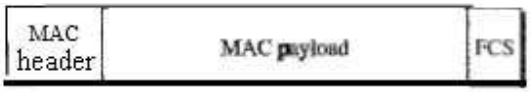
UNIT II

MEDIA ACCESS & INTERNET WORKING

Media access control – Ethernet (802.3) – Wireless LAN' s – 802.11 – Bluetooth – Switching and bridging – Basic Internetworking (IP, CIDR, ARP, DHCP,ICMP)

2.1. Media access control

- IEEE Project 802 has created a sublayer called media access control that defines the specific access method for each LAN.
- For example, it defines CSMA/CD as the media access method for Ethernet LANs and the token- passing method for Token Ring and Token Bus LANs.



MAC frame

Figure 2.1 *MAC frames*

- The lower sublayer that is mostly responsible for multiple access resolution is called the media access control (MAC) layer.
- When nodes or stations are connected and use a common link, called a *multipoint or broadcast link*,
- A multiple-access protocol is used to coordinate access to the link.
- Categorize MAC into three groups. Protocols belonging to each group are shown in Figure a.

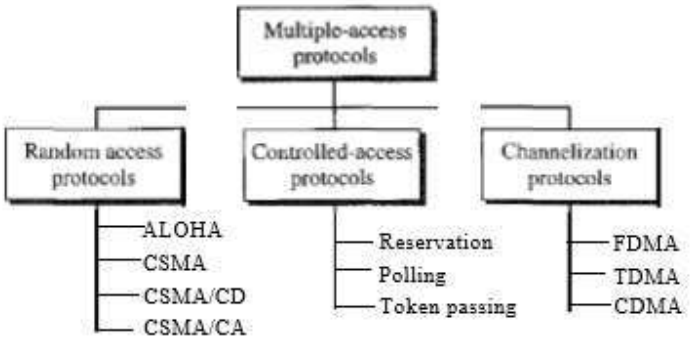


Figure A *Taxonomy ofmultiple-access protocols*

RANDOM ACCESS

- In random access or contention methods, no station is superior to another station and none is assigned the control over another.
- No station permits, or does not permit, another station to send.
- At each instance, a station that has data to send uses a procedure defined by the protocol to make a decision on whether or not to send. This decision depends on the state of the medium (idle or busy).
- Two features give this method its name.
  - ✓ First, there is no scheduled time for a station to transmit. Transmission is random among the stations. That is why these methods are called random access.
  - ✓ Second, no rules specify which station should send next. Stations compete with one another to access the medium.
- If more than one station tries to send, there is an access conflict-collision-and the frames will be either destroyed or modified. To avoid access conflict or to resolve it when it happens, each station follows a procedure that answers the following questions:

- When can the station access the medium?
- What can the station do if the medium is busy?
- How can the station determine the success or failure of the transmission?
- What can the station do if there is an access conflict?

1 ALOHA

- ALOHA, the earliest random access method, was developed at the University of Hawaii in early 1970.
- It was designed for a radio (wireless) LAN, but it can be used on any shared medium.
- The medium is shared between the stations.
- When a station sends data, another station may attempt to do so at the same time.
- The data from the two stations collide and become garbled.

Pure ALOHA

- The original ALOHA protocol is called pure ALOHA.
- The idea is that each station sends a frame whenever it has a frame to send. However, since there is only one channel to share, there is the possibility of collision between frames from different stations.
- There are four stations (unrealistic assumption) that contend with one another for access to the shared channel.
- The figure shows that each station sends two frames; there are a total of eight frames on the shared medium. Some of these frames collide because multiple frames are in contention for the shared channel.
- Figure B shows that only two frames survive: frame 1.1 from station 1 and frame 3.2 from station 3. We need to mention that even if one bit of a frame coexists on the channel with one bit from another frame, there is a collision and both will be destroyed. It is obvious that we need to resend the frames that have been destroyed during transmission.

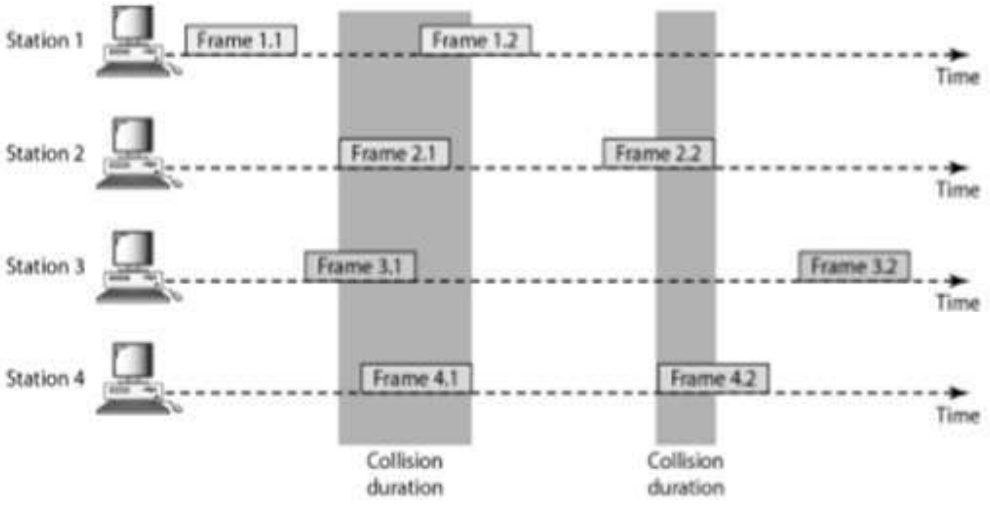


Figure B *Frames in a pure ALOHA network*

- The pure ALOHA protocol relies on acknowledgments from the receiver. When a station sends a frame, it expects the receiver to send an acknowledgment. If the acknowledgment does not arrive after a time-out period, the station assumes that the frame (or the acknowledgment) has been destroyed and resends the frame. A collision involves two or more stations. If all these stations try to resend their frames after the time-out, the frames will collide again.

