- **MAC addresses :** All Ethernet NICs (network interface cards) come with a preassigned 48-bit network address. This address is made up of a 24-bit number that identifies the vendor of the card and a 24-bit number assigned by the vendor to identify the card. The address is globally unique and is called the MAC address because it defines a node on the network at the MAC (Media Access Control) layer.

11.4.2 Ethernet Network Elements

Ethernet LANs consist of network nodes and interconnecting media. The network nodes fall into two major classes:

- . **Data terminal equipment (DTE)**—Devices that are either the source or the destination of data frames. DTEs are typically devices such as PCs, workstations, file servers, or print servers that, as a group, are all often referred to as end stations.
- **Data communication equipment (DCE)**—Intermediate network devices that receive and forward frames across the network. DCEs may be either standalone devices such as repeaters, network switches, and routers, or communications interface units such as interface cards and modems.

The current Ethernet media options include two general types of copper cable : unshielded twisted-pair (UTP) and shielded twisted-pair (STP), plus several types of optical fiber cable.

11.4.3 The Basic Ethernet Frame Format

The IEEE 802.3 standard defines a basic data frame format that is required for all MAC implementations, plus several additional optional formats that are used to extend the protocol's basic capability. The basic data frame format contains the seven fields shown in Figure 11.4.

7	1	6	6	2	46-1500 bytes	4
Pre	SFD	DA	SA	Length Type	Data unit + pad	FCS

Fig. 11.4 The basic IEEE 802.3 Ethernet MAC Data Frame for 10/100Mbps Ethernet

- **Preamble (PRE)**—Consists of 7 bytes. The PRE is an alternating pattern of ones and zeros that tells receiving stations that a frame is coming, and that provides a means to synchronize the frame-reception portions of receiving physical layers with the incoming bit stream.
- **Start-of-frame delimiter (SFD)**—Consists of 1 byte. The SFD is an alternating pattern of ones and zeros, ending with two consecutive 1-bits indicating that the next bit is the left-most bit in the left-most byte of the destination address.
- **Destination address (DA)**—Consists of 6 bytes. The DA field identifies which station(s) should receive the frame. The left-most bit in the DA field indicates whether the address is an individual address (indicated by a 0) or a group address (indicated by a 1). The second bit from the left indicates whether the DA is globally administered

(indicated by a 0) or locally administered (indicated by a 1). The remaining 46 bits are a uniquely assigned value that identifies a single station, a defined group of stations, or all stations on the network.

- **Source addresses (SA)**—Consists of 6 bytes. The SA field identifies the sending station. The SA is always an individual address and the left-most bit in the SA field is always 0.
- **Length/Type**—Consists of 4 bytes. This field indicates either the number of MAC-client data bytes that are contained in the data field of the frame, or the frame type ID if the frame is assembled using an optional format. If the Length/Type field value is less than or equal to 1500, the number of LLC bytes in the Data field is equal to the Length/Type field value. If the Length/Type field value is greater than 1536, the frame is an optional type frame, and the Length/Type field value identifies the particular type of frame being sent or received.
- **Data**—Is a sequence of n bytes of any value, where n is less than or equal to 1500. If the length of the Data field is less than 46, the Data field must be extended by adding a filler (a pad) sufficient to bring the Data field length to 46 bytes.
- **Frame check sequence (FCS)**—Consists of 4 bytes. This sequence contains a 32-bit cyclic redundancy check (CRC) value, which is created by the sending MAC and is recalculated by the receiving MAC to check for damaged frames. The FCS is generated over the DA, SA, Length/Type, and Data fields.

11.4.4 Frame Transmission

Whenever an end station MAC receives a transmit-frame request with the accompanying address and data information from the LLC sublayer, the MAC begins the transmission sequence by transferring the LLC information into the MAC frame buffer.

- The preamble and start-of-frame delimiter are inserted in the PRE and SFD fields.
- The destination and source addresses are inserted into the address fields.
- The LLC data bytes are counted, and the number of bytes is inserted into the Length/Type field.
- The LLC data bytes are inserted into the Data field. If the number of LLC data bytes is less than 46, a pad is added to bring the Data field length up to 46.
- The FCS value is generated over the DA, SA, Length/Type, and Data fields and is appended to the end of the Data field.

After the frame is assembled, actual frame transmission will depend on whether the MAC is operating in half-duplex or full-duplex mode.

The IEEE 802.3 standard currently requires that all Ethernet MACs support half-duplex operation, in which the MAC can be either transmitting or receiving a frame, but it cannot be doing both simultaneously. Full-duplex operation is an optional MAC capability that allows the MAC to transmit and receive frames simultaneously.

Half-Duplex Transmission—The CSMA/CD Access Method

The CSMA/CD protocol was originally developed as a means by which two or more stations could share a common media in a switch-less environment when the protocol does not require central arbitration, access tokens, or assigned time slots to indicate when a station will be allowed to transmit. Each Ethernet MAC determines for itself when it will be allowed to send a frame.

The CSMA/CD access rules are summarized by the protocol's acronym:

- **Carrier sense**—Each station continuously listens for traffic on the medium to determine when gaps between frame transmissions occur.
- **Multiple access**—Stations may begin transmitting any time they detect that the network is quiet (there is no traffic).
- **Collision detect**—If two or more stations in the same CSMA/CD network (collision domain) begin transmitting at approximately the same time, the bit streams from the transmitting stations will interfere (collide) with each other, and both transmissions will be unreadable. If that happens, each transmitting station must be capable of detecting that a collision has occurred before it has finished sending its frame.

Each must stop transmitting as soon as it has detected the collision and then must wait a quasirandom length of time (determined by a back-off algorithm) before attempting to retransmit the frame.

Binary Exponential Backoff Algorithm

When there is a collision, the stations involved in the collision will execute the binary exponential backoff algorithm to reduce the possibility of futher collisions.

1. When a collision is detected, the sender generates a noise burst to insure that all stations recognize the condition and aborts the transmission.
2. Wait 0 or 1 contention period (2 τ, i.e., 2 end-to-end propagation time) before attempting transmission again.
3. If another collision is detected, wait 0, 1, 2 or 3 contention period. And repeat the protocol.
4. In general, wait between 0 and $2^r - 1$ contention periods, where r is the number of collisions encountered.
5. Finally, freeze interval at 1023 contention periods after 10 attempts and give up (report failure) after 16 attempts.

11.4.5 Objective

An Ethernet network consisting of 4 computers is illustrated below. The objective is to send a data packet from Computer 1 to Computer 2 in accordance with the Ethernet's CSMA/CD protocol.

The CSMA/CD protocol is described in the flow chart below. The first step in sending a packet is to check to see if any other computer is in the process of sending a packet (Other

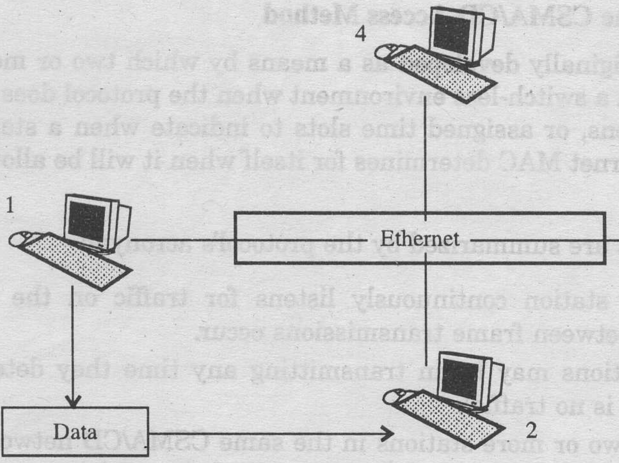


Fig. 11.5 Transmission of data from computer 1 to computer 2

carrier present?). If another computer is sending, then Computer 1 will wait until the network is available. If the network is available, then Computer 1 will start transmitting it's packet

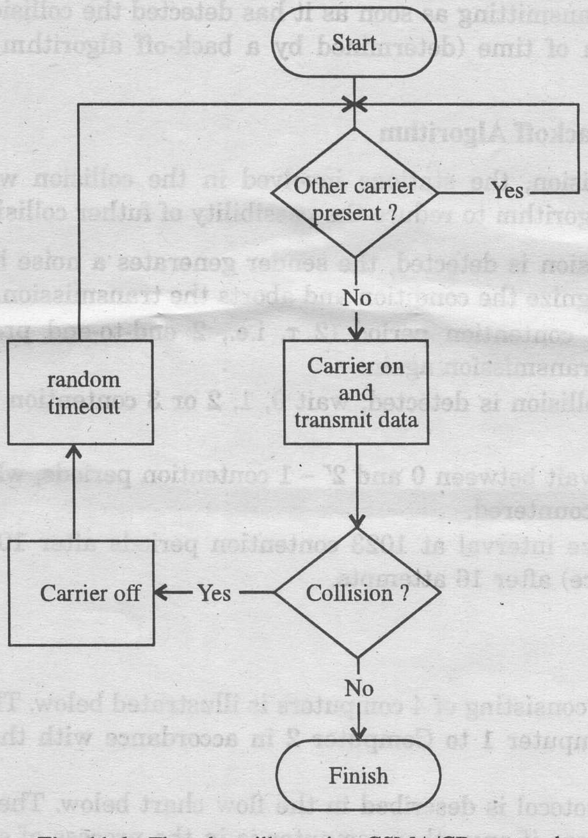


Fig. 11.6 Flowchart illustrating CSMA/CD protocol

(Carrier on and transmit data). It is possible that while Computer 1 was waiting for the network to become available, another computer was doing the same thing. In this event the other computer, which is following the same protocol as Computer 1, will start transmitting at the same time as Computer 1. This event is called a collision, and each computer is able to detect a collision when it occurs. The CSMA/CD response to a collision is to shut down the transmitter immediately, and then wait for a random period of time before trying again. This is analogous to two people accidentally starting to talk at the same time and their reaction to that "collision".

11.4.6 Specifications of Ethernet

Ethernet has evolved over the years to include a number of popular specifications. These specifications are due in part to the media variety, which they employ, such as coaxial, twisted-pair, and fiber-optic cabling. The original Ethernet supports a data rate of 10 megabits per second (Mbps) and specifies these possible physical media:

- 10BASE-2 (Thinwire coaxial cable with a maximum segment length of 185 meters).
- 10BASE-5 (Thickwire coaxial cable with a maximum segment length of 500 meters).
- 10BASE-F (Optical fiber cable).
- 10BASE-T (Ordinary telephone twisted pair wire).
- 10BASE-36 (Broadband multi-channel coaxial cable with a maximum segment length of 3,600 meters).

This designation is an IEEE shorthand identifier. The "10" in the media type designation refers to the transmission speed of 10 Mbps. The "BASE" refers to baseband signalling, which means that only Ethernet signals are carried on the medium (or, with 10BASE-36, on a single channel). The *"T"* represents twisted-pair; the *"F"* represents fiber optic cable; and the "2", "5", and "36" refer to the coaxial cable segment length (the 185 meter length has been rounded up to "2" for 200).

There are a number of new Ethernet standards that define the 100 Mbps Fast Ethernet system. These operate over twisted-pair and fiber-optic media.

The following table summarizes ethernet specifications:

Name	Cable	Max. segment	Nodes/ Seg.	Advantages
10Base5	Thick coax	500 m	100	Good for local area backbone
10Base2	Thin coax	200 m	30	Inexpensive
10Base-T	Twisted pair	100 m	1024	Easy maintenance
10Base-F	Fiber optics	2000 m	1024	Inter-building connection

10Base-T is now the most popular cabling scheme used for local area network. It is :

- easy to maintain.

- reduce the logical distance between stations (all connected at the hup), thus reducing the value of τ.

Fast Ethernet

The original Ethernet was based on a bus topology. Each station connects to the bus anywhere needed with a minimum physical distance one meter apart. Later a better technology is used called 10Base-T for 10 M bps unshielded twisted pair wire with a hub or router in the communications room (essentially Star topology).

Now the new Fast Ehternet took this one step futher, making it 100 M bps network with virtually no change in protocols and wirings. The Fast Ethernet protocol specification was completed by IEEE in 1995.

11.5 Token Ring (IEEE 802.5)

Token Ring is a LAN protocol defined in the IEEE 802.5 where all stations are connected in a ring and each station can directly hear transmissions only from its immediate neighbor. The Token Ring protocol is the second most widely used protocol on local area networks after Ethernet. The protocol deals with the problem of collision, which is defined, as a state where two stations transmit at the same time. In order to avoid the situation of collision there was a need to control the access to the network. This kind of control is possible by the use of a control (permission) called token. The token is passed from one station to another according to a set of rules. The ring consists of ring stations and transmission medium. Data travels sequentially from station to station. Only the station in possession of the token is allowed to transmit data. Each station repeats the data, checks for errors, and copies the data if appropriate. When the data is returned to the sending station, it removes it from the ring. The token Ring protocol supports priorities in transmission. It is implemented setting the priority bits in the Token Ring Frame.

Token Ring is a first and second layer protocol in the OSI seven layer model. The First release of Token Ring version was capable of 4Mbs data transmission rate; the transmission rate was improved later to 16Mbs. Token Ring can be operated on the following media's:

- Unshielded Twisted Pair (UTP).
- Shielded Twisted Pair (STP): Allowing a Max. of 260 stations at 16Mps rings.
- Coaxial cable (Thin\Thick\Broadband).
- Fiber Optics.

What is a Token?

Each token frame is three bytes long:

1. A 1-byte delimiter signals the start of a frame.
2. A 1-byte access control field contains priority information.
3. A 1-byte frame control field distinguishes data frames from control frames.

Note that token frames differ from regular frames by only 1 bit in the frame control field. Thus seizing the token consists of flipping a single bit during the copy operation. After seizing the token, a station may transmit data for no longer than the token holding time, typically 10 ms.

11.5.1 Features of Token Ring

The term "Token Ring" does not necessarily mean that the network is based on the ring topology. Most Token Ring networks use a star topology with central access points called multistation access units (MSAU). Two types of network cable, unshielded twisted-pair (UTP) and shielded twisted-pair (STP) determine the distance and number of workstations on the network. STP provides the greatest flexibility, but costs more. Each computer has a maximum distance of 100 meters (328 feet) from the MSAU when shielded wire is used and 45 meters (148 feet) when unshielded wire is used. Every MSAU can support up to 260 workstations using shielded wire, or up to 72 workstations using unshielded wire. Each ring can also support up to 33 MSAUs.

Token Ring networks can operate at 4 Mbps or 16 Mbps. Most hardware today supports 16 Mbps and can be configured to operate at any speed. With networks of heavy traffic, token passing is the most efficient network architecture. This is due to the fact that the token-passing technology eliminates network collisions because stations are not allowed to simultaneously transmit on the network. However, Ethernet is more effective on networks with light to moderate amounts of traffic.

