receiver/transmitter; all host interfaces will use identical interconnecting cables. Initially, only PDP-11 host interfaces will be built.

The decision to have a different type of host interface for each type of host connected to the Network is contrary to Greenblatt's original idea for his "Chaos-Net", for he proposed that only PDP-11's and LISP machines, with an interface identical to that of a PDP-11, be connected to the Net. Clearly, it is easier to design only a single type of host interface. Greenblatt argued that larger hosts, such as PDP-10's, could be "front-ended" with very small PDP-11's. This would be a relatively easy task, since the PDP-11 Net interface would already have been designed, and the AI Lab has already built many 10-to-11 interfaces. The PDP-11 would be transparent in its operation, shuffling data from the 10 to the Net, and vice versa. However, at least three PDP-10's will be connected to the LCS Net; the unit cost of designing a PDP-10 interface should not be too great. The choice between these alternatives is basically economic, a tradeoff between the cost of a PDP-11 Net interface, a small PDP-11, and a 10-to-11 interface, vs. the cost (including the amortized design cost) of a PDP-10 interface. The decision is up to the owners of the PDP-10's; however, we recommend that the separate PDP-10 host interface be developed.

3.4 Providing an Attachment for Multics

The connection of the Multics system in Building 39 poses a problem. Extending the main coaxial cable of the LCS Network (or a spur) to Building 39 would about double the length of the cable -- all for the sake of a single host (presently; however, see also Section 6, For the Future). It is unlikely, though, that we would take the expensive course of having a new coax installed underground between Tech Square and Building 39. If we did wish to logically extend the Network, we could use the second of the two CATV cables installed by the Laboratory in 1974 in conjunction with installation of the Multics IMP-Host cable (*).

On the other hand, it might be easier to treat the Multics system as a special case in one of two ways:

- Not connect it at all, but have it use the ARPANET Gateway to communicate with local LCS hosts.
- Devise a special host interface scheme, using twisted pair, or possibly packet radio, as the transmission medium between Multics and a special buffered interface at the Tech Square end.

3.5 Packet Size and Data Rate

The data rate on the Ethernet coaxial cable at PARC is very closely tied to the rate at which bits can be transferred directly in and out of the memory

---

(*) The first is being used by the MIT Cable TV system.

of the Alto computers which constitute the majority of the hosts on the Ethernet. The LCS Network should not be so strongly tied to a particular type of host computer. Also, we would like to be able to take advantage of changing technology, increasing the data rate of the Network if it seems advantageous. Providing packet buffers in the host interface aids us in meeting both objectives.

This double benefit is not without cost, however. The hardware of PARC's Ethernet imposes no limitation at all on the length of a message (*). Very long messages, when appropriate, improve the efficiency of utilization of the cable, although they increase the delay seen by other hosts waiting to transmit. Providing packet buffers in the host interface imposes an absolute limit on the length of a message, and the consequences of doing this must be taken into account. Assuming that the network protocol permits only single-packet messages, the terms "message" and "packet" become synonyms, and they are used synonymously in the remainder of this paper.

The selection of a packet size for the network must be made at the time the host interface hardware is designed, as the RAMs for the packet buffers must be included in the design. Examination of traffic in the ARPANET has shown that most messages are either very short (representing interactive I/O, such as terminal input or output) or at the maximum message size (representing fragments of file transfers).

The Transmission Control Protocol used in internetworking specifies a message format with a <u>header</u> that is 256 bits long. If the LCS Network is ever to carry TCP messages (as it well may), the packet size <u>must</u> be greater than 256 bits if absurd amounts of overhead are to be avoided. Thus, assuming that the packet size should be a power of 2, 512 bits is a lower bound for the maximum packet size. This should neatly contain most interactive-type messages.

At the other end of the scale, delay and the cost of the RAMs must be taken into consideration. Efficient file transfer (seen to be one of the major uses of the LCS Network) requires as large a packet size as possible.

(*) However, a maximum message size has been established, by convention, in the software of the hosts on the Ethernet.

If the Network were to operate at 4 Mb/s (pretty much a lower bound on the speed at which it might operate; see discussion below), a 4k bit packet could be transmitted in one millisecond. If fifty hosts were each waiting to transmit such a packet, the last host to transmit would have to wait 50 msec. (clearly, this is a very rough calculation, and does not represent a true worst-case delay; inter-message "quiet time" on the cable, false starts due to collisions, etc., have all been ignored). This delay could be cut in half either by operating at 8 Mb/s, or using 2k bit packets; it could be cut by a factor of 4 if both of these are done. A delay time of 50 msec. under extremely heavy traffic conditions should certainly be acceptable. Thus, 4k bits constitutes a reasonable upper bound for the maximum packet size.

Given the availability of larger RAMs, and the increased efficiency attained with larger packet sizes, there is little motivation for choosing a packet size at the lower end of the range. We choose 2k bits as the packet size for our original implementation.

The data rate at which the LCS Network will operate is limited primarily by the characteristics of the coaxial cable and the circuit elements used to provide isolation between the cable ground reference and the host ground reference. In PARC's Ethernet, this isolation was accomplished with pulse transformers. Greenblatt intended to use opto-isolators for his "Chaos-Net"; however, it is not clear whether opto-isolators currently available can achieve the high data rates desired for the LCS Network. It is expected that, as opto-isolator technology improves, the data rate of the Network can be increased to accomodate increasing traffic.

Ideally, we would like to operate the Network at a rate of 10 Mb/s, which is within the capabilities of the coaxial cable. 8 Mb/s would be a convenient rate, as, with 8-bit bytes, we could speak of a 1 mega<u>byte</u> per second data rate. A data rate of 4 Mb/s is closer to the speed of PARC's Ethernet, a rate at which the pulse-transformer isolation circuitry operates well. Thus, 8 Mb/s can be established as the <u>target</u> data rate for the LCS Network, with 4 Mb/s as a fall-back.

3.6 The ARPANET Gateway

One of the goals of the LCS Network is to provide access to the ARPANET for computers at the Laboratory which are not themselves ARPANET hosts. A computer such as a PDP-11 can be installed as a host on both the ARPANET and the LCS Network to serve as a "gateway" between the two networks. The term "gateway" has been given several different interpretations in the literature and in discussions on inter-networking. According to one "narrow" interpretation, a gateway is a relatively small computer, with relatively simple software, which provides an interface between two networks at the hardware level and at the lowest possible of protocol levels. A "broad" interpretation holds that, in addition to performing the function just mentioned, provides interfaces between the higher-level (e.g. "user-level") protocols of the networks involved.

The "strict", or low-level, gateway functions must be performed in the machine which is physically connected to both networks, for it must perform the task of removing the protocol "wrapper" in which a message arrives encapsulated from its source, and "re-wrapping" it according to the protocol of the destination network. It is important to note, however, that the higher-level functions implied by the broad definition of the term "gateway" need not be performed on the physical gateway machine; these functions can be provided by some other host on one of the networks. Of course, it may be convenient and/or economical to provide both classes of gateway functions on the same machine, but it is important to realize that they need not be, and that the low-level gateway machine ought to be small and simple. The amalgam of all gateway functions, both low- and high-level. wherever performed, are referred to here under the single umbrella term of "the Gateway".

The low-level gateway function is all that is needed when an LCS Network (local) host and an ARPANET ("distant", or "foreign") host can both communicate via the Transmission Control Protocol, which is specifically designed for internetworking. However, a higher-level function must be performed when each host can only communicate in its "native" network protocol. Communication between an ARPANET host and an LCS Network host can most easily to achieved by implementing this function on the machine providing the low-level gateway function, and assigning a portion of the Gateway's

ARPANET socket space to each of the LCS Network hosts. An ARPANET socket identifier is 32 bits long. If the LCS Network adopts a 16 bit host address, this host address can be used as the high-order 16 bits of an ARPANET socket number at the Gateway to specify the target LCS Network host, leaving 16 bits to specify a particular LCS Network socket at the target LCS Network host. As LCS Network socket identifiers are sure to be longer than 16 bits, software designers for LCS Network hosts which choose to not implement the Transmission Control Protocol, but wish to communicate with ARPANET hosts, must be aware of this restriction on use of the host's socket space.

