An Introduction to Local Area Networks

DAVID D. CLARK, MEMBER, IEEE, KENNETH T. POGRAN, MEMBER, IEEE, AND DAVID P. REED

Invited Paper

Abstract—Within a restricted area such as a single building, or a small cluster of buildings, high-speed (greater than 1 Mbit/s) data transmission is available at a small fraction of the cost of obtaining comparable long-haul service from a tariffed common carrier. Local area networks use this low-cost, high-speed transmission capability as the basis for a general-purpose data transfer network. There are two basic issues in local area network design. First, how should the hardware realizing the network be organized to provide reliable high-speed communication at minimum cost? With the low cost of the raw transmission capability, care is required to keep the associated hardware costs correspondingly low. Second, what protocols should be used for the operation of the network? While many protocol problems are common to local area networks and long-haul networks such as the ARPANET, new protocols are required to exploit the extended capabilities of local area networks. This paper addresses these two basic issues. It also considers the interconnection of local area networks and long-haul networks and presents a case study which describes in detail the host computer interface hardware required for a typical local area network.

I. INTRODUCTION

AS ITS NAME IMPLIES, a local area network is a data communication network, typically a packet communication network, limited in geographic scope.[1] A local area network generally provides high-bandwidth communication over inexpensive transmission media. This paper discusses what local area networks are, their structures, the sorts of protocols that are used with them, and their applications. It also discusses the relationship of local area networks to long-haul networks and computer system I/O buses, as well as the impact of these networks on the field of computer communications today.

A. Components of a Local Area Network

Like any other data communication network, a local area network is composed of three basic hardware elements: a *transmission medium*, often twisted pair, coaxial cable, or fiber optics; a *mechanism for control* of transmission over the medium; and an *interface* to the network for the host computers or other devices—the *nodes* of the network—that are connected to the network. In addition, local area networks share with long-haul packet communication networks a fourth basic element: a set of software *protocols*, implemented in the host computers or other devices connected to the networks, which control the transmission of information from one host or device to another via the hardware elements of the network. These software protocols function at various levels, from low-level *packet transport* protocols to high-level application protocols, and are an integral part of both local area networks and their close relatives, long-haul packet communication networks. This combined hardware–software approach to communication serves to distinguish networks, as discussed in this paper, from other arrangements of data communication hardware.

B. Relationship of Local Area Networks to Long-Haul Networks

1) The Evolution of Networking: Local area networks share a kinship with both long-haul packet communication networks and with I/O bus structures of digital computer systems; their structure and protocols are rooted in packet communication, while their hardware technology derives from both networks and computer busses. Local area networks arose out of the continuing evolution of packet communication networks and computer hardware technology. Packet communication techniques have become well known and widely understood in the nine years since development of the ARPANET was begun. Meanwhile, computer hardware has come down in price dramatically, giving rise to environments where, within a single building or a small cluster of buildings, there may be one or more large mainframe computers along with a number of minicomputers, microprocessor systems, and other intelligent devices containing microprocessors. Local area networks evolved to meet the growing demand for high data rate, low-cost communication among these machines.

2) Geographic Scope; Economic and Technical Considerations: Fig. 1 illustrates the geographic scope spanned by long-haul packet networks, local area networks, and computer system busses. Long-haul packet networks typically span distances ranging from meters[2] to tens of thousands of kilometers (for intercontinental packet networks); bus structures used in computer systems range from those of microprocessor systems, which can be as short as 1–10 cm, to those used in large-scale multiprocessor systems, which can be as much as 100 m in length. As Fig. 1 indicates, local area networks span distances from several meters through several kilometers in length.

The first local area networks evolved in environments in which the distances to be spanned by the network were within the range of inexpensive high-speed digital communication technologies. Today, the relationship has been turned around, so that the distance range of local area networks is governed by the distance over which these inexpensive techniques can be used. The result is high-data-rate networks in which the cost of transmission and the cost of control of transmission is very low compared to the costs associated with traditional

Manuscript received April 12, 1978; revised June 30, 1978.

The authors are with Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139.

[1] Conventional Packet communications networks, not limited in geographic scope, are referred to in this paper as "long-haul" networks.

[2] Long-haul packet communication network technology has been used in environments that could be served more effectively, and at lower cost, by local area network technology. This local area use of long-haul network technology, indicated by the shaded area of Fig. 1, is due to the fact that long-haul packet communication technology has been available commercially for several years, while local area network technology has not.

0018-9219/78/1100-1497$00.75

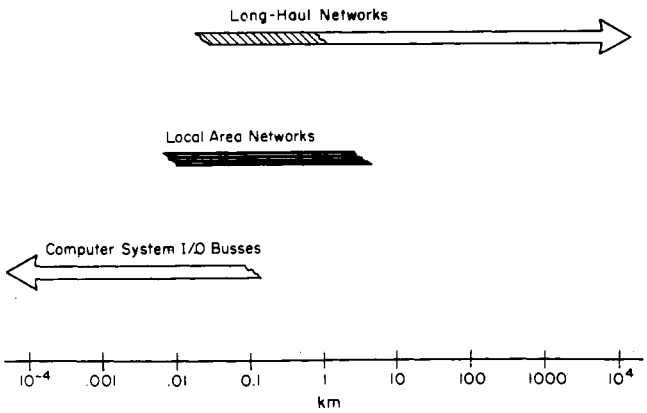

Fig. 1. Geographic range of computer communication networks and I/O buses. The shaded area of the long-haul network bar indicates the distance range for which that technology has been used in the past, but which could be better served, in both cost and performance, by emerging local area network technology.

data communication networks, providing some unique opportunities conventional long-haul networks do not afford.

For long-haul networks, the cost of communication is high. Wide-band common carrier circuits, satellite circuits, and private microwave links are expensive. Long-haul packet communication networks commonly employ moderately expensive (i.e., 50 000-dollar) minicomputers as packet switches to manage and route traffic flow to make the most effective use of the network communication links, delivery of packets to their proper destinations. The geographic characteristics of local area networks yield economic and technological considerations that are quite different. Inexpensive, privately owned transmission media can be used. For example, simple twisted pair can support point-to-point communication in the 1–10-Mbit/s range over distances on the order of a kilometer between repeaters. Coaxial cable, such as low-loss CATV cable, can support either point-to-point or broadcast communication at similar data rates over comparable distances. Typically, base-band signaling is used to place digital signals directly on the medium, rather than by modulation of a carrier. Because the hardware needed to drive and control these transmission media is inexpensive, there is little motivation or need to employ computing power to make the most effective use of the available bandwidth. On the contrary, it is quite reasonable to provide additional bandwidth, or to use a greater fraction of the existing bandwidth, if by doing so some other network cost—either hardware or software—can be reduced.

3) New Opportunities: The economic and technical characteristics of local area networks engender new applications of networking techniques and provide some unique opportunities to simplify traditional networking problems. Many of the constraints that long-haul networks impose on models of communication over a computer network are not present in local area networks. One example is broadcast communication, such as that used in the Distributed Computer System developed at the University of California at Irvine [1].

The high-bandwidth and low-delay attributes of local area networks make possible distributed multiprocessor systems utilizing the sort of information sharing between processors commonly associated with multiprocessor systems sharing primary memory. Local area networks can also be used to provide a central file system for a group of small computers which do not have their own secondary storage; it is even possible to use such a central file system, accessed over the local area network, for swapping or paging—an application made especially attractive by the fact that the cost of local area network interface hardware for a typical minicomputer can be less than the cost of a "floppy disk" drive and its associated controller.

The high bandwidth of local area networks can be exploited to simplify the control structure of communication protocols by removing any motivation to minimize the length of control or overhead information in a packet. Fields of packet headers in local area networks can be arranged to simplify the processing involved in creating or interpreting the packet header, using as many bits as are necessary. There is little need to use "shorthand" techniques often found in the protocols of long-haul networks which necessitate additional table lookups by the receiver of a message. Simplicity also extends to other aspects of local area network protocols, such as schemes for allocation of network bandwidth, flow control, and error detection and correction.

It should be emphasized that local area networks are *not* an off-the-shelf, plug-in panacea for all local area computer communication needs. For the distance range over which they operate, the technology of local area networks holds the promise of doing for computer communications what the hardware innovations of the last five years have done for computing power: they can bring down the cost of high-bandwidth communication and make possible new applications. But they cannot by themselves solve the "software problem," for with low-cost hardware, the costs of software development will dominate the cost of any system development using local area network technology.

C. Interconnection with Other Networks

While some local area networks now in use or under construction are "stand-alone" networks, not connected to other networks, the trend is toward interconnection of local area networks with long-haul networks.

Interconnection can be motivated either by economics or simply by the needs of users of the hosts of a local area network. For example, a local area network can provide an economical means of connecting a number of hosts within a small area to one or more long-haul packet networks. The savings thus obtained is most obvious in the situation where a number of local host computers are to be connected to more than one long-haul network; each computer, rather than being directly connected to every network (an "*M*-by-*N* problem"; see Fig. 2(a)), can be connected only to the local area network, and one host (called the *gateway*) can be connected between the local area network and each of the long-haul networks (Fig. 2(b)). This cost savings can be worthwhile even in a situation in which local hosts are to be connected only to a single long-haul network, for two reasons; first, host interface hardware for local area networks can be less expensive than that for long-haul networks; and, second, only a single port to the long-haul network, rather than one port for each local host, is required.

Examples of motivation for interconnection based on users' needs are as follows.

a) A computer-based mail system, in which messages to and from users of the several hosts of a local area network can be exchanged via a long-haul network.

b) Access to specialized computing resources, occasionally required by the hosts of a local area network but too expensive

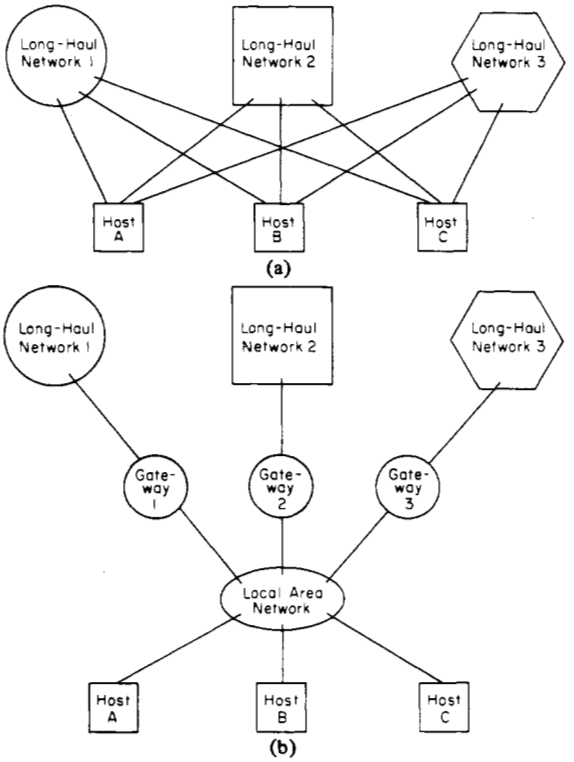

Fig. 2. The "*M* × *N* Problem" and a local area network as one solution to it. In (a), each of three hosts at a particular site is to be connected to three long-haul networks; each host must implement the communication protocols for, and be equipped with a hardware interface to, all three networks; there may be nine different interfaces and nine protocol implementations in all. In (b) each host needs only one hardware interface and one protocol implementation; the gateway machines each handle communication between one long-haul network and the local area network.

to maintain locally; they can be accessed by users of local area network hosts on a fee-for-service basis via a long-haul network.

c) Communication between local area networks maintained by a company at each of its major locations.

The interconnection of a local area network to a long-haul network presents problems, as well as benefits. At some point, the protocols used within the local area network must be made compatible with those of the long-haul network(s). Compatibility can be achieved either by adopting the protocols of a long-haul network for the local area network, or by performing appropriate protocol transformation on messages as they pass through the gateway. Care must be taken with the former approach to ensure that needed capabilities of the local area network are not sacrificed for the sake of ease of the network interconnection. These issues are discussed in greater detail in Section VI of this paper.

II. Why Isn't a Local Area Network Merely a "Big Bus"?

A. Distinctions Between Local Area Networks and Computer Busses

One distinguishing feature of local area networks is the geographic restrictions that permit them to utilize low-cost but very high-bandwidth transmission media. That characterization can also apply to the bus structure of a computer. How, then, is a local area network different from a "big bus"? The significant differences are not, as one might initially expect, topological, technological, or geographic. Rather, the distinction is a philosophical one. A computer bus is usually conceived as connecting together the components of a single computer system; it is difficult to imagine a computer continuing to perform any sort of useful action in the absence of its bus. In contrast, a network is understood to connect together a number of autonomous nodes, each capable of operating by itself in the absence of the network.

1) Defensiveness: This philosophical distinction manifests itself in the management and control strategies of a network, which are far more defensive than the equivalent strategies of a bus are required to be. For example, the reliability issues surrounding a bus are somewhat different from those surrounding a local area network. While both should be reliable, it is usually not critical that a bus continue to work if one if one of the devices attached to it has failed; it is usually acceptable to have the system halt momentarily until the failing device can be manually disconnected from the bus. In contrast, it is presumed that a network will continue to operate despite arbitrary failures of one or more nodes.

The handling of traffic overloads is another example of the defensive nature of a network. One must anticipate, when designing a network, that independently initiated transfers will occasionally demand more bandwidth than the network has available, at which time the network itself must mediate gracefully between these conflicting demands. There is usually no such concern for a computer bus. The problem of insufficient capacity on a computer bus to transfer all the information required is generally solved by reconfiguration of the hardware of the system, or through use of a different programming strategy.