Ring Benefits

- High reliability, the Ring can continue normal operation despite any single fault.
- Bypassing inactive stations.
- Effective use, 95% in Token Ring only whilst 30-40% in Ethernet.
- Excellent traffic handling (17.8 kb in TR, only 15kb in Ethernet.).
- Large maximum frame length.
- High bandwidth efficiency. 70% in Token Ring, 30% in Ethernet.
- Many media choices: UTP, STP, coax, fiber.
- Supports transmission priority.

11.5.2 Frame Format of Token Ring

Token Ring and IEEE 802.5 support two basic frame types : tokens and data/command frames. Tokens are 3-bytes in length and consist of a start delimiter, an access control byte, and an end delimiter. Data/command frames vary in size, depending on the size of the Information field. Data frames carry information for upper-layer protocols, while command frames contain control information and have no data for upper-layer protocols. Both formats are shown in Figure 11.7.

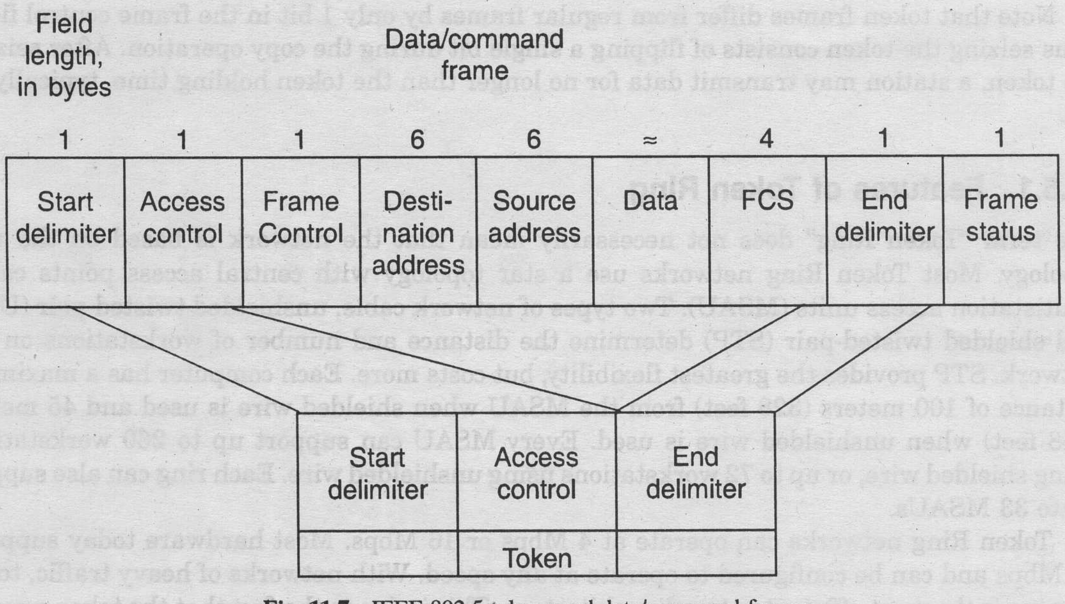


Fig. 11.7 IEEE 802.5 tokens and data/command frames

Token Frame Fields

The three token frame fields illustrated in Figure 11.7 are explained below:

- **Start delimiter**—Alerts each station of the arrival of a token (or data/command frame). This field includes signals that distinguish the byte from the rest of the frame by violating the encoding scheme used elsewhere in the frame.
- **Access-control byte**—Contains the Priority field (the most significant 3-bits) and the Reservation field (the least significant 3-bits), as well as a token bit (used to differentiate a token from a data/command frame) and a monitor bit (used by the active monitor to determine whether a frame is circling the ring endlessly).
- **End delimiter**—Signals the end of the token or data/command frame. This field also contains bits to indicate a damaged frame and identify the frame that is the last in a logical sequence.

Data/Command Frame Fields

Data/command frames have the same three fields as Token Frames, plus several others.

The Data/command frame fields illustrated in Figure 11.7 are explained below:

- **Start delimiter**—Alerts each station of the arrival of a token (or data/command frame). This field includes signals that distinguish the byte from the rest of the frame by violating the encoding scheme used elsewhere in the frame.
- **Access-control byte**—Contains the Priority field (the most significant 3-bits) and the Reservation field (the least significant 3-bits), as well as a token bit (used to differentiate a token from a data/command frame) and a monitor bit (used by the active monitor to determine whether a frame is circling the ring endlessly).

- **Frame-control bytes**—Indicates whether the frame contains data or control information. In control frames, this byte specifies the type of control information.
- **Destination and source addresses**—Consists of two 6-byte address fields that identify the destination and source station addresses.
- **Data**—Indicates that the length of field is limited by the ring token holding time, which defines the maximum time a station can hold the token.
- **Frame-check sequence (FCS)**—Is filled by the source station with a calculated value dependent on the frame contents. The destination station recalculates the value to determine whether the frame was damaged in transit. If so, the frame is discarded.
- **End Delimiter**—Signals the end of the token or data/command frame. The end delimiter also contains bits to indicate a damaged frame and identify the frame that is the last in a logical sequence.
- **Frame Status**—Is a 1-byte field terminating a command/data frame. The Frame Status field includes the address-recognized indicator and frame-copied indicator.

11.5.3 Ring Management

The mechanism of the network operation is consider being the mechanism in the steady state, but before this can take place the ring must be set up. Also if a new station wishes to join an operational ring it must first go through an initialization procedure to ensure that it does not interfere with the correct functioning of the current active ring. Also, during normal operation it is necessary for each active station on the ring to monitor its correct operation and if something is not working well it must take some action to try reestablish a correctly functioning ring. Those functions and others which meant to preserve the correct ring operation are called ring management.

There are two types of stations in the ring :

- Active Monitor (AM) station
- Standby Monitor (SBM) stations

There is only one Active Monitor station per ring. The Active Monitor is the ring manager. All other stations on the ring are Standby Monitor stations. Any station on the ring can be Active Monitor. The Active Monitor is chosen during a process called "Claim Token Process" and after the Active Monitor is chosen all other stations become "Standby Monitors" (SBM).

Active Monitor Duties

- One function of the active monitor is the removal of continuously circulating frames (orphan frames), from the ring. When a sending device fails, its frame may continue to circulate the ring. This can prevent other stations from transmitting their own frames and result in the network locking up. The active monitor can detect these frames, remove them from the ring and generate a new token.
- If a corrupted frame appears on the ring the active monitor can detect it, as it will have an invalid frame format, or an invalid checksum. In this type of case the active monitor drains the token and issues a new one.

- As there are 24-bits in a token the ring must be big enough to hold all of them. On each pass through the stations the token is delayed by one bit. If this one bit delay plus the delay due to the cable length adds up to less than 24-bits, the active monitor inserts extra delay bits so that the token can circulate correctly.

Standby Monitor Duties:

- Check for the presence of an active monitor at regular intervals.
- Go through a token-claiming process to determine a new active monitor if, for any reason the active monitor fails.

Priority System

Token passing is often called deterministic, because you can calculate the maximum time before a workstation can grab the token and begin to transmit. However, you can assign priorities to certain devices that will use the network more frequently. The token itself has a priority value that can be changed. The token is captured by a workstation and an information frame is sent out on the network in place of the token. A workstation with a priority equal to or higher than the priority value in the information frame can then seize the token. When the original token is put back on the network after the workstation is through transmitting, the priority value must be set back to the original value.

Fault Management

A Token Ring network is much more reliable for a number of reasons:

- **Active monitoring :** The active monitor is most likely the first station to be brought up on the Token Ring network. The active monitor performs maintenance functions, such as removing continuously circulated rings and generating new tokens.
- **Beaconing :** *Beaconing* refers to a frame that is sent around the network when a serious network problem occurs. This frame is sent to the nearest active upstream neighbor (NAUN) until the frame stops. When the frame stops, beaconing determines that the next upstream neighbor has a problem. If the station that initiated the beacon receives its own original beacon, it assumes the link failure has been fixed, and then regenerates the token. This is a more effective troubleshooting method compared to the difficulty of determining problems on a bus topology backbone, which gives no clear indication of which station has malfunctioned. However, beaconing is similar to the bus topology in that one break in the link can cause the entire network to stop.
- **Selective removal of stations :** More advanced Token Ring hardware can be used to isolate the failed port and automatically remove it from the ring. This ensures that a faulty station does not affect the rest of the network.

11.5.4 Objective

The objective is to send a data packet from Computer 2 to Computer 4. There are two types of packet: Free Tokens and Busy Tokens. Free tokens contain no data. Busy tokens contain data to be conveyed by the network. When there is no other activity in the network a free token is passed from computer to computer around the ring.

Step 1 : Computer 1

Let us begin our description with computer 1. It has no data to send so when it receives the free token from computer 4, it passes it to computer 2.

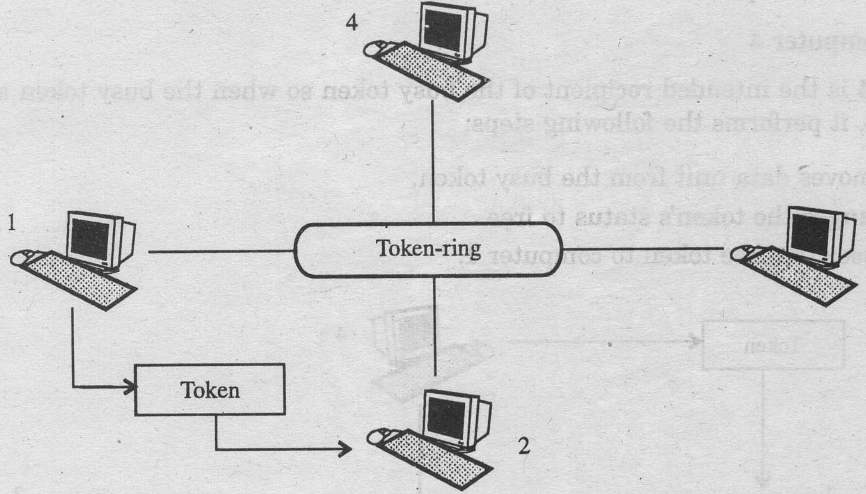


Fig. 11.8 Free token passed from computer 1 to computer 2

Step 2 : Computer 2

Computer 2 does have data to send. It performs the following steps:

- Changes the token status to busy.

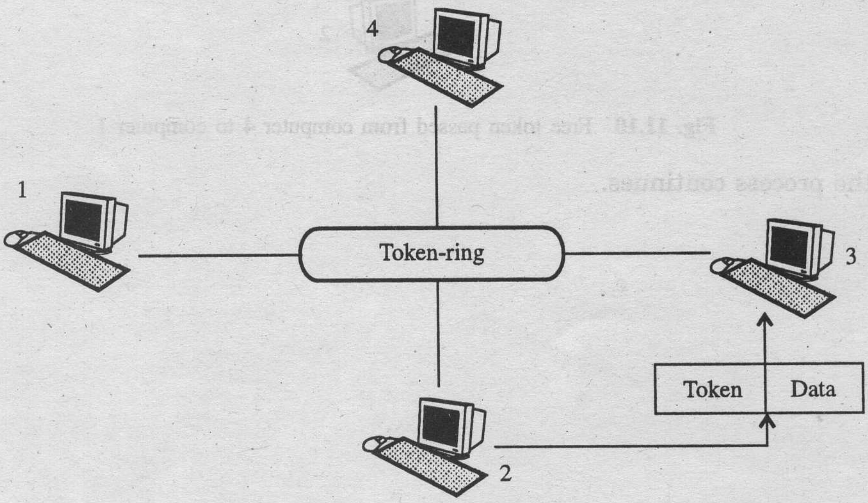


Fig. 11.9 Busy token and data passed from computer 2 to computer 3

- Attaches the data unit to the busy token.
- Passes the busy token and data to computer 3.

Step 3 : Computer 3

Computer 3 has no data of its own to send so when it receives the busy token from computer 4 it passes it to computer 4.

Step 4 : Computer 4

Computer 4 is the intended recipient of the busy token so when the busy token arrives from computer 3, it performs the following steps:

- Removes data unit from the busy token.
- Changes the token's status to free.
- Passes the free token to computer 1.

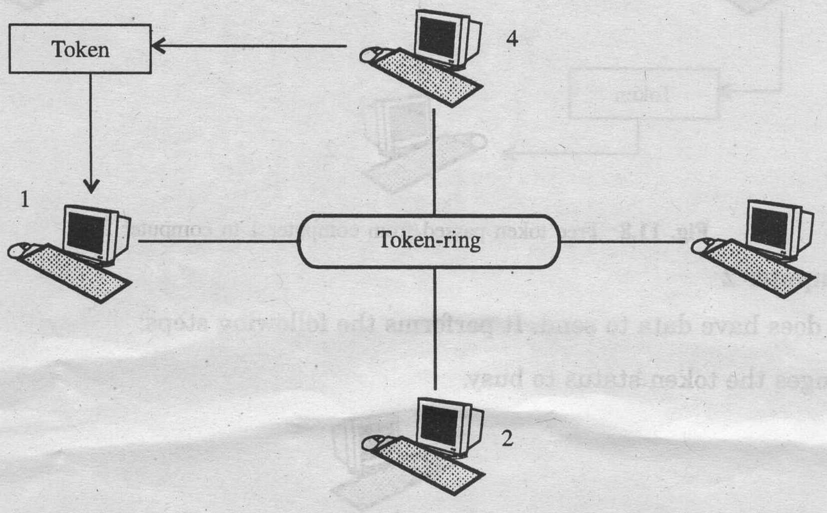


Fig. 11.10 Free token passed from computer 4 to computer 1

And the process continues.

The following flowchart illustrates token access method in a ring network.

