

4

PROTOCOLS

INTRODUCTION

The word protocol has been borrowed from common usage to describe computer communication. In brief, the word means something similar in both instances. It describes conventional social behaviour on the one hand and the orderly exchange of information between computing equipment on the other (see Figure 4.1). One common example of the social analogy is the college classroom. If all the students spoke simultaneously, as they felt the urge, the professor would struggle to make sense of the chaos and valuable information would be lost. For this reason, classroom protocol defines a process of raising hands in which students request that they be permitted to speak, resulting in the orderly exchange of data between several different people.

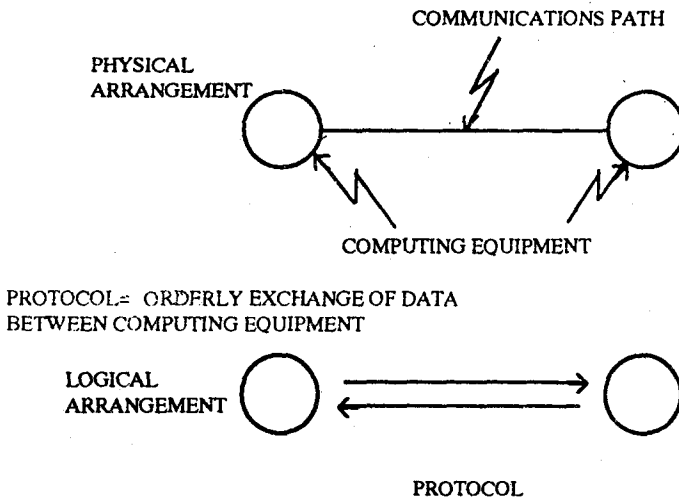


Fig. 4.1 : What is a protocol?

In computing, a protocol is necessary in order for two computers to create a path for exchanging information. The physical path may have some kind of analogy to a digital communication path connecting the two devices. The protocol is merely the logical abstraction of the process which allows two different machines to share information. There are three fundamental functions a protocol performs:

1. Establishing necessary conventions.
2. Establishing a standard communication path.
3. Establishing a standard data element.

It is in the establishment of this path that errors in the data stream may need to be detected. The control of traffic flow over the path may be simple or relatively complex or it may be nonexistant. Finally, conventions are needed for starting and stopping data exchange over the path.

The protocol's final job is to establish standard data elements for use in communication over the path. In this way, the protocol creates a virtual data element to exchange between computing elements. For instance, two computers may wish to swap character streams. Sometimes, they need to deal in a simple data element, such as a letter or memo, while at other times, they deal in entire files. Or systems may be constructed for exchanging a programme or a job between the two machines. Finally, in some applications, the element which needs to be transferred may be as complex as a graphics display.

To grasp the basic elements of a communication protocol, it is necessary to understand a few basic terms. The first is **handshaking**. Handshaking is the controlled two-way transfer of data across an interface. Two devices "shake hands" with each other via a sequence of interlocking steps. In doing this, one unit of information may be transferred (see Figure 4.2).

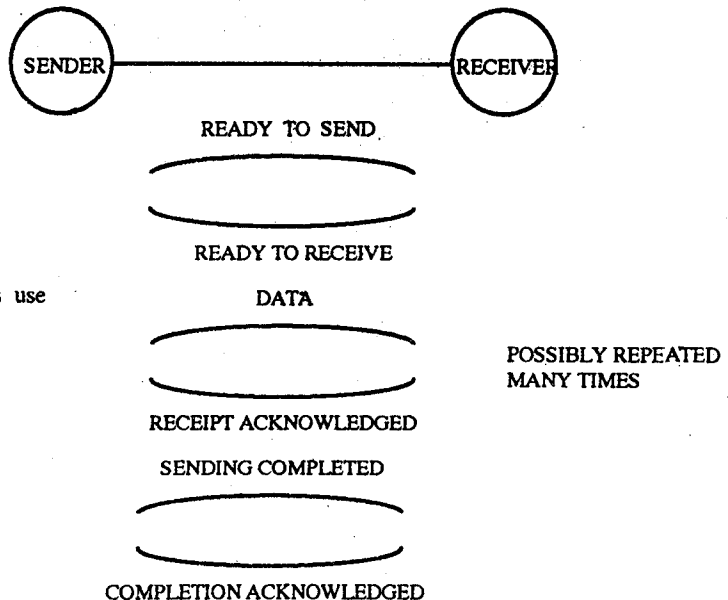


Fig. 4.2 : Protocols use "handshaking"