The remaining major problem is that of an ARPANET user desiring a service from an LCS host -- a service such as Server TELNET, File Transfer Protocol Server, or Message (mail) Service. The "contact sockets" for these services are standardized for all ARPANET hosts; user programs know how to perform an Initial Connection Protocol (ICP) to the appropriate standard socket. If contact sockets were to be established for an LCS Network host in the socket space at the Gateway assigned to that host, their socket numbers would be different from the standard, and user programs at other ARPANET hosts would not be able to obtain the services desired from the LCS Network host in the usual, straightforward manner. A better solution is to have user programs at ARPANET hosts make their service requests of the LCS Network Gateway itself; the Gateway, then, must dispatch the service request to the proper LCS Network host. In effect, the Gateway must operate an ARPANET Server TELNET, File Transfer Protocol Server, etc. on behalf of all LCS Network hosts. How this can be done is described below.

TELNET service can be handled quite nicely, for it can be assumed that somewhere at the other (ARPANET) end of the connection a human user is sitting at a terminal. The Gateway can therefore transmit an LCS Network greeting, and query the user as to which LCS Network host he wishes to communicate with. The Gateway can then establish an ordinary LCS Network connection to the Server TELNET at the desired host.

The File Transfer Protocol (FTP) Server is more difficult. Here, the dialogue between the user program and the server program is pre-defined; there is no FTP user command to "hand off" or relay an FTP connection to a sub-network host. Moreover, while it is possible to embed the desired LCS

Network host name in the "file pathname" supplied in an FTP file transfer command, the structure of the File Transfer Protocol, and its common usage in the ARPANET, makes this difficult. A careful look must be taken at what can be done here, although it may turn out that the best that can be hoped for is an awkward solution.

Better accomodation can be made for the ARPANET Message, or mail, Service. At present, this is a specialized subset of the File Transfer Protocol, usually operated by an automaton at the user end. The Gateway can maintain a directory of all persons who receive ARPANET messages on the LCS Network, and, by using this directory, route incoming messages from the ARPANET to the appropriate LCS Network host. The Gateway thus serves as a "forwarder" of messages from one network to the other.

As can be seen from the foregoing discussion, the gateway functions, while not trivial, are basically straightforward and well understood. They are an important piece of the LCS Network; it is expected that they will be implemented soon after the Network itself is operational.

3.7 Re-transmission Strategy

When a collision occurs in the Ethernet, the host interface involved must "back off", delay awhile, and then re-transmit its packet. Of course, if the two host interfaces involved in the collision each wait the same length of time before re-transmitting, the collision is likely to recur. It is the task of the re-transmission algorithm implemented in the host interface to vary the re-transmission delay to minimize the probability of a second collision.

Metcalfe has indicated that, with the relatively low traffic rate on the Ethernet at PARC, almost any algorithm will work; that is, the difference in overall network performance between a "perfect" re-transmission algorithm which never results in a second collision, and a far-from-perfect algorithm which sometimes does yield a second collision, is virtually undetectable. A "smart" re-transmission algorithm can, however, reduce the average delay necessary before re-transmission without greatly increasing the probability of a second collision. Greenblatt has proposed a fairly elaborate scheme for his "Chaos-Net" in which hosts are located along the coaxial cable in numerical order (or close to it); a host interface bases its re-transmission delay on the difference between its host number and the host number of the last host which successfully transmitted a message (which thus must be remembered by all host interfaces).

The ordering of the hosts along the cable aids in minimizing the required post-collision delay time for all hosts; the host interface which is closer (both physically and in host number) may begin transmitting with impunity soon after the collision, for the more distant host interface will delay longer, see the beginning of the transmission of the closer host interface, and defer to it. In effect, Greenblatt's scheme, after a collision, replaces the *contention* of hosts waiting to transmit with a *total ordering* of hosts *a la* the Ring Network.

The host number differential scheme may be used without the ordered arrangement of hosts on the cable, with an attendant increase in the minimum post-collision delay time. It is not clear whether the ordering of hosts along the cable presents any real handicap with a single-segment cable; however, it is impossible to accomplish with the spine/spur arrangement. Since we anticipate that the LCS Network cable will eventually be a spine/spur cable, we must reject the ordered host number approach. The host number differential scheme can, however, be adopted for the LCS Network.

3.8 Modulation Scheme

In the Ethernet, the word "modulation" is used to describe the process by which the bits of a packet are placed onto the coaxial cable transmission medium. No actual carrier is being modulated in the traditional sense; in fact, the process could better be described as an encoding of the data into signal elements as is done, for example, in the recording of data on magnetic media such as disk or tape. Modulation, then, means the conversion of the bits of a packet into a signal suitable for transmission on the coaxial cable.

The use of terms from the radio communication field provides a useful, if somewhat misleading, analogy. The Ethernet *does* have a transmission medium; it has a transmitter and receiver, which can be said to perform modulation and demodulation of the signals broadcast on the transmission medium. In fact, with a properly-chosen encoding, or modulation scheme, it is easy to detect the presence of a stream of bits on the cable and, continuing the analogy, detect the presence of a "carrier".

Any one of a number of encoding schemes can be used. The basic requirements are:

- The scheme must be self-timing; that is, it should be possible to derive timing information from the stream of encoded signal elements on the cable.
- The signal element must include a state change each bit time, regardless of the pattern of data. This prevents a long string of zeros, ones, or other repetitive pattern from appearing the same as the absence of any message.

This second requirement implies a third:

- The process of "demodulating" the stream of encoded signal elements should result in a "carrier detect" signal which can be used to determine when a message is being transmitted.

PARC's Ethernet uses what Metcalfe terms "phase encoding", often called "NRZI" encoding, which is widely used in recording magnetic media. Figure 5 shows the NRZI encoding of the bit string "10011". Essentially, NRZI encoding consists of transmitting the complement of the bit value for half a bit time, followed by the sense of the bit value for half a bit time. This results in a mid-bit-time transition which denotes the value of the bit: a positive-going transition represents a one, and a negative-going transition represents a zero. There may or may not be another transition between bit-times, depending upon the sequence of bits being transmitted (see Fifure 6).

Other encoding schemes may produce fewer state transitions, on the average, resulting in a signal of somewhat narrower bandwidth. The phase encoding scheme, however, is simple and effective, and the coaxial cable can handle the required bandwidth; therefore, we propose that it be adopted for the LCS Network.

4. HARDWARE DESCRIPTION

As mentioned previously, the LCS Network will have three major hardware components: the coaxial cable transmission medium, the host interface, and the receiver/transmitter. Each of these will now be discussed in greater detail.

4.1 Coaxial Cable Transmission Medium

The LCS Network will use a type of coaxial cable used in CATV applications; experiments conducted by Tom Knight of the M.I.T. AI Lab this past summer have demonstrated that it has better, more uniform pulse transmission and attentuation characteristics than either of the two coaxial calbe types (RG 11/U, type "foam", and RG 59/U) used in PARC's Ethernet. A single logical segment of coax will be used; physically, the coax will run from receiver/transmitter, with the transceiver providing a "through" connection from one segment to the next, as well as a tap for the receiver/transmitter itself. The cable must be terminated in its characteristic impedance at each end, and the cable shield must be solidly grounded at one location. The cable drivers and receivers of the receiver/transmitter (described below) must be referenced to this cable shield ground.