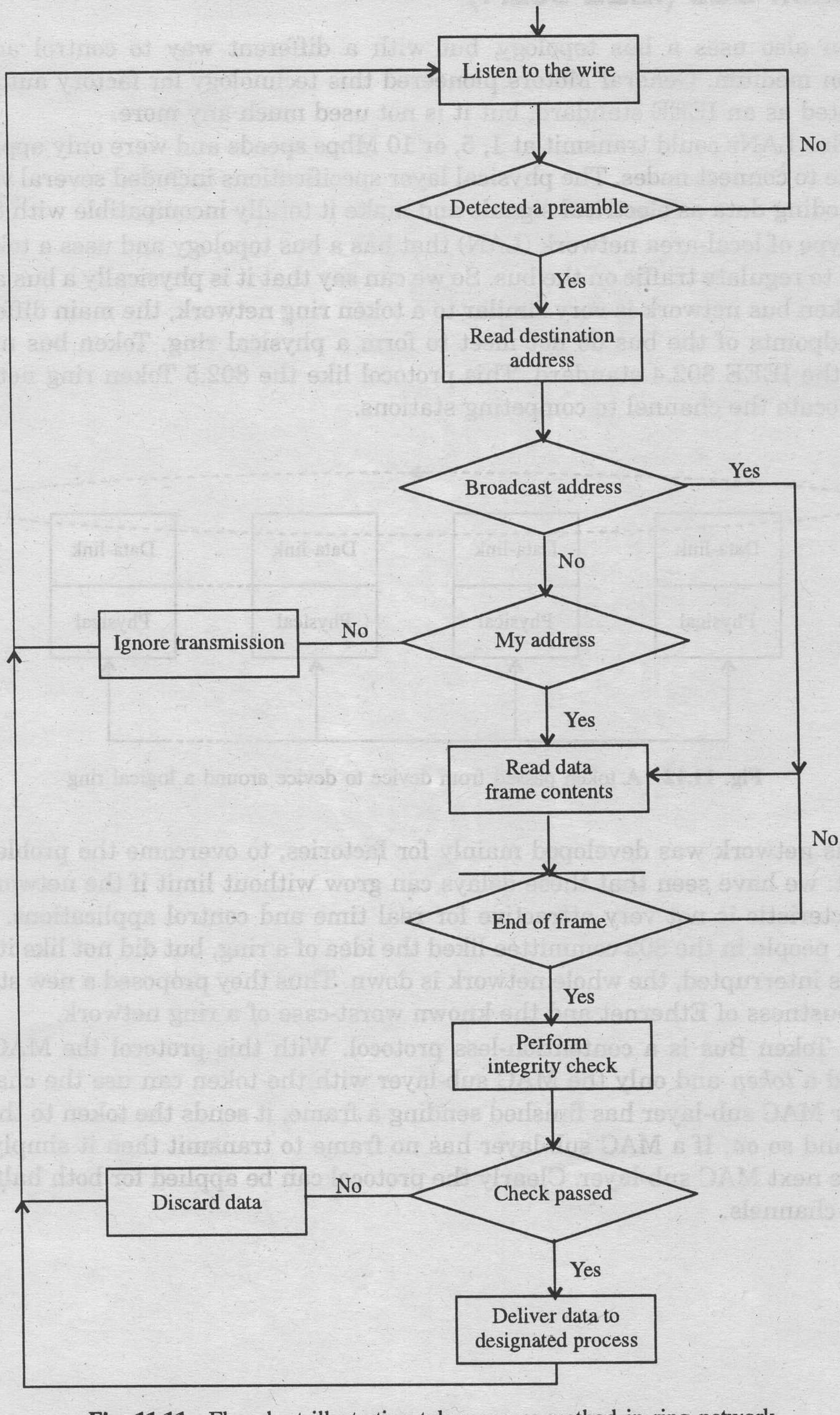


Fig. 11.11 Flowchart illustrating token access method in ring network

11.6 Token Bus (IEEE 802.4)

This version also uses a bus topology, but with a different way to control access to the transmission medium. General Motors pioneered this technology for factory automation and it was adopted as an IEEE standard, but it is not used much any more.

Token Bus LANs could transmit at 1, 5, or 10 Mbps speeds and were only approved to use coaxial cable to connect nodes. The physical layer specifications included several very complex ways of encoding data as electrical signals and make it totally incompatible with 802.3 LANs.

It is a type of local-area network (LAN) that has a bus topology and uses a token -passing mechanism to regulate traffic on the bus. So we can say that it is physically a bus and logically a ring. A token bus network is very similar to a token ring network, the main difference being that the endpoints of the bus do not meet to form a physical ring. Token bus networks are defined by the IEEE 802.4 standard. This protocol like the 802.5 Token ring network, use a token to allocate the channel to competing stations.

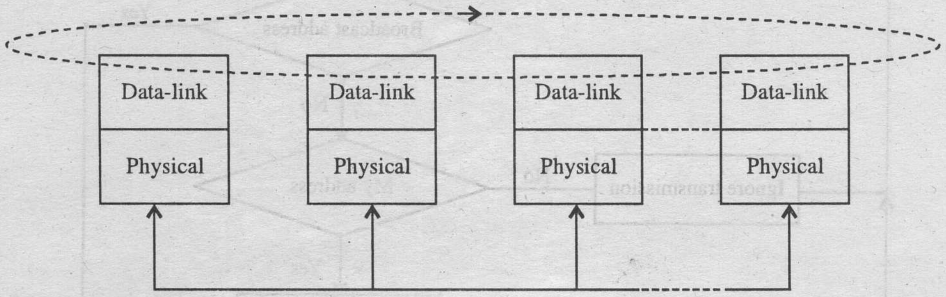


Fig. 11.12 A token passed from device to device around a logical ring

This bus network was developed mainly for factories, to overcome the problem of delays in Ethernet: we have seen that these delays can grow without limit if the network is loaded. This characteristic is not very attractive for real time and control applications. The factory automation people in the 802 committee liked the idea of a ring, but did not like its weakness: if the ring is interrupted, the whole network is down. Thus they proposed a new standard that has the robustness of Ethernet and the known worst-case of a ring network.

Logical Token Bus is a contention-less protocol. With this protocol the MAC sub-layers pass around a *token* and only the MAC sub-layer with the token can use the channel. When a particular MAC sub-layer has finished sending a frame, it sends the token to the next MAC sub-layer, and so on. If a MAC sub-layer has no frame to transmit then it simply passes the token to the next MAC sub-layer. Clearly the protocol can be applied for both half duplex and full duplex channels.

11.6.1 Token Bus Frame Format

Field	Preamble	Start of frame delimiter	Frame control	Destination address	Source address	Data	Checksum bits	End delimiter
Size (bytes)	1	1	1	6	6	0–8174	4	1

Fig. 11.13 Token bus frame format

Token Bus fields and their uses:

- **Preamble**—A special bit pattern that marks the beginning of a frame.
- **Start of frame delimiter**—A special bit pattern that marks the start of the frame itself.
- **Frame control**—A 1-byte field used to indicate if the frame contains actual data or if it is a control message.
  - □ For data frames, it carries the frame's priority, indicator requiring the destination to acknowledge the receipt of the frame.
  - □ For control frames, the filed is used to specify the frame type, including token passing, letting new station enter the ring, allowing station to leave, and other maintenance functions.
- **Destination Address**—An address specifying a specific destination station, a group of stations, or all stations in the LAN. This address can be 16 bits or 48 bits in length, but all stations in the LAN must adhere to one format or the other.
- **Source Address**—The address of the originating station. This address has the same length requirements as the destination address.
- **Length of data**—Indicates exactly how much data is in the frame (since it can vary).
- **Data**—The data to be delivered by the frame, it can range from 0 to 8174 bytes. Note that the 802.4 data section can be much longer than the 802.3 data section. The data field can be upto 8182 bytes long when 2-byte address is used; it can be upto 8174 bytes long when 6-byte address is used.
- **Checksum**—32-bit CRC error detection information for finding errors in the frame.
- **End of frame delimiter**—A special bit pattern that marks the end of the frame. The last two bits of this field signal if the frame is the last frame to be transmitted and whether any station has detected an error in the frame.

11.6.2 Token Bus Operation

- There is a special frame called the token that contains a unique bit pattern. This token is passed from node to node in a predetermined way along the bus.
- Only the node that has the token is allowed to transmit a frame and all other nodes must wait for their turn. Once a node receives the token, it can transmit a frame if it

needs to. If a node has several small frames to send, it can send them all at one time, so long as it does not take longer than a preset time limit.

- If there are no frames to be transmitted, the token is immediately passed to the next node.
- Since only one node can transmit at a time, collisions are impossible.
- The physical topology is arranged as a bus, but the logical topology is a ring. Each node on the bus knows its "successor" node in the logical ring, and passes the token to that successor node. Data frames are passed along the bus in the same way. This allows a token bus LAN to be connected in a simpler way than a physical ring, but still act like one.
- Nodes on a token bus LAN can assign different priorities to frames and they will deliver all high priority frames before they send any of the lower priority frames. This allows nodes to guarantee that they can send real-time data every time that they get the token and, if they have any time left, send other data as well.

Stations are added to an 802.4 bus by an approach called response windows:

- While holding the token, a node issues a solicit-successor frame. The address in the frame is between it and the next successor station.
- Token holder waits one window time (slot time, equal to twice the end-to-end propagation delay).
- If no response, the token is transferred to the successor node.
- If response, a requesting node sends a set-successor frame and token holder changes its successor node address. Requested node receives token, sets its addresses, and proceeds.

A node can drop out of the transmission sequence. Upon receiving a token, it sends a *set-successo*r frame to the predecessor, which orders the next node to give the token hereafter to its successor.

Priority Scheme

When the ring is initialized, stations are inserted into it in order of station address, from highest to lowest. The token bus defines four priority classes, 0, 2, 4, 6 for traffic, with 0 the lowest and 6 the highest. Conceptually, each station internally being divided into four substations, one at each priority level. As input comes into the MAC sublayer from above, data are checked for priority and routed to one of the four substations. Thus each substation maintains its own queue. When the token comes into the station over the cable, it is passed internally to the priority 6 substation, which may begin transmitting frames, if it has any. When it is done (or when its timer expires), the token is passed internally to the priority 4 substation. Setting the timer properly we can ensure that a guaranteed fraction of the total token holding time can be allocated to priority 6 traffic. The lower priorities will have to live with what is left over. Priority 6 traffic is guaranteed a known fraction of the network bandwidth and can be used to implement real-time traffic.

Overall Performance of Token Based Protocols

Token based protocols (token ring and token bus) are essentially contention-free protocols. They rely on coordination among stations.

- Under light load, few stations want to transmit. But they still have to wait for token to pass around. Token based protocols perform poorer than contention based protocol such as Ethernet at light load.
- Under heavy load, all stations want to transmit, thus coordination can avoid collision. Token based protocols perform better than contention based protocol such as Ethernet.

This is analog to waiting lines in a restaurant. When few people are around, the extra zig-zag lines are anoying. When many people want to eat, these lines help organize the crowd and people can get through fast.

11.7 Comparison Of Various IEEE Standards

11.7.1 Token Ring IEEE 802.5 Vs Ethernet IEEE 802.3

- Token Ring networks are deterministic in nature-nodes may only transmit at certain well defined times. Result is high bandwidth efficiency. Up to 90% in Token Ring, 40% in Ethernet.
- Token Ring performance does not deteriorate to the same extent as Ethernet when network traffic increases. This means that at high loads, the presence of collisions of data frames on Ethernet networks becomes a major problem and can seriously affect the throughput.
- By its nature Token Ring has a higher reliability, the ring can continue normal operation in most cases despite any single fault.
- Ethernet has an advantage over Token Ring in that the cost of network equipment is lower for Ethernet. Token Ring networks tend to be more expensive to set up and maintain than Ethernet, although hardware costs for Token Ring are decreasing.
- Advances in Ethernet technology have tended to be much more rapid than Token Ring. Gigabit Ethernet being an example of this. Token Ring technologies are being developed however to allow data transfer rates of 100Mbps using technologies such as Token Ring Switching.

11.7.2 Token Ring—IEEE 802.5 Vs Token Bus—IEEE 802.4

- These standards although similar in some respects were developed with different applications in mind. Token Bus (IEEE 802.4), networks use a Bus technology and do not have a centralized active monitor. Token Bus was designed with large factories in mind where machines and equipment would be moving around under computer control. Network failures could have serious consequences and so had to be avoided at all costs. On the other hand the Token Ring network standard was designed with office automation in mind, where a failure once in a rare while could be tolerated as the price for simpler system.

- Token Bus is not as deterministic as Token Ring but like Token Ring it has excellent throughput and efficiency at high load.
- Token Bus uses broadband transmission and cabling which obviously gives it a bandwidth advantage over Token Ring but the downside is the equipment costs (modems, wideband amplifiers, etc). The IEEE 802.4 protocol is also extremely complex compared to IEEE 802.5 and it has a substantial delay at low loads.

11.7.3 Comparison of 802.3, 802.4, and 802.5

Advantages of 802.3

- Low maintenance (low management complexity)
- Easy installation (in comparison with 802.4 and 802.5)
- Widely used (cheap components and a large amount of trained personnel)
- Low latency under low loads—get quick access when not many stations are transmitting—have to wait for a token with the other protocols

Disadvantages of 802.3

- Non-deterministic
- Utilization falls under a heavy load
- Lack of priorities
- Lack of acknowledgements
- Minimum frame length
- Efficiency drops as speed increases

Advantages of 802.4

- Uses cable TV equipment (no creation of new equipment)
- Deterministic
- Has priorities
- No minimum packet size (if you only have 1-bit to send, only send 1-bit)
- High utilization under a heavy load
- Efficiency doesn't drop with increased speed

Disadvantages of 802.4

- High management complexity
- Large analog component (since uses cable TV equipment)
- Not widely used
- Low efficiency under low loads
- Not suited for fiber optics

Advantages of 802.5

- Can use fiber
- Deterministic
- Has priorities
- Utilization high under loads
- No frame size restrictions

Disadvantages of 802.5

- High management complexity
- Breaks in the ring
- Installation (wire centers improve this my preventing you from having to break the ring to install a new station)
- High latency under low loads (must wait for the token to propagate around)

11.8 Distributed Queue Dual Bus—DQDB (IEEE 802.6)

Distributed Queue Dual Bus (DQDB), specified in the IEEE 802.6 standard, is a data-link layer communication protocol for Metropolitan Area Networks (MANs). DQDB is designed for data as well as voice and video transmission based on cell switching technology. DQDB, which permits multiple systems to interconnect using two unidirectional logical buses, is an open standard that is designed for compatibility with carrier transmission standards.

For a MAN to be effective it requires a system that can function across long, city-wide distances of several miles, have a low susceptibility to error, adapt to the number of nodes attached and have variable bandwidth distribution. Using DQDB, networks can be thirty miles long and function in the range of 34 Mbps to 155 Mbps. The data rate fluctuates due to many hosts sharing a dual bus as well as the location of a single host in relation to the frame generator, but there are schemes to compensate for this problem making DQDB function reliably and fairly for all hosts.

The **distributed queue dual bus (DQDB)** network uses a different kind of MAC method based on the use of a distributed queuing algorithm called **queued-packet distributed-switch (QPSX)** and a slotted ring arrangement. It uses two unconnected unidirectional buses, which are normally implemented as a series of point-to-point segments. DQDB also expects the use of optical fibre links.

11.8.1 Architecture of DQDB

The DQDB is composed of a two bus lines with stations attached to both and a frame generator at the end of each bus. The connections are normally point-to-point, but are often depicted in the tapped-bus type configuration as shown in Figure 11.14. The buses run in parallel in such a fashion as to allow the frames generated to travel across the stations in opposite directions. Both these buses always have a constant number of slots circulating on them. A slot on one bus is copied to the other.