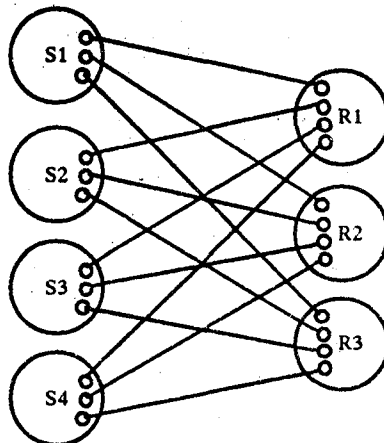
A second fundamental concept is the idea of standards, which can give users more flexibility in selecting and interconnecting equipment. Standardisation drastically reduce system development time and maintenance while allowing for evolution as computing needs change.

For two computing elements to communicate, conventions must be established. The agreed-on protocol convention determines the nature of the data representation, the format and speed of the data representation over the communication path and any sequence of control messages which are sent. A control message may be the "hello" which initiates the connection, or any other supervisory or control step. In other words, protocol conventions range from describing what a zero (0) and a one (1) look like to the control messages which "start" and "stop" data traffic.

The protocol can also build a standard communication path between computing devices. This entails translating the physical realities of the path between the two devices into a more useful virtual communication path; ideally, this is a medium suited to both pieces of equipment.

In establishing this virtual communication path, several items may need to be defined. For example, it may be necessary to have an addressing structure over the path which allows for communication with another device or with several others. A terminal and its connected computer may need to address other computers or printers. It is here that a level of priority may be defined. Messages flowing over the path may need to be sequenced or they may not.

Suppose you have four different kinds of information sources, such as mainframe computers, minicomputers, etc. Perhaps these source devices need to exchange data with three different types of receivers (including terminals, graphics terminals, word processors and personal computers). Unless there is a standard shared protocol, you will need 12 different protocols to permit all the possible connections. This is illustrated in Figure 4.3. Such a system demands that each protocol must serve a special purpose.



AD HOC COMMUNICATION : 12 DIFFERENT PROTOCOLS
24 PROTOCOL IMPLEMENTATIONS

Fig. 4.3 : Protocols without standards

The preferred solution for interconnecting equipment entails introducing a standard protocol which would require only 7 implementations (4 sources plus 3 receivers). This is shown in Figure 4.4.