The maximum overall length of the cable is limited by the attenuation of the cable and by the impact of longer cable transit times on network performance. As the length of the cable increases, the longer transit time increases the maximum collision detection time; under heavy traffic conditions, this will result in an increase in the percentage of time wasted in the transmission of portions of packets which will collide. Neither of these factors should affect the initial network, however, and the length of the cable can be determined solely by our initial needs.

The suggested cable layout for the building is shown in Figure 7. The ninth floor loop is intended to be run under the false floor in the machine room portion of the ninth floor, and above the ceiling of the corridor in the office sections of the ninth floor. On the eighth and fifth floors, a loop will be run above the corridor ceiling. On the third and second floors, a half-loop will be run, also over the ceiling. If desired, the tail end of the cable can be dropped down to the first floor.

4.2 The Receiver/Transmitter

The receiver/transmitter is shown in Figure 8. Structurally, it is a fairly simple device, but it plays a very important role. Basically, it couples signals from the host interface to the cable, and vice versa, while providing electrical isolation between them. All signals on the cable are repeated via the cable to the host interface by the the receiver/transmitter; the receiver/transmitter does not distinguish signals intended for its host from others on the cable. The receiver/transmitter performs one additional vital function: when the host interface to which it is connected is transmitting, it compares the signal with which it is driving the cable with the signal it is receiving from the cable; a difference in the two signals indicates that some other host is also transmitting, and the receiver/transmitter signals its own host interface, via its interface cable, that a collision has occurred.

In order to maintain electrical isolation between the cable and the host interface, the receiver/transmitter must have two power supplies, one powering the coaxial cable drivers and receivers (referenced to cable ground), and one for the host interface drivers and receivers (referenced to host, local ground). It is proposed that low voltage AC (9, 12 or 24 V) be supplied on the interconnecting cable from the host interface, and that the receiver/transmitter include a small transformer with several secondary windings, feeding a separate power supply for each of the two sections of the receiver/transmitter.

The basic design of the receiver/transmitter is being performed by Tom Knight and Jack Holloway of the AI Lab; the same receiver/transmitter will be used for both the AI "Chaos-Net" and the LCS Network.

4.3 The Host Interface

The host interface, diagrammed in Figure 9, is divided into several parts. On one hand, it interfaces to the host's I/O structure, and on the other hand, to the receiver/transmitter. In between are the packet buffers, one transmit and two receive (to provide double buffering on the receive side, as described below), and the control logic, including a crystal oscillator to control the transmit and receive data rates. The host I/O circuitry is the only portion of the interface that is host dependent. By replacing this portion of the host interface, it is possible to interface to different types of host computers, as mentioned in Section 3.3.

Internally, the host interface deals with eight-bit bytes. The packet buffers, in particular, are eight-bit byte RAM's. Outgoing bytes are converted to serial form as they are passed to the receiver/transmitter, while the incoming bit stream is converted to eight-bit bytes as it is received from the receiver/transmitter. If the connected host's I/O interface handles eight-bit bytes comfortably, all is well. If some other byte size is more appropriate, the interface's host I/O circuitry must make an appropriate conversion.

The receiver/transmitter logic of the host interface monitors cable activity as received from the receiver/transmitter (as noted in the previous section, signalling between the receiver/transmitter and the host interface utilizes differential transmission over shielded twisted pair). As a message goes by on the cable, the control logic of the interface compares its first bytes -- its destination address -- with the local host address, and, if it matches, the message is routed into an empty receive packet buffer. If there is no empty receive packet buffer, the message cannot be received, is therefore ignored by the host interface, and must be retransmitted later by its sender.

When the entire message has been received, and its checksum has been verified, it is passed from the receive packet buffer to the host via the host I/O circuitry. This transfer can take place at any rate convenient to the host. The use of packet buffering within the host interface enables the cable-to-interface and interface-to-host transfer to proceed independently of each other; the use of double packet buffering on the receive side of the host interface permits the reception of a second message close on the heels of the first, eliminating the otherwise-guaranteed discard of the second message.

The host can initiate the transfer of a message into the transmit packet buffer whenever it is empty. A cyclic checksum is computed as the message is gated byte-by-byte into the buffer, and the checksum bytes are appended to the message stored in the buffer. Once the buffer has been loaded the control logic of the interface will wait until the cable is quiet, and then will begin to transmit the message through the receiver/transmitter. If the receiver/transmitter detects a collision, as described in the previous section, the host interface aborts transmission of the packet, and directs the receiver/transmitter to "jam" the cable momentarily to ensure that the conflicting host interface also detects the collision. The message to be sent is still in the transmit buffer, however, and the host need not be bothered. After an appropriate delay (see discussion of the retransmission algorithm in Section 3.7), transmission of the packet is begun again.

Thus, on the transmit side, the packet buffer not only permits different data rates for the host-to-interface and interface-to-cable transfers, but eliminates an interaction with the host in the event of a collision.

5. NETWORK PROTOCOLS

It is not enough, of course, to provide common hardware for all the hosts which wish to connect to the LCS Network; some standard protocols must be developed as well. It has been a goal of the work done so far to ensure that the hardware described in the previous sections can support the sorts of protocols we will have for the LCS Network. David Reed, working as a DSR Staff member at the Laboratory this past summer, has taken charge of protocol design and development, lending his expertise to this important area. A report on the work he has done will be published shortly in a companion CSR RFC.

5.1 Criteria for Protocol Design

The nature of the buffered Ethernet greatly influences the design of protocols for the LCS Network. For example, not every message transmitted by a host successfully reaches its destination. Even with collision detection and resulting automatic re-transmission implemented entirely within the host interface, some messages will not be properly received due to glitches in the network resulting in checksum errors, or because the receive packet buffers of the destination host are full. Thus, network protocols must require that messages be acknowledged. Since a message in a sequence of messages may be "missing", and later retransmitted, messages may be received out of order. A host may be tardy in acknowledging receipt of a message, causing the sender to retransmit; this will result in receipt of multiple copies of a message. All of these situations must be provided for in the protocol design. In order to properly implement these protocols, we assume that all hosts in the network are computers with at least a modest amount of processing power, and that none are merely hardware controllers (this assumption will be examined later).

To summarize, then, protocols for the LCS Network must provide for:

- positive acknowledgement of received messages
- sequencing
- retention of unacknowledged messages in host buffers

- re-transmission
- discarding of duplicate messages

In implementing these protocols, we assume that we may take advantage of

- a "modest" amount of processing power

5.2 The Host-Host Protocol

In the ARPA Network, four levels of protocol are often at work, operating upon user's stream of data. This complex layering of protocols has given rise to concern over the design of protocols for other networks, and engendered a desire to minimize the number of separate layers of protocol in newly-designed networks. We presently intend to have a single-level protocol in the LCS Network provide a user process with the ability to communicate with user processes at other network hosts; this single-level Host-Host protocol will provide functionality similar to that of the ARPANET Host-Host protocol, which has two other levels of protocol beneath it.

Both byte-stream- (virtual connection) and message-oriented ("datagram") protocols have their proponents. We hope to develop a Host-Host protocol which can be used in both ways. Dave Reed has suggested that the protocol be "port"-oriented, and this, too, merits investigation.

The LCS Network will be connected to the ARPANET via a gateway computer, so the design of the Host-Host protocol for the Network must take into account the fact that it will be carrying messages to the gateway for dispatch to the ARPANET or to other networks, and must embed, encapsulate or otherwise deal with the ARPANET Host-Host protocol and other protocols as well, such as the Transmission Control Protocol (TCP) now being used in internetworking experiments.

The Host-Host protocol will carry through the eight-bit-byte architecture which we saw at the hardware level; message header information fields will be assigned in eight-bit bytes, and the amount of data contained in a message will be expressed in eight-bit bytes. The protocol will include a sequencing mechanism, as mentioned in the previous section.