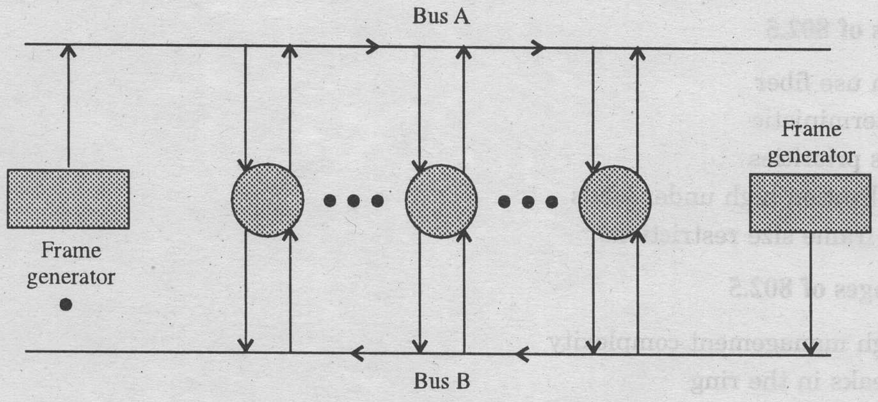


Fig. 11.14 Architecture of DQDB

The two buses in a DQDB network transmit cells in opposite directions and each node is connected to both buses.

The DQDB LAN or MAN transports data in fixed size **cells**, which happen to look very much like ATM cells (discussed in chapter 8). However, as DQDB offers a MAC service, the MAC frame may need to be segmented into several cells before transmission. The cell structure for DQDB is almost identical to that of an ATM cell. This similarity between the cell structure has been made so that the DQDB will be compatible with B-ISDN.

Octets	5	1	1	44	1	1
	Header	ST	MID	Information	LEN	CRC

CRC coverage

ST = Segment type MID = Message identifier
LEN = Data length CRC = Cyclic redundancy check

Fig. 11.15 A DQDB cell

BIT

8	7	6	5	4	3	2	1	OCTET
ACF								1
VCI								2
VCI								3
VCI				PT		CLP		4
HEC								5

VPI = Virtual path identifier PT = Pay load type
VCI = Virtual channel identifier CLP = Cell loss priority
HEC = Header error control ACF = Access control field

Fig. 11.16 A DQDB cell header

A DQDB cell differs from an ATM cell in its header and its payload. In the header of the DQDB cell there is no virtual path identifier (VPI) and the virtual channel identifier (VCI) has an additional 4-bits. The first 8 bits of the DQDB header form an access control field (ACF) (in ATM this is the first 8-bits of the VPI). Also, the next 4-bits (the final 4-bits of the VPI in ATM) form the first 4-bits of the VCI.

Also, the **CELL PAYLOAD** is structured as discussed below:

- **Segment type (ST)** : Identifies the cell as one of the following:
  - □ **Single segment** : Only this segment (no MAC fragmentation was required).
  - □ **First segment** : The first cell of a segmented MAC frame.
  - □ **Intermediate segment** : The intermediate cells in a fragmented MAC frame.
  - □ **Last segment** : The final cell of a segmented MAC frame.
- **Message identifier (MID)** : The MID is the same for all DQDB cells from the same MAC frame. This allows the identification of intermediate segments.
- **Information** : (part of) The MAC frame contents.
- **Length (LEN)** ; The length of the information field.
- **CRC** : Covering everything the whole cell payload.

The **CELL HEADER** contains the following information:

- **Access control field (ACF)** : This contains the BUSY and REQUEST bits that are used in the operation of the QPSX mechanism. The BUSY bit indicates the slot is in use. The REQUEST bit is set in a slot by a node that is waiting to transmit.
- **Virtual channel identifier (VCI)** : This is not used in a DQDB MAN as there are no logical connections which require multiplexing—the ST and MID fields ,in the payload are used instead. In a DQDB LAN, there is the possibility for applications to make use of this field.
- **Payload type (PT)** : (same as ATM) It identifies payload type for OAM (Operations, Administration and Maintenance). Also reserved bits for explicit congestion control.
- **Cell loss priority (CLP)** : (same as ATM) It is used for buffer management.
  - □ If set (1) then the cell may be discarded according to network conditions—congestion control.
  - □ If not set (0) cell cannot be discarded.
- **Header error control (HEC)** : CRC for the header. It is used to detect and correct errors in the header.

11.9 CSMA/CA Medium Access Protocol (IEEE 802.11)

Introduction

The IEEE 802.11 standard defines the physical layer and media access control (MAC) layer for a wireless local area network. This standard is increasingly being deployed in wireless LAN communication.

Shared broadcast channels are often used in local area networks (LANs). Both wired 802.3 Ethernet network and wireless 802.11 LAN network must co-ordinate transmissions onto the shared communication channel. In case of 802.3 Ethernet, the shared channel is the shared wire whereas in case of wireless 802.11 LAN the shared channel is the radio frequency. The media access control (MAC) protocol co-ordinates the transmission. Although the IEEE 802.11 standard belongs to the same standard family as wired 802.3 Ethernet, it has a significantly different media access protocol. While Collision Detection works well on Ethernet, they cannot be used in wireless LAN.

The media access control of IEEE 802.11 is using carrier sense multiple access with collision avoidance (CSMA/CA) as the fundamental access. In 802.11 carrier sense (CS) is performed both at physical layer (physical carrier sensing) and at the MAC layer (virtual carrier sensing).

The IEEE 802.11 MAC protocol does not implement collision detection. There are a few reasons for this:

- Implementing a Collision Detection Mechanism would require the implementation of a Full Duplex radio, capable of transmitting and receiving at once. This approach would increase the price significantly.
- In radio systems received signal is weak compared to transmitted signal and therefore collision detection cannot be done by simple comparison.
- Even if one had collision detection and sensed no collision when sensing, a collision could still occur at the receiver. The reason for this is the two following properties of the wireless channel.

The 802.11 standard includes a basic Distributed Co-ordination Function (DCF). The DCF is the fundamental access method used to support asynchronous data transfer on the best effort basis. As specified in standards, the DCF must be supported by all the stations in a basic service set (BSS). The DCF is based on CSMA/CA.

There are two techniques used for packet transmitting in DCF:

- The default one is a two-way handshaking mechanism, also called basic access method. The destination station transmits a positive acknowledgement (ACK) message to signal a successful packet transmission.
- The other optional mechanism is a four-way handshaking access method, which uses the request-to-send/clear-to-send (RTS/CTS) technique to reserve the channel before data transmission.

Carrier Sensing Protocol with Collision Avoidance

The carrier sensing family of protocols is characterized by sensing the carrier and deciding according to it whether another transmission is ongoing. All the carrier sense multiple access (CSMA) protocols share the same philosophy: when a user generates a new packet the channel is sensed and if found idle the packet is transmitted immediately.

These kinds of algorithms are very effective when the medium is not heavily loaded, since it allows users to transmit with a minimum delay. However, there is always a chance of several users transmitting at the same time (i.e., collision), because the users sensed the medium free and decided to transmit at once. Because of the difficulties in detecting collisions at a wireless receiver, the IEEE 802.11 protocol tries to avoid collisions, rather than detect and recover from collisions.

The CSMA/CA protocol is designed to reduce the collision probability at the points where collisions would most likely occur. The highest probability of a collision exists when the medium has become idle after a busy state. This is because several users could have been waiting for the medium to be available again. This is the situation that necessitates a random back-off procedure to resolve medium contention conflicts. Also, the use of IFS (Inter Frame Space) helps to resolve the problem. CSMA/CA provides some carrier sense functions to avoid collisions.

11.9.1 IEEE 802.11 Architectures

In IEEE's proposed standard for wireless LANs (IEEE 802.11), there are two different ways to configure a network

- Ad-hoc.
- Infrastructure.

In the ad-hoc network, computers are brought together to form a network "on the fly". As shown in Figure 11.17, there is no structure to the network; there are no fixed points; and usually every node is able to communicate with every other node. A good example of this is the aforementioned meeting where employees bring laptop computers together to communicate and share design or financial information. Although it seems that order would be difficult to maintain in this type of network, algorithms such as the spokesman election algorithm (SEA) have been designed to "elect" one machine as the base station (master) of the network with the others being slaves. Another algorithm in ad-hoc network architectures uses a broadcast and flooding method to all other nodes to establish who's who.

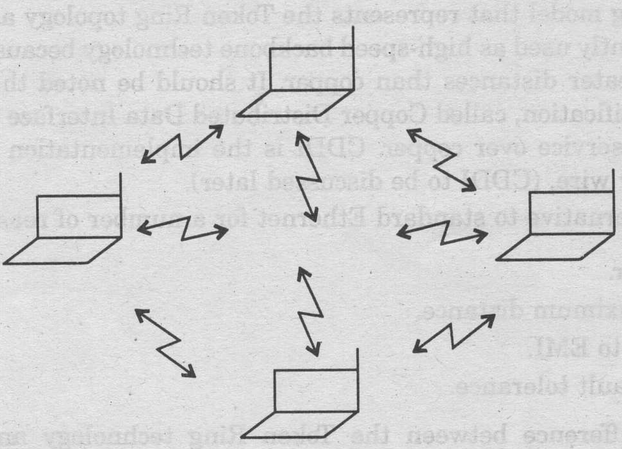


Fig. 11.17 Ad-hoc network

As shown in Figure 11.18, the second type of network structure used in wireless LAN's is **the infrastructure**. This architecture uses fixed network access points with which mobile nodes can communicate. These network access points are sometime connected to landlines to widen the LAN's capability by bridging wireless nodes to other wired nodes. If service areas overlap, handoffs can occur. This structure is very similar to the present day cellular networks around the world.

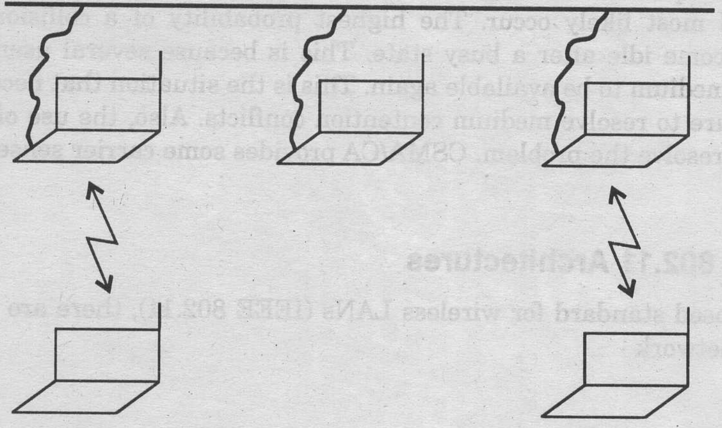


Fig. 11.18 Infrastructure network

11.10 Fiber Distributed Data Interface (FDDI)

Another high-speed token-passing network architecture is the Fiber Distributed Data Interface (FDDI). FDDI is a new standard for token ring using fiber for individual links. This network is much faster, extremely fault tolerant and can cover more distance than Token Ring. This technology uses fiber-optic cabling to reach speeds of 100-Mbps. Although FDDI lacks a Project 802 standard, it roughly maps to the IEEE 802.3 and 802.5. FDDI most represents the 802.5 token-passing model that represents the Token Ring topology as indicated earlier.

FDDI is frequently used as high-speed backbone technology because of its support for high bandwidth and greater distances than copper. It should be noted that relatively recently, a related copper specification, called Copper Distributed Data Interface (CDDI), has emerged to provide 100-Mbps service over copper. CDDI is the implementation of FDDI protocols over twisted-pair copper wire. (CDDI to be discussed later).

FDDI is an alternative to standard Ethernet for a number of reasons:

- Much faster.
- Greater maximum distance.
- Resistance to EMI.
- Improved fault tolerance.

The biggest difference between the Token Ring technology and FDDI is the use of fiber-optic cable, which increases the transmission speeds to 100-Mbps. A much greater maximum distance of two kilometers between workstations now exists within a FDDI network. This is due in part to the fiber-optic cabling used which also resists *electromagnetic interference (EMI)*, which has always plagued copper-based cabling. These features have made FDDI an excellent choice for connecting LAN's or buildings together. In addition to high speed and resistance to EMI, fiber-optic cable cannot be tapped like copper-based cable can. However, the cost is still beyond reach of most networks to offer a FDDI solution for local area networks.

FDDI also introduces dual, counter-rotating rings. Workstations can connect to one ring or both, which still allows the workstation to communicate even if one of the rings fails. Although Token Ring has the ability to implement fault-tolerant techniques with redundant rings, many Token Ring networks use only one ring.

11.10.1 Features of FDDI

1. Supports data rates of 100-Mbps over 200 km.
2. Uses LED rather than lasers for safety reason.
3. Calls for less than 10-10 bit error rate. This means we worry much less about checksums etc.
4. In addition to data frames, FDDI also permits special synchronous frames for real-time data such as voice. The synchronous frames are generated every 125µ second to provide the 8000 samples per second PCM data.

11.10.2 FDDI Transmission Media

FDDI uses optical fiber as the primary transmission medium, but it also can run over copper cabling. As mentioned earlier, FDDI over copper is referred to as *Copper-Distributed Data Interface (CDDI)*. Optical fiber has several advantages over copper media. In particular, security, reliability and performance all are enhanced with optical fiber media because fiber does not emit electrical signals. A physical medium that does emit electrical signals (copper) can be tapped and therefore would permit unauthorized access to the data that is transiting the medium. In addition, fiber is immune to electrical interference from radio frequency interference (RFI) and electromagnetic interference (EMI). Fiber historically has supported much higher bandwidth (throughput potential) than copper, although recent technological advances have made copper capable of transmitting at 100 Mbps. Finally, FDDI allows 2 km between stations using multimode fiber and even longer distances using a single mode.

FDDI defines two types of optical fiber: single-mode and multimode. A *mode* is a ray of light that enters the fiber at a particular angle. *Multimode* fiber uses LED as the light-generating device, while *single-mode* fiber generally uses lasers.

11.10.3 FDDI Frame Format

The FDDI frame format is similar to the format of a Token Ring frame. FDDI frames can be as large as 4,500 bytes. Figure 11.19 shows the frame format of an FDDI data frame and token.