- Pure ALOHA dictates that when the time-out period passes, each station waits a random amount of time before resending its frame. The randomness will help avoid more collisions. We call this time the back-off time T_B.
- Pure ALOHA has a second method to prevent congesting the channel with retransmitted frames. After a maximum number of retransmission attempts K_{max}'s a station must give up and try later.
- Figure C shows the procedure for pure ALOHA based on the above strategy.
  - ✓ The time-out period is equal to the maximum possible round-trip propagation delay, which is twice the amount oftime required to send a frame between the two most widely separated stations (2 x T_p).
  - ✓ The back-off time T_B is a random value that normally depends on K (the number of attempted unsuccessful transmissions). The formula for TB depends on the implementation. One common formula is the binary exponential back-off.

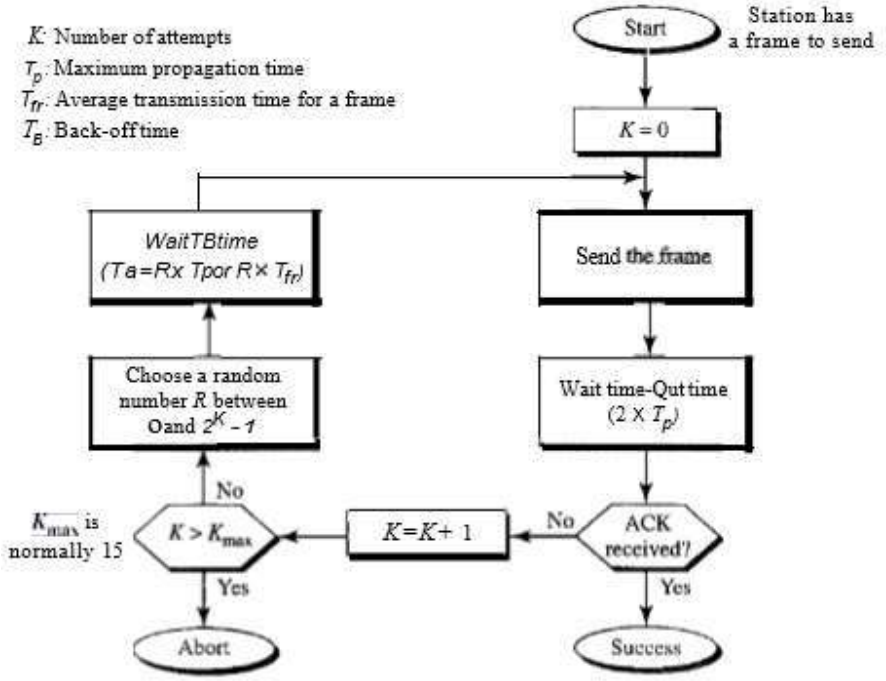


Figure C *Procedure for pure ALOHA protocol*

  - ✓ In this method, or each retransmission, a multiplier in the range 0 to 2^K - 1 is randomly chosen and multiplied by T_p (maximum propagation time) or T_{fr} (the average time required to send out a frame) to find T_B Note that in this procedure, the range of the random numbers increases after each collision. The value of K_{max} is usually chosen as 15.

Example

The stations on a wireless ALOHA network are a maximum of 600 km apart. If we assume that signals propagate at 3 x 108 m/s, we find Tp = (600 x 105) / (3 x 108) = 2ms. Now we can find the value of T_B for different values of K.

a) For K = 1, the range is {0, 1}. The station needs to generate a random number with a value of 0 or 1. This means that TB is either ms (0 * 2) or 2 ms (l * 2), based on the outcome of the random variable.

b) For K =2, the range is {0, 1, 2, 3}. This means that TB can be 0, 2, 4, or 6ms, based on the outcome of the random variable.

c) For K =3, the range is to, {1,2,3,4,5,6,7}. This means that TB can be 0,2,4, ... , 14ms, based on the outcome of the random variable.

d) We need to mention that if K > 10, it is normally set to 10.

Vulnerable time

- Let us find the length of time, the vulnerable time, in which there is a possibility of collision. We assume that the stations send fixed-length frames with each frame taking T_{fr} s to send.
- Figure D shows the vulnerable time for station A.

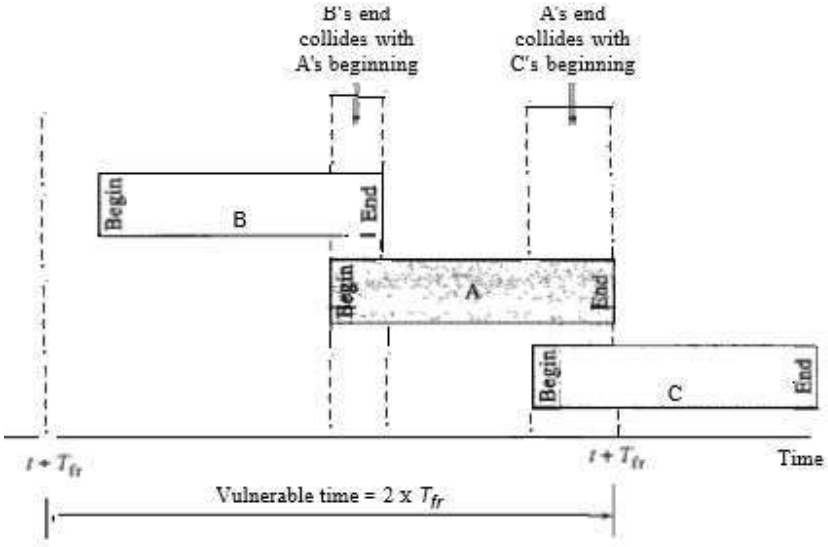


Figure D *Vulnerable time for pure ALOHA protocol*

- Station A sends a frame at time t. Now imagine station B has already sent a frame between t - T_{fr} and t. This leads to a collision between the frames from station A and station B.
- The end of B's frame collides with the beginning of A's frame. On the other hand, suppose that station C sends a frame between t and t + T_{fr}. Here, there is a collision between frames from station A and station C. The beginning of C's frame collides with the end of A's frame.
- Vulnerable time, during which a collision may occur in pure ALOHA, is 2 times the frame transmission time.

$$\textit{PureALOHA vulnerable time} = 2 \times T_{fr}$$

Example

A pure ALOHA network transmits 200-bit frames on a shared channel of 200 kbps. What is the requirement to make this frame collision-free?

Solution

Average frame transmission time T_{fr} is 200 bits/200 kbps or 1 ms. the vulnerable time is 2 x 1 ms =2 ms. This means no station should send later than 1 ms before this station starts transmission and no station should start sending during the one 1-ms period that this station is sending.