5.3 User-Level Protocols

So far, no mention has been made of user-level protocols such as the ARPANET's TELNET and File Transfer Protocols. Since the Host-Host protocol for the LCS Network will provide a stream of eight-bit bytes, the ARPANET TELNET protocol can be easily adopted, reducing the amount of software which must be implemented for each LCS Network host. Similarly, a subset of the ARPANET File Transfer Protocol can be used initially for file transfer on the LCS Network, although a modified version of the special inter-ITS file transfer protocol could also be used. A new user level protocol oriented towards fast, efficient file transfer can be developed later, when it is needed for use with the proposed central file system machine.

5.4 Protocols for Micro-Hosts

We now go back and examine an assumption made earlier, namely, that all hosts have at least a "modest" amount of processing capability. It is very likely that as time goes on we may wish to connect smaller devices to the LCS Network, devices which do not, for example, have sufficient buffer space to store a sufficient number of un-acknowledged messages. An example of such a device might be a high-speed terminal connected directly to a micro-processor based network interface (see Figure 10).

For such "micro-hosts" it would be desirable to use a restricted version of the network protocol, with very strict rules regulating the buffering of messages and their acknowledgement and retransmission. A micro-host must be able to communicate with "regular" hosts as well as with other micro-hosts. This suggests that a restricted "micro-host protocol" should, in fact, be a subset of the regular Host-Host protocol, with the use of the subset between any pair of hosts initiated and controlled by either host of the pair. This is, to be sure, a complication in the initial protocol design, but one which should pay off in the long run.

6. FOR THE FUTURE

The initial implementation of the LCS Network will not fill all of the Laboratory's needs. Other topics, as well, need investigation. They can be developed, after the initial implementation of the network, as time and resources permit. These topics are discussed briefly here.

6.1 Central File System

One of Greenblatt's original motivations for development of his "Chaos-Net" was his desire to provide a central file system for his LISP machines. The desirability of such a central file system, accessible rapidly and efficiently to smaller host computers without file systems of their own, does not diminish in the larger context of the LCS Network. For the initial implementation of the Network, the file system of one or more of the PDP-10's can serve as a "central" file system. This is not an ideal permanent solution, however, as the ITS file system does not provide some of the needed features, such as adequate access control; in addition, the present ARPANET File Transfer Protocol is somewhat cumbersome, making access to files less efficient than is desirable for the central file system application.

Clearly, it would be advantageous in the long run to construct a special-purpose system, which does nothing but provide an efficient central file system for the LCS Network. The construction of such a system is not merely a trivial programming task, however; good, solid design is needed, and the amount of effort required should not be underestimated.

6.2 Extension of the LCS Network

In Section 3, we discussed the tradeoffs between the single segment and spine/spur approaches to construction of the initial LCS Network, and opted for the single segment approach. Once the initial network becomes operational, it will definitely be worth investigating the spine/spur approach, and developing the necessary repeaters. This relatively simple experience with a multi-segment network will prepare us for the inevitable more complex long-distance extensions of our network.

One such extension would be a network segment in the EECS buildings, Buildings 36 and 38. The distance between those buildings and the Laboratory suggests that the EECS segment cannot be a simple spur of the LCS Network, as the transmission delay caused by the long distance would result in a markedly higher collision rate. An appropriate interconnecting scheme will have to be developed. The EECS segment could be, for example, the second sub-network of an eventual campus-wide data network (the LCS Network being the first).

The M.I.T. Cable TV system is now in place, with a number of spare channels. We should investigate the possibility of using CATV channels for communication between distant network segments. We should also investigate how CATV channels can best be used for high-bandwidth data communication, as a straightforward approach to using CATV channels for Ethernet segment interconnection requires four of the cable's channels.

6.3 High-Bandwidth Data Communication Policy for M.I.T.

The preceding discussion leads to another point. M.I.T. currently has no unified campus-wide policy regarding high bandwidth data communication. The development of the LCS Network -- and its extension to the main campus -- will give us considerable expertise in this area. We should be able to encourage and participate in the development of such a policy for M.I.T.

In addition, if we plan carefully now, the protocols and addressing scheme developed for the LCS Network should be applicable to a unified, campus-wide data network composed of sub-networks using a variety of network hardware technologies. The LCS network would thus be the premier sub-network of this campus-wide network.

6.4 Development of a Terminal Concentrator System

The acquisition of a large number of terminals in the 10kb/s speed range, and the adoption of a computing life-style which requires access to a variety of small host computers on the LCS Network, provide impetus for the development of a terminal concentrator system to serve the entire Laboratory.

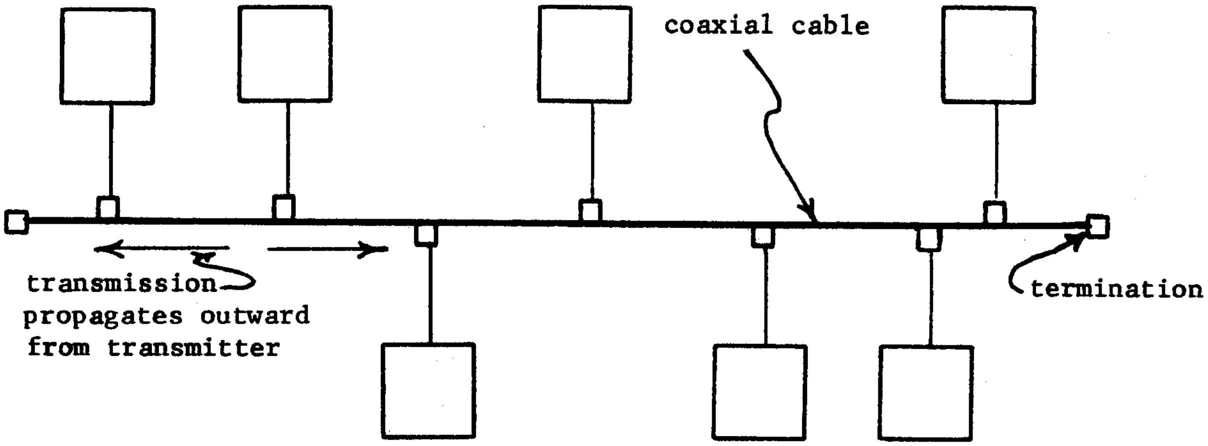
Neither of these conditions exists today, but we are clearly headed in these directions. The PDP-11/40 attached to the KL-10, equipped with additional memory and an additional DH-11 terminal controller, can be programmed to provide network access for the VT-52′s we now have; use of the ARPANET TIP is also possible, although the TIP cannot handle any more high- or medium-speed terminals than it already has. At some point, therefore, we will need to design and construct a system which provides the functions and performance we need in an economical fashion.