2) Generality: The intended generality of a computer bus and a network is a second philosophical distinction between the two. For example, the protocols that control communication on a local area network are often designed with the explicit intention that messages can be exchanged between a local network and a long-haul network, an idea that is usually missing from the addressing and control structure of a computer bus. For another example, networks usually transmit variable size messages, while buses often transfer single, fixed-size words.

Another sort of generality that serves to distinguish a bus from a network is the nature of the interface that each provides to the nodes attached to it. A computer bus often has a specialized interface, oriented towards the addressing and control architecture of a particular computer. A network, on the other hand, usually attempts to provide an interface equally suitable for a wide variety of computers and other devices. In this respect, the interface is often less efficient than a bus interface, but is easier to implement for arbitrary devices. Fraser describes one plausible specification for a general network interface [2].

3) Minor Distinctions: Current realizations of networks and busses suggest other differences which are much less relevant. Busses are often even more "local" than our definition of a local area network. A computer bus is often highly parallel, with separate control, data, and address lines; networks tend to carry this information serially over a single set of lines. On the other hand, the idea of a computer bus which is completely serial is very attractive in the design of microprocessor systems, both to reduce component pinout and to eliminate problems of skew on parallel lines.

B. The IEEE Instrumentation Bus as a Border-Line Case

The IEEE Instrumentation Bus [3] is a good example of a communication medium that lies on the boundary between a network and a bus. In certain respects, this bus resembles a

local area network, since it has a general interface capable of interconnecting a variety of instruments and computers, each of which operates with a certain degree of autonomy. Philosophically, however, the Instrumentation Bus is indeed very much a bus, since its specification is clearly based on the assumption that all of the devices connected to a particular bus are intended to operate harmoniously as one system to perform a single experiment under the control of one particular experimenter. The experimenter, not the bus itself, is expected to ensure that the capacity of the bus is not exceeded, and to detect and remove failing nodes which disable the bus. Also, the addressing structure of the Instrumentation Bus, as defined, is not extendable, so that the idea of connecting several of these busses together to make a larger network is difficult to realize. This limitation on addressing may prove a hindrance to experimenters who wish to use the Instrumentation Bus as a component of a larger interconnected array of computers and experimental equipment.

III. Topologies and Control Structures for Local Area Networks

The introduction of this paper identified three hardware components of a local area network: the transmission medium, a mechanism for control, and an interface to the network. This section will discuss the first two of these, which together provide the lowest-level functionality of the network, the ability to move messages from place to place in a regulated manner.

A. Network Topology

Network topology is the pattern of interconnection used among the various nodes of the network. The most general topology is an unconstrained graph structure, with nodes connected together in an arbitrary pattern, as illustrated in Fig. 3. This general structure is the one normally associated with a packet-switched network; its advantage is that the arrangement of the communication links can be based on the network traffic. This generality is a tool for optimizing the use of costly transmission media, an idea which is not germane to local area networks. Further, this generality introduces the unavoidable cost of making a routing decision at each node a message traverses. A message arriving at a node cannot be blindly transmitted out of all the other links connected to that node, for that would result in a message that multiplied at every node and propagated forever in the network. Thus each node must decide, as it receives a message, on which link it is to be forwarded, which implies a substantial computation at every node. Since this general topology is of no significant advantage in a local area network, and does imply a degree of complexity at every node, local area network designers have identified a variety of constrained topologies with attributes particularly suited to local area networks. We shall consider three such topologies: the *star*, the *ring*, and the *bus*.

1) The Star Network: A star network, illustrated in Fig. 4(a), eliminates the need for each network node to make routing decisions by localizing all message routing in one central node. This leads to a particularly simple structure for each of the other network nodes. This topology is an obvious choice if the normal pattern of communication in the network conforms to its physical topology, with a number of secondary nodes communicating with one primary node. For example, the star is an obvious topology to support a number of terminals communicating with a time-sharing system, in which case the central node might be the time-sharing machine itself.

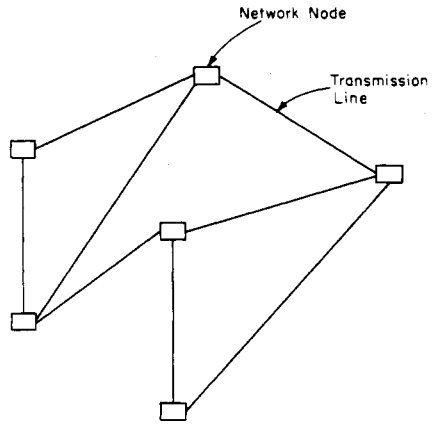

Fig. 3. Unconstrained topology. Each node receiving a message must make a routing decision to forward the packet to its final destination.

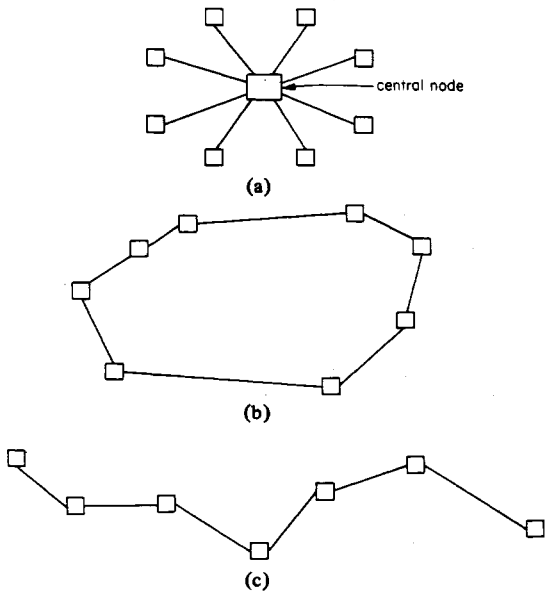

Fig. 4. Examples of constrained topologies. (a) The star. (b) The ring. (c) The bus.

If, however, the normal pattern of communication is not between one primary node and several secondary nodes, but is instead more general communication among all of the nodes, then reliability appears as a possible disadvantage of the star net. Clearly, the operation of the network depends on the correct operation of the central node, which performs all of the routing functions, and must have capacity sufficient to cope with all simultaneous conversations. For these reasons, the central node may be a fairly large computer. The cost and difficulty of making the central node sufficiently reliable may more than offset any benefit derived from the simplicity of the other nodes.

2) Ring and Bus Networks: The ring and bus topologies attempt to eliminate the central node on the network, without sacrificing the simplicity of the other nodes. While the elimination of the central node does imply a certain complexity at the other nodes of the net, a decentralized network can be constructed with a surprisingly simple structure of the nodes. In the ring topology, illustrated in Fig. 4(b), a message is passed from node to node along unidirectional links. There are no

routing decisions to be made in this topology; the sending node simply transmits its message to the next node in the ring, and the message passes around the ring, one node at a time, until it reaches the node for which it is intended. The only routing requirement placed on each node is that it be able to recognize, from the address in the message, those messages intended for it. Similarly, in the bus structure pictured in Fig. 4(c), there are no routing decisions required by any of the nodes. A message flows away from the originating node in both directions to the ends of the bus. The destination node reads the message as it passes by. Again, a node must be able to recognize messages intended for it.

B. Network Control Structures

Both the ring network and the bus network introduce a problem, not immediately apparent in the star net, of determining which node may transmit at any given time. The mechanism for control, the second component of the network as listed in the introduction, performs this determination. This task is not difficult in the star network; either the central node has sufficient capacity to handle a message for every node simultaneously, or it may poll each of the other nodes in turn to determine if that node wishes to transmit. Both the ring and the bus topology, lacking any central node, must use some distributed mechanism to determine which node may use the transmission medium at any given moment.

1) Daisy Chain, Control Token, and Message Slots: There are a variety of control strategies suitable for the ring topology, based on the general idea that permission to use the net is passed sequentially around the ring from node to node. In what is often called a *daisy chain network*, dedicated wires are used to pass the control information from one node to the next. Alternatively, the control information may be a special bit pattern transmitted over the regular data channel of the ring. For example, the network for the Distributed Computing System uses an 8-bit *control token* that is passed sequentially around the ring [4]. Any node, upon receiving the control token, may remove the token from the ring, send a message, and then pass on the control token. A third strategy for ring control is to continually transmit around the network a series of *message slots*, sequences of bits sufficient to hold a full message. A slot may be empty or full, and any node, on noticing an empty slot passing by, may mark the slot as full and place a message in it. This strategy was described by Pierce [5], and has been used in the Cambridge Net [6] and in the network described by Zafiropulo and Rothauser [7]. This technique is not completely decentralized, since one node must initially generate the slot pattern.

2) Register Insertion: Another control strategy particularly suited for the ring topology is called *register insertion*. In this technique, a message to be transmitted is first loaded into a shift register. The network loop is then broken and this shift register inserted in the net, either when the net is idle or at the point between two adjacent messages. The message to be sent is then shifted out onto the net while any message arriving during this period is shifted into the register behind the message being sent. Since the shift register has then become an active component of the network, no further messages can be sent by this node until the register is switched back out of the ring. This can only be done at a moment when there is no useful message in the register. One obvious way to remove the register from the network is to allow the message transmitted by the node to pass all the way around the network and back into its shift register. At the moment when the message is again contained in the register, both message and register can be simultaneously removed from the network. If this technique is not used, or if the message is damaged and does not return intact back to its original sender, it is necessary to wait for the network to become idle before removing the buffer from the network.

The performance characteristics of the register insertion technique are rather different from those of the previously discussed techniques, since the total delay encountered in the network is variable, and depends on the number of messages currently being sent around the net. Further, a message to be sent is inserted in front of, rather than behind, a message arriving at a node. One analysis [8] indicates that the register insertion network may, under certain circumstances, have better delay and channel utilization characteristics than either the control token or message slot strategy. The register insertion strategy is complex, however, especially in the technique for removing the register from the network when the transmitted message does not return.

The register insertion technique was initially described by Hanfer *et al.* [9], in a paper that discusses a variety of operating modes that can be achieved using this technique. In the initial description of the Cambridge Network [10], register insertion was proposed as a control strategy, because it ensured a fair share allocation of network bandwidth. Since a node that has transmitted one message cannot transmit a second until the first message has passed completely around the ring, and the node has removed its register from the network, every other node has a chance to send one message before a given node may transmit a second. However, circulating message slots were finally chosen as the control mechanism of the Cambridge Network. Slots, if suitably employed, ensure fair share allocation, and the slot mechanism reduces the number of components whose failure can disrupt network operation, since there is no buffer in series with the net. Thus the slot scheme seemed more reliable. The register insertion technique is currently being used as the control strategy in the Distributed Loop Computing Network (DLCN) described by Liu and Reames [11].

3) Contention Control: A bus topology also requires a decentralized control strategy. One very simple control strategy that has been used for bus networks is *contention*. In a contention net, any node wishing to transmit simply does so. Since there is no control or priority, nothing prevents two nodes from attempting to transmit simultaneously, in which case a *collision* occurs, and both messages are garbled and presumably lost. The contention control strategy depends on the ability of a node to detect a collision, at which point it waits a random amount of time (so that the same collision will not recur), and then retransmits it message. Assuming that network traffic on the average consumes only a small percentage of the available bandwidth, the number of collisions and retransmissions will be reasonably small. The essential local area network characteristic of inexpensive bandwidth makes this strategy well suited to a local network. The bandwidth wasted in order to keep the channel utilization low is a small price to pay in return for the very simple mechanisms that must be implemented at each node: a timer capable of generating a random distribution, and some means of detecting collisions.

A variety of strategies have been used to detect collisions. The first use of a contention packet network was the ALOHA

Net [12], not a local area network, but a network using radio transceivers to connect together computer terminals to a computer center on the island of Oahu. The techniques for detecting a collision in this network was very simple: the transmitting node started a timer when it transmitted the message, and if an acknowledgment for the message had not been received when the timer expired, the message was retransmitted. The disadvantage of this collision detection scheme is that it leads to a very low theoretical upper limit on the percentage of channel capacity which can be utilized without causing the network to overload with retransmission traffic [13].

A strategy which greatly increases the maximum effective transmission capacity of the network is to listen before transmitting, which changes the whole pattern of network operation. A collision will now occur only if two nodes attempt to transmit at nearly the same instant, because if one node has started sufficiently in advance of the other so that its signal has propagated over the transmission medium to that other node. the other node will hear that signal and will refrain from transmitting.

A further embellishment of this idea is to listen, *while* transmitting as well. This permits colliding nodes to detect the collision much more promptly than if they detected the collision only by noticing the absence of an acknowledgment. This strategy not only reduces the delay caused by a collision, it makes the transmission medium available sooner, as well, since colliding nodes can cease transmitting as soon as they detect a collision. The strategy of listening while transmitting is not suitable for terrestrial radio, because the transmitter overloads the receiver, but is quite reasonable when transmitting over wire or cable. This strategy is used in the ETHERNET, developed at the Xerox Palo Alto Research Center [14].

C. Combinations of Topology and Control Structure

We have identified three network topologies: the star, the ring, and the bus topology, and three control strategies: ring control, contention control, and centralized control. It is important to note that any control strategy can be used with any topology. Several interesting combinations are described in the following paragraphs.

A variation on the use of a control token, suitable for a bus topology, is described by Jensen [15]. Every node is provided with a list of the order in which each may send, since the bus itself imposes no natural order. A special signal on the bus causes every node to move to the next entry on the list. The node named by that entry may send a message if it has one, after which it must in turn send the special signal. A mechanism is required to recover synchronism should one or more nodes miss the special signal, so that the lists get out of step. This scheme has the interesting advantage that some interfaces can be entered in the list more often than others, so that they receive a larger proportion of the bandwidth.

A bus topology using a daisy chain ring control strategy is a common means of implementing a computer bus. An example is the UNIBUS architecture of the Digital Equipment Corporation PDP-11 [16].