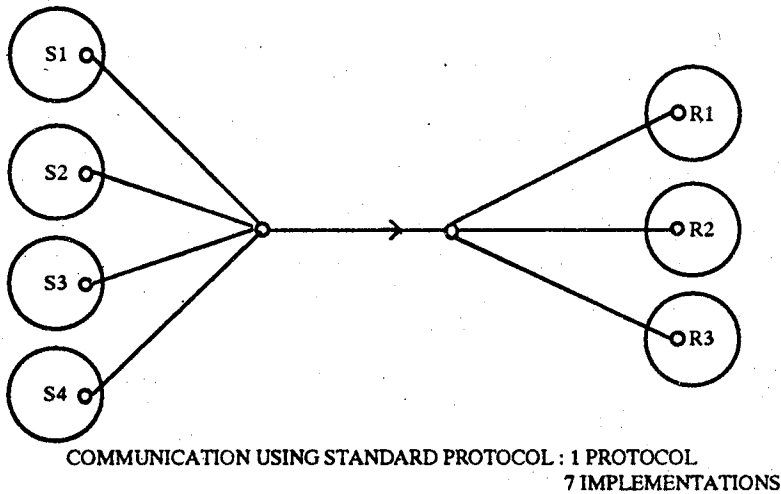
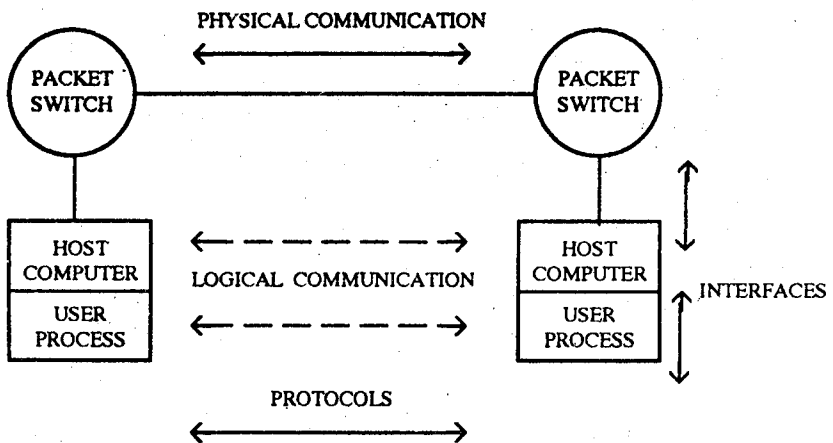


Fig. 4.4 : A standard protocol

Finally, there is a fundamental difference between protocols and interfaces. Protocol is the set of rules for communication between similar processes. Interface refers to a set of rules between dissimilar processes. Also, an interface is a physical connection between two devices or processes, while a protocol is a logical concept only. For example, there can be an interface between a host computer and a packet-switching node. In turn, there can be a host-to-node protocol as well as a host-to-host protocol. Figure 4.5 illustrates this concept.



PROTOCOLS : RULES FOR COMMUNICATION BETWEEN SIMILAR PROCESS
 INTERFACES : RULES FOR COMMUNICATION BETWEEN DISSIMILAR PROCESS

Fig. 4.5 : Protocols and interfaces

LAN PROTOCOLS

Protocols are the formal rules and conventions governing the exchange of information between computers, defined to provide reliable and efficient transfer of information. Without protocols to guide the orderly exchange of data between points in a network, there would be chaos, not communication.

Detailed protocols are required to precisely define the format in which data and system messages are to be sent; describe how a message is addressed; and govern network traffic flow by controlling priority, routing and sequencing of messages. Only when two devices agree on the specific conventions to be used can conversation take place.

Qualitatively, protocols may be said to insure that the system is capable of "useful" work. As part of the task, the protocols must be implemented uniformly throughout the network. For computer-to-computer communication, the protocols are, of necessity, complex.

No protocol works in isolation. Rather, it functions as part of the total set of instructions which determine the operations of a device or network. Each set of protocols is designed to work under different conditions and to satisfy different requirements.

The range of possible methods for passing messages between computers is enormous. The following are specific communication protocols of interest for local area networking:

- * Contention:
 - Simple contention
 - Carrier Sense Multiple Access (CSMA)
 - Carrier Sense Multiple Access with Collision Detection
(CSMA/CD)
 - Carrier Sense Multiple Access with Collision Avoidance
(CSMA/CA)
- * Polling
- * Token Passing

PROTOCOL EVALUATION FACTORS

Precise protocol specification could fill several books with complex mathematical formulas; we will avoid mathematics in favour of general functional descriptions. When selecting a protocol, the following factors may influence your choice:

Message length: How long a message may be passed between workstations at any one time?

- Traffic volume: How many messages can be passed?
- Network size constraints: How large can a network using the protocol be?
- Performance: Under what conditions does the protocol perform well? Poorly?
- Overhead: How much traffic capacity is required to pass control messages?
- Access Delay: Does the protocol have built-in delays before a workstation can access the network?
- Station Failures: What happens to the network if a workstation fails?
- Expansion: How easily can the protocol accommodate additional workstations?