6.5 Attachment of the IPC Multics System to the LCS Network

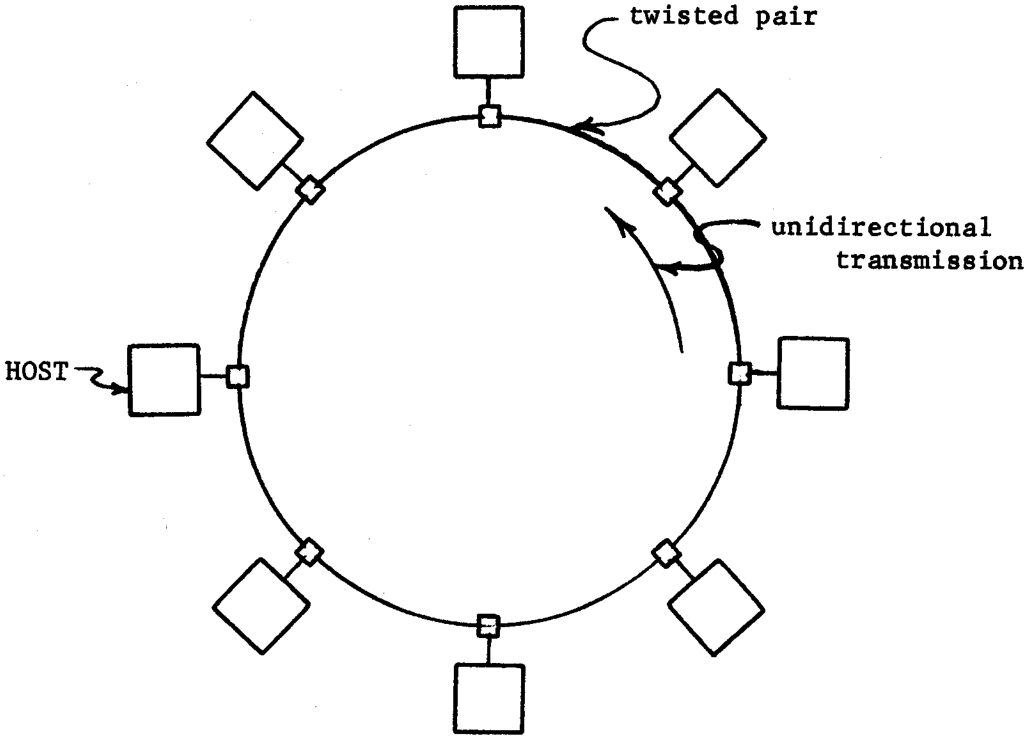
As mentioned in Section 3, the attachment of the IPC Multics System in Building 39 to the LCS Network requires special attention. Most likely it will not be done with the initial implementation of the LCS Network, and will thus come under the category of "future plans".

6.6 An LCS Message Service

One idea which merits consideration even without a local network and terminal concentrator, but is especially attractive with them, is that of a Message Service System for the Laboratory. This is an area in which ARPA has expressed considerable interest. Work in this area has been done on the DMS PDP-10 by LCS/PTD, on Multics by LCS/CSR, and on TENEX (under direct ARPA sponsorship) by BBN. In addition, this summer ARPA is sponsoring the development of a message service system at Rand for the PDP-11 UNIX operating system. A Message Service System for the Laboratory could be part of a similar system proposed for the EECS department, or could be a completely different venture.