The throughput for pure ALOHA is S =G x e^{-2G}. The maximum throughput S_{max} =0.184 when G=(1/2).

Example

A pure ALOHA network transmits 200-bit frames on a shared channel of 200 kbps. What is the throughput if the system (all stations together) produces

a. 1000 frames per second
b. 500 frames per second
c. 250 frames per second

Solution

The frame transmission time is 2001200 kbps or 1 ms.

a. If the system creates 1000 frames per second, this is 1 frame per millisecond. The load is 1. In this case S =G x e^{-2G} or S =0.135 (13.5 percent). This means that the throughput is 1000 X 0.135 =135 frames. Only 135 frames out of 1000 will probably survive.
b. If the system creates 500 frames per second, this is (1/2) frame per millisecond. The load is (112). In this case S = G x e^{-2G} or S = 0.184 (18.4 percent). This means that the throughput is 500 x 0.184 =92 and that only 92 frames out of500 will probably survive. Note that this is the maximum throughput case, percentagewise.
c. If the system creates 250 frames per second, this is (1/4) frame per millisecond. The load is (1/4). In this case S =G x e^{-2G} or S =0.152 (15.2 percent). This means that the throughput is 250 x 0.152 = 38. Only 38 frames out of250 will probably survive.

SlottedALOHA

- Pure ALOHA has a vulnerable time of 2 x T_{fr}. This is so because there is no rule that defines when the station can send.
- A station may send soon after another station has started or soon before another station has finished. Slotted ALOHA was invented to improve the efficiency of pure ALOHA.
- In slotted ALOHA we divide the time into slots of T_{fr}'s and force the station to send only at the beginning of the time slot. Figure E shows an example of frame collisions in slotted ALOHA.

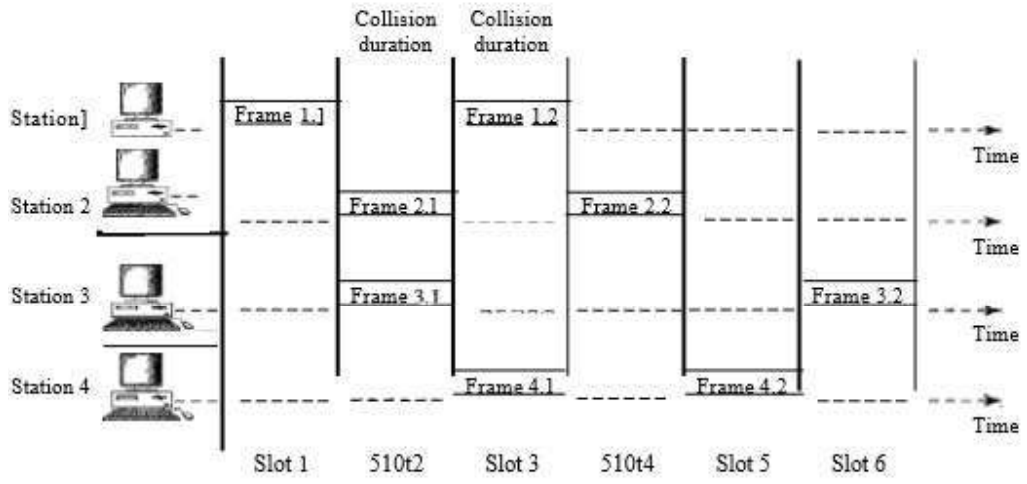


Figure E *Frames in a slottedALOHA network*

- Because a station is allowed to send only at the beginning of the synchronized time slot, if a station misses this moment, it must wait until the beginning of the next time slot. This means that the station which started at the beginning of this slot has already finished sending its frame.
- Figure F shows that the vulnerable time for slotted ALOHA is one-half that of pure ALOHA.

Slotted ALOHA vulnerable time =T_{fr}

- Throughput: It can be proved that the average number of successful transmissions for slotted ALOHA is S = G x e^{-G}. The maximum throughput S_{max} is 0.368, when G = 1. 370

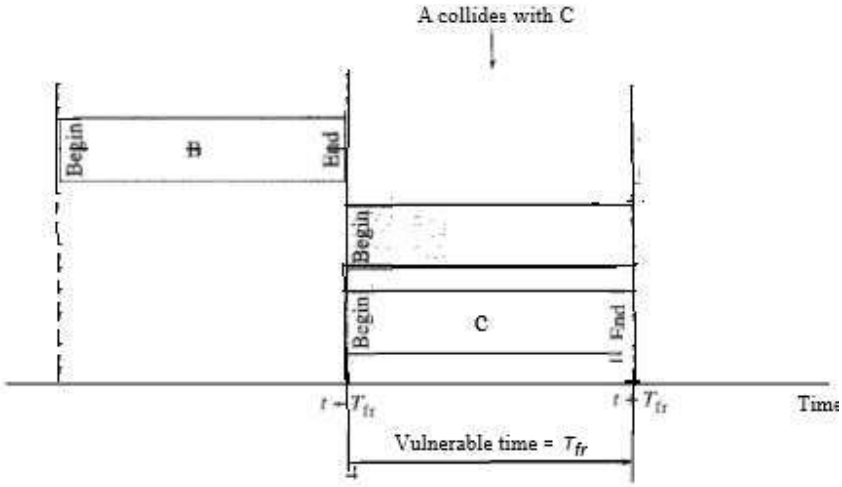


Figure F *Vulnerable time for slotted ALOHA protocol*

Example

A slotted ALOHA network transmits 200-bit frames using a shared channel with a 200-kbps bandwidth. Find the throughput if the system (all stations together) produces

a. 1000 frames per second
b. 500 frames per second
c. 250 frames per second

Solution

This situation is similar to the previous exercise except that the network is using slotted ALOHA instead ofpure ALOHA. The frame transmission time is 200/200 kbps or 1ms.

a. In this case G is 1. So S =G x e^{-G} or S =0.368 (36.8 percent). This means that the throughput is 1000 x 0.0368 =368 frames. Only 368 out of 1000 frames will probably survive. Note that this is the maximum throughput case, percentagewise.

b. Here G is 1/2. In this case S =G x e-G or S =0.303 (30.3 percent). This means that the throughput is 500 x 0.0303 =151. Only 151 frames out of500 will probably survive.

c. Now G is 1. In this case S =G x e^{-G} or S =0.195 (19.5 percent). This means that the throughput 4 is 250 x 0.195 = 49. Only 49 frames out of250 will probably survive.

