September 1981

CS-TR-81-01

A LOCAL COMPUTER NETWORK

by

Jed B. Marti

Department of Comp. and Inf. Science
The University of Oregon
Eugene, Oregon 97403

ABSTRACT

The operation of a carrier sense multiple access / carrier detect (CSMA/CD) network is described. The operation of the network interface module is described in detail.

A LOCAL COMPUTER NETWORK

The Carrier Sense Multiple Access / Carrier Detect (CSMA/CD) local network scheme is popular for a number of reasons. As outlined by Cotton [1], the interface is simple, reliability of network interfaces does not influence overall network reliability, and the transmission medium is passive. The network interfaces can be easily built into single VLSI chips or as small microprocessor based units out of 'off the shelf' parts.

This paper describes a low cost CSMA/CD network implemented within the Department of Computer and Information Science at the University of Oregon. The first section presents the network configuration, the second the specific message formats, the third message contents, the fourth implementation of collision avoidance and the final the use of conferences in clusters of machines.

1.Ø THE NETWORK CONFIGURATION

The network is a coaxial cable with connections soldered directly to it and a set of message processors which serve as interfaces to network sites. The network is a low performance but very inexpensive version of the Alhoa, Ethernet, and NBSNET networks [2-4]. Up to 254 message processing units may be attached to the network. The host processor may be a terminal, a micro or minicomputer, or a

large mainframe. A portion of the network appears schematically in Figure 1.

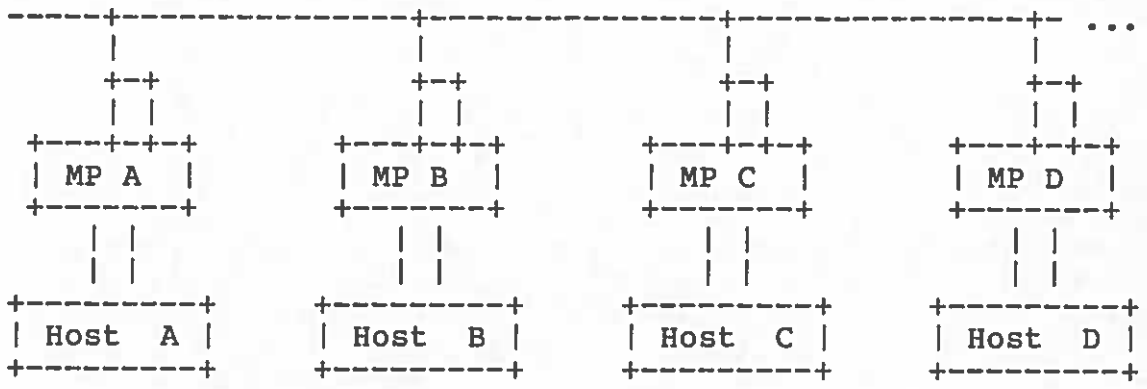

Figure 1. Host to network connections.

Connections to the network are single wire half duplex serial lines capable of asynchronous operation at 50,000 baud. The host to message processor interface is either a serial asynchronous full duplex line capable of 38,400 baud or half duplex 8 bit parallel transmission at 100,000 bytes per second.

A message processor (IMP for short) is a Z80 microcomputer with 2k bytes of ROM and 1k bytes of RAM for program and data storage. This provides the capability of message buffering, virtual circuit establishment and so on without direct host intervention. Likewise, a common network protocol and hardware interface can be established for a heterogeneous network of hosts. The use of a small computer for 'front ending' network communication has considerable historical precedent in telecommunications [5].

To send a message, an IMP watches the network connection for a short period of time. If the line is not being used, a message is sent and all IMPs attached to the line receive it. The IMP for which the message is intended sends it to its attached host and all others ignore it. If another IMP starts sending at the same time, both detect the error and restart later at random times.

The next level of network protocols requires the establishment of a 'virtual circuit', a logical connection between two message processors and their hosts. Such connections may be established between many hosts to establish conferences.