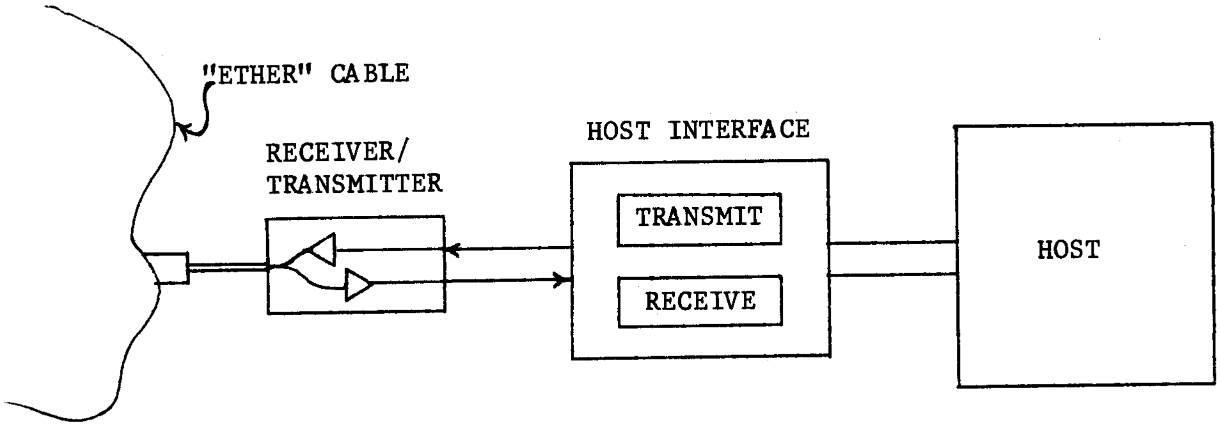


ETHERNET

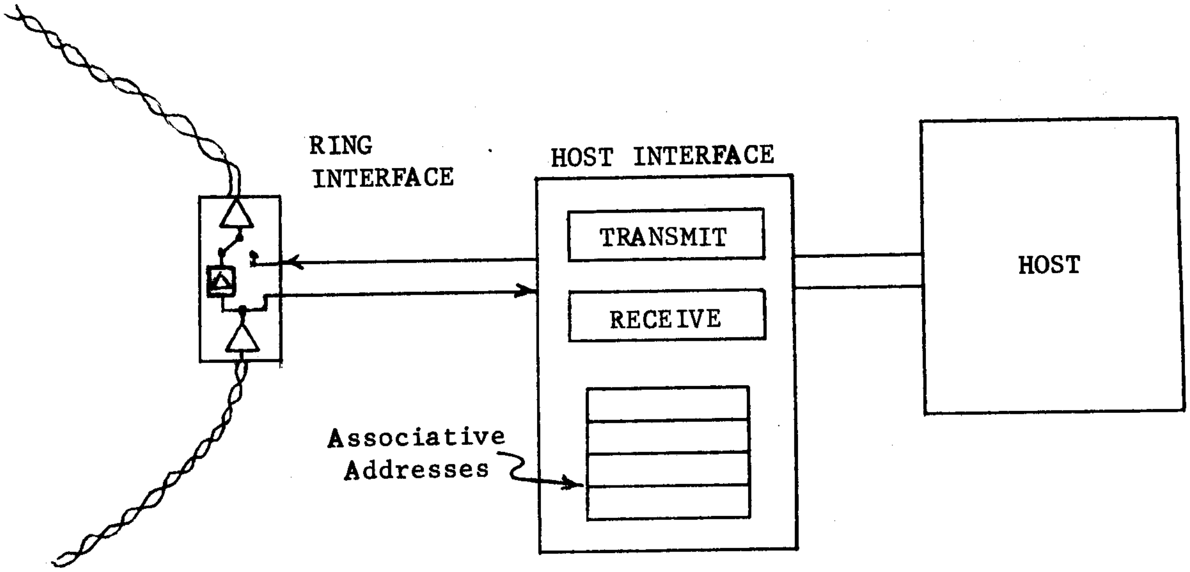


RING NETWORK

Figure 1: Comparison of Ethernet and Ring Network Architecture



ETHERNET



RING NETWORK

Figure 2: Comparison of Ethernet and Ring Network Host Interface

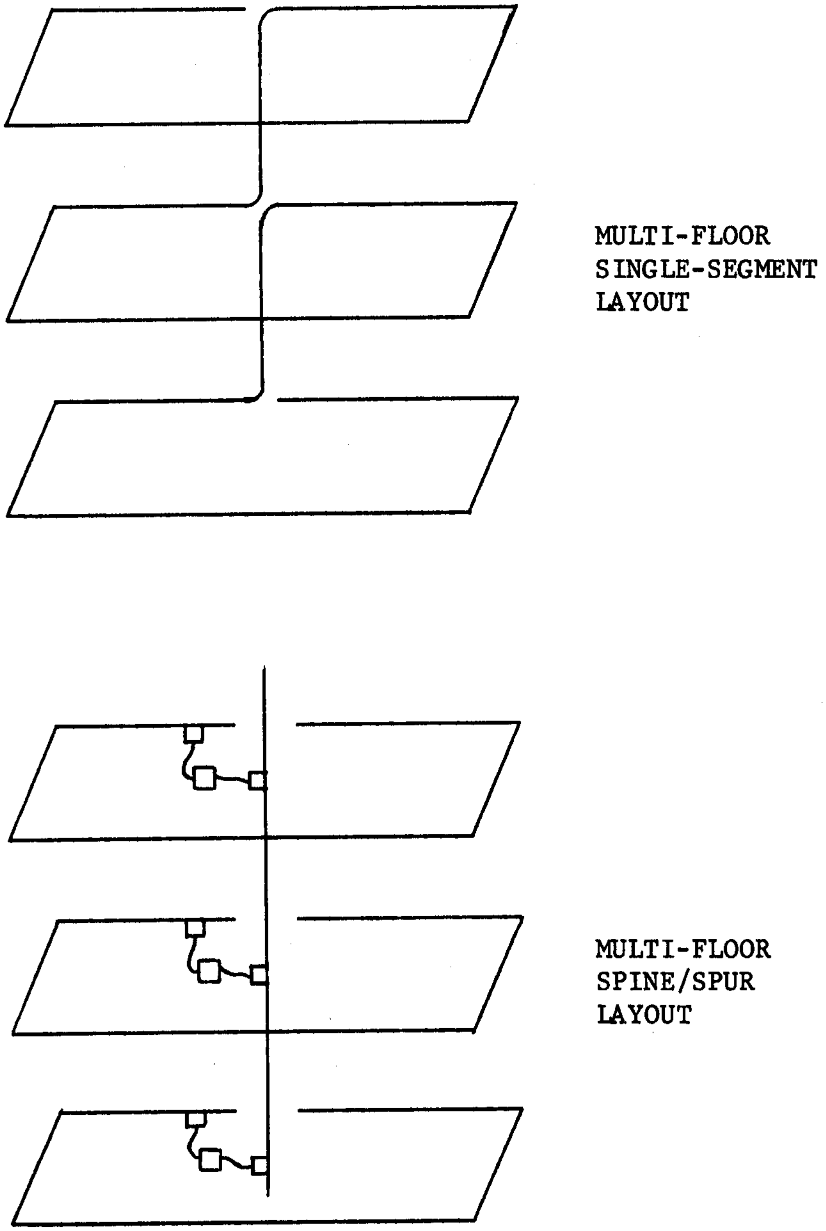


Figure 3: Two arrangements for the Ethernet coaxial cable transmission medium.