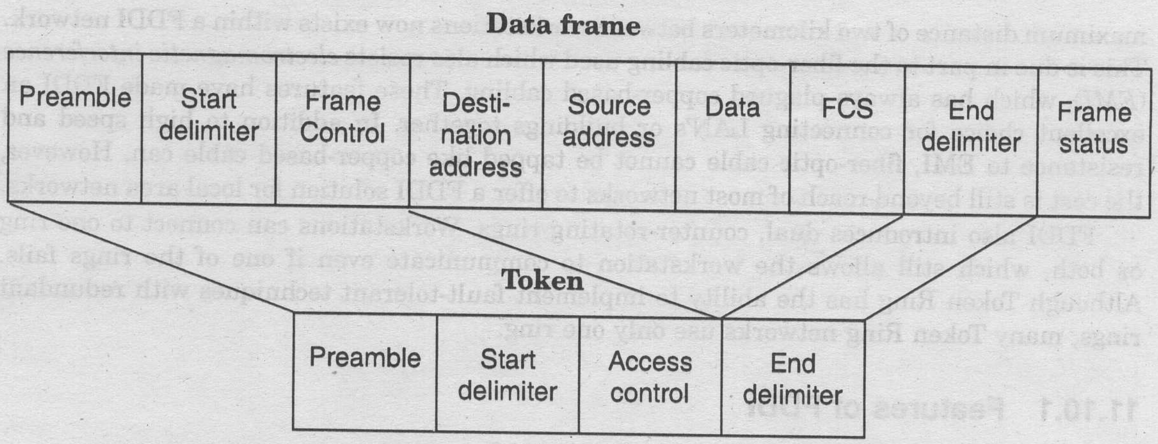


Fig. 11.19 The FDDI frame is similar to that of a token ring frame

FDDI Frame Fields

The following descriptions summarize the FDDI data frame and token fields illustrated in Figure 11.19.

- **Preamble (PA)**—16 (or more) IDLE symbols. Gives a unique sequence that prepares each station for an upcoming frame.
- **Start Delimiter (SD)**—Indicates the beginning of a frame by employing a signaling pattern that differentiates it from the rest of the frame. The 2 symbols *J* and *K* are used to show the start of the frame and also to allow interpretation of correct symbol boundaries.
- **Frame Control (FC)**—Indicates the size of the address fields and whether the frame contains asynchronous or synchronous data, among other control information. 2 symbols indicating whether or not this is an information frame or a MAC frame (e.g., the token), with some additional control information for the station identified by the DA.
- **Destination Address (DA)**—Contains a unicast (singular), multicast (group) or broadcast (every station) address. As with Ethernet and Token Ring addresses, FDDI destination addresses are 6 bytes long.
- **Source Address (SA)**—Identifies the single station that sent the frame. As with Ethernet and Token Ring addresses, FDDI source addresses are 6 bytes long.
- **Data**—Contains either information destined for an upper-layer protocol or control information.
- **Frame Check Sequence (FCS)**—Is filled by the source station with a calculated cyclic redundancy check value dependent on frame contents (as with Token Ring and Ethernet). The destination address recalculates the value to determine whether the frame was damaged in transit. If so, the frame is discarded.
- **End Delimiter**—Contains unique symbols; cannot be data symbols that indicate the end of the frame.

- **Frame Status**—Allows the source station to determine whether an error occurred; identifies whether the frame was recognized and copied by a receiving station.

11.10.4 Operation Of FDDI

FDDI's operation is similar to that of token ring. FDDI uses dual-ring architecture with traffic on each ring flowing in opposite directions (called counter-rotating). The primary purpose of the dual rings is to provide superior reliability and robustness. Figure 11.20 shows the counter-rotating primary and secondary FDDI rings.

The dual rings consist of a primary and a secondary ring. During normal operation, the primary ring is used for data transmission and the secondary ring remains idle. Stations attached to the FDDI may be connected to both rings, **dual attach stations (DASs)** or only to the primary ring, **single attach station (SASs)**. Although, the stations are logically attached in a ring, the physical connection is more conveniently realized in a hub-star fashion by using wiring concentrators.

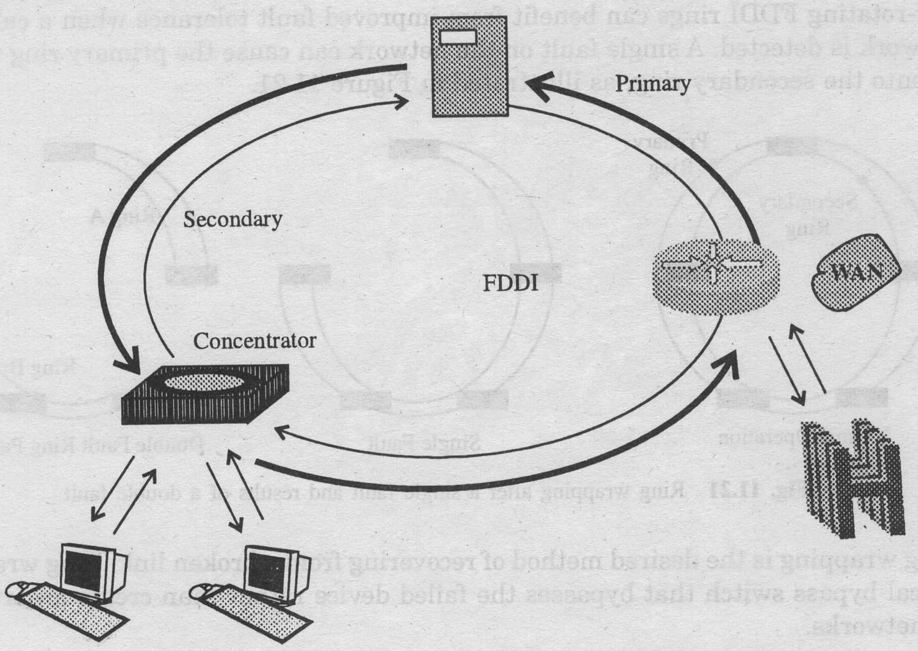


Fig. 11.20 FDDI uses counter-rotating primary and secondary rings

Access Method of FDDI—Token Passing

Token passing is the network access method for the FDDI architecture and still behaves as the original token-passing implementation. Of course, the fiber-optic cable has increased the speed at which the token travels around the network. Priorities can still be assigned to workstations and the active monitoring still remains. One reason FDDI can produce higher transmission rates than Token Ring is because a FDDI network can have several frames on the network at once. Once an FDDI station is done transmitting, the token is released. The

FDDI source does not have to wait for successful acknowledgements from the destination before the token is released. As with Token Ring, FDDI is well suited for high volumes of network traffic. This is due to the token-passing technology which eliminates collisions on the network.

SAS and DAS stations

A single-attached station (SAS) is a workstation or device that is attached to only one of the fiber-optic cable rings. This workstation can continue communicating with stations only on the same segment if the ring were to fail. A dual-attached station (DAS) is attached to both fiber-optic cable rings and can continue communicating with the remaining stations if one of the rings fails and can redirect traffic onto the secondary ring.

Beaconing and Wrapping

Beaconing still functions just as described in the section on Token Ring. However, a new technique known as *wrapping* has been introduced with the FDDI network. The dual, counter-rotating FDDI rings can benefit from improved fault tolerance when a cable fault on the network is detected. A single fault on the network can cause the primary ring to wrap the signal onto the secondary ring, as illustrated in Figure 11.21.

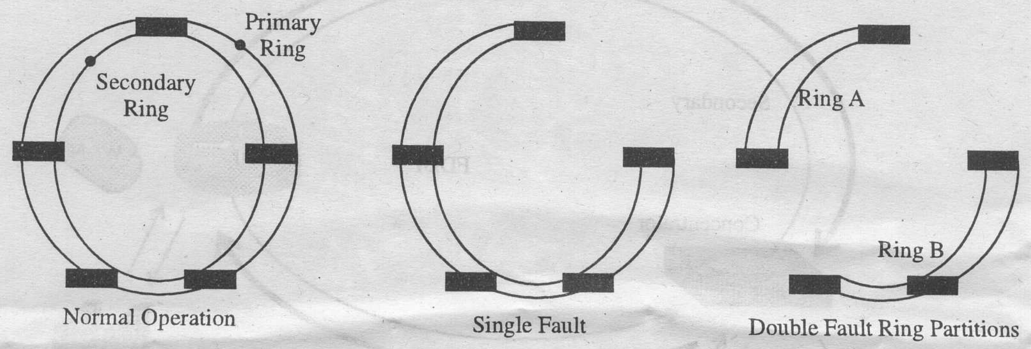


Fig. 11.21 Ring wrapping after a single fault and results of a double fault

Ring wrapping is the desired method of recovering from a broken link. Ring wrapping uses an optical bypass switch that bypasses the failed device rather than create a partitioned set of two networks.

FDDI Fault Tolerance

FDDI provides a number of fault-tolerant features. In particular, FDDI's dual-ring environment, the implementation of the optical bypass switch and dual-homing support make FDDI a resilient media technology.

Dual Ring

FDDI's primary fault-tolerant feature is the *dual ring*. If a station on the dual ring fails or is powered down or if the cable is damaged the dual ring is automatically wrapped (doubled back onto itself) into a single ring. When the ring is wrapped, the dual-ring topology becomes a single-ring topology. Data continues to be transmitted on the FDDI ring without performance impact during the wrap condition.

11.10.5 FDDI Specifications

FDDI specifies the physical and media-access portions of the OSI reference model. FDDI is not actually a single specification, but it is a collection of four separate specifications, each with a specific function. Combined, these specifications have the capability to provide high-speed connectivity between upper-layer protocols such as TCP/IP and IPX, and media such as fiber-optic cabling.

FDDI's four specifications are stated below:

- **Media Access Control (MAC) :** The MAC specification defines how the medium is accessed, including frame format, token handling, addressing, algorithms for calculating cyclic redundancy check (CRC) value and error-recovery mechanisms.
- **Physical Layer Protocol (PHY) :** The PHY specification defines data encoding/decoding procedures, clocking requirements, and framing, among other functions.
- **Physical-Medium Dependent (PMD) :** The PMD specification defines the characteristics of the transmission medium, including fiber-optic links, power levels, bit-error rates, optical components and connectors.
- **Station Management (SMT) :** The SMT specification defines FDDI station con-

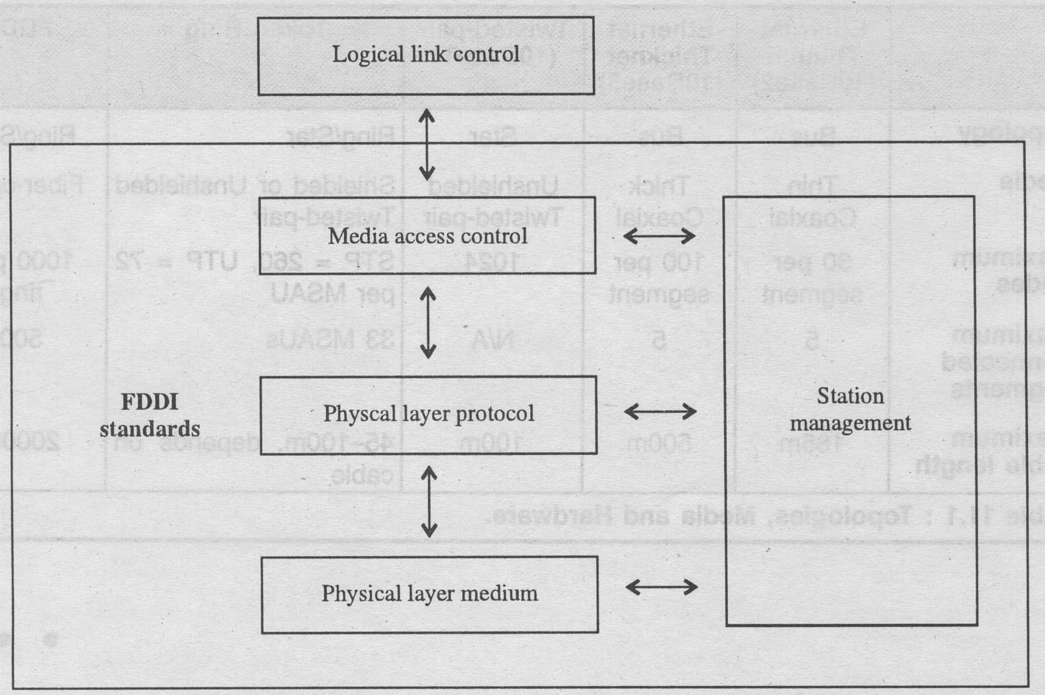


Fig. 11.22 FDDI Specifications map to the OSI hierarchical model

figuration, ring configuration and ring control features, including station insertion and removal, initialization, fault isolation and recovery, scheduling and statistics collection.

FDDI is similar to IEEE 802.3 Ethernet and IEEE 802.5 Token Ring in its relationship with the OSI model. Its primary purpose is to provide connectivity between upper OSI layers of common protocols and the media used to connect network devices. Figure 11.22 illustrates the four FDDI specifications and their relationship to each other and to the IEEE-defined Logical Link Control (LLC) sublayer. The LLC sublayer is a component of Layer 2, the MAC layer, of the OSI reference model.

11.11 Copper Distributed Data Interface (CDDI)

Copper Distributed Data Interface (CDDI) is a technology just like FDDI, except that copper-based media such as UTP or STP is used rather than fiber-optic cabling. However, the cost is greatly reduced, as well as the maximum cable lengths. The fiber-optic cable length used in FDDI is two kilometers, where a CDDI copper cable is limited to 100 meters.

Like FDDI, CDDI provides data rates of 100 Mbps and uses dual-ring architecture to provide redundancy. CDDI supports distances of about 100 meters from desktop to concentrator.

Table 11.1 Summarizes the topologies the media they employ and the hardware they require.

	Ethernet Thinnet (10Base2)	Ethernet Thicknet (10Base5)	Twisted-pair (10BaseT)	Token Ring	FDDI
Topology	Bus	Bus	Star	Ring/Star	Ring/Star
Media	Thin Coaxial	Thick Coaxial	Unshielded Twisted-pair	Shielded or Unshielded Twisted-pair	Fiber-optic
Maximum nodes	30 per segment	100 per segment	1024	STP = 260, UTP = 72 per MSAU	1000 per ring
Maximum connected segments	5	5	N/A	33 MSAUs	500
Maximum cable length	185m	500m	100m	45–100m, depends on cable	2000m

Table 11.1 : Topologies, Media and Hardware.

● ● ●

12

Network Layer

Introduction

The network layer is concerned with getting packets from the source all the way to the destination. This function contrasts with the goal of the data link layers whose purpose is to just transmit the bits from one end of a wire to the other end. The network layer design issues include the service provided to the transport layer, routing of packets through the subnet, congestion control and connection of multiple networks together.