CONTENTION

Contention is what happens at a staff meeting when several people start to talk at the same time. In contention protocols, no "policeman" controls usage of the communication channels.

All workstations on a contention network share a common transmission channel. Messages are broadcast on that channel and may be overheard by all attached workstations (see Figure 4.6). A workstation responds only to messages with its address: messages intended for different destinations are ignored. When not responding to a specific message, workstations are passive, simply listening in on the channel rather than being actively involved in transmitting messages.

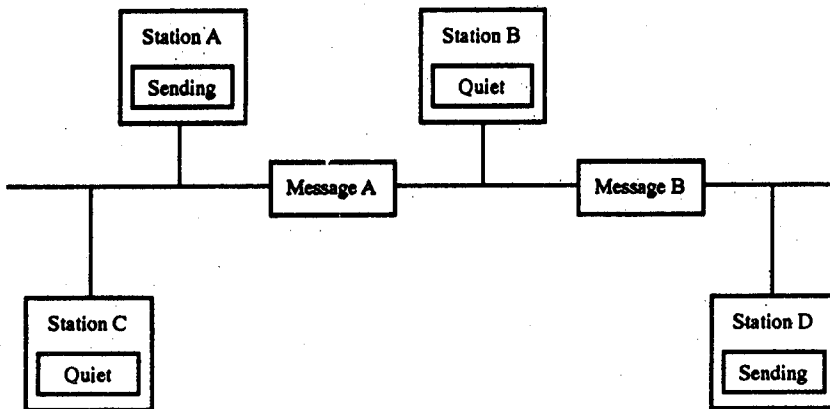


Fig. 4.6 : Contention

Messages to be transmitted are converted to packets and are sent when ready, without verifying the availability of the channel. When transmission of a packet from one workstation overlaps with that of another, collision occurs. Colliding packets, with their embedded message, are destroyed.

While the basic contention protocol makes no provision for knowing if another message is already underway, it does provide for acknowledging the successful receipt

of a packet. If the originating workstation does not receive an acknowledgement, it assumes that transmission was garbled or destroyed. The sending workstation waits a random amount of time and then retransmits the packet. The waiting time must be random or the same messages will collide repeatedly.

In some cases, the receiving workstation receives only part of a packet. The receiver may then return a negative acknowledgement to the originator, requesting retransmission.

More than any other network, the contention network is characterised by **bursty traffic**: the time interval needed for each transmission is short in relation to the interval between transmission.

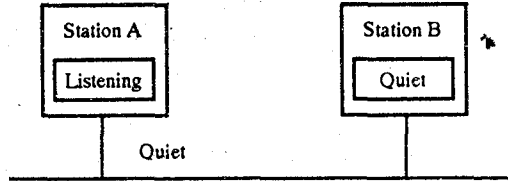
CONTENTION EVALUATION FACTORS

Smooth functioning of a contention-based network is dependent upon high availability of the transmission media and a low collision rate. The network is characterised by the following:

- **Message length:** Messages are divided into short packets in order to reduce the amount of data that must be rebroadcast after collisions. Normally, the original message is also fairly short.
- **Traffic volume:** Contention protocols are designed for networks with low traffic volume, that is, one with few time. Low traffic volume implies a limited number of attached workstations.
- **Network length constraints:** The longer the network, the greater the chance of collision. Contention networks are limited by the time needed for a signal to travel the length of the transmission media and have an acknowledgement returned (that is, propagation delay).
- **Performance:** Contention networks are most effective under light to medium load. Performance under those conditions is excellent. Under heavy load, a contention network tends to be unstable, with service rapidly degrading.
- **Overhead:** Contention networks have high overhead because of collisions and the need to acknowledge the successful receipt of messages.
- **Access Delay:** Delay on the network is generally moderate to long, depending on traffic. Delay under heavy load can be significantly higher than load alone would seem to dictate.
- **Station Failures:** Because operation of the network is not dependent on the presence or absence of any one workstation, failure of a workstation inconveniences only its users. Rarely does failure of a single station disrupt service on the whole network.
- **Expansion:** Addition of new workstations is relatively easy because to be included in the network, the workstation simply must recognise its own unique address. Expansion may be achieved with minimal disruption of the network.