This scheme has several ramifications. Only one message can be in progress at any time. If more than one message is being broadcast simultaneously, an error will occur and both messages will be resent later. A message sent by a processor will be received by all other message processors. Each message processor will determine whether or not the host will have access to it. Removing a node from the network will have no effect on the operation of the network nor will powering down any node or message processor. Finally, network statistic collection is easily accomplished by placing a message processor in 'promiscuous' mode where all messages are received and the information present is passed to the host processor [6].

2.0 MESSAGE FORMAT

A message can be as short as 6 bytes or as long as 134. All messages are formatted as depicted in Figure 2.

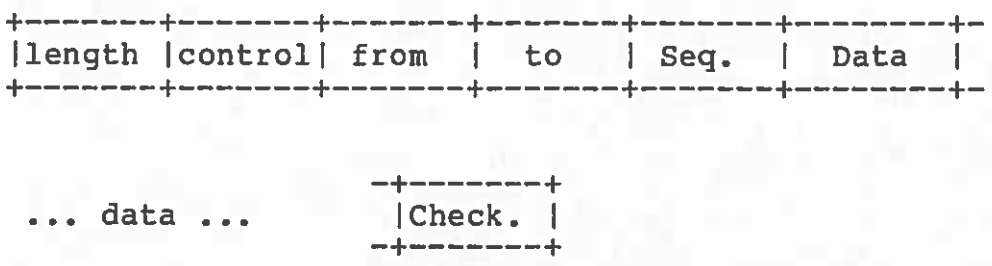

Figure 2. Standard message packet.

The minimum length of a message is 6 bytes as all fields except the data are required. The 'length' field is a single byte which represents the total number of bytes in the message including the length field. The 'control' field contains information for both the host and message processor concerning virtual circuit establishment and handshaking protocols for end of file, accepted message and so on. The 'from' field contains an 8 bit identification of the message processor sending the message and the 'to' field the identity of the destination processor. The 'Seq.' or sequence field contains a sequence number which helps in the verification of the correct arrival of packets. The sequence field is interpreted by the host processor only. The data portion of a message is optional. Up to 128 bytes of data may be sent.

The error check byte is an 8 bit checksum of all previous bytes (except itself) and is used to determine whether or not a message should be resent. A message can come to grief by two means. It may be overwritten by a message from another message processor, or a hardware error on the network may cause the loss of a bit. These errors will most likely be detected by the IMP hardware and then resent without interference by the host. Checksum errors detected by the host cause error message packets to be sent as the acknowledgement.

3.0 MESSAGE PROCESSOR SOFTWARE

A message processor runs two mutually exclusive processes: sending to the network and sending to the host. Two receiving subprocesses are interrupt driven. The four functions correspond to messages traveling to and from the network and to and from the connected host (see Figure 3).

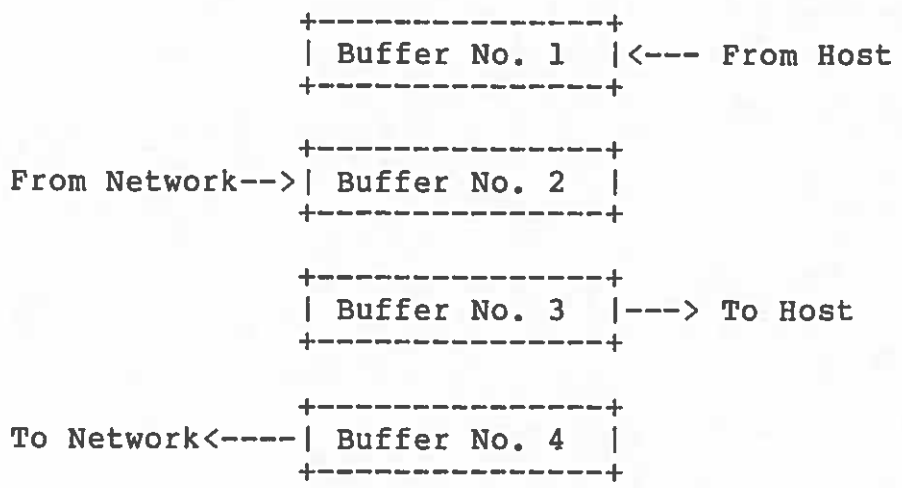

Figure 3. Messages and Buffers.