A ring topology controlled by a contention strategy produces a network with some very promising attributes, being explored in an experimental contention ring currently under development at the Laboratory for Computer Science of the Massachusetts Institute of Technology. In a bus topology contention network, collisions most commonly occur immediately following the end of a message, for at that moment all of the nodes that have refrained from transmitting during the previous message simultaneously attempt to seize the bus. In a ring topology contention net, the unidirectional flow of messages from node to node provides a natural ordering of all nodes wishing to transmit at the end of a previous message. Thus the contention ring will experience a much lower collision rate for a given degree of channel utilization. Further, in a contention ring it is very easy to implement the concept of listening while transmitting in order to detect collisions promptly. One way of operating a ring topology is for the transmitting node to place the message on the ring and also to remove the message from the ring. The message flows all the way around the ring; it is not removed by the recipient. A collision is detected when a transmitting node discovers that the message it is removing from the ring is different from the one it is sending.

It is also possible to devise a ring network with centralized control. Such a network was proposed by Farmer and Newhall [17]; the SPIDER network, described by Fraser [18] also has this structure. Line control protocols such as SDLC [19] use centralized control to regulate both a ring topology and multidrop line, a topology that somewhat resembles a bus.

D. Reliability Characteristics of Ring and Bus

The chief motivation for the ring and the bus topology was to avoid a potential reliability problem with the central node of the star. Reliability considerations arise both from the topology and the control strategy of the network. The contention control strategy has an inherent reliability advantage over the ring control strategies described above, for in any ring control strategy there is some entity, be it a control token or an explicit signal on a wire, which is passed from node to node to indicate which node currently has the right to transmit. The control strategy must always take into account the possibility that a transient error will destroy this entity. For example, a control token may be destroyed by a noise burst on the transmission medium. Therefore, any ring control strategy must be prepared to restart itself after a transient error by regenerating the permission to send and bestowing it uniquely upon one of the nodes. Unfortunately, with a completely decentralized control strategy, it is very difficult to determine with certainty that the control entity has been lost, and it is even more difficult to decide which node should take it upon itself to recreate the control entity. Thus one must either use some sort of contention scheme to deal with error recovery, as is done in the Distributed Computing System network, or have a single node provide a centralized mechanism for restart, as is done in the Cambridge Network.

In contrast, almost any transient failure in a contention control network has exactly the same effect as a collision, and is thus dealt with automatically. If a message is garbled it must be retransmitted, but no long-term failure of the network results. This is one of the very appealing attributes of the contention control strategy. On the other hand, contention control does require that the recipient detect a garbled message, and be able to request a retransmission if the original transmitter has for some reason failed to discover that the message was garbled. Thus higher level protocols must provide mechanisms which ensure reliable recovery. In fact, attention to reliability at higher levels is required regardless of the control strategy of the network; this is not a particular disadvantage of the contention strategy.

The topology of a network, as well as its control strategy, influences its reliability. The ring topology requires that each node be able to selectively remove a message from the ring or pass it on to the next node. This requires an active repeater at each node, and the network can be no more reliable than these active repeaters. The active repeater must also play a role in implementing the control strategies used on a ring, for almost without exception the control strategies depend on the ability of a node to modify a message is it passes by. For example, in the control token strategy, the token is removed from the network by modifying its last bit as it passes by, so that it is no longer recognized as a control token. Thus there can be a significant amount of logic at each node whose failure disables the network. The repeater portion of a ring network interface should, therefore, be made very reliable, with a reliable power supply, to reduce its probability of failure. In the Cambridge Network, repeaters are powered from the ring, so that the network is immune to loss of local power at any node. Another technique that enhances the reliability of the network is to provide a relay at each repeater that can mechanically remove it from the network in the event of a failure, including a local power failure.

The bus topology, on the other hand, does not require the message to be regenerated at each node. The bus is a passive medium, with each node listening. A node can fail without disrupting the bus so long as it fails in a manner that presents a high impedance to the bus. A bus network node that is designed to operate in this way is described later in this section.

This analysis suggests that a network constructed with a bus topology could be more reliable than one constructed with a ring topology. While both have active elements whose failure can disrupt the network, the bus components must fail in a particular way, so as to drive and disrupt the bus, whereas almost any failure of the ring repeater will disable the ring. However, not all the reliability issues favor the bus topology. If, for example, the transmission medium is subjected to a catastrophic disruption, such as a lightning strike or an errant cross connection to the power lines, one can expect all electronic components connected to the medium to be destroyed. In the case of a ring, this will be one set of line drivers and one set of receivers. In the case of the bus, every node may be damaged. While in both cases the net will be disabled, the bus may require much longer to repair. Practical experience, although somewhat limited and not well reported in the literature, suggests that with care *both* topologies can be made sufficiently reliable that the possibility of failure can essentially be ignored in a practical system. The various factors previously discussed may tend to favor a passive bus or an active repeater design in some particular case. However, we believe that the most significant factor in hardware reliability is the quality and care in the engineering design. In this context it is worth noting that certain design problems present in a bus do not arise in a ring (Several such problems are discussed in the case study of a bus interface in the next section.). Thus it may be somewhat easier to design a ring than a bus.

E. Patterns of Data Flow; Addressing

So far we have identified three criteria for choosing one of the three constrained network topologies: simplicity, reliability, and the need for a network control strategy. There is one additional criterion of importance—the patterns of data flow that each topology will support. Message flow in the arbitrary graph structure discussed at the beginning of this section is inherently point-to-point. That is, a message inserted into the network flows over some subset of the available links and is routed to some eventual destination. There is an alternative pattern, in which any message placed in the network is automatically broadcast to all nodes, each node examining the address on the message and copying it as appropriate. As we will discuss later, many applications of a local area network can benefit greatly from the ability to send messages in a broadcast mode. Thus there is an advantage to a topology that naturally supports broadcast at the low level.

The bus topology is inherently a broadcast medium, as there is no way to selectively route a message along the bus. The ring topology can either be used in a point-to-point mode or a broadcast mode. In the point-to-point mode, the message is transmitted around the ring until it reaches the recipient, who then removes it. In broadcast mode, the message passes completely around the ring and is removed by the original sender. The star configuration, also, can operate as either a point-to-point or a broadcast network.

The possibility of operating a network in the broadcast mode raises the question of what addressing mechanism is used on the network to identify the recipient of a message. The simplest addressing strategy, a pre-assigned, wired-in, fixed size address for each node, is easily implemented, but precludes the use of multidestination addressing. Instead, it forces all entities communicating over the network to be aware of the low-level routing of messages. An alternative to fixed addressing is associative addressing, so called because the address recognition mechanism in a node's network interface is implemented using an associative memory. In the simplest form of associative addressing, each interface contains a set of addresses for which the interface is a destination. Attached to a broadcast network, the interface listens to each packet as it is transmitted, and picks up those packets that are addressed to one of the addresses contained in the interface. A multidestination packet can be sent by using an address recognized by several interfaces. It is also possible to move a destination from one node to another transparently with this scheme. These features may be used to great advantage in the design of distributed system software, as discussed by Mockapetris and Farber [20]. Two disadvantages of this scheme are 1) its implementation requires more complex address recognition hardware containing host-loadable memory for the addresses, and 2) it is more difficult to determine the cause of failure when messages are not delivered.

Mockapetris has designed a more general associative addressing scheme [21] that allows host interfaces to match subfields of the message destination address against subfields of addresses contained in the host interface name table. For example, see Fig. 5. Host *A* will receive all messages addressed to destinations with the first four bits zero. Host *B* will receive all messages with the first two bits zero, and the fifth bit one. The additional power gained by this scheme is the ability to specify large classes of destinations with a small amount of memory in the interface.

The comparison of point-to-point and broadcast transmission in a ring topology raises the interesting question of how a message is removed from the ring. In a point-to-point mode, each node must examine the message before deciding whether to remove it. Thus each node must buffer enough of the message to see the destination address before passing the message

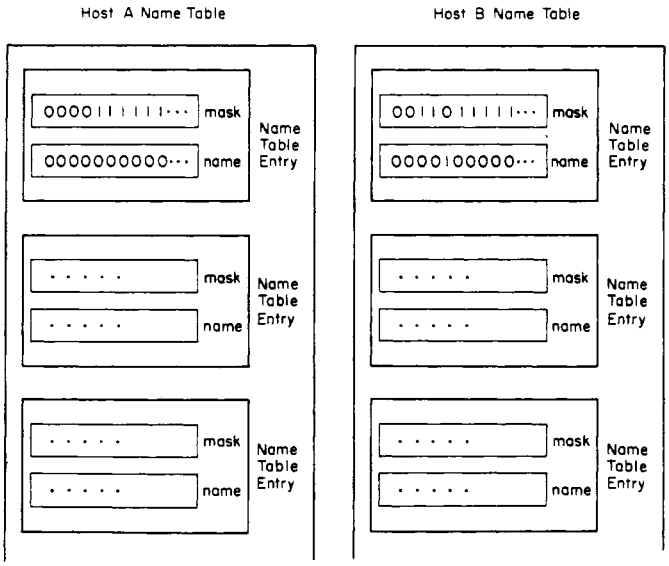

Fig. 5. Name Tables for Host Associative Addressing. Zeros in host A's first name table entry's mask select the first four bits of destination addresses for comparison. Destination addresses whose first four bits match the first entry's name are thus accepted by host A. Zeros in host B's first name table entry's mask select the first, second, and fifth bit for comparison. Destination addresses of the form 00XX1XXX ··· are thus accepted by host B.

on. Considerable delay will be introduced by that buffer unless a special addressing technique is used which allows this decision to be made in 1-bit time. In contrast, it is possible to build a broadcast net in which the message can be removed without examining it first. In the network for the Distributed Computing System, there is one control token, so there is one message on the network at any time. Thus a transmitting node knows that the next message it sees will be its own, and can remove it without examination. In the Cambridge Network, several message slots circulate on the net, so the transmitting node must do something slightly more complicated to determine which slot contains the returning message to be removed. In fact, the technique used is to count the number of slots going by. When the correct number has passed, the next slot is marked as empty.[3] Again, no buffering is required. In fact, if techniques such as this are used for message removal, it is possible to build a network in which the only delay at each node is due to gate propagation delays.

F. Transmission Media

Most local area networks now in use or being designed use bit-serial transmission over either coaxial cable or twisted pair. The geographic limitations of local area networks are precisely those encountered when attempting to send high-speed digital information over such wire or cable. The goal of simplicity has led to the use of baseband signaling, the simplest class of modulation schemes, as the means of encoding data. Thus if one is interested in achieving the highest bit rates or the longest distances, one will choose coaxial cable because of its more uniform impedance characteristics. On the other hand, it is much easier to splice new nodes in between two existing nodes if a twisted pair is used.

[3] To determine the total number of slots in the ring, each node makes use of a unique pattern occuring only once on the sequence of slots circulating on the ring, counting the number of slots between arrivals of this pattern.

One promising candidate for local area data transmission is cable television (CATV) technology. CATV makes tremendous bandwidth available; its wide-spread utilization tends to make CATV system components low in cost. In many cases, CATV may already be installed, and a network can be produced using some of the channels of the existing equipment. The MITRE Corporation has developed MITRIX, a CATV network with a bus topology and centralized control [22], and an alternative with decentralized contention control [23].

Fiber optics is another promising candidate for local area data transmission. Transmission of signals for a small number of kilometers (but significantly longer than possible using wire or cable) using bit rates between 1 and 20 Mbit/s appears to be a fairly simple task for fiber-optic technology; other characteristics of optical fibers such as their high noise immunity and inherent ground isolation make the technology even more tempting. However, transmission using fiber-optic technology is inherently unidirectional, which seems to eliminate the bus topology at the present time. An interesting and challenging problem is the design of a high-speed bus topology contention network using fiber optics as the transmission medium.

Radio broadcast has been demonstrated for a local network using packet switching [24]. Other ideas have even been suggested such as communication between computers using light signals reflected off a mirrored ceiling or a blimp. In general, the physical transmission medium for a local area network should be reliable, simple, inexpensive, high-speed, noise-free, and physically robust. It should also be easy to install, maintain, and reconfigure. There is room for further creativity in this area.

IV. The Nature of Hosts and their Interfaces

The hardware of a local area network is keyed to high performance at low cost. In a decentralized local area network, the interface hardware associated with a host generally provides all the transmission control and address recognition circuitry that is required. Because of the desire to connect low-cost minicomputer and microprocessor systems to local area networks, there is a good deal of motivation to make the host interface hardware as inexpensive as possible. Ultimately, a good portion of a host interface for a local area network could be implemented as a single large-scale integrated circuit (LSI) chip.

A. Hardware Structure of a Host Interface

Generally, the host interface hardware for a network may be viewed as having two parts: a *network-oriented* part that performs whatever transmission control functions are required for the network, and a *host-specific* part that fits into the I/O structure of a particular type of host computer and controls the exchange of data between the host and the network-oriented portion of the interface.

The simple architecture and control structures of local area networks aid in reducing the complexity and cost of the network-oriented portion of the interface. However, the situation can be quite different for the host-specific portion of the interface, as microprocessor systems, minicomputers, and large-scale systems present a wide range of I/O interface complexity. Interfaces for microprocessor systems tend to be the least complex, because of the simple bus structures of microprocessor systems, and the availability of LSI peripheral interface circuits. Large-scale systems tend to require the most

complex interfaces, as one might expect. Interfaces for minicomputer systems tend to be more complex than one might expect at first glance, largely due to the need for a "direct-memory access" type of interface required by a high-bandwidth peripheral device, which the host interface for a local area network is.