12.1 Design Issues for the Network Layer

The network layer has been designed with the following goals:

1. The services provided should be independent of the underlying technology. Users of the service need not be aware of the physical implementation of the network. This design goal has great importance when we consider the great variety of networks in operation. In the area of Public networks, networks in underdeveloped countries are nowhere near the technological skill of those in the countries like the US or Ireland. The design of the layer must not disable us from connecting to networks of different technologies.
2. The transport layer (that is the host computer) should be shielded from the number, type and different topologies of the subnets it uses. That is, all the transport layer want is a communication link; it need not know how that link is made.
3. Finally, there is a need for some uniform addressing scheme for network addresses.

With these goals in mind, two different types of service emerged: Connection oriented and connectionless. A connection-oriented service is one in which the user is given a "reliable" end-to-end connection. To communicate, the user requests a connection, then uses the connection to his hearts content, and then closes the connection. A telephone call is the classic example of a connection-oriented service.

In a connection-less service, the user simply bundles his information together, puts an address on it and then sends it off, in the hope that it will reach its destination. There is no guarantee that the bundle will arrive. So a connection less service is one reminiscent of the postal system. A letter is sent, that is, put in the post box. It is then in the "postal network" where it gets bounced around and hopefully will leave the network in the correct place, that is, in the addressee's letter box. We can never be totally sure that the letter will arrive, but we know that there is a high probability that it will and so we place our trust in the postal network.

Overview of Other Network Layer Issues

The network layer is responsible for routing packets from the source to destination. The *routing algorithm* is the piece of software that decides where a packet goes next (e.g., which output line, or which node on a broadcast channel).

For connectionless networks, the routing decision is made for each datagram. For connection-oriented networks, the decision is made once, at circuit setup time.

1. Routing Issues

The routing algorithm must deal with the following issues:

- **Correctness and simplicity :** Networks are never taken down; individual parts (e.g., links, routers) may fail, but the whole network should not.
- **Stability :** If a link or router fails, how much time elapses before the remaining routers recognize the topology change? (Some never do..)
- **Fairness and optimality :** An inherently intractable problem. Definition of optimality usually doesn't consider fairness. Do we want to maximize channel usage? Minimize average delay?

When we look at routing in detail, we'll consider these routing issuses again.

2. Congestion

The network layer also must deal with congestion:

- When more packets enter an area than can be processed, delays increase and performance decreases. If the situation continues, the subnet may have no alternative but to discard packets.
- If the delay increases, the sender may (incorrectly) retransmit, making a bad situation even worse.
- Overall, performance degrades because the network is using (wasting) resources processing packets that eventually get discarded.

3. Internetworking

Finally, when we consider internetworking—connecting different network technologies together—one finds the same problems, only worse:

- Packets may travel through many different networks.
- Each network may have a different frame format.
- Some networks may be connectionless, other connection oriented.

12.2 The Internal Organization Of The Network Layer

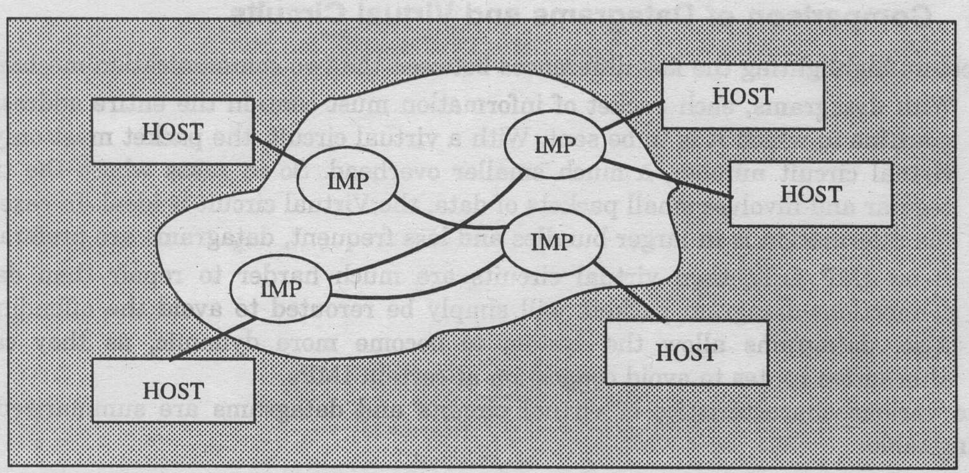


Fig. 12.1 General shape of a network

The above figure shows the general shape of a network. Your machine is called the host, and it is connected to the **IMP (Interface Message Processor)**. The terms *packet switch node, intermediate system* and *data switching exchange* are all synonyms for IMP. As you can see from the figure, you communicate with other hosts through a number of IMP's—these act as nodes which forward your messages to the correct destination.

We know of the two different classes of service offered by the network layer-connection oriented and connectionless. It is instructive now to see how these services are implemented within the network. The two mechanisms used are:

- Virtual Circuits.
- Datagrams.

Virtual Circuits

Virtual circuits are used to provide a connection-oriented service. This kind of service is ideal in situations where many packets of information are to be sent on the same route—rather than go through the effort of routing each packet separately, we set up a virtual path.

To do this, each IMP maintains a table of forwarding addresses for each virtual path. So, if the IMP receives a message, it can tell from the line on which it received that through which virtual circuit the message is on. The message can then be forwarded appropriately. On termination of the connection, each IMP removes the Virtual Circuit from its table, thus freeing it for further use.

Datagrams

For a connectionless service, datagrams are used. Each packet of information is bundled into a datagram with its destination address. Instead of each IMP having a table for virtual paths, it has a table of lines—which line to use for each IMP. In fact, these tables are used already for the Virtual Circuit service, for setting up the circuit in the first place.

12.2.1 Comparison of Datagrams and Virtual Circuits

Some points highlighting the key differences between the two communication mechanisms:

- With datagrams, each packet of information must contain the entire address of the machine to which it is to be sent. With a virtual circuit, the packet must only hold a virtual circuit number, a much smaller overhead. So in cases where the traffic is regular and involves small packets of data, the Virtual circuit is ideal. In cases where the information is in larger bundles and less frequent, datagrams are preferable.
- If an IMP fails, then virtual circuits are much harder to repair than datagram connections-datagram packets will simply be rerouted to avoid the offending node. Also, datagrams allow the system to become more dynamic, as they can take alternative routes to avoid congestion at certain IMPs.

The various characteristics of virtual circuits and datagrams are summarized in the following table:

	Virtual Circuit	Datagram
Connection Setup	Before Data Flow	Not required
Addressing	Network address mapped to Virtual Circuit Identifier	Source and destination network address included on each
State	Required for each Virtual Circuit	No state storage needed in routers
Routing	Selected at connection setup	Selected for each packet
Node Failure	Terminates all virtual circuits in the switch	Only packets in the router queue are lost
Complexity	Within the network switches	Within the host computers

12.3 Routing

Routing is moving information across a network from source to destination. Along the way, at least one intermediate node is typically encountered. Routing is often contrasted with bridging which seems to accomplish precisely the same thing. The primary difference between the two is that bridging occurs at Layer 2 (the link layer) of the OSI reference model, while routing occurs at Layer 3 (the network layer).

This distinction provides routing and bridging with different information to use in the process of moving information from source to destination. As a result, routing and bridging accomplish their tasks in different ways and, in fact, there are several different kinds of routing and bridging.

12.3.1 Routing Components

Routing involves two basic activities : determination of optimal routing paths and the transport of information groups (typically called *packets*) through a network. The latter of these is referred to as *switching*. Switching is relatively straightforward. Path determination, on the other hand, can be very complex.

Path Determination

A metric is a number assigned to a parameter and is used to quantify how "good" or "bad" that parameter is for a task. For example, path length is used by routing algorithms to determine the optimal path to a destination. To aid the process of path determination, routing algorithms initialize and maintain routing tables, which contain route information. Route information varies depending on the routing algorithm used.

Routing algorithms fill routing tables with a variety of information. Destination/next hop associations tell a router that a particular destination can be gained optimally by sending the packet to a particular router representing the "next hop" on the way to the final destination. Routing tables can also contain other information, such as information about the desirability of a path. Routers compare metrics to determine optimal routes. Metrics differ depending on the design of the routing algorithm being used. Routers communicate with one another (and maintain their routing tables) through the transmission of a variety of messages. The routing update message is one such message. Routing updates generally consist of all or a portion of a routing table. By analyzing routing updates from all routers, a router can build a detailed picture of network topology. A link-state advertisement is another example of a message sent between routers. Link-state advertisements inform other routers of the state of the sender's links. Link information can also be used to build a complete picture of network topology. Once the network topology is understood, routers can determine optimal routes to network destinations.

Metrics

Routing tables contain information used by switching software to select the best route. But how, specifically, are routing tables built? What is the specific nature of the information they contain? How do routing algorithms determine that one route is preferable to others?

Routing algorithms have used many different metrics to determine the best route. Sophisticated routing algorithms can base route selection on multiple metrics, combining them in a single (hybrid) metric. All of the following metrics have been used :

1. Path Length.

2. Reliability.
3. Delay.
4. Bandwidth.
5. Load.
6. Communication Cost.

Path Length

Path length is the most common routing metric. Some routing protocols allow network administrators to assign arbitrary costs to each network link. In this case, path length is the sum of the costs associated with each link traversed. Other routing protocols define hop count, a metric that specifies the number of passes through networking products (such as routers) that a packet must take to route from a source to a destination.

Reliability

Reliability, in the context of routing algorithms, refers to the reliability (usually described in terms of the bit-error rate) of each network link. Some network links may go down more often than others. Once down, some network links may be repaired more easily or more quickly than other links. Any reliability factors can be taken into account in the assignment of reliability ratings. Reliability ratings are usually assigned to network links by network administrators. They are typically arbitrary numeric values.

Delay

Routing delay refers to the length of time required to move a packet from source to destination through the network. Delay depends on many factors, including the bandwidth of intermediate network links, the port queues at each router along the way, network congestion on all intermediate network links and the physical distance to be traveled. Because it is a collection of several important variables, delay is a common and useful metric.

Bandwidth

Bandwidth refers to the available traffic capacity of a link. All other things being equal, a 10-Mbps Ethernet link would be preferable to a 64-kbps leased line. Although bandwidth is a rating of the maximum attainable throughput on a link, routes through links with greater bandwidth do not necessarily provide better routes than routes through slower links. If, for example, a faster link is much busier, the actual time required to send a packet to the destination may be greater through the fast link.

Load

Load refers to the degree to which a network resource (such as a router) is busy. Load can be calculated in a variety of ways, including CPU utilization and packets processed per second. Monitoring these parameters on a continual basis can itself be resource intensive.

Communication Cost

Communication cost is another important metric. Some companies may not care about performance as much as they care about operating expenditures. Even though line delay might be longer, they will send packets over their own lines rather than through public lines that will cost money for usage time.

12.3.2 Routing Protocols

A routing protocol communicates global topological information to each router, allowing it to make local decisions. Due to frequent changes and occasional failures of network elements, the routing protocol asynchronously updates routing tables at every router or switch controller. The updates are done through periodic exchange of routing table information between routers. This ensures that they have a consistent "view" of the network's topology.

The main requirements of any routing protocol are:

1. **Ensuring that tables at different routers are consistent :** Routing tables must be consistent so that routes can be found via a concatenation of local forwarding decisions.
2. **Minimizing the size of the routing table :** The size of the table affects the cost and efficiency of the routers. It is expected that the size of the routing tables will grow more slowly than the size of the network.
3. **Minimizing control messages :** Routing protocols require the use of control messages. Passing control messages is an overhead of routing operations and must be minimized (providing that their functionality is not reduced).
4. **Robustness :** Misrouting messages, so that they do not reach their destinations (entering a black hole) or causing loops and oscillations, must be kept to a minimum.

Routing protocols are broken into the Exterior Routing protocols that are used in the main part of the Internet and the Interior Routing protocols used within a trusted environment.

12.4 Routing Algorithms

Routing algorithms can be differentiated based on several key characteristics. First, the particular goals of the algorithm designer affect the operation of the resulting routing protocol. Second, there are various types of routing algorithms. Each algorithm has a different impact on network and router resources. Finally, routing algorithms use a variety of metrics that affect calculation of optimal routes.

Routing algorithms often have one or more of the following design goals:

- Optimality.
- Simplicity and low overhead.
- Robustness and stability.
- Rapid convergence.
- Flexibility.

Optimality

Optimality refers to the ability of the routing algorithm to select the "best" route. The best route depends on the metrics and metric weightings used to make the calculation. For example, one routing algorithm might use number of hops and delay, but might weight delay more heavily in the calculation. Naturally, routing protocols must strictly define their metric calculation algorithms.

Simplicity

Routing algorithms are also designed to be as simple as possible. In other words, the routing algorithm must offer its functionality efficiently, with a minimum of software and utilization overhead. Efficiency is particularly important when the software implementing the routing algorithm must run on a computer with limited physical resources.

Robustness

Routing algorithms must be robust. In other words, they should perform correctly in the face of unusual or unforeseen circumstances such as hardware failures, high load conditions and incorrect implementations. Because routers are located at network junction points, they can cause considerable problems when they fail. The best routing algorithms are often those that have withstood the test of time and proven stable under a variety of network conditions.

Rapid Convergence

Routing algorithms must converge rapidly. Convergence is the process of agreement, by all routers, on optimal routes. When a network event causes routes to either go down or become available, routers distribute routing update messages. Routing update messages permeate networks, stimulating recalculation of optimal routes and eventually causing all routers to agree on these routes. Routing algorithms that converge slowly can cause routing loops or network outages.

Figure 12.2 shows a routing loop. In this case, a packet arrives at Router 1 at time t1. Router 1 has already been updated and so knows that the optimal route to the destination calls for Router 2 to be the next stop. Router 1 therefore forwards the packet to Router 2. Router 2 has not yet been updated and so believes that the optimal next hop is Router 1. Router 2 therefore forwards the packet back to Router 1. The packet will continue to bounce back and forth between the two routers until Router 2 receives its routing update or until the packet has been switched the maximum number of times allowed.

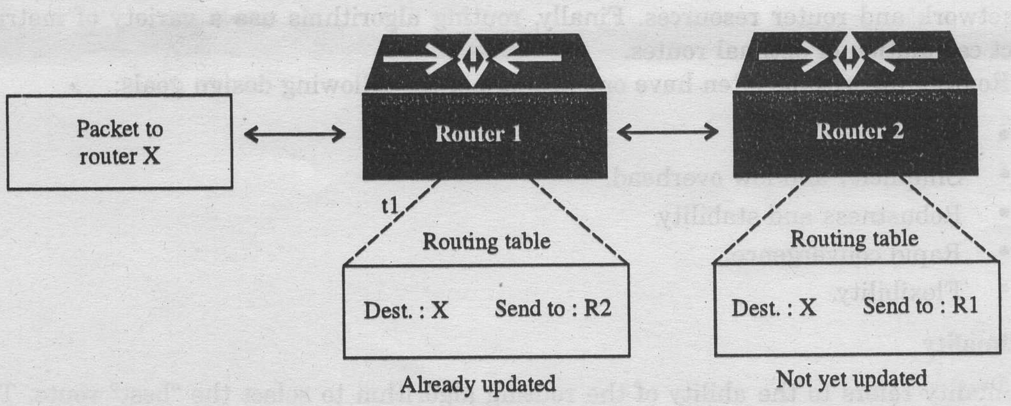


Fig. 12.2 Slow convergence and routing loops

Flexibility

Routing algorithms should also be flexible. In other words, routing algorithms should quickly and accurately adapt to a variety of network circumstances. For example, assume that a network segment has gone down. Many routing algorithms, on becoming aware of this problem, will quickly select the next-best path for all routes normally using that segment. Routing algorithms can be programmed to adapt to changes in network bandwidth, router queue size, network delay, and other variables.