3.1 Receiving Network Messages

Messages are received and stored by an asynchronous interrupt driven process. The message processor is notified when a message is complete so that further processing may take place. If a message is complete, the following functions are performed. The 'to' field is compared against the number stored in the message processor as the identity of the host. If they are equal, then the message is for this message processor's host. If not, the buffer pointers are reset and the message is ignored. If the fields agree and the virtual circuit between the 'from' and 'to' processors has been established, then the 'send to host' process is signalled to awaken and another input buffer is established. If no virtual circuit has been established, and the message control byte says to do so, then the message

processor establishes one. If a virtual circuit has been established but with a processor different than the one of the message, the message is ignored and the buffer pointers are reset.

3.2 Transmitting Messages To The Network

Messages are received from the host processor in the format mentioned above. These messages are then broadcast according to the following strategy. First, a period of dead time or 'carrier detection' is established. If there are no messages being transmitted on the network for a predetermined period of time, transmission may begin. If the network is busy, the processor waits this specified amount of time after the last byte of the current message has been received. This time will be randomly computed and increased as the network becomes congested. When sending may finally begin, bytes are transmitted at the highest possible speed to the destination processor. At the same time, all message processors on the network including the sending one are receiving that same message. If a receiver error is detected by the message processor hardware the process is stopped and after the random amount of time, is restarted. If the message is sent without error, the host is notified of this occurrence and this process goes to sleep until awakened by the host receiving process.

3.3 Receiving Messages From The Host

The host communicates with the message processor using data packets or in a single character mode. They are buffered for transmission by the network message sender. Only when the message is complete is the network message sender awoken. The message processor is notified of proper completion by the receiving host which returns a 'valid data' reply message. Resending because of a collission is handled by the sending IMP.

A special mode can be entered by a message from the host whereby only single lines terminated by a carriage return are buffered and no error or timeout checking is performed. In this case a single line character is converted into a packet to the established virtual circuit and sent off. The return packet may be of any length and has its control and checksum stripped from it before being sent to the host. Depending upon the configuration, a carriage return, line feed sequence or just a carriage return may be sent.

3.4 Sending To The Host

When a packet has been received from another message processor and identified as being for this host, the host sender process is awoken for transmission of the packet to the host. If the special mode has been invoked by the host,

the control information is stripped from the message before transmission. When the message has been sent to the host, this process goes to sleep.

3.5 Control Codes

The message packet structure is rigorously enforced in all communication on the network. Each packet has a control code which indicates how far it is to travel and what function it is to serve.

The distance a message must travel can be either of three alternatives: between a host and its message processor; between a host and a remote message processor; or between two hosts. These will be known as HM, HMM', HMM'H' message. Messages traveling in the opposite direction are MH, M'MH, and H'M'MH messages (see Figure 4).