2. Carrier Sense Multiple Access (CSMA)

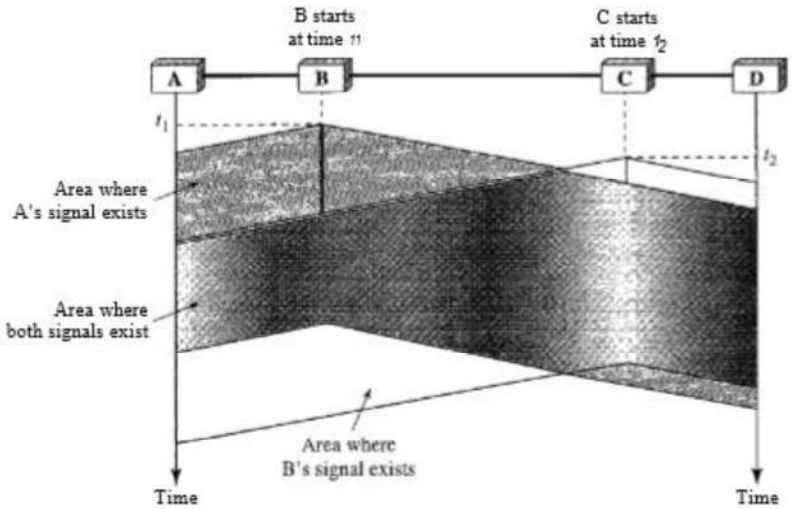


Figure G *Space/time model of the collision in CSMA*

- To minimize the chance of collision and, therefore, increase the performance, the CSMA method was developed. The chance of collision can be reduced if a station senses the medium before trying to use it.
- CSMA is based on the principle "sense before transmit" or "listen before talk." CSMA can reduce the possibility of collision, but it cannot eliminate it.
- The reason for this is shown in Figure G, a space and time model of a CSMA network. Stations are connected to a shared channel (usually a dedicated medium).
- The possibility of collision still exists because of propagation delay; a station may sense the medium and find it idle, only because the first bit sent by another station has not yet been received. At time t1station B senses the medium and finds it idle, so it sends a frame. At time t2 (t2> t1) station C senses the medium and finds it idle because, at this time, the first bits from station B have not reached station C. Station C also sends a frame. The two signals collide and both frames are destroyed.

Vulnerable Time

- The vulnerable time for CSMA is the propagation time T_p. This is the time needed for a signal to propagate from one end of the medium to the other.
- When a station sends a frame, and any other station tries to send a frame during this time, a collision will result.
- But if the first bit of the frame reaches the end of the medium; every station will already have heard the bit and will refrain from sending.
- Figure H shows the worst case. The leftmost station A sends a frame at time t1 which reaches the rightmost station D at time t1 + T_p. The gray area shows the vulnerable area in time and space.

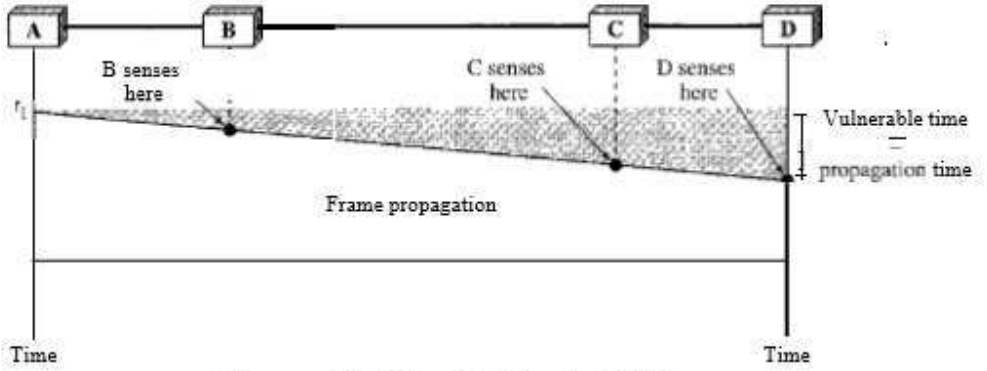


Figure H *Vulnerable time in CSMA*

Persistence Methods

- What should a station do if the channel is busy? What should a station do if the channel is idle? Three methods have been devised to answer these questions: the 1-persistent method, the non persistent method, and the p-persistent method.
- Figure I shows the behavior of three persistence methods when a station finds a channel busy.
- *1-Persisten*t: The 1-persistent method is simple and straightforward. In this method, after the station finds the line idle, it sends its frame immediately (with probability I). This method has the highest chance of collision because two or more stations may find the line idle and send their frames immediately.

- *Nonpersistent* : In the nonpersistent method, a station that has a frame to send senses the line. If the line is idle, it sends immediately. If the line is not idle, it waits a random amount of time and then senses the line again. The non persistent approach reduces the chance ofcollision because it is unlikely that two or more stations will wait the same amount of time and retry to send simultaneously. However, this method reduces the efficiency of the network because the medium remains idle when there may be stations with frames to send.
- *p-Persistent* : The p-persistent method is used if the channel has time slots with a slot duration equal to or greater than the maximum propagation time. The p-persistent approach combines the advantages of the other two strategies. It reduces the chance of collision and improves efficiency. In this method, after the station finds the line idle it follows these steps:

1. With probabilityp, the station sends its frame.
2. With probability q = 1 - p, the station waits for the beginning of the next time slot and checks the line again.
   a. If the line is idle, it goes to step 1.
   b. If the line is busy, it acts as though a collision has occurred and uses the back- off procedure.

- Figure J shows the flow diagram of thee persistent methods.