B. Approaches to Attachment of a Host to a Network

Over the entire range of computer systems, from microprocessors to large-scale systems, there is little difference between the complexity of the host-specific portion of an interface for a local area network and for a long-haul network. With the reduced complexity of the network-oriented portion of a local area network interface, and the low cost of transmission medium of a local area network, domination of the cost of attachment of a host computer to a network shifts from network-related costs, in the case of long-haul network, to host-related costs, for a local area network. This shift has two major effects: first, it becomes practical—that is, economically justifiable—to connect microprocessor systems to a network, and, second, it causes those responsible for the attachment of large-scale hosts to a local area network to examine their approach very carefully.

One approach to the interfacing of large-scale systems to long-haul networks that has become popular among those who do not wish to develop specialized hardware interfaces (until packet network interfaces become standard offerings of large-scale system vendors) or modify vendor-supplied operating system software is to interconnect a packet network to the system via a *front-end processor*, usually a minicomputer, which has an appropriate packet network interface and which connects to the large-scale host system in a way that mimics a standard method of attachment to the system, such as a group of remote interactive terminal lines, or a remote job entry (RJE) port. With such an attachment, the host is usually limited to utilizing only that portion of the network's functional capabilities which correspond to those of the standard attachment being mimicked. This front-end approach is less satisfactory for attachment of a local area network to a large-scale host, for, although a large-scale host system may be even better able to utilize the high data rate offered by the local area network than a minicomputer or microprocessor, actual data rates available through standard interfaces mimicking RJE or interactive terminal ports are meager by comparison. In addition, the protocols and applications used with and envisioned for local area networks are less well matched to standard interactive and RJE ports than are those of long-haul packet networks. In short, more of the potential of the local area network is lost through front-ending.

Although development of specialized hardware and software to interface a large-scale host system to a local area network may initially be the more expensive path to follow, it is likely to be the most fruitful in the long run; for, with properly designed interface, the high-speed local area network and the large-scale host system are particularly well matched. Such an interface is virtually a necessity in applications in which the large-scale host serves as a central data repository, as a specialized or centralized information processing resource, or in tightly coupled distributed processing systems.

C. Case Study of a Local Network Interface

We have discussed a number of alternatives for the topology, control strategy, and interface hardware of a local area network. We shall now examine a particular local network interface, both to see how the various design issues were resolved in the particular unit, and to gain an overall perception of its complexity. Our example is the Local Network Interface, or LNI, originally developed at the University of California at Irvine and now being used as part of the local area network under development at the Laboratory for Computer Science of the Massachusetts Institute of Technology [24]. This interface can be made to operate a control token ring network, a contention ring network, or a contention bus network with only small modifications. These varied capabilities will enable us to perform experiments comparing these different network control strategies in the same operational environment. By examining the modifications needed to achieve each of these operational modes it is possible to perceive the similarities and differences required to support each of them.

1) Host-Specific Part: The structure of the local network interface (LNI) follows the general plan outlined in Section IV-B above, comprising a host-specific part and a network-oriented part. The first implementation of the LNI is for a Digital Equipment Corporation PDP-11 minicomputer host; therefore, the host-specific portion of the initial LNI is a full-duplex direct memory access (DMA) interface connected to the PDP-11 UNIBUS [16]. With this interface, the LNI can transfer data to or from the memory of the PDP-11 without the intervention of the processor. Although there are simpler forms of I/O interfaces for the PDP-11; namely, programmed I/O and interrupt-driven I/O, the data rate of the LCS Network (initially, 1 Mbit/s, with an eventual goal of 4–8 Mbit/s) requires a DMA interface to ensure that the PDP-11 processor will be available for tasks other than servicing data transfers to and from the LNI.

a) Registers: A PDP-11 full-duplex DMA interface is a surprisingly complex device. As implemented for the LNI, it contains ten registers directly addressable by the PDP-11 processor, including two 16-bit and two 18-bit counter registers:

Command
Status
Transmitted Data
Received Data
Transmit Address (lower 16 bits)
Transmit Address (upper 2 bits)
Receive Address (lower 16 bits)
Receive Address (upper 2 bits)
Transmit Byte Count
Receive Byte Count.

The Transmitted and Received Data Registers allow the LNI to be used in an interrupt-driven or programmed I/O mode for testing purposes. The Command Register is a path to flip-flops in the network-oriented portion of the LNI which controls its operation; setting bits in the Command Register initializes transmission of messages over the network, enables receipt of messages, etc. Similarly, the Status Register is a path to flip-flops in the LNI which indicate the success, failure, or other status of the LNI.

b) Input/output transactions: In a typical DMA transaction, the PDP-11 processor loads the memory address of the first byte to be transmitted into the Transmit Address register, and loads the Transmit Byte Count Register with the number of bytes to be transmitted. The processor then sets a bit in the

Command Register to initiate a transmit operation. The DMA interface requests memory cycles of the PDP-11, transferring data bytes from PDP-11 memory into a first-in-first-out (FIFO) buffer in the LNI. Once the buffer is full, further transfers from the PDP-11 take place only when actual transmission over the network has begun, as data are shifted out of the FIFO buffer. The setup of the DMA for receipt of data from the network is similar: the Receive Address and Receive Byte Count Registers are set, and the LNI enabled for input. Data are transferred into the PDP-11 only when a message addressed to this network host begins to arrive. Good programming practice suggests that an input operation should *always* be pending, and, unless complex I/O buffer strategies are used, the input byte count register should be set to permit receipt of a message of the maximum expected length (a "full packet"). A DMA operation terminates either when the byte count goes to zero, or upon a signal from the network-oriented portion of the LNI (e.g., a complete received message contained fewer bytes than the maximum initially set in the Byte Count Register). Terminations are generally signalled to the PDP-11 via an interrupt, although interrupts may be inhibited, under the control of a bit in the command register.

c) Advantages of full-duplex operation: The full-duplex nature of the interface makes it possible to initiate a transmit operation while a receive operation is pending. This is important for a network interface, especially a local network interface, as the receiver of a message has no control over when that message will arrive. A host may initiate a transmit DMA operation, with the network-oriented portion of the interface awaiting an opportunity to begin actual transmission of the message over the network, when a message may arrive on the network addressed to that host. If the host cannot perform a receive DMA transaction while the transmit operation is in progress, the message addressed to the host will be lost (unless the network-oriented portion of the interface can buffer entire messages).

d) Interfaces for other host computers: The PDP-11 DMA interface described here is typical of the host-specific part of a local area network interface for minicomputers, and for some microprocessor systems as well. DMA interfaces for microprocessors are available as LSI chips, although they are generally half-duplex interfaces, and two would be required for the full-duplex operations described here. Later versions of the MIT-UCI LNI will provide host interfaces for the PDP-10 and LSI-11; the PDP-10 interface will be a program-interrupt, I/O bus interface, while the LSI-11 bus interface will be similar to the PDP-11 UNIBUS interface described here.

2) Network-Oriented Part: Like any interface for a local area network, the network-oriented portion of the LNI (shown in Fig. 6) performs four basic functions:

1) *Control of transmission:* It observes transmissions on the network, determining when it may send a message of its own.

2) *Control of reception:* It observes transmissions on the network, looking for incoming messages. Incoming messages addressed to this host are transferred to the host-specific portion of the interface.

3) Address recognition: It determines whether or not a message detected by the interface is addressed to this host.

4) Signal conditioning: It provides appropriate transformations between the logic-level signals of the interface and signals appropriate to the network transmission medium. Examples are: differential voltage signals on twisted pair; bipolar pulses on coaxial cable; light pulses on optical fibers.

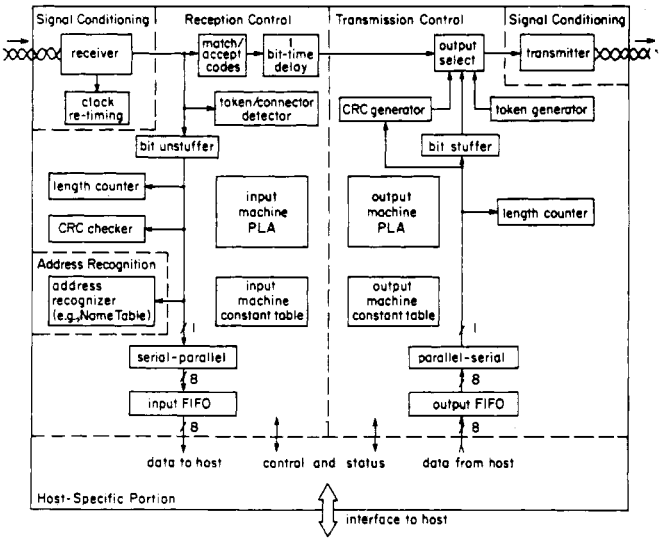

Fig. 6. Block diagram of the Local Network Interface (LNI). The five major components described in the text are separated by dashed lines; the Signal Conditioning Section appears in two parts, at the upper left and upper right. The arrows depict data flow; control, vested in the Programmed Logic Arrays (PLA's) of the input and output machines, is not indicated.

In the LNI control of transmission and control of reception are each achieved with a sequential state machine composed of a state counter, field-programmable logic array (FPLA), header field length and data length counters, and programmable read-only memories (PROM's) which serve as constant tables providing the lengths of header fields. The actions taken by these machines, and their sequence, can be changed by reprogramming the FPLA. The lengths of the various fields of a packet header can be changed by reprogramming the constant table PROM's. The initial version of the LNI implements a ring network similar to the original UC-Irvine DCS ring network. When an LNI is not transmitting a message, it serves as a repeater, retransmitting, one bit-time later, each bit it receives. It recovers clock from the incoming signal and synchronizes its transmit clock to the recovered clock.

When the transmission control section of the LNI has a message to send, it waits for the passage of the *control token*, a particular bit pattern, on the ring. Since the LNI introduces only one bit-time of delay into the ring, the token detector circuit must itself be a small sequential machine. The transmission control section inverts the last bit of the token to transform it into a *connector* which indicates to each LNI on the ring that a message follows. The transmission control mechanism then ceases repeating bits it has received, and instead transmits its own message. The transmit clock is decoupled from the recovered clock, and the transmitting LNI sets the timing of the ring.

The *output machine* of the transmission control section of the LNI follows each field of the packet header as it is transmitted, noting in particular the *length* field of the packet. After all the data of the packet have been transmitted, the output machine outputs a 16-bit cyclic redundancy checksum (CRC) followed by a match/accept field and a new control token. While this is taking place, the *input machine* of the reception control section is following the message, which has traveled around the ring and returned to the transmitting LNI. It verifies the received checksum, as it would for any message it would receive for this host, and passes the received checksum to the output control machine, to verify that it is

identical to the one transmitted. This, together with the fact that the input machine detects extraneous tokens or connectors appearing in the middle of a message, provides a means of detecting ring errors due to faulty LNI's which may have begun to transmit a message without waiting for a token.

When the LNI is *not* transmitting a message of its own, the input machine monitors data arriving on the ring and passes them to the output machine to be repeated. When the token detector detects a *connector*, the input machine begins to follow the fields of the message. The destination address fields are passed, a bit at a time, to the address recognition section of the LNI. In addition, the reception control section of the LNI assembles the bits into 8-bit bytes and passes them to a FIFO buffer. When the entire destination address field has been received, the address recognition section indicates whether or not the message is for the host. If it is, the input machine signals the host specified part of the LNI that the data in the FIFO may be passed to the host; if not, it clears the FIFO and does not load subsequent data bits into the FIFO.

The input machine compares the checksum which follows the data of the received message with the checksum it has computed; checksum errors are reported to the host, if the message was addressed to the host. Following the checksum, the machine modifies the match/accept bits to indicate that the message was received. Neither the checksum nor the match/accept bits are placed into the FIFO to be passed to the host.

The address recognition section of the LNI is a sophisticated associative memory *name table* [21] which can be loaded by the host. It is an essential element of the UC-Irvine Distributed Computing System concept, in which the name table is loaded with the names of processes currently located in the host. The LNI name table, and the destination address fields of packets on the ring, contain masks as well as names, to facilitate the addressing of classes or groups of processes. From the point of view of the LNI as an example of a local network interface, the name table contains more sophistication than is necessary: it could be replaced by a simple mechanism which recognizes a single address, either wired into the unit or encoded in a PROM.

With either the name table or the single address recognizer, address recognition is done on a bit-by-bit basis, since that is how the data are presented to the address recognition circuit by the reception control section. The address recognition circuit indicates to the reception control section, after all the address bits have been processed, whether the address fields match—whether the message is addressed to this host.

The signal conditioning section of the LNI is quite straightforward; a simplified version of it is shown in Fig. 7. The transmission medium of the ring network is a shielded twisted pair; differential signals are placed on the pair by the transmitter of one LNI, and current flow on the pair is detected by high-speed optocouplers placed across the pair at the next LNI, which also serve to terminate the pair. Two optocouplers are used, connected across the pair in opposite directions, one to detect current flow in each direction. During the first half of each bit-time, the transmitter output driving each side of the pair is low, so that no current flows in the pair and neither optocoupler turns on; during the second half of the bit-time one transmitter output goes high if the value of the data bit is 1, and the other goes high

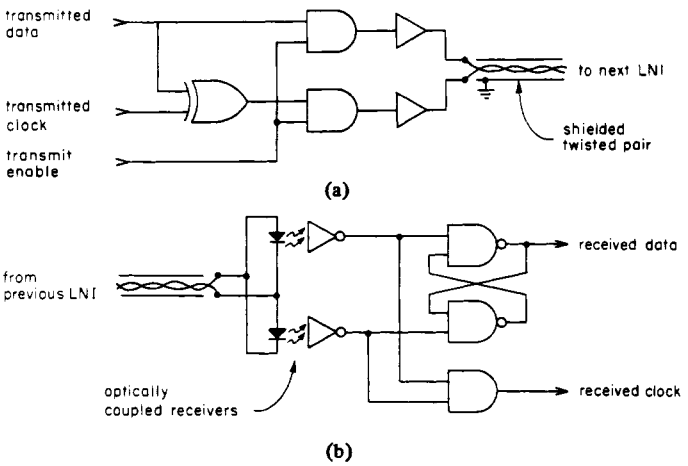

Fig. 7. Simplified schematic diagram of the Signal Conditioning Section of the LNI. (a) The transmitter circuit. (b) The receiver circuit.

if the bit to be transmitted is a 0. Thus the direction of current flow in the pair during the second half of a bit-time indicates the value of the data bit; the current flow is detected by one or the other of the optocouplers, one coupler turning on to indicate a 1, and the other turning on to indicate a 0. The couplers drive a simple latch, the output of which presents recovered data to the reception control section of the LNI, with a half-bit-time of delay. The logical OR of the outputs of the two couplers yields the recovered clock: 0 for the half-bit-time the pair is quiesient, and 1 for the half-bit-time during which there is current flow on the pair.