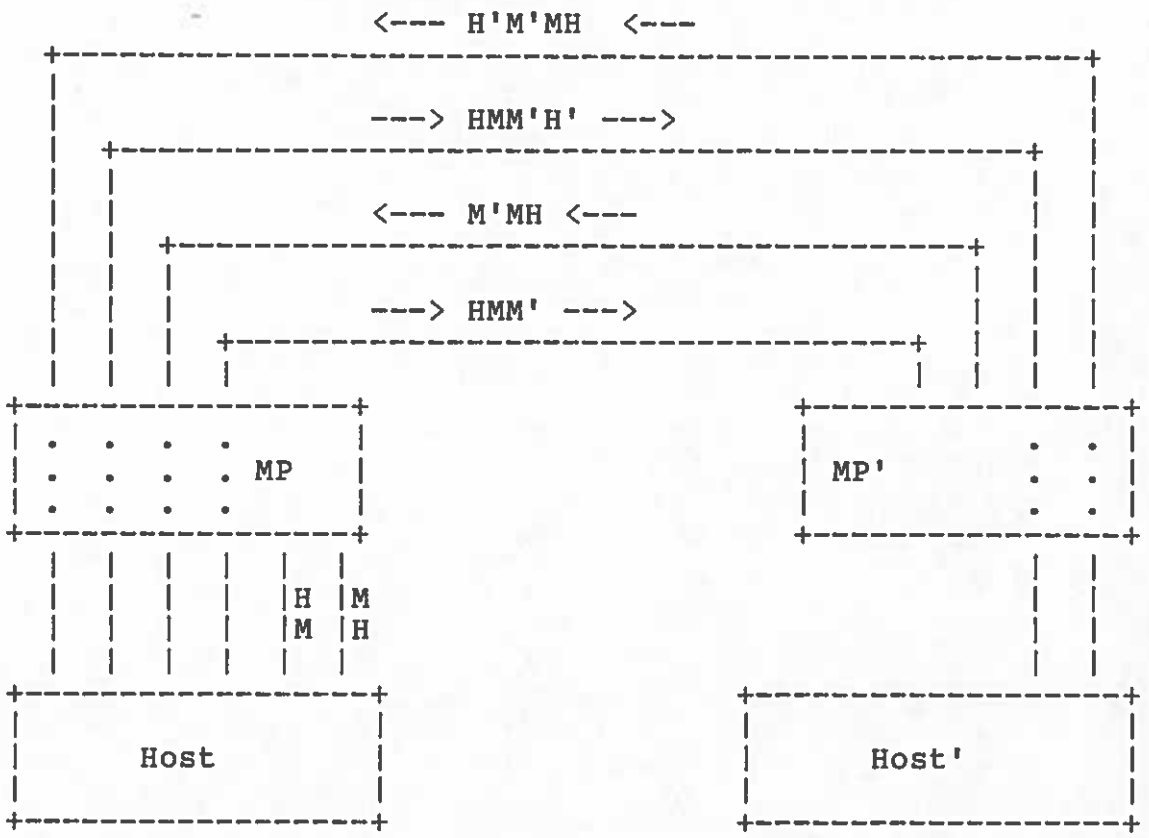

Figure 4. Message routes and distances.

The control codes included in each packet dictate direction, distance and function.

3.5.1 HM - MH Control Codes -

These packets are for transmission between a host and its associated message processor only. They are used to synchronize the two, establish local identity and control virtual circuit establishment. The control codes will be given mnemonically with system uniformity being controlled by a central implementation.

OC-LOCAL - Open a local connection. This message is sent from a host to its message processor and to notify it that a virtual circuit is about to be established with another site and that messages from other sites are to be refused. The SYS-LOCAL response packet will be returned as acknowledgement of this message.

CC-LOCAL - Close local connection. The function of this packet is to break the local end of a virtual circuit either because an attempt to make circuit was not completed or a circuit is being terminated. The remote connection must be broken first by the CC-REMOTE message or the MP there will be left hanging. The SYS-LOCAL response packet will be returned as acknowledgement of this message.

SYS-LOCAL - Response to a local message. This MH message is set as a response to a CC-LOCAL, OC-LOCAL and QUERY-LOCAL message sent from the host. The packet contains the identity number of the message processor the identity of the current virtual circuit partner, the current packet/character mode status, and the current backoff interval.

OS-LOCAL - Open Special mode locally. For hosts which are not capable of receiving packets without stripping off control information, this mode is either automatic or entered when special network programs terminate. The mode

is terminated by an escape character (ASCII 27) sent by the host.

QUERY-LOCAL - Causes the SYS-LOCAL packet to be returned to the host.

3.5.2 HMM' - M'MH Control Codes -

These control packets serve to establish a virtual circuit between two message processors and the controlling host. Refusal to establish a remote connection is silence of the remote IMP rather than a message response.

OC-REMOTE - Open Connection remote. This message is sent to a remote message processor to complete the establishment of a virtual circuit. If no response is returned the destination processor either has another virtual circuit or is refusing messages because its host is not on. The RR-REMOTE message is sent in response to this message if the circuit is established.

CC-REMOTE - Close Connection remote. This message is sent to close the virtual circuit to a remote message processor. The RR-REMOTE response will be returned by the destination message processor if the connection is properly closed.

CM-REMOTE - Causes the listener IMP to switch into character

mode.

PM-REMOTE - Causes the listener IMP to switch into packet mode.