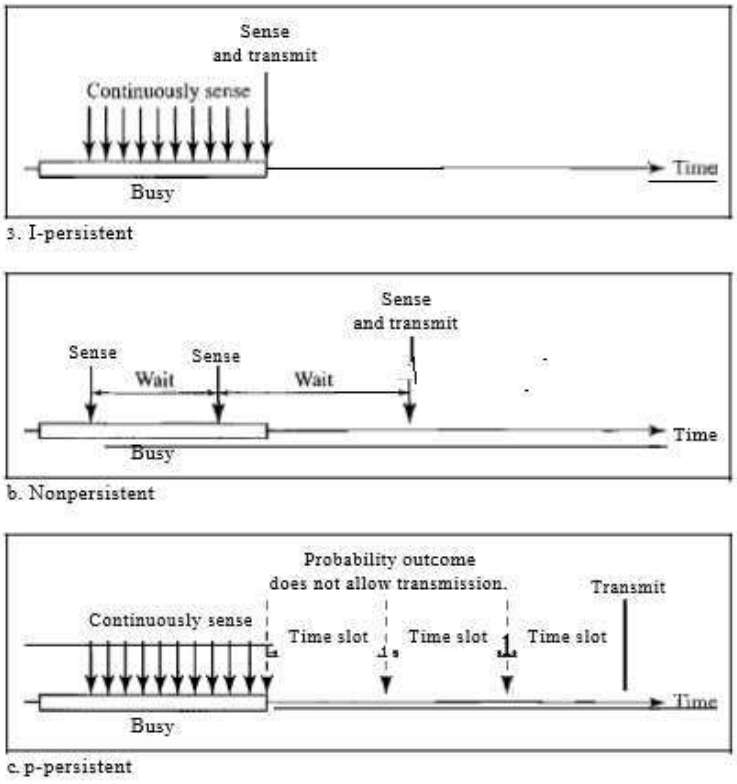


Figure I *Behavior ofthree persistence methods*

Carrier Sense Multiple Access with Collision Detection (CSMA/CD)

- Carrier sense multiple access with collision detection (CSMA/CD) augments the algorithm to handle the collision.
- In this method, a station monitors the medium after it sends a frame to see if the transmission was successful. If so, the station is finished. If, however, there is a collision, the frame is sent again.

- Let us look at the first bits transmitted by the two stations involved in the collision. Although each station continues to send bits in the frame until it detects the collision, we show what happens as the first bits collide. In Figure K, stations A and C are involved in the collision.

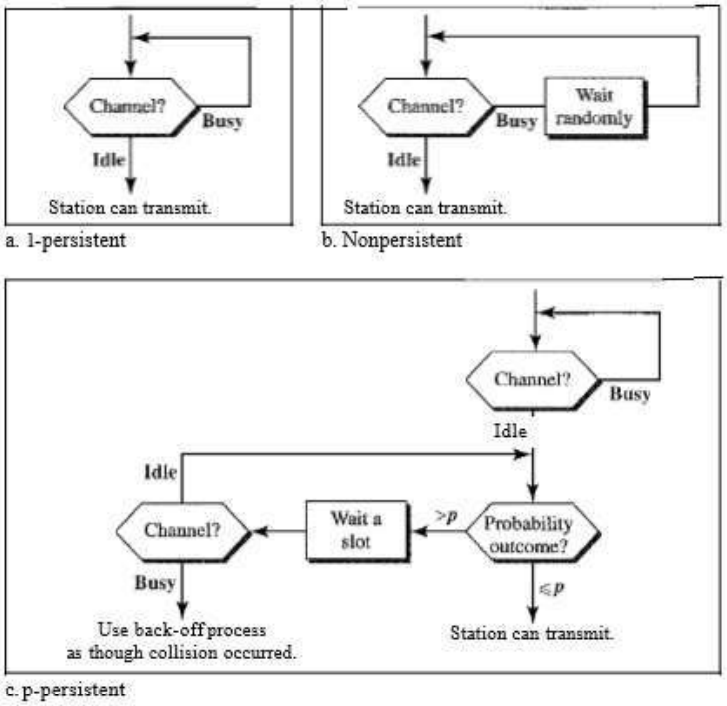


Figure J *Flow diagram for three persistence methods*

  - ✓ At time t1, station A has executed its persistence procedure and starts sending the bits of its frame.
  - ✓ At time t2, station C has not yet sensed the first bit sent by A. Station C executes its persistence procedure and starts sending the bits in its frame, which propagate both to the left and to the right.
  - ✓ The collision occurs sometime after time t2. Station C detects a collision at time t3 when it receives the first bit of A's frame. Station C immediately (or after a short time, but we assume immediately) aborts transmission. Station A detects collision at time t4 when it receives the first bit of C's frame; it also immediately aborts transmission. Looking at the figure, we see that A transmits for the duration t4 – t1; C transmits for the duration t3 - t2.
  - ✓ The length of any frame divided by the bit rate in this protocol must be more than either of these durations. At time t4, the transmission of A's frame, though incomplete, is aborted; at time t3, the transmission of B's frame, though incomplete, is aborted.
  - ✓ Now that we know the time durations for the two transmissions, we can show a more complete graph in Figure L.

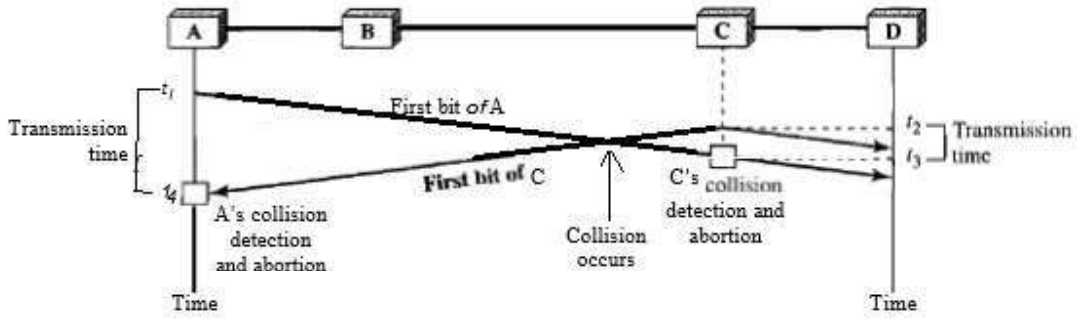


Figure K *Collision of the first bit in CSMA/CD*

Minimum Frame Size

- Before sending the last bit of the frame, the sending station must detect a collision, if any, and abort the