3) Modifying the LNI for Other Types of Networks: The claim was made earlier in this section that the structure of the LNI was representative of the structure of interfaces for various types of local area networks, and that the LNI could readily be modified to operate either a contention bus network or a contention ring network. We now investigate how this can be done, to further illuminate the nature of local area network interfaces.

Little change needs to be made in the host-specific part of the LNI, as the nature of data interchange with the host remains the same. Different types of networks may require somewhat different command bits, or report different status bits, but, since the most of the bits of command and status registers of the host-specific part of the LNI are but paths to and from flip-flops in the LNI's network-oriented portion, these changes have little impact on the host-specific part of the LNI.

The several sections of the network-oriented portion of the LNI are affected to varying degrees. Least affected is the address recognition section; its function remains the same: to examine the bits of the address field of an arriving message to see if it addressed to this host. Although the DCS ring network and the LNI have been described in the literature as containing a name table associative memory, and contention bus networks such as the Xerox PARC ETHERNET have been described as having a fixed address recognition mechanism, these are design decisions based on the intended initial applications of the network technology, rather than choices dictated by the nature of the technology itself.

The changes made to the transmission control section of the LNI are the most illustrative. In the ring network previously described, the LNI output machine section must wait for a token to pass before initiating the transmission

of a message. In a contention bus or contention ring network, the output machine may transmit only when the network is quiet. The "token present" signal is replaced by a "network quiet" signal. In the ring network, the reception control section signals the transmission control section if it detects another token in the midst of its receipt of the message the transmission control section sent; this has its analogue in the collision detection capability of the contention network. In both cases, the LNI must abort transmission of its message and take corrective action. In the ring network this is an error condition, an exception; more than one control token is present in the ring. In the contention network, a collision is an expected event. Both situations can be handled by the LNI reporting the event to host software, which can attempt to restart a token on the ring, in the ring network case, or apply a retransmission backoff algorithm in the contention network case.

A better solution for the contention network is to modify the transmission control section to execute a simple retransmission backoff algorithm in hardware. This requires that the entire message remain accessible to the transmission control section without host intervention. The FIFO buffer cannot be used in this situation; a complete packet buffer which is not erased until the message has been successfully transmitted is an appropriate alternative.

Two features of the ring network LNI's transmission control section are not needed in the contention bus network version: the data repeater which passes bits from the receive side of the LNI to its transmit side when the LNI is not transmitting a message, and the token generator which places a new token or connector onto a quiescent ring. Of course, the connector is a brief sequence of bits, and there is no good motivation to delete it from the beginning of messages transmitted by the contention bus version of the LNI. In fact, retention of the connector at the head of a message results in fewer changes to the input machine of the LNI. It can use its token/connector detector to signal the beginning of an incoming message. Its function remains the same, for the most part; extra connectors detected in the middle of a message indicate a collision, just as they do for the ring network version. However, in the contention bus network, because bits are not repeated from one LNI to another, there is no way to set the match/accept bits for the benefit of the transmitting LNI, and the match/accept field of the message cannot be used.

The signal conditioning section of the LNI undergoes an interesting transformation. For a contention ring network, of course, the signal conditioning section remains the same. However, for a contention bus network, the logic levels of the LNI must be converted to appropriate signal levels and waveforms for the coaxial cable of the bus. This is done in a two-step process. First, a cable transceiver is added to the configuration. To minimize impedence mismatches, reflections, etc., the transceiver is located immediately adjacent to the network cable, and is often packaged separately from the LNI.[4] It is connected to the cable either directly, or via

[4] This has become common practice in local area networking; the networking transmission medium is generally *not* brought into the racks, equipment bays, etc., of a host computer where it would be subject to accidental disconnection and other physical abuse that could disrupt the entire network. Instead, the connection point for a host is designed to be physically stable: a box on the wall, above a false ceiling, etc.

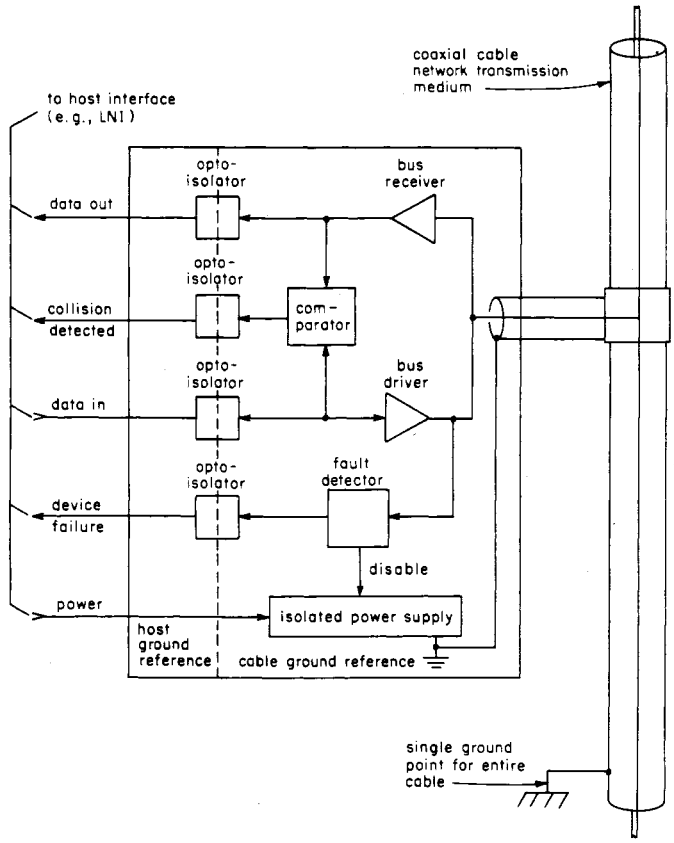

Fig. 8. A typical bus transceiver. The opto-isolators and isolated power supply permit the drivers and receivers to be referenced to cable ground; the cable, in turn, is grounded at only one point along its length, eliminating problems that would result if each transceiver tied the cable to local host ground.

a short stub cable attached to the main cable via a tap. Second, since the transceiver is located adjacent to the network bus cable, and the LNI is located next to its host, an appropriate transmission scheme must be selected to span the intervening distance. For distances up to 30 ft or so, "single-ended" drivers and receivers will suffice. For better reliability, greater distances, or both, differential signals over a shielded twisted pair can be used—just as in the transmission medium of the ring network itself. So, the signal conditioning section of the original LNI can be modified to interconnect the LNI and the cable transceiver.

4) The Cable Transceiver: The care taken in the design of a cable transceiver for a contention bus network will strongly influence the overall reliability and performance of the network. Therefore, we conclude our case study by examining a hypothetical contention bus cable transceiver, shown in Fig. 8, that is similar to one designed and built for the CHAOS Network at the MIT Artificial Intelligence Laboratory; it is typical of transceivers built for various contention bus networks.

The cable transceiver performs the following functions:

1) transmission (cable driving);
2) reception;
3) power and ground isolation;
4) collision detection;
5) transceiver fault detection ("watchdog").

The first three of these constitute part of the signal conditioning function described previously.

The basic design principle of the transceiver is that it must present a high impedence to the bus except when it is transmitting and actually driving the bus. This is essential to the operation of the contention bus network; a large number of receivers on the bus must not present impedence lumps or in any way interfere with a transceiver which is actively transmitting.

The receiver must be able to detect and properly receive signals from the most distant point on the bus; in addition, it must be able to detect a colliding signal while its companion transmitter is itself driving the bus. This requirement impacts the choice of an encoding scheme for data transmitted on the bus. A number of data encoding schemes can be used, all of which require that the transmitter be able to place the transmission medium in two distinct states. At first glance, it might seem that *three* states could be used: the quiescent, high-impedence state, to indicate that no transmission is in progress, and two active driver states, for example $+V$ and $-V$. However, with two active driver states, when two or more network nodes attempt to transmit simultaneously, the cable will be driven to different voltage levels at different points. This has two effects. First, it places a severe load on drivers. Second, it makes the detection of a colliding signal more difficult than it needs to be. On the other hand, if the transceiver drives the cable to some voltage to represent one signaling state, and represents the other signaling state by *not* driving the cable, the problem of overloaded drivers is eliminated, and the task of collision detection is greatly simplified. Collision detection is accomplished looking at the bus during the transmitter's quiescent state. Any signal present during that time must come from another transceiver, and constitutes a collision. The transceiver can detect an incoming signal with 20-dB attenuation, which corresponds to about 1 km of the particular cable used.

The transceiver must be able to cope with ground potential differences at the various network hosts. Isolation is accomplished by high-speed optocouplers and an isolated power supply which enables the major circuit elements of the transceiver to be referenced to cable ground, rather than local host ground. Finally, the fault detection, or watchdog circuit examines the output of the driver to guard against transceiver failures which drive the bus and disrupt the network. The signaling states used by the transceiver result in the driver being quiescent approximately 50 percent of the time; if the driver remains on steadily for several bit-times, it is deemed to be faulty, and the fault detector disconnects its power, which, of course, returns the driver to its high-impedence state.

5) Complexity of the Local Network Interface: In its present form, the LNI comprises about 350 TTL SSI and MSI integrated circuits, apportioned as follows:

PDP-11 full-duplex DMA	100
Name table controller	25
Name table cells (8 provided)	90
Network-oriented portion	120
Test and diagnostic	15
Total	350

The count of 120 chips for the network-oriented portion of the LNI, excluding the associative name table, is well within the capabilities of current large-scale integration. As the field of local area networking matures, and standards are arrived at, it is likely that integrated circuit manufacturers will add local area network controllers to their product lines, to take their place alongside other LSI data communication chips which are already available, making high-performance local area network technology available at a very reasonable cost.

V. Protocols for Local Area Networks

As in long-haul networks, local area network protocols can be divided into two basic levels—low-level protocols and high-level protocols. At each level, the characteristics of local networks impact effects on protocol design and functionality.

A. Low-Level Protocols

The term *low-level* protocol identifies the basic protocols used to transport groups of bits through the network with appropriate timeliness and reliability. The low-level protocols are not aware of the meaning of the bits being transported, as distinct from higher level application protocols that use the bits to communicate about remote actions. Two aspects of local area networks have a very strong impact upon the design of low-level protocols. First, the high performance achievable purely through hardware technology enables the simplification of protocols. Second, low-level protocols must be designed to take advantage of and preserve the special capabilities of local networks, so that these capabilities can be utilized, in turn, by higher level application protocols. We will explore these two issues in this section.

1) Simplicity: Local area networks must support a wide variety of hosts, from dedicated microprocessors to large time-sharing systems. The existence of extremely simple hosts (such as microprocessor-based intelligent terminals, or even microprocessor printer controllers) leads to a desire for simple, flexible, low-level protocols that can be economically implemented on small hosts, while not compromising the performance of large hosts. Supporting a variety of hosts also leads to a difficult software production and maintenance problem that can be ameliorated somewhat by having a protocol that is simple to implement for each new kind of host. Although quite a variety of hosts has been attached to long-haul networks such as the ARPANET, the problem of software development has not been too severe, since each individual host in such environments usually has a software maintenance and development staff. In the local area network context where a variety of computers are all maintained by a small programming staff, the arguments for simplicity in protocol design are far stronger in our view.

In a long-haul network, complexity results from strategies that attempt to make as much of the costly network bandwidth as possible available for transport of high-level data. The costs of a local area network are concentrated instead in the host interfaces, the hosts themselves, and their software. Two factors lead to the simplicity of low-level local area network protocols.

a) Unrestricted use of overhead bits: Bandwidth is inexpensive in a local area network; there is little motivation to be concerned with protocol features designed to reduce the size of the header or overhead bits sent with each message. This is in contrast to protocols developed for networks making the more conventional assumption that bandwidth is expen-

sive. For example, the ARPANET NCP host-to-host protocol [26] initiates a connection using a 56-bit (net, host, socket) identifier for the destination, but then goes through a negotiation so that instead of sending this 56-bit value on subsequent messages, a 32-bit (net, host, link) value can be sent instead. It is not clear whether this conservation of bits is appropriate even in a long-haul network; in a local area network, where bandwidth is inexpensive, it is clearly irrelevant. Other examples of ways in which extra header space can be used to simplify processing include:

1) having a single standard header format with fields in fixed locations, rather than having optional fields or multiple packet types; field extraction at the host can be optimized, reducing processing time;
2) using addresses that directly translate into addresses of queues, buffers, ports, or processes at the receiver without table lookup.

b) Simplified flow control, etc.: The low transmission delay inherent in local area networks, as well as their high data rate, can eliminate the need for complex buffer management, flow control, and network congestion control mechanisms. Consider, for example, flow control: the problem of assuring that messages arrive at the recipient at the rate it can handle, neither too fast, so that its buffers overflow, nor too slow, so that it must wait for the next message after processing the previous one. In a long-haul network, a receiver typically allocates to the transmitter enough buffer space for several messages following the one currently processed by the receiver, so that messages can be placed in transit well before the receiver is ready to process them. Considerable mechanism is required to keep the sender and the receiver properly synchronized under these circumstances. In a local area network, the delay will typically be low enough for a much simpler flow control mechanism to be employed. For example, one can use the very simple strategy of not sending a message until the recipient has explicitly indicated, by a message in the other direction, that it is ready for it. In contrast, a network using communication satellites has such a high transmission delay that very complex predictive flow control algorithms must be used to obtain reasonable data throughput.