Refinements on contention procedures are used by many of the current microcomputer local networks. These refinements are: Carrier Sense Multiple Access (CSMA) in Figure 4.7; Carrier Sense Multiple Access with Collision Detection (CSMA/CD); and Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA).

Step 1 :



Step : 2

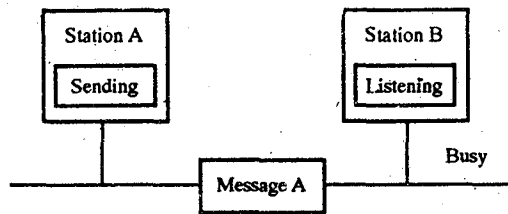


Fig. 4.7 : CSMA

Each will be discussed in separate sections. For all practical purposes, characteristics of these refinements are identical to the characteristics of simple contention.

CARRIER SENSE MULTIPLE ACCESS

Carrier Sense Multiple Access (CSMA) is a polite staff meeting, with colleagues beginning to talk only when no one else is talking. As in simple contention, members of the network share a single communication channel (see Figure 4.7).

Before information is sent, the workstation "listens" - usually on a secondary frequency - to sense whether any other workstation is using the primary transmission channel (the "carrier"). Only when the line is clear will the workstation transmit.

If a workstation becomes ready to transmit while another workstation is active, it detects the signal passing on the cable and does not send its message until the current transmission is complete. For microcomputer networks the waiting station has two options, depending on system design:

1. It can continually sense the channel while waiting for the busy signal to cease and then transmit immediately. This is called **persistent carrier sense**, as the terminal actively waits to seize the channel as soon as it becomes free. If other workstations are equally persistent, a collision may occur immediately after the busy signal ceases.
2. Alternatively, if the channel is sensed busy, the terminal reschedules its

transmission for a later time, using a random delay, and tries again **nonpersistent carrier sense**. Fewer collisions occur, resulting in higher throughput. However, delays may be slightly longer, at least in networks with low channel utilisation.

In addition to transmitting its message on the main channel, the active workstation broadcasts a carrier-sense signal on the secondary channel to inform other workstations that the line is busy.

After transmitting, the workstation waits for an acknowledgement, indicating that transmission was successful. If no acknowledgement is received or if a negative acknowledgement (indicating unsuccessful transmission) is received, the workstation assumes a collision has occurred. The workstation then waits a random amount of time before starting the process again.

In a CSMA network, collision between transmitting workstations is still inevitable. A "ready" signal on the secondary channel does not necessarily mean that the network is free of other traffic. Because of the length of time required for a signal to travel the channel (propagation delay), two or more workstations may sense an idle line simultaneously, and thus both attempt to transmit at the same time. If the propagation delay is short, the information the workstation hears by monitoring the channel is sufficiently current to permit a useful decision. The probability of success will be significantly higher than in simple contention. If, however, the information is old, that is, the propagation delay is long, CSMA offers only slight improvement over plain contention.

The secondary channel which carries the busy tone does require some bandwidth. This bandwidth is generally minimal. There is a brief delay involved in sensing the busy signal.

CARRIER SENSE MULTIPLE ACCESS WITH COLLISION DETECTION

Carrier Sense Multiple Access with Collision Detection (CSMA/CD) provides the proper etiquette for the times when polite colleagues inadvertently start talking at the same instant. At our theoretical staff meeting, both speakers would stop and wait for the other to continue. The one who resumes first would have the floor.

In CSMA/CD, in addition to sensing whether the transmission channel is in use before beginning to transmit, workstations monitor the link during transmission. When a collision is detected, transmission is halted.

As in the other contention-based protocols, the message is retransmitted after a brief interval. For CSMA/CD, the interval may be either random or pre-defined as a unique period for each workstation. Because of the ability to listen before and during transmission, the number of collisions is relatively low. Successive collisions between the same workstations is rare. Additionally, since transmission ceases as soon as a collision is detected, less delay occurs.