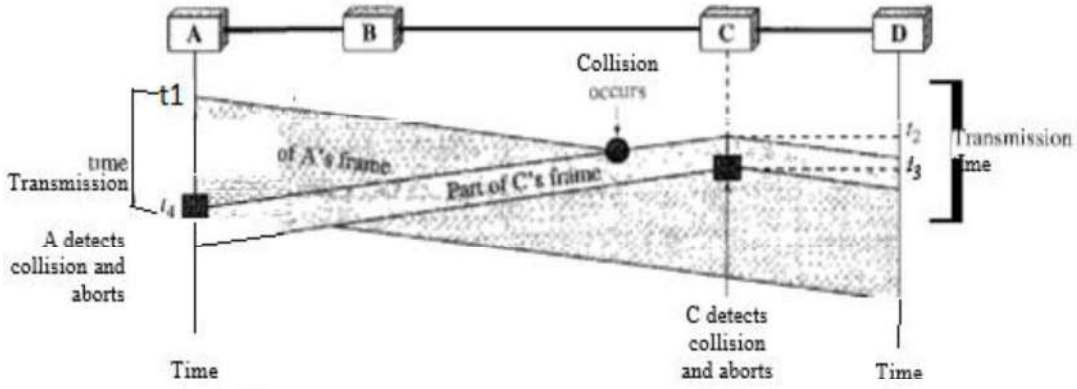


Figure L *Collision and abortion in CSMAICD*

transmission. This is so because the station, once the entire frame is sent, does not keep a copy of the frame and does not monitor the line for collision detection.

- The frame transmission time T_{fr} must be at least two times the maximum propagation time Tp.
- To under- stand the reason, let us think about the worst-case scenario. If the two stations involved in a collision are the maximum distance apart, the signal from the first takes time T_p to reach the second, and the effect of the collision takes another time T_p to reach the first. So the requirement is that the first station must still be transmitting after $2T_p$.

Example

A network using CSMA/CD has a bandwidth of 10 Mbps. If the maximum propagation time (including the delays in the devices and ignoring the time needed to send a jamming signal, as we see later) is 25.611S, what is the minimum size of the frame?

Solution

The frame transmission time is T_{fr} =2 x T_p =51.2 µs. This means, in the worst case, a station needs to transmit for a period of 51.2 µs to detect the collision. The minimum size of the frame is 10 Mbps x 51.2µs =512 bits or 64 bytes.

Procedure

- Now let us look at the flow diagram for CSMA/CD in Figure M.
- Three *differences from ALOHA*.
  - ✓ The first difference is the addition of the persistence process. We need to sense the channel before we start sending the frame by using one of the persistence processes we discussed previously (nonpersistent, I-persistent, or p-persistent). The corresponding box can be replaced by one of the persistence processes shown in Figure J.
  - ✓ The second difference is the frame transmission. In ALOHA, we first transmit the entire frame and then wait for an acknowledgment. In CSMA/CD, transmission and collision detection is a continuous process. We do not send the entire frame and then look for a collision. The station transmits and receives continuously and simultaneously (using two different ports).
    - o We use a loop to show that transmission is a continuous process.

- We constantly monitor in order to detect one of two conditions: either trans- mission is finished or a collision is detected. Either event stops transmission.
- When we come out of the loop, if a collision has not been detected, it means that transmission is complete; the entire frame is transmitted. Otherwise, a collision has occurred.

✓ The third difference is the sending of a short *jamming signal* that enforces the collision in case other stations have not yet sensed the collision.

Energy Level

The level of energy in a channel can have three values: zero, normal, and abnormal.

- *At the zero level*, the channel is idle.
- *At the normal level*, a station has successfully captured the channel and is sending its frame.
- *At the abnormal level*, there is a collision and the level of the energy is twice the normal level. A station that has a frame to send or is sending a frame needs to monitor the energy level to determine if the channel is idle, busy, or in collision mode. Figure N shows the situation.

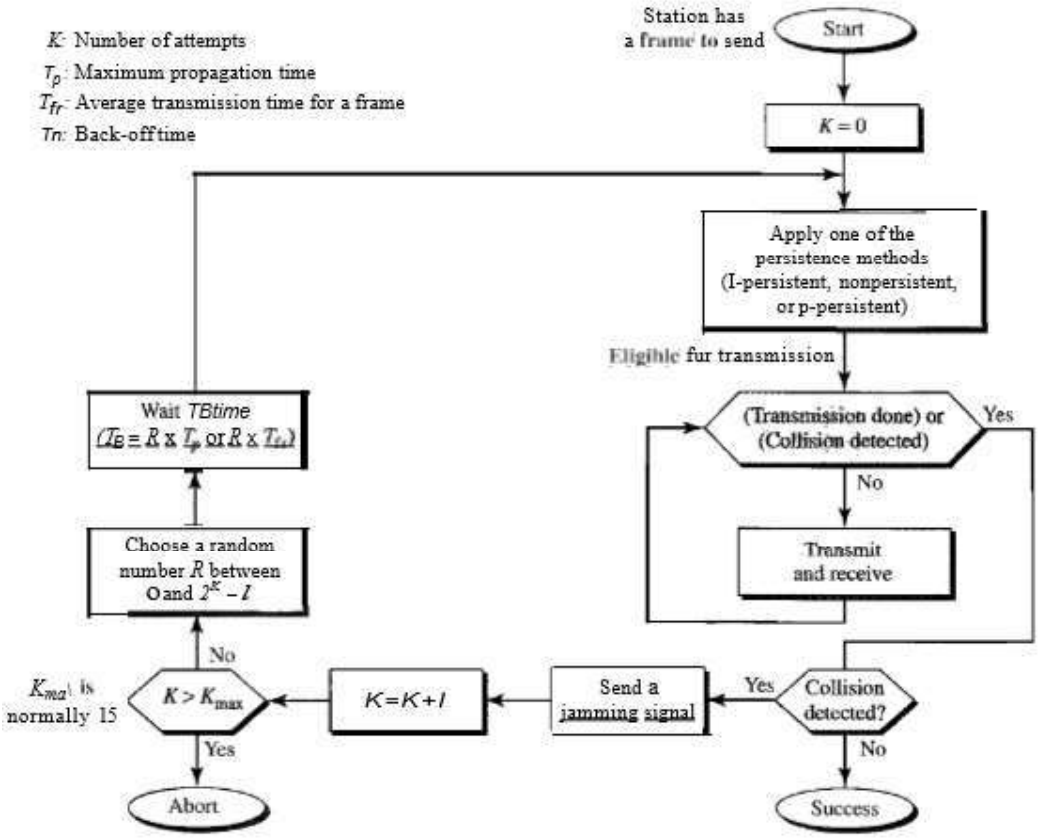


Figure M *Flow diagramfor the CSMAlCD*

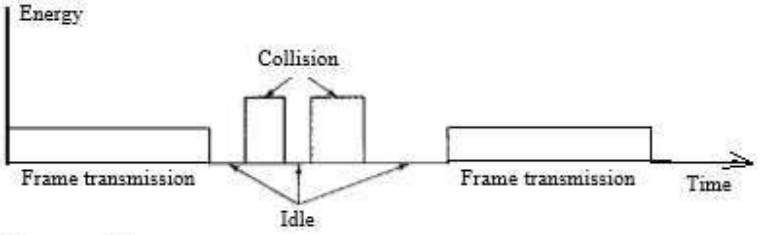


Figure N *Energy level during transmission, idleness, or collision*

Throughput : The throughput of CSMA/CD is greater than that of pure or slotted ALOHA. The maximum throughput occurs at a different value of G and is based on the persistence methods.