RR-REMOTE - Respond to Remote message. This packet is sent as a response to an OC-REMOTE message as an indication that a virtual circuit has been established or in response to a CC-REMOTE message to indicate that the connection has been closed.

3.5.3 HMM'H' - H'M'MH Control Codes -

These packets contain data to be communicated between two hosts. The contents of these messages are completely determined by the host. In all host to host transactions, the host which initiates the entire dialog is called the sender and the other host is called the listener.

DS-HOST - Data Send to listener. This packet carries data from the sending to the listener host. The virtual circuit must be established or the message will be ignored by the MP at listener side. The response to this message is either DA-HOST for proper receipt or DERR-HOST if a checksum error is detected by the listener.

DA-HOST - Data Acknowledge to sender. This message is

initiated by the listener host to acknowledge the proper receipt of a message from the sender. The data portion of this message should be empty.

DERR-HOST - Data Error response to sender. This packet is initiated by the listening host to indicate that a message received has an invalid checksum and should be resent. Resending is up to the sending host and not the MP. The MP detects and resends packets when collissions occur but not otherwise. It is up to the individual hosts to perform or not the CRC checking.

EOF-HOST - End-of-file sender. This message is initiated by the sending host to indicate the end of a file being transferred between processors. A DA-HOST message should be sent by the listener to indicate that the message has been properly received.

3.6 Checksum

Each message created by a host or MP has as the last byte a checksum which is the sum of all characters in the message truncated to 8 bits. The value is computed by adding the 8 bit value of all characters to be sent (except the checksum of course) to an 8 bit accumulator truncating any overflow. The checksum is recomputed on receipt of a message and compared to the one sent. If they do not match

the message is in error and should be resent.

The sequence number is a further error check. Communicating host level processes use these to insure proper sequential arrival of messages.

4.0 NETWORK LOADING

When two or more message processors try to send a message simultaneously, a collision occurs. The carrier detection scheme prevents most of these collisions from occurring, but there is about a 1.5 character width window during which two processors can still collide. Such collissions become more probable as longer messages are used and network load goes up. During a long message more messages processors are liable to be waiting to send a message making collissions at the end of this message highly probable. The error detection circuitry of the UART's in the message processor will probably detect two colliding messages very quickly. In the unlikely event that the processors start within about one quarter of a serial bit width of each other (about 5 microseconds), the checksum errors will most likely detect what the hardware missed. In the extremely unlikely event that the processors start within 5 microseconds of each other and are sending the same message (almost impossible given the constraints on message format since they both shouldn't have the same from field) there is no way of detecting the error which in this case will manifest itself

as having come from a host resulting from ORring the bits of the from fields together.

As the network load increases, the possibility of the first type of collision increases. To prevent two or more message processors from continually bumping into each other as they try to send a message, a random time interval will be added to the waiting period. To accomplish this, a random number computed by host to message processor traffic will be continually computed. Thus one of the two processors will most likely start far enough ahead of the other for it to sense that it cannot start sending.

To prevent the network from being saturated by collisions, the time interval between restarts will be increased up to a maximum of about one half a second.

List of References

1. Cotton, I. W., 'Technologies for Local Area Computer Networks', Computer Networks 4 (1980), pp. 197-208.

2. Abramson, N., 'The Aloha System', AFIPS Conf. Proc., Vol. 37, 1970 FJCC, AFIPS Press, Montvale, N.J., 1970, pp. 281-285.

3. Metcalfe, R. N., Boogs, D. R., 'Ethernet: Distributed Packet Switching for Local Computer Networks', CACM, July 1976, Volume 19, No. 7, pp. 395-404.

4. Carpenter, R. J., Sokol, J., 'Serving Users with a Local Area Network', Computer Networks 4 (1980), pp. 209-214.

5. Newport, C.B., Ryzlak, Jan, 'Communication Processors', Proceedings of the IEEE, Vol. 60, No. 11, November (1972), pp. 1321-1332.

6. Shoch, J. F., Hupp, J. A., 'Measured Performance of an Ethernet Local Network', CACM, December (1980), Vol. 23, No. 12, pp. 711-721.