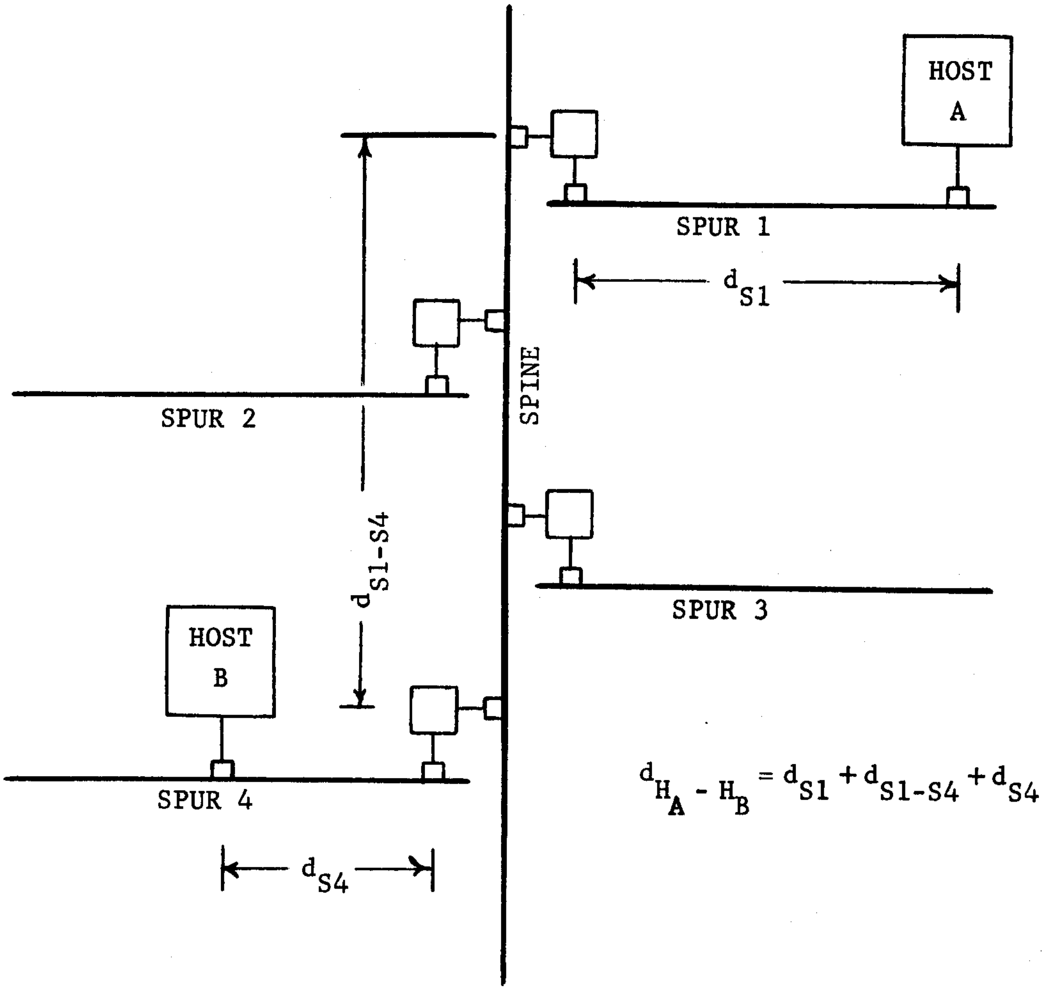


Figure 4: "Distance" between hosts, spine/spur scheme

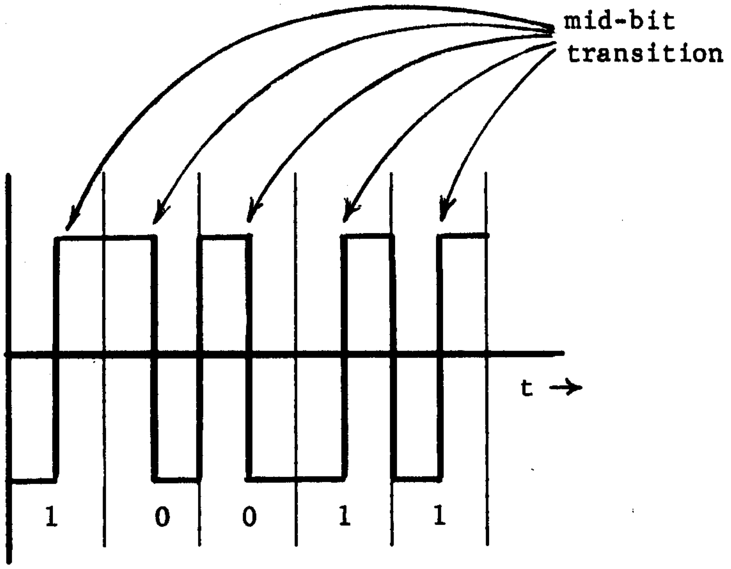


Figure 5: NRZI Encoding

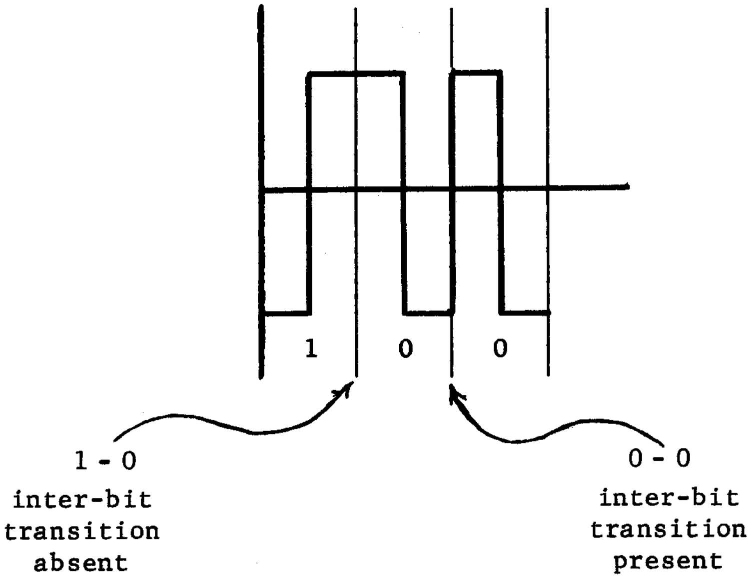


Figure 6: Transitions between bit-times, NRZI Encoding

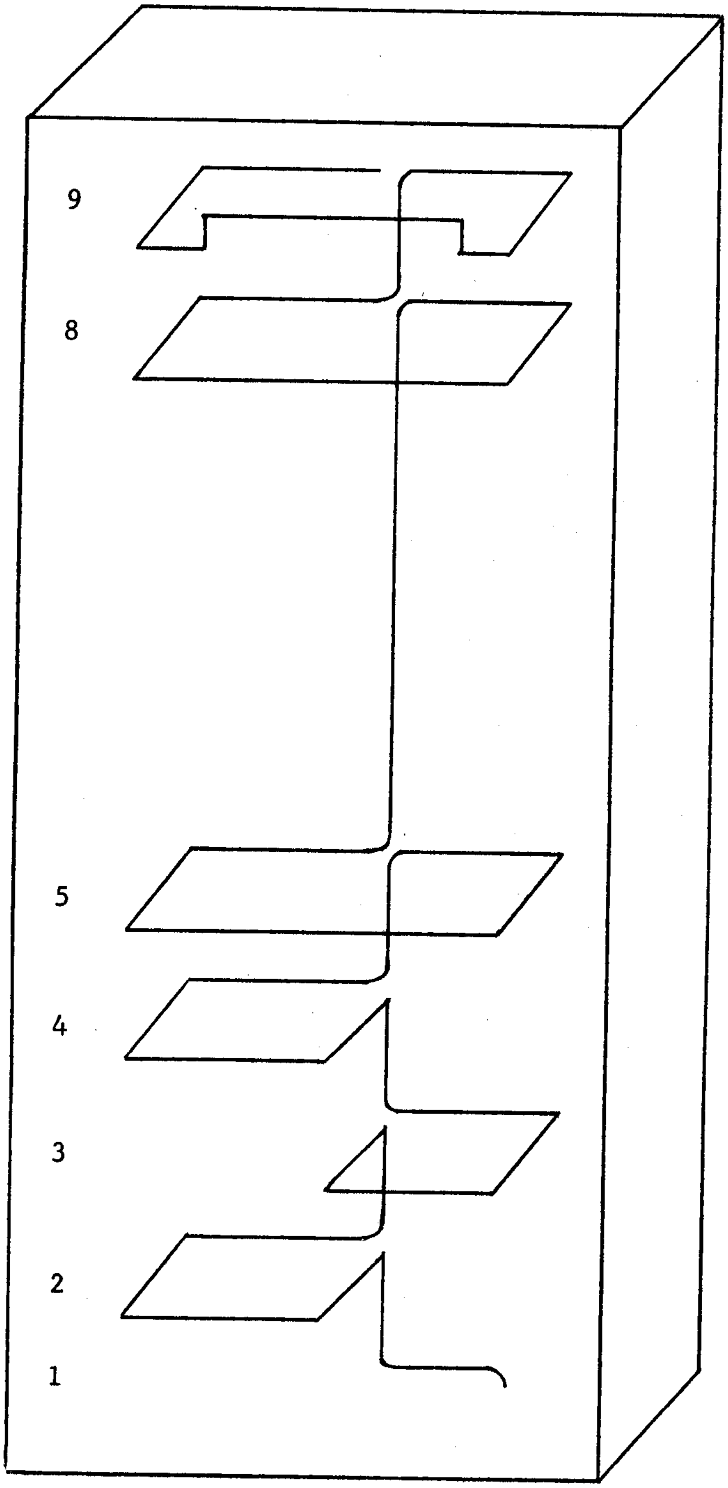


Figure 7: Cable Layout for 545 Technology Square.