Classes of Routing Algorithms

Routing algorithms can be classified as:

- Non-adaptive algorithms
- Adaptive algorithms

If a network is stable in its topology and traffic flow a **non-adaptive algorithm** is used. In this case all routes are computed initially and never change. This relieves the nodes from having to monitor changes and compute new routes. Alternately adaptive algorithms attempt to make routing decisions based on current traffic and topology.

Adaptive algorithms use such dynamic information as current topology, load, delay, etc. to select routes. Also known as dynamic routing. A dynamic algorithm can be run either periodically or in direct response to topology or link cost changes. While dynamic algorithms are more responsive to network changes, they are also more susceptible to problems such as routing loops and oscillation in routes. Adaptive algorithms can be further divided in the following types:

1. **Isolated :** Each router makes its routing decisions using only the local information it has on hand. Specifically, routers do not even exchange information with their neighbors.
2. **Centralized :** A centralized node makes all routing decisions. Specifically, the centralized node has access to global information.
3. **Distributed :** Algorithms that use a combination of local and global information.

In non-adaptive algorithms, routes never change once initial routes have been selected. Also called static routing. In static routing algorithms, routes change very slowly over time, often as a result of human intervention (for example, a human manually editing a router's forwarding table).

Obviously, adaptive algorithms are more interesting, as non-adaptive algorithms don't even make an attempt to handle failed links. Only two types of routing algorithms are typically used in the Internet: a dynamic global link state algorithm and a dynamic decentralized distance vector algorithm.

12.4.1 Shortest Path Algorithm

A path is the route taken by a data packet traveling between two network nodes. A network is typically highly complex so the correct path will not be obvious. Routing algorithms must overcome this complexity and make useful path choices. The notion of a *shortest path* is necessary when evaluating routing choices.

Dijkstra's algorithm (or variation) is used to find the shortest path. The basic idea of this algorithm is :

- Choose the source and put nodes connected to source in list to consider.
- From the list to consider choose the nearest node.

Distance Metrics

Shortest path routing requires that a cost metric be assigned to each link, so that that metric can be used to measure path lengths. Possible metrics: number of hops along a path, link capacities along a path, propagation delays along a path, per bit cost along a path or hybrids. In the following example we will consider two different metrics and find the path in each case.

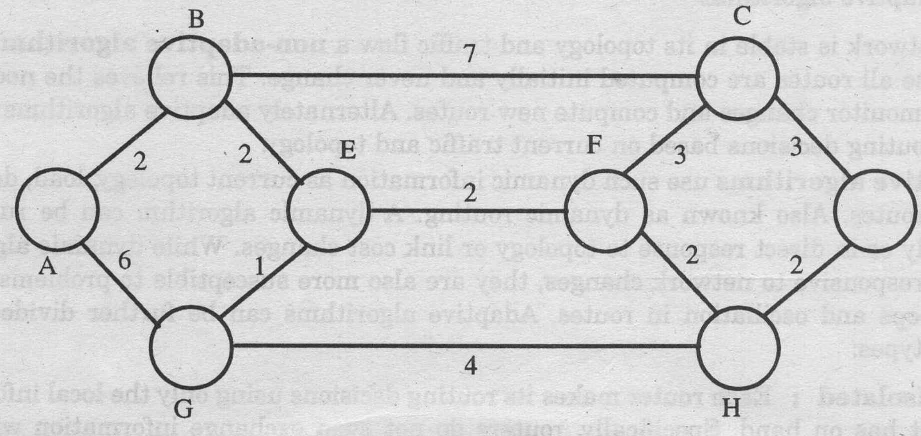


Fig. 12.3 Network with path lengths

An obvious metric of path length is number of hops. In the above example ABC is shorter than ABEF.

Many other metrics are commonly used; the ones chosen will greatly affect how optimal paths are found. The numbers on the links, on the network shown above, represent physical distance, it can be seen that ABEF is shorter than ABC. Other common metrics include mean queueing and transmission times, bandwidth, cost and average traffic. The labels in the above figure could be the result of a function of some or all of these metrics.

12.4.2 Flooding

Flooding is a form of isolated routing. It does not select a specific route. When a router receives a packet, it sends a copy of the packet out on each line (except the one on which it arrived):

- To prevent packets from looping forever, each router decrements a hop count contained in the packet header. Whenever the hop-count decrements to zero, the router discards the packet.

- To reduce looping even further:
  - □ add a sequence number to each packet's header.
  - □ each router maintains a private sequence number. When it sends a new packet, it copies the sequence number into the packet, and increments its private sequence number.
- For each source router S, a router:
  - □ keeps track of the highest sequence number seen from S.
  - □ whenever it receives a packet from S containing a sequence number lower than the one stored in its table, it discards the packet.
  - □ otherwise, it updates the entry for S and forwards the packet on.

Flooding wastes network resources by transmitting excessive numbers of packets.

Flooding is used mainly to transmit control information that must reach all nodes, but

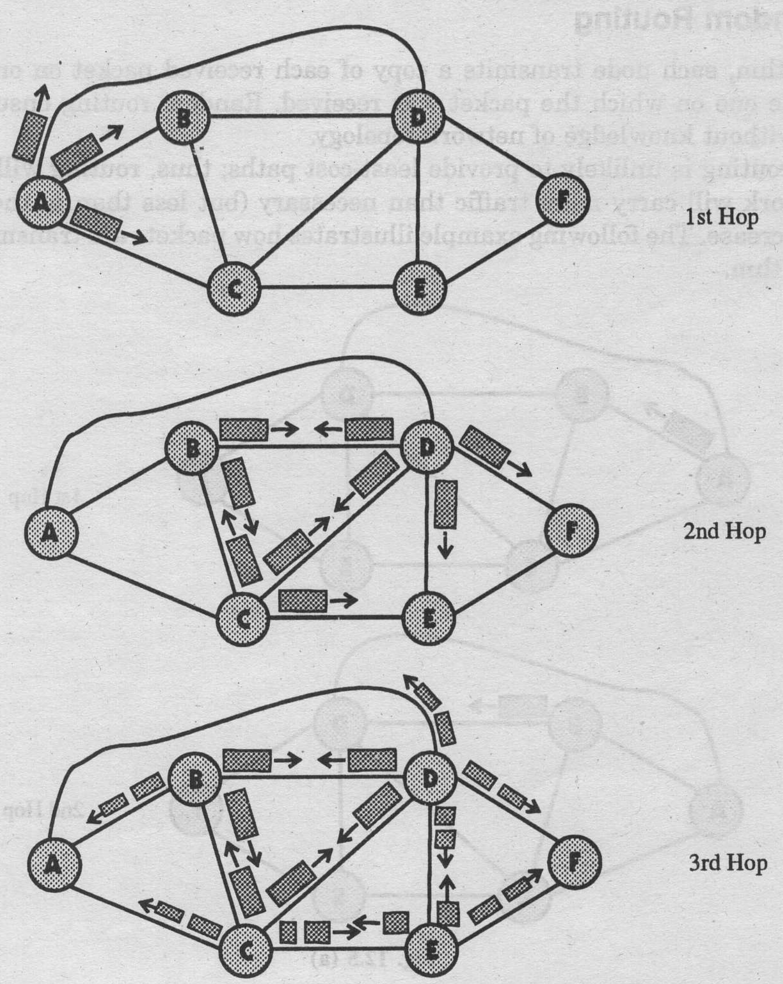


Fig. 12.4 An example illustrating flooding algorithm

when the network topology is not known. Figure 12.4 illustrates how packets are transmitted in flooding algorithm.

Flooding has several important uses:

- In military applications, the network must remain robust in the face of (extreme) hostility.
- Sending routing updates, because updates can't rely on the correctness of a router's routing table.
- Theoretical-chooses all possible paths, so it chooses the shortest one.

Another alternative of flooding is *selective flooding*, in which a router sends packets out only on those lines in the general direction of the destination. That is, don't send packets out on lines that clearly lead in the wrong direction.

12.4.3 Random Routing

In this algorithm, each node transmits a copy of each received packet on one outgoing link other than the one on which the packet was received. Random routing ensures good traffic distribution without knowledge of network topology.

Random routing is unlikely to provide least-cost paths; thus, routing will be sub-optimal and the network will carry more traffic than necessary (but less than in the flooding case). Delays will increase. The following example illustrates how packets are transmitted in random routing algorithm.

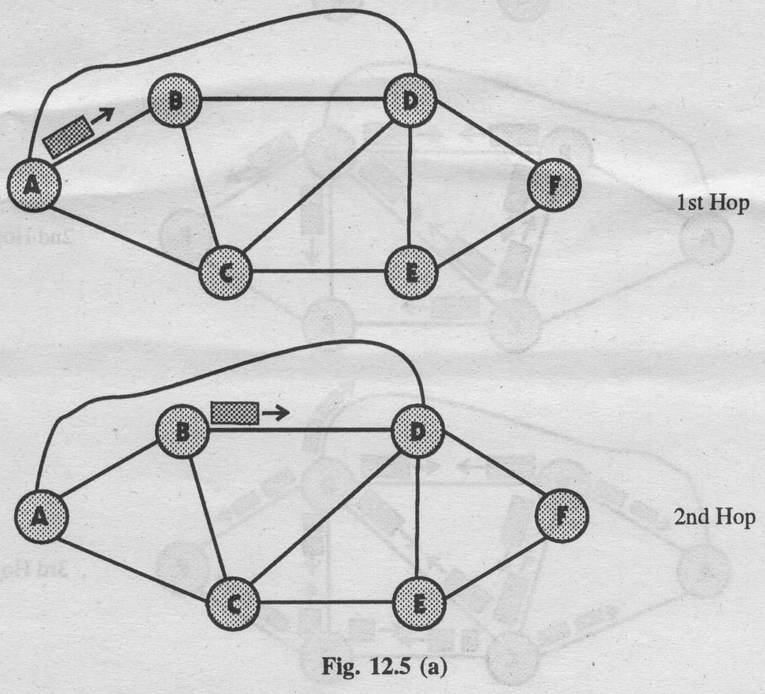


Fig. 12.5 (a)

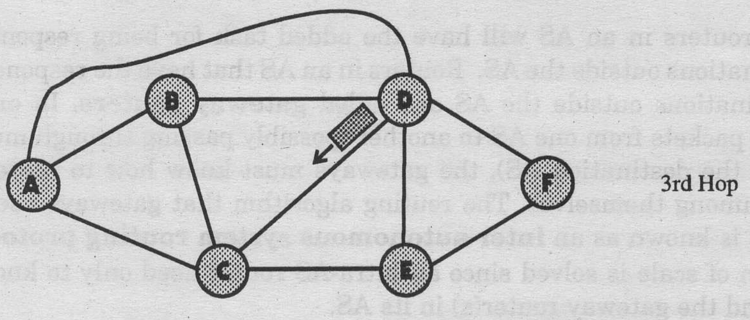


Fig. 12.5 (b) An example illustrating random routing

12.4.4 Flow Based Routing

The algorithms we've examined so far have been based on heuristics rather than on some sort of fundamental theory. Another approach is to develop a mathematical model of the network and define the routing problem as minimizing some objective function. The approach:

- Assume that traffic flows remain constant over time.
- Assume that we have an estimate of the traffic flowing between each router-router pair.
- Assume that we know the network's topology and the capacity of each link.

Our goal becomes: Find a set of routes that minimizes end-to-end delay. One possibility is to let cost be the sum across all links of the average delay multiplied by the amount of traffic carried by that link.

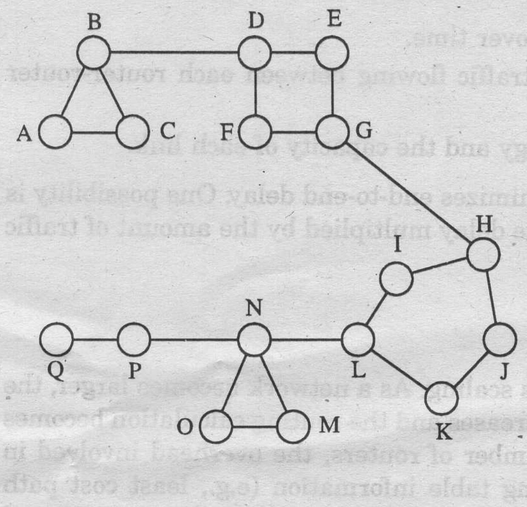
12.4.5 Hierarchical Routing

One of the fundamental issues regarding routing is scaling. As a network becomes larger, the amount of information that must be propagated increases and the routing calculation becomes increasingly expensive. Due to increase in the number of routers, the overhead involved in computing, storing and communicating the routing table information (e.g., least cost path changes) becomes prohibitive. Today's public Internet consists of millions of interconnected routers and more than 50 million hosts. Storing routing table entries to each of these hosts and routers would clearly require enormous amounts of memory. The overhead required to broadcast link state updates among millions of routers would leave no bandwidth left for sending the data packets! A solution is to have some routers do the routing for others: Hierarchical routing.

This problem can be solved by aggregating routers into "regions" or "autonomous systems" (ASs). Routers within the same AS all run the same routing algorithm (e.g., a LS or DV algorithm—described later on) and have full information about each other. The routing algorithm running within an autonomous system is called an **intra-autonomous system routing protocol**. It will be necessary, of course, to connect ASs to each other and thus one or more of the routers in an AS will have the added task for being responsible for routing packets to destinations outside the AS. Routers in an AS that have the responsibility of routing packets to destinations outside the AS are called **gateway routers**. In order for gateway routers to route packets from one AS to another (possibly passing through multiple other ASs before reaching the destination AS), the gateways must know how to route (i.e., determine routing paths) among themselves. The routing algorithm that gateways use to route among the various ASs is known as an **inter-autonomous system routing protocol**.