CARRIER SENSE MULTIPLE ACCESS WITH COLLISION AVOIDANCE

Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) is analogous to several people wishing to contribute information at a meeting, and therefore all raising their hands at the same time. At the meeting, the moderator selects the next speaker. In CSMA/CA, the protocol determines who speaks next.

A station with a message to transmit monitors the medium and waits for the line to be available. When the channel is clear, the workstation signals its intention to broadcast. If multiple workstations are waiting, the order of precedence is determined by a pre-established table.

Just as human meetings tend to be biased in favour of allowing the main speaker or resident expert the greatest opportunity to talk, most CSMA/CA schemes are biased in favour of the lower-numbered workstations. That is, after any transmission, the workstation designated as first by the table has the right to transmit. If it has no message to send, or if it fails to transmit within a pre-defined time for any reason, the next workstation has the chance, and so on.

Once any workstation transmits, the network begins again at the top of the list of precedence. Refinements are possible. Some implementations are designed to avoid having the network dominated by any one workstation: a station which has just transmitted may not be allowed to transmit again until all other stations have had an opportunity.

To accommodate multmessage dialogue, the workstation receiving a message may have the first right to transmit. If it does reply, the original sender would again have a chance. Unfortunately, two workstations may seize the medium, with one station sending messages and the other replying.

In the case when no workstation has a message to transmit, the network may reinitialise, starting again with the first workstation. In other cases, the network may enter a free-for-all period, in which the first workstation to transmit gains the channel. During the contention period, collisions are allowed.

POLLING

Polling (see Figure 4.8) involves the central control of all workstations in a network. The central, or primary, workstation acts like a teacher going down the rows of the classroom asking each student for homework. When one student has answered, the next is given a chance to respond.

A polling network contains two classes of workstations, the primary workstation (also termed the central controller or server), and the multiple secondary workstations connected to it. A buffer that can temporarily store messages is associated with each secondary workstation. When a workstation has information to transmit, the data is passed to the buffer. The message is held until the workstation is polled by the central controller.

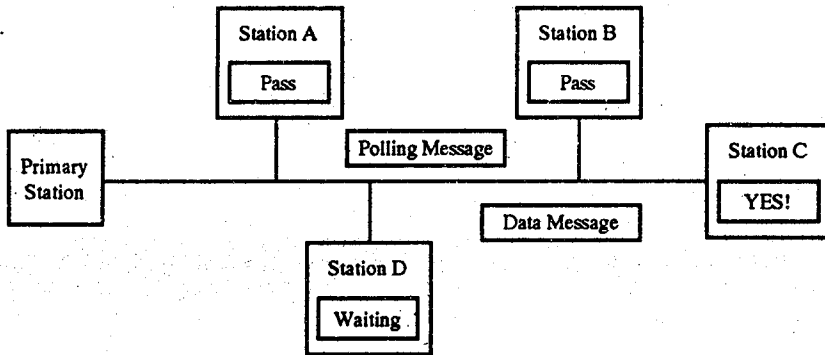


Fig. 4.8 : Polling

The primary workstation queries each secondary in turn to determine if it has a message to transmit. If the answer is affirmative, the workstation is either given permission to transmit immediately or assigned a transmission time. The amount of time a workstation may have in which to transmit once channel access is gained is determined by system parameters.

If the workstation does not have data to transmit, it still must respond with a short control message. Rather than returning a message to the primary, some networks allow a polled workstation with no data to send to pass the polling signal to the next secondary station.

Each time a workstation is polled, the primary workstation must wait for a response to be returned. After a workstation responds, the next station is polled. The primary workstation determines which workstation has access to the network at any one time.

There are two possibilities for the path of a message from source to destination workstation:

1. All messages may be required to pass to the central workstation, which routes them to their destinations.
2. Messages may be sent directly from the originator to their destinations.

In either case, communication between workstations is possible only under the direction of the polling computer.