It is crucial to understand that other factors may obviate these simplifications. While the data rate and delay characteristics of a local area network can render it essentially instantaneous, its speed cannot eliminate the intrinsic disparity that may exist between the capabilities of two hosts that wish to communicate with each other. These disparities may not show up when the two hosts are communicating through a long-haul network whose characteristics are so constraining that the principal problem is dealing with the restrictions of the network. While protocols for local area networks need not include mechanisms designed to cope with the limitations of the network itself, it is still necessary to design protocols with sufficient generality to cope with disparities between the capabilities of machines wishing to communicate through the network. Such disparities include:

1) mismatch between the rate at which hosts can generate and absorb data;
2) host delay between the time a packet is received and the time it is successfully processed and acknowledged:
3) amount of buffer space available at the sender and the receiver.

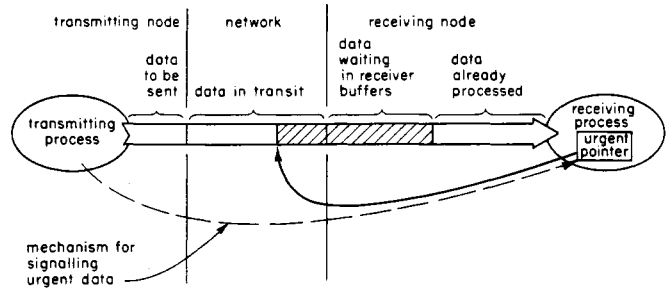

Fig. 9. The urgent pointer mechanism. By transmitting a new, larger value of the urgent pointer, a pointer into the data stream, a sender can indicate the data bufferred in the sender, network, and receiver are holding up data that must be processed quickly. The receiver can then adjust his use of the data stream flow control to process the bufferred data until the urgent data is processed. The shaded area indicates the location of potentially urgent data specified by a particular urgent pointer value.

Further, considerable effort may be required to modify host software to provide a suitable interface to the network. If one were to consider the simple flow control mechanism mentioned earlier, where a message is sent in the reverse direction requesting transmission of each message as it is needed, one would discover that in many cases the scheme was unworkable, not because the network introduced intolerable delays, but because the hosts communicating with each other themselves introduced excessive delay. In a large host with a time-shared operating system, for example, the real time that elapses from the time a message is received, one or more processes are scheduled in response to this message, and that process runs, to the time a message is sent in response, could well run into a large number of milliseconds, milliseconds during which the other host is forced to wait.

c) Example of protocol simplification: The low-level protocol initially proposed for the Laboratory for Computer Science Network at MIT is an example of the sort of protocol that results when simplicity of mechanism is a primary design goal. The Data Stream Protocol (DSP) was based on the Transmission Control Protocol (TCP) used in internetworking experiments sponsored by the Defense Advanced Research Projects Agency [27], but evolved from original TCP due to the continuing desire to simplify the protocol features, packet formats, and implementation strategies. Most of these simplifications have subsequently been incorporated into the TCP.

One specific example is the mechanism used to signal *interrupts* and other urgent messages that are logically part of the sequence of data in a virtual circuit. The basic model is that the sender occasionally wants to signal the receiver that all data in the stream preceding the signal (buffered somewhere in the network) must be scanned immediately in order to respond promptly to some other important signal. A mechanism is provided whereby a pointer into the data stream is maintained at the receiver, which can be moved, when the sender chooses, to point to a more recently transmitted piece of data. This pointer, called the *urgent pointer*, can be used to indicate the point in the data stream beyond which there is no more urgent data. (See Fig. 9.) The urgent pointer can be implemented in two ways, depending upon the nature of the host receiving the message. In the case of a simple (e.g., microprocessor) host dedicated to a task that processes the incoming stream as it arrives, the host need not process the urgent pointer, since by design, all data, urgent or not, are processed as quickly as possible. In contrast, on a large time-shared host, data need not be processed until either

a) the process to receive the data is scheduled and requests input, or b) the urgent pointer points to data not already received by the process. In case b) an interrupt is sent to the receiving process, indicating that data should be read and processed until the urgent pointer is past. The corresponding mechanism in TCP required that a host be capable of understanding and responding to a special interrupt signal in the data stream, even if the signal had no meaning to the host in its particular application of TCP. The urgent pointer, then, is a simple mechanism that meets the needs of sophisticated host implementations without placing an excessive burden on unsophisticated hosts.

2) Special Capabilities: The other aspect of low-level protocols for local area networks to be discussed is the manner in which protocols must be structured to take advantage of, and provide to higher levels, the unique capabilities of local networks. Conventional low-level protocols have provided a function best characterized as a bidirectional stream of bits between two communicating entities—a *virtual circuit*. The virtual circuit is implemented by a process that provides sequenced delivery of packets at the destination. While a virtual circuit is one important form of communication, two others easily provided by a local network are very useful in a variety of contexts. These are *message exchange* communication, where the packets exchanged are not viewed as being members of a sequence of packets but are rather isolated exchanges, and *broadcast* communication in which messages are sent not to one particular recipient but to a selected subset of the potential recipients on the network.

a) Message exchange: A typical example of a message exchange is the situation in which one message asks a question and another provides the answer. For example, if there are a large number of services provided by nodes connected to a local net, it is disadvantageous to maintain, on every node, a table giving all of the addresses of these, for whenever a change is made in the network address of any service, every node's table will need to be revised. Rather, it may be advantageous to maintain, as a network service, a facility which will take the name of a desired entity and give back its network address. Clearly, the pattern of communication with this service is not one of opening a connection and exchanging a large number of messages, but instead is a simple two-message exchange, with a query of the form "What is the address of such and such a service?" and a reply of similarly simple form. While a virtual circuit *could* be used for this exchange, it is unneeded and uses excessive resources.

b) Broadcast: The example given above demonstrates the need for a broadcast mechanism. If the service described above is intended to provide the address of network services, how can we find the address of this service itself? An obvious solution is to broadcast the request for information. The query then takes the form "Would anyone who knows the address of such and such a network service please send it to me?" There are many other examples, some apparently trivial but nonetheless very useful, for support of broadcast queries in a local network. A microprocessor with no calendar clock may broadcast a request for the time of day. A new host attached to the network for the first time may broadcast a message announcing its presence, so that those who maintain tables may discover its existence and record the fact. Broadcast mechanisms in the low-level protocols can also be quite useful in implementing higher level protocols for such applications as document distribution to multiple host nodes, and for speech and video conference calls.

Why are these alternative models of communication not commonly found in traditional networks? The first, and perhaps most important reason is that long-haul networks have not been extensively exploited for applications in which computers directly query other computers with individual, self-contained queries. Instead, the major use of long-haul networks has been for long-term, human-initiated interactions with computers, such as direct terminal use of a remote computer, or long-term attachments of remote job entry stations. Such human interactions usually involve many message exchanges between sender and receiver, so that the extra delay and cost of initial setup of a virtual circuit is insignificant—perhaps even recovered by reducing redundant information in each message. As new applications such as distributed data base systems become more important, these alternative models will become important in long-haul networks, but long-lived connections between terminals and host computers continue to dominate the usage.

The second reason is precisely that discussed in the previous section concerning the relative simplicity of protocols for local area networks—a variety of functions performed in conventional networks are very difficult to understand except in the context of a sequence of ordered messages (a virtual circuit) exchanged between two nodes. For example, flow control is normally handled in network protocols by placing an upper bound on the number of messages which may be flowing at any one time between the sender and the receiver. This concept has meaning only in the restricted case where the sender and the receiver are a well-identified pair exchanging a sequence of messages. There is no obvious equivalent of flow control that can be applied to situations where sender and receiver communicate by sending arbitrary unsequenced messages, or where a sender broadcasts to several receivers. Similarly, if efficiency requires use of the shorthand version of an address for communication between the sender and the receiver, this clearly implies that the sender and the receiver have negotiated this address, and agree to use it over some sequence of messages. Again, this idea makes no sense if communication is isolated in unsequenced messages.

Another problem that is traditionally handled in the context of a sequence of messages is the acknowledgment to the sender that the receiver has correctly received a message. If messages are sequenced, acknowledgment can be very easily done by acknowledging the highest member of the sequence that has been successfully received. If messages bear no relationship to each other, then each must be identified uniquely by the sender, and acknowledged uniquely by the receiver. This increases the complexity and overhead of acknowledgment. However, in most cases where message exchange communication is the appropriate underlying communication model, no acknowledgment mechanism is required of the low-level protocol at all. For example, if a microprocessor system asks the time of day, it is not at all necessary to acknowledge that the query has been successfully received; the receipt of the correct time is sufficient acknowledgment. Similarly, a request for a network address is acknowledged by a return message that contains the desired address. Depending on a low-level acknowledgment message to handle all failures can be dangerous, for it may lead to the practice of assuming that acknowledgment of receipt of a message implies that the message was processed at a high level.

In the broadcast context, it is difficult to formulate a useful definition of acknowledgment that can be supported by a low-level protocol. What does it mean to say that a broad-

cast message has been successfully received? By one of the possible recipients? By all of the possible recipients? One appropriate strategy is to rely on the high-level application to deal with these problems as a part of its normal operation, rather than have the low-level protocol concern itself with issues of flow control or acknowledgment at all.

3) Protocol Structure: Based on the previous observations, a two-layer structure is a very natural one for low-level protocols in a local area network. The bottom layer should provide the basic function of delivering an addressed message to its (one or many) destinations. This level corresponds to the concept of a *datagram* network [28]. It should also take on the responsibility of detecting that a message has been damaged in transit. To this end it may append a checksum to a message and verify the checksum on receipt. However, this layer probably should not take on the responsibility of ensuring that messages are delivered, and delivered in the order sent, since different applications have different needs and requirements for these functions. The first layer might be implemented entirely in hardware; however, if the packet size, addressing structure, or routing topology of the hardware is not sufficient to provide adequate message size, process addressing, or broadcast selectivity, some software help will be needed to make up the difference.

Above this first layer should be made available a variety of protocols. One protocol should support a virtual circuit mechanism, since a virtual circuit is definitely the appropriate model for a great deal of the communication that will go on in any network, local or otherwise. As alternatives to the virtual circuit protocol, there should be mechanisms for sending isolated messages, for message exchange communication, and additional alternatives to provide support for message models other than the ones we have discussed here. For example, transmission of digitized speech requires a communication model with some but not all of the attributes of the virtual circuit; in particular, reliability is of less concern than timeliness of arrival.

B. Applications of Local Area Networks; Higher Level Protocols

In the previous section we considered low-level protocols for a local area network. These protocols exist, of course, to support higher level protocols, which, in turn, support user applications. In this section we will consider a number of applications for which local area networks are suited.

1) Access to Common Resources: The model of computing most common over the last few years is that of a large centralized computer, with the only remote components being terminals and, perhaps, a few other I/O devices. Line control protocols such as SDLC [19] were created to serve this sort of arrangement. A simple but very important application of a local area network is to generalize this picture very slightly to include more than one central computer. As the total workload grows to exceed the capacity of a single machine, a common solution is to procure a second machine, and to divide the applications and workload between the two. The communication problem to be solved in this arrangement is simple but critical—to allow an individual terminal to have access to both of the central machines. A local area network can solve this problem, and provide some additional capabilities as well. For example, if the central facility has specialized I/O devices such as plotters or microfilm writers, they can be placed on the local area network and made accessible to both central machines—an advantage if a device is expensive and is not heavily enough loaded to justify having one for each computer. Further, I/O devices can be placed remote from the central site but convenient to users; for example, a line printer can be placed near a cluster of users.

This pattern of sharing among several computers can be expanded to include more than just I/O devices. In fact, the network can be used to move computations from one machine to another in order to spread the computing load equally. The high speeds available in the local area networks make this sort of load leveling much more practical than do the bandwidths traditionally available on long-haul networks.

2) Decentralized Computing: A wide variety of new uses for a local area network arises if the computing power available is not strongly centralized. Let us consider the alternative of a computing environment consisting of a large number of relatively small machines, each dedicated to a small number of users or a small number of tasks. In the extreme, we can look to the future and imagine the day when each user has a computer on his desk instead of a terminal. Such a completely distributed computing environment by no means eliminates the need for an interconnecting network, for users will still need to exchange information. Data files containing the results of one person's computation will need to be shipped through the local area network to be used as input to other tasks. Users will wish to communicate with each other by exchanging computer mail, as is now done over the ARPANET [29]. Users will still want access to specialized resources which cannot be provided to each user, resources such as large archival storage systems, specialized output devices such as photo typesetters, or connection points to long-haul networks. All of these features can be made available through the local area network.

3) Protocol and Operating System Support: The applications outlined in the previous paragraph can be supported by high-level protocols very similar to the ones already in existence in the ARPANET: TELNET for logging into a remote system through the network, and File Transfer Protocol for exchanging data between machines [26]. When one examines how these protocols might be modified to take advantage of the special attributes of a local area network, for example, its higher speed, one discovers that the problem is not one of modifying the protocols, but of modifying the operating system of the hosts connected to the network so that the services available through the network appear to be a natural part of the programming environment of the operating system. The File Transfer Protocol in the ARPANET, for example, is usually made available to the user as an explicit command which he may invoke to move a file from one machine to another. As part of this invocation he may be required to identify himself at the other machine, and give explicit file names in the syntax of the local and the foreign machine, describing exactly what action he wishes to perform.