- The value of p in the p-persistent approach.
- For *I-persistent* method the maximum throughput is around 50 percent when G =1.
- For *non persistent* method, the maximum throughput can go up to 90 percent when G is between 3 and 8.

Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)

- The basic idea behind CSMA/CD is that a station needs to be able to receive while transmitting to detect a collision.
- When there is no collision, the station receives one signal: its own signal. When there is a collision, the station receives two signals: its own signal and the signal transmitted by a second station. To distinguish between these two cases, the received signals in these two cases must be significantly different.
- In a wired network, the received signal has almost the same energy as the sent signal because either the length of the cable is short or there are repeaters that amplify the energy between the sender and the receiver. This means that in a collision, the detected energy almost doubles.
- In a wireless network, much of the sent energy is lost in transmission. The received signal has very little energy.
- Therefore, a collision may add only 5 to 10 percent additional energy. This is not useful for effective collision detection.
- We need to avoid collisions on wireless networks because they cannot be detected. Carrier sense multiple access with collision avoidance (CSMA/CA) was invented for this network. Collisions are avoided through the use of CSMA/CA's three strategies: the interframe space, the contention window, and acknowledgments, as shown in Figure O.

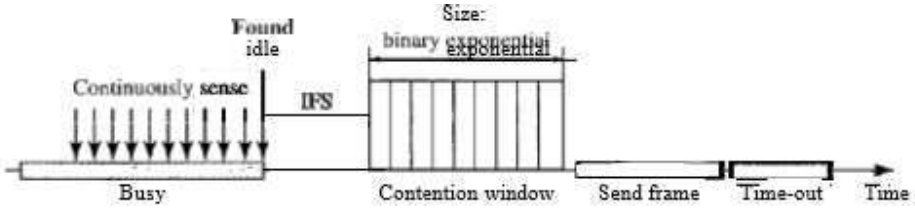


Figure O *Timing in CSMAICA*

Inter Frame Space (IFS)

- First, collisions are avoided by deferring transmission even if the channel is found idle. When an idle channel is found, the station does not send immediately. It waits for a period of time called the inter frame space or IFS. Even though the channel may appear idle when it is sensed, a distant station may have already started transmitting. The distant station's signal has not yet reached this station.
- The IFS time allows the front of the transmitted signal by the distant station to reach this station. If after the IFS time the channel is still idle, the station can send, but it still needs to wait a time equal to the contention time (described next). The IFS variable can also be used to prioritize stations or frame types. For example, a station that is assigned a shorter IFS has a higher priority.
- In CSMA/CA, the IFS can also be used to define the priority of a station or a frame.

Contention Window

- The contention window is an amount of time divided into slots. A station that is ready to send chooses a random number of slots as its wait time. The number of slots in the window changes according to the binary exponential back-off strategy. This means that it is set to one slot the first time and then doubles each time the station cannot detect an idle channel after the IFS time. This is very similar to the p-persistent method except that a random outcome defines the number of slots taken by the waiting station.
- In CSMA/CA, if the station finds the channel busy, it does not restart the timer of the contention window; it stops the timer and restarts it when the channel becomes idle.

Acknowledgment

The data may be corrupted during the transmission. The positive acknowledgment and the timeout timer can help guarantee that the receiver has received the frame.

Procedure

Figure P shows the procedure. The channel needs to be sensed before and after the IFS. The channel also needs to be sensed during the contention time. For each time slot of the contention window, the channel is sensed.

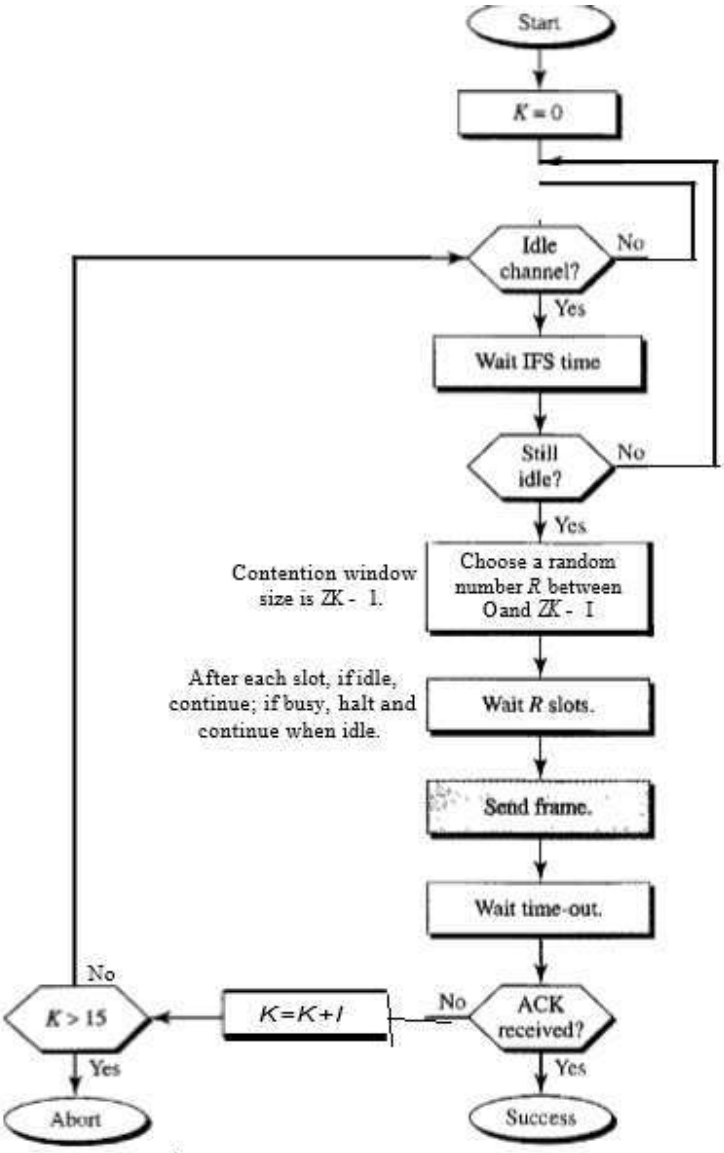


Figure P *Flow diagramfor CSMAICA*

If it is found idle, the timer continues; if the channel is found busy, the timer is stopped and continues after the timer becomes idle again.