Variations on polling tend to be concerned with how often workstations are queried. The basic protocol calls for all stations to have equal opportunity to broadcast. This is not always so. In some networks, workstations considered to be very active or to have priority may be polled several times within a single cycle. In other cases, an inactive device may not be polled every cycle. A third alternative is that the frequency with which individual workstations are polled may be varied to reflect their current activity level.

Polling techniques can be said to maintain a tighter control over the network than do contention-based protocols.

POLLING EVALUATION FACTORS

The polling network is characterised by the following:

Message Length: Allowable message length tends to be longer than in contention networks, as no workstation can pre-empt the network through frequent, lengthy messages. However, if all workstations have long messages, the transmission delay is high.

Traffic Volume: Polling networks support moderate to high traffic volume, limited primarily by the need to wait for permission to transmit. Direct conflict for time to transmit, as in contention techniques, is avoided. Therefore, a large number of workstations can share the common channel.

Network Length Constraints: The distance between workstations and the overall length of the network is limited by the transmission medium, rather than by the polling protocol. As in any network, the greater the length, the longer the time required for messages to travel between sending and receiving stations.

Performance: Polling networks perform best under moderate load. Under heavy load, transmission delays may become unacceptably long. Polling is inefficient for networks with light loads. In the extreme case, most network traffic may be comprised of polling signals and negative acknowledgements, rather than actual messages.

Overhead: Administrative overhead on a polling network is high. The query and response use a measurable part of the total network capacity. Furthermore, in many polling networks, the server unit cannot be used as a workstation.

Access Delay: Delays in most polling networks are relatively long. In most implementations, a workstation is polled only once each cycle. If the network is very large, delay may be unacceptably long.

Station Failures: Little or no disruption of the network is caused by a failed secondary workstation. The inactive workstation is "invisible" to the network; that is, it simply fails to respond when polled. However, if the central workstation fails, all communication on the network ceases.

Expansion: In order for the network to be expanded, the primary workstation must be informed and the order of polling revised to reflect the addition. Therefore, expansion is more complex than expansion of a contention network.

TOKEN PASSING

Token passing can be seen as the children's game of hot potato in reverse. Like the players, the network continuously circulates a special bit pattern known as a token. Rather than being out if you are holding the object being passed, holding the token confers the right to communicate. Only the workstation holding the token can put a message onto the network. Control of the network is decentralised.

Each token contains network information, comprising of a header, a data field and a trailer (see Figure 4.9).

Empty Token :



In Use :

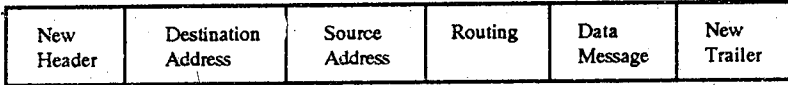


Fig. 4.9 : Token

When a workstation that wants to transmit receives an empty token, it inserts routing information, inserts the data and sends the token on a complete circuit of the network.

The workstation holding the token may transmit messages up to a specified maximum length. If it does not have anything to communicate, it passes the token to the next station in the network. Figure 4.10 illustrates token passing.

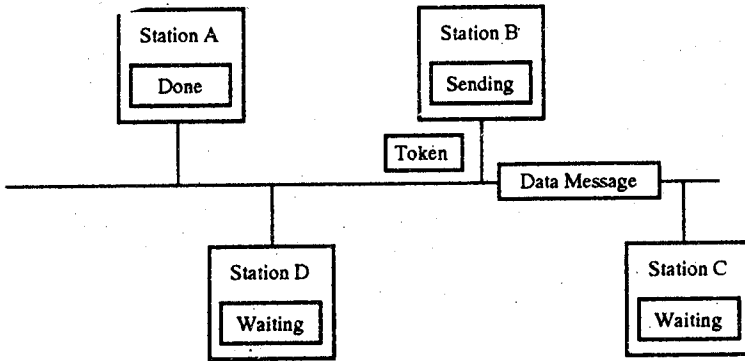


Fig. 4.10 : Token Passing

In most token passing networks, the token passes from one workstation to the one immediately adjacent. However, the token may pass from workstation to workstation in a sequence established at network implementation time, without requiring nodes to be physically adjacent. In such implementations the workstation knows the address of the next workstation to receive the token, as well as its own address.