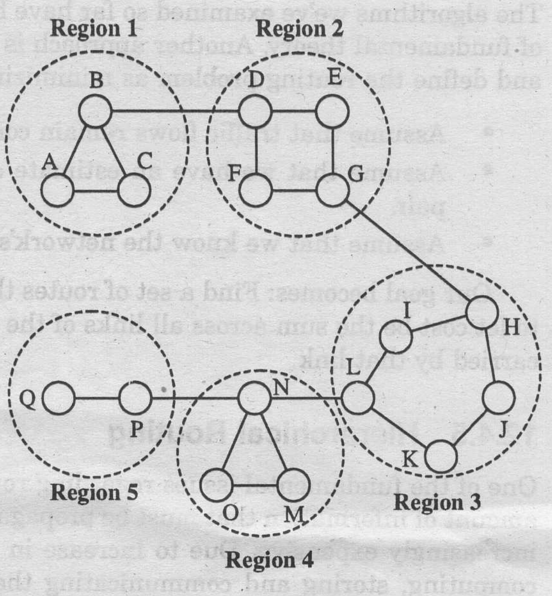
The problem of scale is solved since an intra-AS router need only to know about routers within its AS and the gateway router(s) in its AS.

Hierarchical routing is an approach that hides information from far-away nodes, reducing the amount of information a given router needs to perform routing. The following example explains hierarchical routing:



Simple Network

Fig 12.6 (a)



Network divided into 5 regions

Fig 12.6 (b)

The first table is A's routing table in case of simple network and second table shows hierarchical routing when network is divided into 5 regions.

In this type of routing, the tables can be summarized, so network efficiency improves. The above example shows two-level hierarchical routing. We can also use three- or four-level hierarchical routing.

In three-level hierarchical routing, the network is classified into a number of **clusters**. Each cluster is made up of a number of regions and each region contains a number or routers. Hierarchical routing is widely used in Internet routing and makes use of several routing protocols.

Destination	Line	Weight
A	—	—
B	B	1
C	C	1
D	B	2
E	B	3
F	B	3
G	B	4
H	B	5
I	C	5
J	C	6
K	C	5
L	C	4
M	C	4
N	C	3
O	C	4
P	C	2
Q	C	3

Destination	Line	Weight
A	—	—
B	B	1
C	C	1
Region 2	B	2
Region 3	C	2
Region 4	C	3
Region 5	C	4

Advantage : Scaling—Each router needs less information (table space) to perform routing.

Disadvantage : Sub optimal routes—The average path length increases because there may be a shorter path that bypasses the entry points, but we don't use it.

Hierarchical routing can be extended to multi-levels.

Example : Telephone system:

- Area code identifies a region.
- Area code plus the first three digits identify the central office within a specific region.

12.4.6 Broadcasting

Sending a packet to all destinations simultaneously is known as broadcasting. There are several ways to implement broadcasting:

1. **In broadcast networks, the implementation is trivial :** Designate a special address as the "all hosts address". Send a unicast packet to each destination. However, this approach makes poor use of resources.

2. **Flood packets to all nodes :** Flooding generates many packets and consumes too much bandwidth.
3. **Use multi-destination routing :** Each packet contains a list (or bitmap) of all destinations and when a router forwards a packet across two or more lines, it splits the packet and divides the destination addresses accordingly. This approach is similar to sending unicast packets, except that we don't send individual copies of each messages. However, the copy operations slow down the ability of a router to process many packets.
4. **Use a spanning tree :** If the network can be reduced to a tree (e.g., only one path between any two pairs of routers), copy a packet to each line of spanning tree except the one on which it arrived. It works only if each router uses the same spanning tree.
5. **Reverse Path Forwarding (RPF) :** Use a sink tree (assume sink/source trees are the same). When a packet arrives from router *X*, if the packet arrived on a line of the sink tree leading to *X*, the packet is traveling along the shortest path, so it must be the first copy we've seen. Copy the packet to all outgoing lines of the sink tree. If the packet arrives on another line, assume that the packet is a copy, it didn't arrive on the shortest path and discard it.

RPF is easy to implement and makes efficient use of bandwidth.

12.4.7 Dynamic Routing

The are two types of Dynamic Route Selection:

- Distance Vector Routing
- Link-State Routing

Basically, Distance Vector protocols determine best path on how far the destination is, while Link State protocols are capable of using more sophisticated methods taking into consideration link variables, such as bandwidth, delay, reliability and load. Distance Vector protocols judge best path on how far it is. Distance can be hops or a combination of metrics calculated to represent a distance value. Distance-vector routing protocols are simple and efficient in small networks and require little, if any management. However, they do not scale well and have poor convergence properties, which has led to the development of more complex but more scalable link-state routing protocols for use in large networks.

12.4.7.1 Distance vector routing

Distance-Vector is an algorithm that uses a direction to any link in the interconnection network to determine the best route. Distance Vector Routers advertise their presence to other routers on the network. Periodically, each router broadcasts their routing table information. Other Routers update their Routing Tables with this information. This process is effective but inefficient. Changes ripple through the network from router to router and take a while for all routers to become aware of the changes. Also, these broadcasts cause network traffic that can affect performance, more so on large networks with many routers.

The name distance vector is derived from the fact that routes are advertised as vectors of (distance, direction), where distance is defined in terms of a metric and direction is defined in terms of the next-hop router. For example, "Destination A is a distance of 5 hops away, in the direction of next-hop router *X*". As this statement implies, each router learns routes from its neighboring routers' perspectives and then advertises the routes from its own perspective. Because each router depends on its neighbors for information, which the neighbors in turn may have learned from their neighbors and so on, distance vector routing is sometimes facetiously referred to as "routing by rumor".

Distance-vector routing makes this assumption:

Each router knows the identity of every other router in the network.

The two popular Distance Vector routing protocols are the:

1. Routing Information Protocol (RIP)
2. Interior Gateway Routing Protocol (IGRP).

RIP : RIP is a relatively old but still commonly used interior gateway protocol created for use in small, homogeneous networks. It is a classical distance-vector routing protocol. With RIP, routers periodically exchange entire tables. Because this is inefficient, RIP is gradually being replaced by a newer protocol called Open Shortest Path First (OSPF).

RIP is a widely used protocol for managing router information within a self-contained network such as a corporate local area network or an interconnected group of such LANs. RIP uses broadcast User Datagram Protocol (UDP) data packets to exchange routing information. There are various metrics that Routing Protocols use to rate the value of different routes. The RIP uses the hop count that is the number of routers that can be traversed in a route.

IGRP : IGRP is a Cisco proprietary, distance-vector, routing protocol used by routers to exchange routing information. The goal of IGRP was to create a robust protocol for routing within an autonomous system (AS).

Working of distance-vector routing protocol

A very simple distance-vector routing protocol works as follows:

1. Initially, the router makes a list of which networks it can reach and how many hops it will cost. In the outset this will be the two or more networks to which this router is connected. This table is called a routing table. Example of routing table of a simple network is shown below:

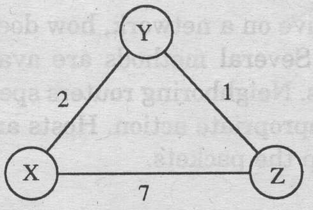


Fig. 12.7 A simple network

Routing table for X:

Destination	Cost	Next HOP
Y	2	Direct
Z	7	Direct

2. Periodically the routing table is shared with other routers on each of the connected networks via some specified inter-router protocol. This information is only shared in-between physically connected routers ("neighbors"), so routers on other networks are not reached by the new routing tables yet.
3. A new routing table is constructed based on the directly configured network interfaces, as before, with the addition of the new information received from other routers.
4. Bad routing paths are then purged from the new routing table. If two identical paths to the same network exist, only the one with the smallest hop-count is kept.
5. The new routing table is then communicated to all neighbors of this router. This way the routing information will spread and eventually all routers know the routing path to each network, which router it shall use to reach this network and to which router it shall route next.

Common Characteristics of Distance Vector Routing

A typical distance vector routing protocol uses a routing algorithm in which routers periodically send routing updates to all neighbors by broadcasting their entire route tables.

- **Periodic Updates**

Periodic updates means that at the end of a certain time period, updates will be transmitted. This period varies from protocol to protocol and usually in the range of 10 seconds to 90 seconds. At issue here is the fact that if updates are sent too frequently, congestion may occur; if updates are sent too infrequently, convergence time may be unacceptably high.

- **Neighbors**

In the context of routers, neighbors always mean routers sharing a common data link. A distance vector routing protocol sends its updates to neighboring routers and depends on them to pass the update information along to their neighbors. For this reason, distance vector routing is said to use hop-by-hop updates.

- **Broadcast Updates**

When a router first becomes active on a network, how does it find other routers and how does it announce its own presence? Several methods are available. The simplest is to send the updates to the broadcast address. Neighboring routers speaking the same routing protocol will hear the broadcasts and take appropriate action. Hosts and other devices uninterested in the routing updates will simply drop the packets.

- **Full Routing Table Updates**

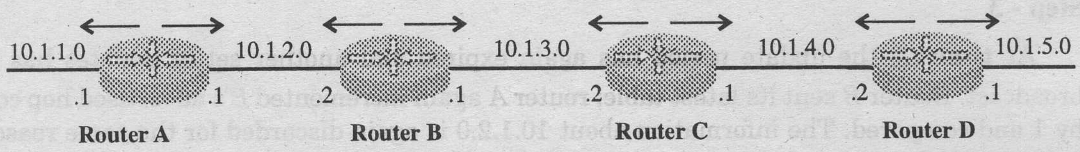
Most distance vector routing protocols take the very simple approach of telling their neighbors everything they know by broadcasting their entire route table. Neighbors receiving these updates glean the information they need and discard everything else.

Example of distance vector algorithm

Figure 12.8 shows a distance vector algorithm in action and is explained in the following four steps. In this example, the metric is hop count.

Step - 1

At time t_0, routers A through D have just become active. Looking at the route tables across the top row, at t_0 the only information any of the four routers has is its own directly connected networks. The tables identify these networks and indicate that they are directly connected by having no next-hop router and by having a hop count of 0. Each of the four routers will broadcast this information on all links.



	Router A			Router B			Router C			Router D		
	NET	VIA	HOPS	NET	VIA	HOPS	NET	VIA	HOPS	NET	VIA	HOPS
t_0	10.1.1.0	...	0	10.1.2.0	...	0	10.1.3.0	...	0	10.1.4.0	...	0
	10.1.2.0	...	0	10.1.3.0	...	0	10.1.4.0	...	0	10.1.5.0	...	0
t_1	10.1.1.0	...	0	10.1.2.0	...	0	10.1.3.0	...	0	10.1.4.0	...	0
	10.1.2.0	...	0	10.1.3.0	...	0	10.1.4.0	...	0	10.1.5.0	...	0
	10.1.3.0	10.1.2.2	1	10.1.1.0	10.1.2.1	1	10.1.2.0	10.1.3.1	1	10.1.3.0	10.1.4.1	1
				10.1.4.0	10.1.3.2	1	10.1.5.0	10.1.4.2	1			
t_2	10.1.1.0	...	0	10.1.2.0	...	0	10.1.3.0	...	0	10.1.4.0	...	0
	10.1.2.0	...	0	10.1.3.0	...	0	10.1.4.0	...	0	10.1.5.0	...	0
	10.1.3.0	10.1.2.2	1	10.1.1.0	10.1.2.1	1	10.1.2.0	10.1.3.1	1	10.1.3.0	10.1.4.1	1
	10.1.4.0	10.1.2.2	2	10.1.4.0	10.1.3.2	1	10.1.5.0	10.1.4.2	1	10.1.2.0	10.1.4.1	2
				10.1.5.0	10.1.3.2	2	10.1.1.0	10.1.3.1	2			
t_3	10.1.1.0	...	0	10.1.2.0	...	0	10.1.3.0	...	0	10.1.4.0	...	0
	10.1.2.0	...	0	10.1.3.0	...	0	10.1.4.0	...	0	10.1.5.0	...	0
	10.1.3.0	10.1.2.2	1	10.1.1.0	10.1.2.1	1	10.1.2.0	10.1.3.1	1	10.1.3.0	10.1.4.1	1
	10.1.4.0	10.1.2.2	2	10.1.4.0	10.1.3.2	1	10.1.5.0	10.1.4.2	1	10.1.2.0	10.1.4.1	2
	10.1.5.0	10.1.2.2	3	10.1.5.0	10.1.3.2	2	10.1.1.0	10.1.3.1	2	10.1.1.0	10.1.4.1	3

Fig. 12.8 Distance vector protocols converge hop-by-hop

Step - 2

At time t_1, the first updates have been received and processed by the routers. Look at router A's table at t_0. Router B's update to router *A* said that router *B* can reach networks 10.1.2.0 and 10.1.3.0, both 0 hops away. If the networks are 0 hops from *B*, they must be 1 hop from *A*. Router *A* incremented the hop count by 1 and then examined its route table. It already knew about 10.1.2.0 and the hop count (0) was less than the hop count *B* advertised, (1), so *A* disregarded that information.

Network 10.1.3.0 was new information, however, so *A* entered this in the route table. The source address of the update packet was router *B*'s interface (10.1.2.2) so that information is entered along with the calculated hop count.

Notice that the other routers performed similar operations at the same time t_1. Router *C*, for instance, disregarded the information about 10.1.3.0 from *B* and 10.1.4.0 from *C* but entered information about 10.1.2.0, reachable via *B*'s interface address 10.1.3.1 and 10.1.5.0, reachable via *C*'s interface 10.1.4.2. Both networks were calculated as 1 hop away.

Step - 3

At time t_2, the update period has again expired and another set of updates has been broadcast. Router *B* sent its latest table; router *A* again incremented *B*'s advertised hop counts by 1 and compared. The information about 10.1.2.0 is again discarded for the same reason as before. 10.1.3.0 is already known and the hop count hasn't changed, so that information is also discarded. 10.1.4.0 is new information and is entered into the route table.

Step - 4

At time t_3, the network is converged. Every router knows about every network, the address of the next-hop router for every network and the distance in hops to every network.

Thus Distance vector algorithms provide road signs to networks. They provide the direction and the distance, but no details about what lies along the route. And like the sign at the fork in the trail, they are vulnerable to accidental or intentional misdirection.

Problems of distance-vector protocols

Distance-vector protocols perform well in small internetworks fewer than 15 routers wide. Distance-vector protocols are preferred over link-state protocols in internetworks of this size due to their easy configuration. In addition, distance-vector protocols are easier to troubleshoot and understand. Problems do not arise until the internetwork becomes larger and more complex.

To understand the limitations of distance-vector protocols, you must first understand how they work. When a network change, such as a down link is detected, a distance-vector protocol updates its routing table and broadcasts the entire routing table to all other routers every 30 to 90 seconds when operating in a steady state. As the internetwork grows, you can expect to see the following problems when using distance-vector protocols:

- Convergence time increases.
- Routing loops occur.