This particular view of file transfer has two disadvantages. First, there is a lot of overhead associated with moving a file. Much of the delay in moving the file seen by the user has nothing to do with the time required to send the data itself through the network, but is rather the time spent establishing the connection, identifying the user at the other site, etc. Second, the file system on the local computer understands nothing about the existence of files accessible through the network. No matter how sophisticated the local file system

is, in terms of keeping track of the various files that the user cares about, it requires explicit user intervention in order to reach through the network and retrieve a file from another machine. The use of a high-speed local area network will not eliminate any of these problems, but will instead make even more obvious to the user the overhead that the protocol imposes on the transfer of data. Clearly, what is needed is a further integration of the local area network into the file system and user authentication mechanism of the individual operating systems, so that interchange of information between the various machines can be done with less direct user intervention. Some attempts have been made to do this within the context of the ARPANET. RSEXEC is an example of a protocol which makes files on various TENEX operating systems in the ARPANET appear to the user to exist on a single machine [30].

The design of operating system structures to take full advantage of the capabilities of local area networks represents the current edge of research in this area. Examples of operating systems that incorporate a high-speed local area network into their architecture are the Distributed Computing System [31], the Distributed Loop Operating System [11], and MININET [32].

VI. Interconnection of Local Area Networks with other Networks

A. Motivation for Interconnection

As was mentioned earlier, a local area network will be only a part of the overall communication system used by the hosts attached to it. A very important use of the local area network can be to provide an interconnection between hosts attached to a local area network and other networks such as long-haul packet-switched networks and point-to-point transmission links. The advantage of this method of interconnection is reduced cost, by taking advantage of the fact that connection of a host to a local area network is relatively inexpensive. Instead of connecting all machines directly to the long-haul network, one can connect all the host computers to the local area network, with one machine, the *gateway*, connected to both the local area network and the long-haul network.

B. Protocol Compatibility

There are two pitfalls that should be avoided when planning for the interconnection of a local area network with a long-haul network. On the one hand, long-haul networks currently cannot provide all of the functions that local area networks can. If a local area network is initially designed to serve only the function of connecting hosts to a long-haul network, the protocols of the local network may be designed to serve only the needs of communicating with the long-haul network, and may not support the other functions that make a local area network especially attractive. On the other hand, if a local network is initially designed with no thought given to the possibility that it may be interconnected with another network, the protocols designed for it may lack the necessary generality. For example, the addressing structure used on the local area network may not be able to express destinations outside the local network. In either case, the only after-the-fact solution is to implement a second set of protocols for the local area network, so that different protocols are used for intercommunication with long-haul networks and for local services. This proliferation of protocols is undesirable, as it adds to the cost of software development associated with each new host added to the local area network. To avoid these pitfalls, it is important that all the functions a local area network is to provide must be considered from the very inception of the design of the network, and the protocols for the network must be designed to support that entire range of functionality.

Fortunately, initial experiments with protocols for local area networks suggest that a uniform approach to protocol design can support both specialized local network functions and interconnection with other networks, provided that both functions are envisioned from the start. Although the protocols used in the local area network must be made slightly more general to handle the internetworking situation, there is no interference with the realization of the purely local network functions. For example, a more general address field must be used to specify the destination of a message, but the only overhead implied if this same addressing structure is used for purely local messages is additional bits in the message to hold a presumably larger address. Since bandwidth is inexpensive, the bits "wasted" on this larger address are presumably irrelevant.

A slightly more difficult problem, one that is still being studied, is the problem of speed matching between the local area network and the long-haul network. As this paper has characterized the difference between local nets and long-haul nets, it is reasonable to presume that the local network will have a much higher data rate. If a host sends a large number of packets into the local area network with an ultimate destination to be reached through the long-haul network, the packets may arrive at the gateway much faster then the gateway can pass them to the long-haul network. Some mechanism will be required to prevent the gateway from exhausting its buffer space. The speed matching problem is not unique to the gateway between the local area network and the long-haul network; it occurs any time two networks of differing speed are connected together. (The problem may be more extreme here, though, due to the greater speed difference that can be encountered between local area and some long-haul networks. Satellite networks with speeds comparible to local networks are quite conceivable, yet are a long-haul technology.) A general discussion of the problems of internetworking, and some proposed solutions can be found in a companion paper by Cerf and Kirstein in this issue [33].

At the next higher level of protocol, one finds facilities that support various communications models, such as virtual circuits, broadcast, and message exchange. In interconnecting to a long-haul network we are chiefly forced to deal with a virtual circuit model, since that is the only pattern of communication usually supported by commercial long-haul networks. Here, it is appropriate to use a virtual circuit protocol in the local area network as similar as possible to that used in the long-haul network, so that translation between the two is easy. Although there is not as much practical experience available in the area of network interconnection as could be desired, it appears that one can develop a virtual circuit protocol for a local area network that is a compatible subset (in the sense of using compatible packet formats and control algorithms) of a suitable long-haul virtual circuit protocol. This means that it is not necessary to implement two complete virtual circuit protocols, one for internal local network use and the other for communication out through

the local net. It leaves unanswered the question of how the additional features, such as complex flow control, buffering, and speed matching required for the long-haul protocol should be implemented. One approach would be to implement them in every host that desires to communicate over the long-haul network; this implies a programming burden for every machine. An alternative would be to implement the additional functions in the gateway machine that interconnects the local area network to the long-haul network. This would add considerable complexity to the gateway, for it will have to cope with such problems as the speed differential between the two networks without having the benefit of the flow control mechanisms normally used for this purpose in the long-haul network. At this time, it is not clear whether the gateway can assume the entire responsibility for augmenting a local network virtual circuit protocol with the functions required for communication through a long-haul network.

It would be advantageous to make sure the local area network protocols are also compatible with other communication models, such as single message exchange or selective broadcast, that may become available on commercial long-haul networks in the future. However, this presupposes that the long-haul networks attached to the local area network use a two-layer low-level protocol implementation such as that described for the local area network, and if the long-haul networks do use such an implementation, that they provide an interface that allows direct use of the datagram layer. Many current long-haul networks do not provide that interface.

VII. The Subnetwork Concept

Resting midway between the monolithic, single-technology, local area network and the internetworking environment is an approach to local area networking that we term the *subnetwork concept*, which provides for a mix of network technologies within a uniform addressing and administrative structure.

A. General Approach

A local area network can be composed of a collection of subnetworks, possibly implemented with various network technologies and perhaps with various transmission rates, but using identical software protocols, compatible packet sizes, and a single overall homogeneous address space.[5] These subnetworks are interconnected by *bridges*, which are midway in complexity between the repeaters used in a multisegment contention bus network (ETHERNET) and the gateway processor used between networks in an internetworking environment. This general structure is indicated in Fig. 10. A bridge links two subnetworks, generally at a location at which they are physically adjacent, and selectively repeats packets from each of them to the other, according to a "filter function."[6] In addition, since they buffer the packets they repeat, they can perform a speed-matching function as well.

B. Advantages of Subnetworking

The subnetworking concept enables a variety of technologies and data rates to be utilized in a single local area network, each to its best advantage. For example, a network could

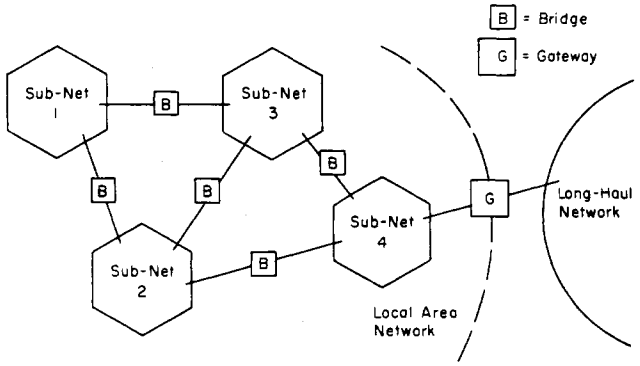

Fig. 10. The subnetwork concept. Here, a local area network is composed of a number of subnetworks, linked in some fashion by bridges. The subnetworks, though of differing technologies, share one address space, and the same protocols are used over the entire network. Thus, the bridges can be simpler than the gateway which connects the local area network to the long-haul network. Viewed externally, from outside the dashed line in the figure, the local area network appears to be monolithic.

be constructed with a contention bus subnetwork, perhaps using coaxial cable originally installed for CATV, and with a ring subnetwork, using twisted pair which can be easily installed in a crowded laboratory environment. These two subnetworks could be of different data rates; the bridge between the two will handle the speed difference between them.

Subnetworking also provides an orderly means for handling growth in traffic. Local area networks perform best, providing high throughput with low delay, when they are not heavily loaded. As traffic on a local area network grows with time, if a higher speed technology is not available, it may be desirable to split the network into two or more interconnected subnetworks. Since the bridges which interconnect the subnetworks are selective in their repeating of packets "across the bridge," not all packets from a subnetwork will flow to all other subnetworks, and the traffic density on each subnetwork will be less than that of the original monolithic network. If the partitioning of the hosts into subnetworks can be done along the lines of "communities of interest," such that a group of hosts with high traffic rates among themselves but with substantially lower traffic rates to other hosts are placed in the same subnetwork, traffic across the bridges will be minimized, and a greater fraction of all packets will stay within their subnetwork of origin.

C. Bridges

A bridge, depicted in Fig. 11, contains:

- two network interfaces, one appropriate to each of the two subnetworks it interconnects,
- a limited amount of packet buffer memory, and
- a control element, which implements an appropriate filter function to decide which messages to "pull off" one subnetwork and buffer until it has an opportunity to retransmit it to the other subnetwork.

The topology of the subnetworks interconnected by a bridge determines the complexity of its filter function. A bridge with a simple filter function can be implemented using a finite state machine as its control element; a complex filter function, which may involve a periodic exchange of information among bridges on the network to determine correct routing, may require the capabilities of a microprocessor [34].

A bridge *must* buffer packets since, upon receiving a message from one subnetwork which it decides to repeat to the other

[5] The subnetwork concept, as we describe it, is a generalization of an approach suggested by Pierce [5] for use with multiple loops or rings.

[6] The concept of the filter function is introduced in the "filtering repeaters" described by Boggs and Metcalfe [14].

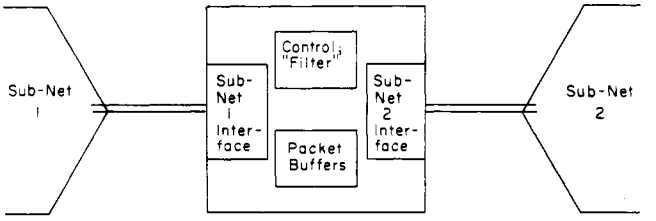

Fig. 11. The structure of a bridge. A bridge would most naturally be located at a point where the two subnetworks it interconnects have been made physically adjacent.

subnetwork, it must wait for an opportunity to transmit on that subnetwork, according to the control structure of that subnetwork. Packet buffers also aid a bridge in handling instantaneous cross-bridge traffic peaks during which the traffic offered by one subnetwork exceeds the available capacity of the other. This situation can arise if the bridge interconnects subnetworks of dissimilar data transmission rates, or subnetworks of drastically different traffic densities. However, if the sustained cross-bridge traffic offered is greater than the target subnetwork can handle, the bridge must discard packets. This is an acceptable course of action, as local area network protocols are generally prepared to handle lost packets.

D. Transparency

The subnetwork structure of a local area network should be transparent, both to the hosts on the local area network and to the "outside world"—other networks to which the local area network may be connected via gateways. A host on the local area network wishing to transmit a packet to another need have no knowledge of whether that host is on the same subnetwork, in which case the packet will be received by the destination host directly, or whether the destination host is on another subnetwork, in which case the packet is retransmitted by one or more bridges. In particular, no ordinary data packets *are ever addressed* to a bridge; rather, packets are simply addressed to their destination hosts, and may be picked up by a bridge and passed along through other subnetworks, finally reaching their destinations. This is a key distinction between subnetworking, with bridges, and internetworking, with gateways: in internetworking, a host about to transmit a packet must realize that the host to which it is addressed is on a different network. The sending host must transmit the message in a local network "wrapper" to an appropriate gateway, which "unwraps" it, performs protocol conversions, if any, packet fragmentation, etc., as necessary, and then transmits the message into the other network. In subnetworking, protocols are identical over all subnetworks, and packet sizes are compatible, so that neither protocol conversion nor fragmentation takes place in the bridges. Finally, as was mentioned above, a packet is directly addressed to its destination host, not to a bridge, for hosts do not know that the local area network is composed of subnetworks.

E. Impact on Network Characteristics

Splitting a local area network into subnetworks has little impact on the key characteristics of the network. From the point of view of the users and hosts of the network, addressing is affected only slightly, if at all. Bridges must determine whether or not a packet should be picked up for retransmission; one way to aid bridges in this determination is to include a subnetwork field in the address of each host. Other routing techniques which have no impact at all on addressing (such as complete table look-up of host addresses by the bridges) can be implemented, although usually at the expense of greater complexity within the bridges.

Splitting a local area network into subnetworks should have no effect on the protocols of the network. One exception is if a particular subnetwork technology provides a hardware acknowledgment of delivery of a packet (as in the DCS Ring Network) [2]; this acknowledgment may only indicate successful receipt by a bridge. However, not all network technologies provide hardware acknowledgments, and, in a network of mixed technologies, host-to-host acknowledgments will generally be provided by software protocols. Traffic is, of course, affected by subnetworking in a positive way. Splitting a local area network into subnetworks in a judicious way can minimize the overall traffic of the network; bottlenecks can be eliminated by using higher bandwidth technologies for affected subnetworks.