All the workstations on the network read the address in an occupied token; if it is for a different workstation, it is passed on unchanged and unread. At the destination, the receiving workstation reads the message, marks the token as copied or rejected, and continues passing it. Only the workstation which has placed a particular

message onto the network may remove that message. If a workstation is disassociated from the network, it simply does not read the message.

When the token returns to its original sender, the message is removed. The token is marked as empty and forwarded to the next workstation. The sender can either save the message, in order to compare it with the original data as part of a network reliability monitoring scheme, or discard it. Acknowledgement or lack thereof notifies the sender of the status of the message. If the receiving workstation is absent from the network, no acknowledgement will be received. If the message has been rejected because it is garbled, the sending workstation can retransmit it.

Complex error-recovery protocols are required to recognise and recover from events, such as lost or garbled tokens, the failure of a workstation to forward the token; or any time that no token exists, such as at network start-up. In such cases, a method of generating a token and beginning to circulate it must be specified. Controlled contention or a priority access scheme based on workstation address may be used to re-establish token ownership.

Token passing ensures relatively tight control over the network. The elimination of inefficiencies caused by collisions between contending units is a major advantage for users.

Figure 4.11 gives comparative performance of token rings with that of CSMA/CD. Table 4.1 lists advantages and disadvantages of CSMA/CD vis-a-vis token bus.

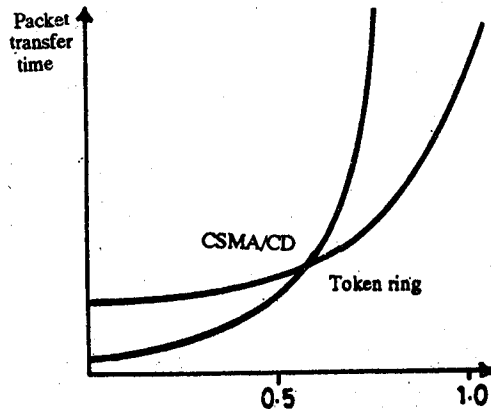


Fig. 4.11 : Performance of token rings and CSMA/CD

TABLE 4.1 : CSMA/CD versus Token Bus

Advantages	Disadvantages
	<i>CSMA/CD</i>
Simple algorithm	Collision detection requirement
Widely used	Fault diagnosis problems
Fair access	Minimum packet size
Good performance at low to medium load	Poor performance under very heavy load

Biased to long transmissions

Token Bus

Excellent throughput performance
Tolerates large dynamic range
Regulated access

Complex algorithm
Unproven technology

TOKEN PASSING EVALUATION FACTORS

- **Message Length:** Token passing can accommodate moderate to long messages. Since messages are embedded within the token, multiple types of data may be handled.
- **Traffic Volume:** Traffic on a token passing network can be quite high. Because each workstation may broadcast a single message on each turn, availability of the network is equitable. There is little possibility that any one workstation will claim more than its fair share of network capacity.
- **Network Length Constraints:** Token passing networks are limited primarily by the transmission medium; not by the protocol.
- **Performance:** Token passing networks perform well under most conditions. Given a moderate to high traffic load when the network is in almost constant use, performance is excellent.
- **Overhead:** Administrative overhead on token networks is comparatively high.
- **Access Delay:** Under any given volume of traffic, line delays on the token network tend to be constant and predictable. Heavy traffic creates a moderate delay. Under moderate traffic conditions, delay is short.
- **Station Failure:** In older token rings, failure of any one workstation blocked all transmission. Recently developed token rings have eliminated this problem.
- **Expansion:** Expanding a token passing network is a complex process which may involve re-wiring the network to include the new workstations and identifying a new token circulation pattern. During expansion, service on the network may be disrupted for extended periods.