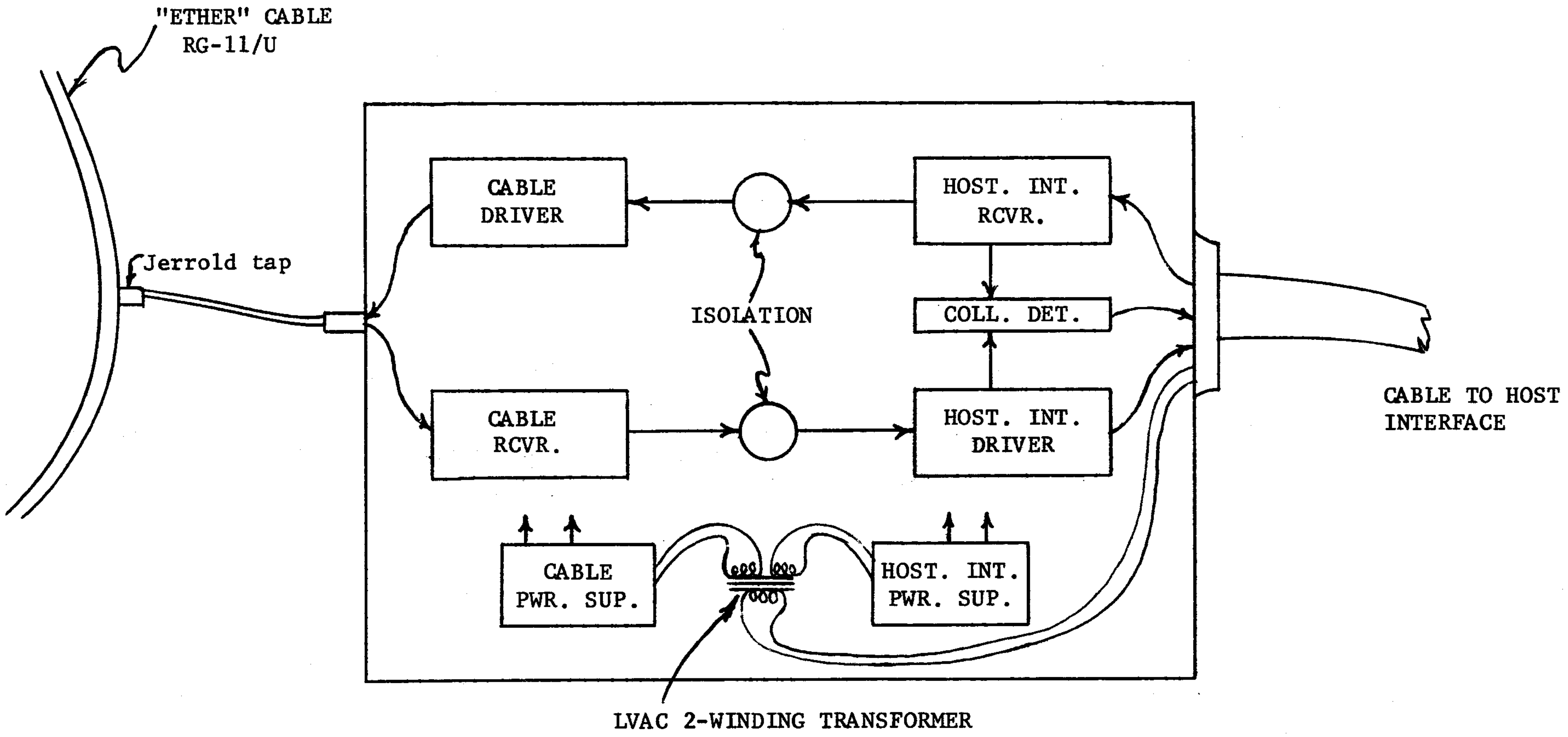


Figure 8: RECEIVER/TRANSMITTER

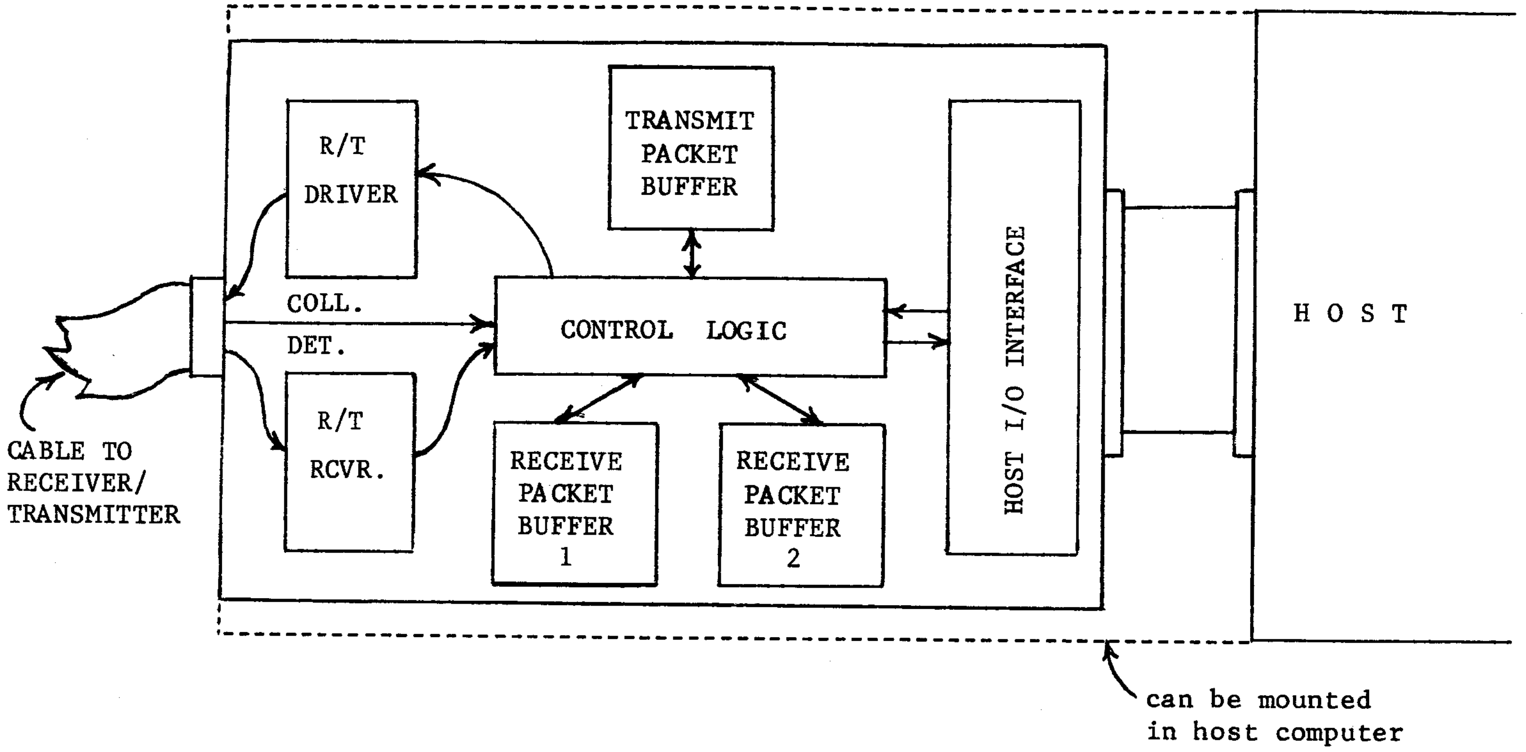


Figure 9: HOST INTERFACE

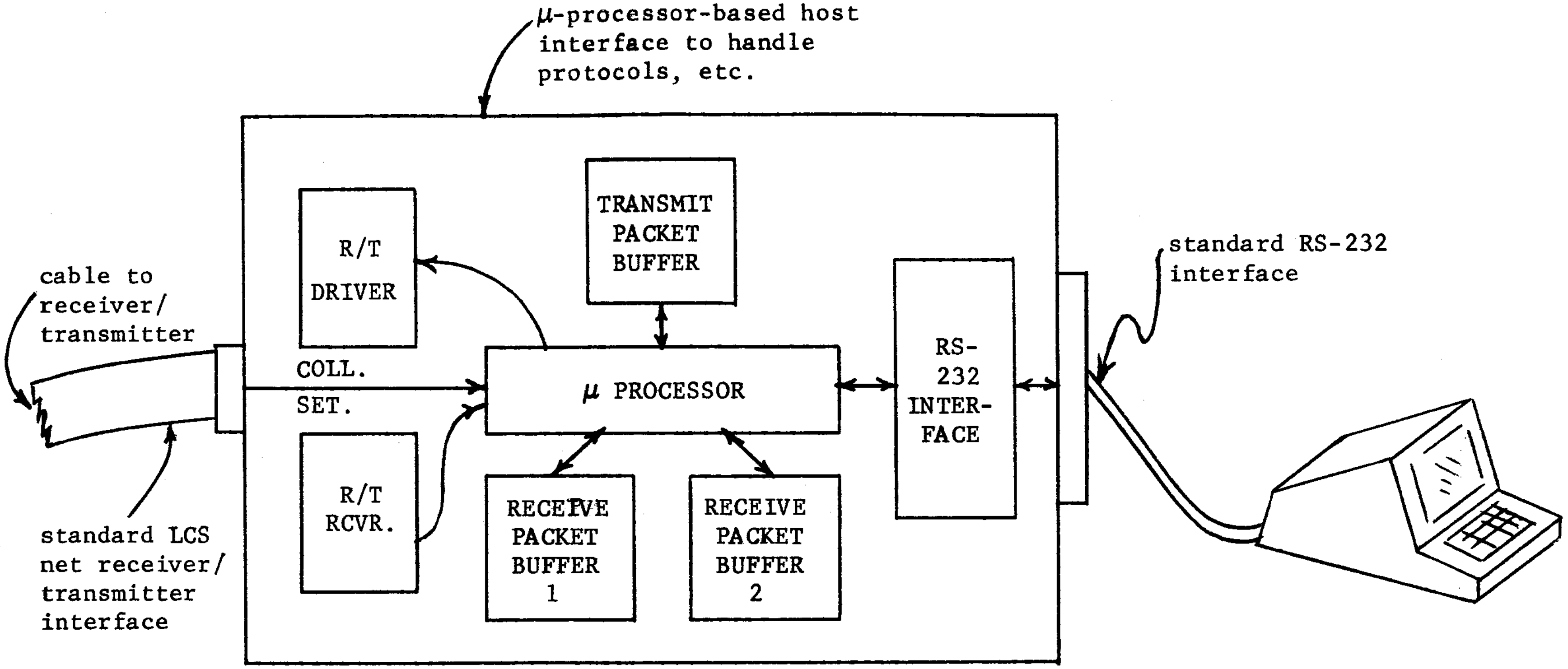


Figure 10: Micro-Processor Controller for direct connection of terminal.