F. The Long-Distance Bridge

There are situations in which it is necessary to interconnect two subnetworks of a local area network which cannot be brought physically adjacent to one another so that an ordinary bridge may be connected between them. An example of this would be a local area network on a university campus, with a major research laboratory across town. The laboratory may be beyond the range of a twisted-pair ring network or a coaxial cable contention bus network; or it may be within range, but it may be impossible for the university to install its own cables between them.[7] The off campus research laboratory can be given its own subnetwork, connected to the main campus subnetwork via a specialized *long-distance bridge*.

A long-distance bridge is made up of two *half-bridges* at either end of a suitable full-duplex point-to-point communication link, such as a high-bandwidth common carrier circuit, an optical link, or a private microwave link (Fig. 12). Some other network technology such as packet radio could be used to derive this point-to-point link as desired.[8] Each half-bridge contains an appropriate interface to its subnetwork, packet buffers, and a controller. In addition to its filtering function, the controller of a half-bridge regulates the flow of data over the communication link between the two halves of the bridge. Of course, it is possible that the bridge communication link may be of lower bandwidth than the two subnetworks it interconnects. Additional packet buffers at each half-bridge can help to smoothe out traffic peaks, but if the communication link is a bottleneck, the long-distance bridge must discard packets just as an ordinary bridge does when it is overloaded.[9]

[7] Although common carriers such as the Bell System operating companies are moving in the direction of leasing wire pairs for transmission of digital signals with customer-provided equipment, these circuits are not intended for use at the high bandwidth of local area networks, and are generally routed through central offices rather than point-to-point.

[8] Although we do not discuss it further in this paper, there is an interesting philosophical issue whether the intervening network should be viewed in the internetworking context using gateways or as a point-to-point link within a single bridge.

[9] If the bottleneck created by the communication link of a long bridge is severe, the local area network advantages of high-bandwidth communication with low delay will be forfeited.

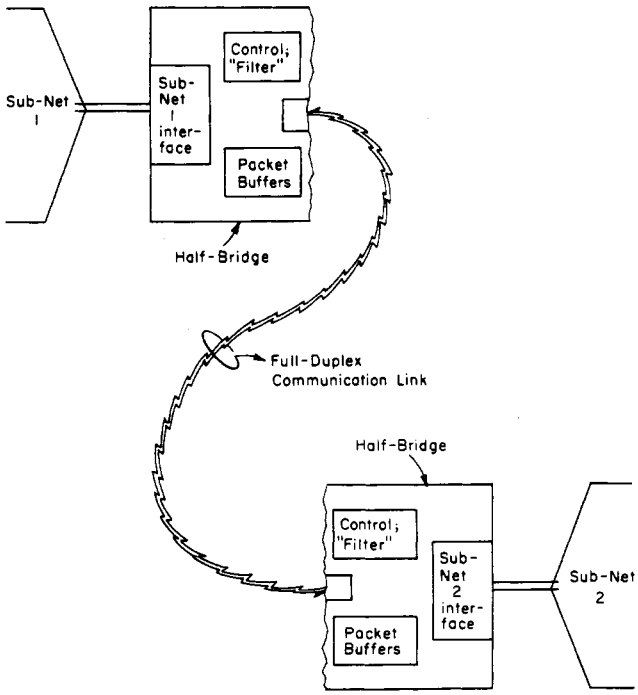

Fig. 12. The "long bridge." In this case, the two subnetworks cannot be made physically adjacent, so a half-bridge is attached to each, and a full-duplex communication link is employed to interconnect the two half-bridges. The control and filter functions, and the packet buffers, are replicated in each half-bridge.

VIII. Conclusion

The utilization of a technological innovation often occurs in two stages. In the first stage, the innovation is exploited to perform better the same tasks that were already being performed. In the second stage, new applications are discovered, which could not be reasonably performed or even forseen prior to the innovation. Local area networks are now on the threshold of this second stage. While there is still much room for creativity in improving the innovation itself—reducing the cost of the network interface and increasing its speed and convenience—the real challenge lies in identifying new sorts of applications that a local area network can make possible.

Current trends in hardware costs encourage abandoment of a single large computer in favor of a number of smaller machines. This decentralization of computing power is, for many applications, a natural and obvious pattern. In many information processing applications, for example, the information itself is distributed in nature, and can most appropriately be managed by distributed machines. Distributed applications can only be constructed, however, if it is possible to link their machines together in an effective manner. Subject to their geographical limitations, local area networks offer a very effective and inexpensive way to provide this interconnection. The greatest impact of local area networks will come with the development of operating systems that integrate the idea of distribution and communication at a fundamental level.

The impact of local area networks on the decentralization of computing is sociological as well as technological. Operational control of centralized computers has traditionally been vested in the staff of a computer center. The trend toward decentralized computing greatly increases the autonomy of individual managers in the operation of their machines, and appears to reduce the need for a centralized staff of computer managers. The communication capability made available by local area networks will serve to bind these decentralized machines together into a unified information processing resource. The effectiveness of this resource can be measured by the degree of coherence it achieves, which, in turn, depends upon the care and foresight put into the design of the local area network and the development of standards for communication at all levels. It is in the identification of areas in which standards are needed, and in their development, that the staff of the "computer center" of the future will find its work.

References

[1] D. J. Farber and K. C. Larson, "The system architecture of the distributed computer system—The communications system," presented at the *Symposium on Computer Networks* (Polytechnic Institute of Brooklyn, Brooklyn, NY, Apr. 1972).

[2] A. G. Fraser, "On the interface between computers and data communications systems," *Commun. Ass. Comput. Mach.*, pp. 31-34, July 15, 1969.

[3] IEEE Instrumentation and Measurements Group, *IEEE Standard Digital Interface for Programmable Instrumentation*, IEEE Standard 488, 1975.

[4] D. C. Loomis, "Ring communication protocols," University of California, Department of Information and Computer Science, Irvine, CA, Tech. Rep. 26, Jan. 1973.

[5] J. R. Pierce, "Network for block switching of data," *Bell Syst. Tech. J.*, vol. 51, pp. 1133-1143, July/Aug. 1972.

[6] A. Hopper, "Data ring at computer laboratory, University of Cambridge," in *Computer Science and Technology: Local Area Networking*. Washington, DC, Nat. Bur. Stand., NBS Special Publ. 500-31, Aug. 22-23, 1977, pp. 11-16.

[7] P. Zafiropulo and E. H. Rothauser, "Signalling and frame structures in highly decentralized loop systems," in *Proc. Int. Conf. on Computer Communication* (Washington, DC), IBM Res. Lab., Zurich, Switzerland, pp. 309-315.

[8] G. Babic and T. L. Ming, "A performance study of the distributed loop computer network (DCLN)," in *Proc. Computer Networking Symp.*, (National Bureau of Standards, Gaithersburg, MD, December 15, 1977), pp. 66-76.

[9] E. R. Hafner *et al.*, "A digital loop communication system," *IEEE Trans. Commun.*, p. 877, June 1974.

[10] M. V. Wilkes, "Communication using a digital ring," in *Proc. PACNET Conf.* (Sendai, Japan, August 1975), pp. 47-55.

[11] M. T. Liu and C. C. Reames, "Message communication protocol and operating system design for the distributed loop computer network (DLCN)," in *Proc. 4th Annu. Symp. Computer Architecture*, pp. 193-200, Mar. 1977.

[12] N. Abramson, "The ALOHA system," University of Hawaii Tech. Rep. No. B72-1, Jan. 1972; also *Computer Communication Networks*. Englewood Cliffs, NJ: Prentice-Hall, 1972.

[13] R. M. Metcalfe, "Packet communication," M.I.T., Project MAC, Tech. Rep. 114, Cambridge, MA, Dec. 1973.

[14] D. R. Boggs and R. M. Metcalfe, "Ethernet: Distributed packet switching for local computer networks," *Comm. Ass. Comput. Mach.*, vol. 19, no. 7, pp. 395-404, July 1976.

[15] E. D. Jensen, "The Honeywell experimental distributed processor—An overview," *Computer*, Jan. 1978.

[16] Digital Equipment Corporation, *PDP-11 Processor Handbook*. Maynard, MA: Digital Equipment Corporation, 1973.

[17] W. D. Farmer and E. E. Newhall, "An experimental distributed switching system to handle bursty computer traffic," in *Proc. ACM Symp. Problems in the Optimization of Data Communication Systems* (Pine Mountain, GA, Oct. 1969), pp. 31-34.

[18] A. G. Fraser, "A virtual channel network," *Datamation*, pp. 51-56, Feb. 1975.

[19] *IBM Synchronous Data Link Control General Information*, GA27-3093-0, File GENL-09, IBM Systems Development Division, Publications Center, North Carolina, 1974.

[20] P. Mockapetris and D. J. Farber, "Experiences with the distributed computer system," submitted to the *J. Distributed Processing*, 1978.

[21] P. Mockapetris, "Design considerations and implementation of the ARPA LNI name table," Univ. California, Dep. Information and Computer Sci., Tech. Rep. 92, Irvine, CA, Apr. 1978.

[22] D. G. Willard, "A time division multiple access system for digital communication," *Comput. Des.*, vol. 13, no. 6, pp. 79-83, June 1974.

[23] N. B. Meisner *et al.*, "Time division digital bus techniques implemented on coaxial cable," in *Proc. Computer Networking Symp.* (National Bureau of Standards, Gaithersburg, MD, Dec. 15, 1977).
[24] R. E. Kahn, S. A. Gronemeyer, J. Burchfiel, and R. C. Kunzelman, "Advances in packet radio technology," this issue, pp. 1468-1496.
[25] P. Mockapetris *et al.*, "On the design of local network interfaces," *Informat. Process.*, vol. 77, pp. 427-430, Aug. 1977.
[26] *ARPANET Protocol Handbook*, Network Information Center, SRI International, Menlo Park, CA, NIC 7014, revised Jan. 1978.
[27] V. Cerf and R. Kalin, "A protocol for packet network interconnector," *IEEE Trans. Commun.*, vol. COM-25, No. 1, pp. 169-178, May 1974.
[28] L. Pouzin, "Virtual circuits vs. datagrams-Technical and political problems," in *AFIPS Conf. Proc.* (National Computer Conf., June 1976), p. 483.
[29] D. H. Crocker *et al.*, "Standard for the format of ARPA network text messages," ARPA Network RFC 733, NIC 41952, Nov. 21, 1977.
[30] R. H. Thomas, "A resource sharing executive for the ARPANET," *AFIPS Conf. Proc.*, vol. 42 (Nat. Computer Conf. and Exposition, 1973), pp. 155-163.
[31] D. J. Farber and F. H. Heinrich, "The structure of a distributed computer system-The distributed file system," in *Proc. Int. Conf. on Computer Communication* (Washington, DC, 1972), pp. 364-370.
[32] E. G. Manning and R. W. Peebles, "A homogeneous network for data sharing communications," Computer Communications Network Group, University of Waterloo, Waterloo, ON, Tech. Rep. CCNG-E-12, Mar. 1974.
[33] V. G. Cerf and P. T. Kirstein, "Issues in packet network interconnection," this issue, pp. 1386-1408.
[34] S. L. Ratliff, "A dynamic routing algorithm for a local packet network," S.B. thesis, M.I.T., Department of Electrical Engineering and Computer Science, Cambridge, MA, Feb. 1978.

Enhanced Message Addressing Capabilities for Computer Networks

JOHN M. McQUILLAN, MEMBER, IEEE

Invited Paper

Abstract-Three message addressing modes are described:
1) Logical addressing, in which a permanently assigned address denotes one or more physical addresses. This permits multiple connections from the subscriber to the network, as well as other functions.
2) Broadcast addressing, in which a message is addressed to all subscribers.
3) Group addressing and multidestination addressing, in which a message carries the name of a list of addresses, or the list itself.
These methods facilitate many new ways of using computer networks. The paper focuses on two basic issues for each method: efficiency and reliability, and recommends implementation approaches in each case. Significant performance improvements are possible if these addressing methods are implemented with efficient delivery mechanisms. A distinction is made between virtual circuit and datagram systems; virtual circuits are superior for logical addressing, while datagrams are preferable for broadcast, group, and multidestination addressing.

I. INTRODUCTION

HOW SHOULD one user of a network address messages to other users? The answer to this question is fundamental in defining the appearance of the network to its users. For example, does one user have to know exactly where the other is located, or just the region of the network, or is the address independent of location? Can he identify himself to the network or does the network know who he is automatically? If self-identification is possible, can he have several addresses corresponding to several roles or functions? Can he have multiple connections to the network, and can he move from one location to another without changing his address(es)? Can he send a single message to a group or list of other users (e.g., a mailing list) automatically? Can he set up "conference calls" with other users, and join conferences in progress? Can he send a message to all other users?

These questions are important for several reasons: some addressing modes allow functions which would not be available otherwise (e.g., the ability to send a message to a distribution list without knowing the identity or location of the members of the list), and which are essential for certain types of users and applications. Furthermore, these addressing capabilities offer opportunities for efficient implementations that would not exist otherwise (e.g., a message addressed to a group can be transmitted with fewer packets than the equivalent separately addressed messages). The topic of addressing has received surprisingly little attention to date; the present paper indicates that it may be a fruitful area for further work.

Manuscript received May 15, 1978; revised July 3, 1978.
The author is with Bolt Beranek and Newman, Inc., Cambridge, MA 02138.

0018-9219/78/1100-1517$00.75 © 1978 IEEE