3. CONTROLLED ACCESS

In controlled access, the stations consult one another to find which station has the right to send. A station cannot send unless it has been authorized by other stations.

1. Reservation

- In the reservation method, a station needs to make a reservation before sending data.
- Time is divided into intervals. In each interval, a reservation frame precedes the data frames sent in that interval.
- If there are N stations in the system, there are exactly N reservation mini slots in the reservation frame. Each minislot belongs to a station. When a station needs to send a data frame, it makes a reservation in its own minislot.
- The stations that have made reservations can send their data frames after the reservation frame.
- Figure Q shows a situation with five stations and a five minislot reservation frame.
  - ✓ In the first interval, only stations 1, 3, and 4 have made reservations.
  - ✓ In the second interval, only station 1 has made a reservation.

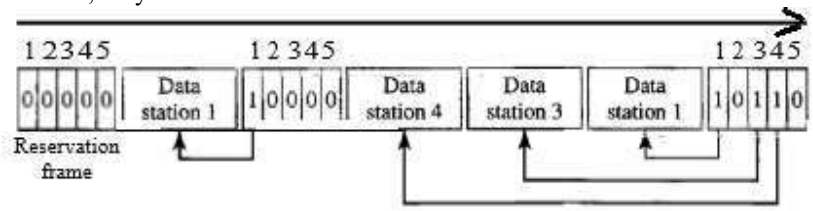


Figure Q *Reservation access method*

a. Polling

- Polling works with topologies in which one device is designated as a primary station and the other devices are secondary stations.
- All data exchanges must be made through the primary device even when the ultimate destination is a secondary device. The primary device controls the link; the secondary devices follow its instructions. It is up to the primary device to determine which device is allowed to use the channel at a given time. The primary device, therefore, is always the initiator of a session (see Figure R).

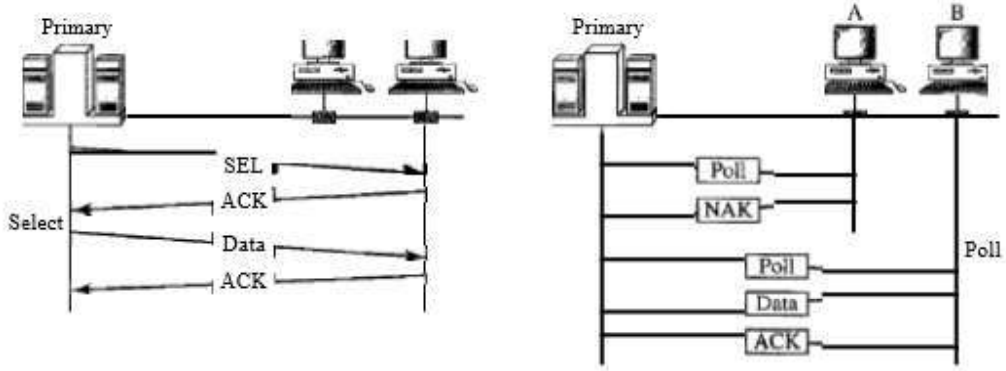


Figure R *Select and poll functions in polling access method*

- If the primary wants to receive data, it asks the secondaries if they have anything to send; this is called poll function. If the primary wants to send data, it tells the secondary to get ready to receive; this is called select function.

b. Select

- The select function is used whenever the primary device has something to send.
- The primary controls the link. If the primary is neither sending nor receiving data, it knows the link is available. If it has something to send, the primary device sends it. What it does not know, however, is whether the target device is prepared to receive. So the primary must alert the secondary to the upcoming transmission and wait for an acknowledgment of the secondary's ready status.
- Before sending data, the primary creates and transmits a select (SEL) frame, one field of which includes the address ofthe intended secondary.

c. Poll

- The poll function is used by the primary device to solicit transmissions from the secondary devices.
- When the primary is ready to receive data, it must ask (poll) each device in turn if it has anything to send.
- When the first secondary is approached, it responds either with a NAK frame if it has nothing to send or with data (in the form of a data frame) if it does.
- If the response is negative (a NAK frame), then the primary polls the next secondary in the same manner until it finds one with data to send.
- When the response is positive (a data frame), the primary reads the frame and returns an acknowledgment (ACK frame), verifying its receipt.

2. Token Passing

- In the token-passing method, the stations in a network are organized in a logical ring(there is a predecessor and a successor).
- The predecessor is the station which is logically before the station in the ring; the successor is the station which is after the station in the ring. The current station is the one that is accessing the channel now.
- The right to this access has been passed from the predecessor to the current station. The right will be passed to the successor when the current station has no more data to send. But how is the right to access the channel passed from one station to another? In this method, a special packet called a *token* circulates through the ring. The possession of the token gives the station the right to access the channel and send its data.
- When a station has some data to send, it waits until it receives the token from its predecessor. It then holds the token and sends its data.
- When the station has no more data to send, it releases the token, passing it to the next logical station in the ring. The station cannot send data until it receives the token again in the next round. In this process, when a station receives the token and has no data to send, it just passes the data to the next station.
- Token management is needed for this access method. Stations must be limited in the time they can have possession of the token. The token must be monitored to ensure it has not been lost or destroyed.
- For example, if a station that is holding the token fails, the token will disappear from the network. Another function of token management is to assign priorities to the stations and to the types of data being transmitted. And finally, token management is needed to make low-priority stations release the token to high- priority stations.

Logical Ring

- In a token-passing network, stations do not have to be physically connected in a ring; the ring can be a logical one.

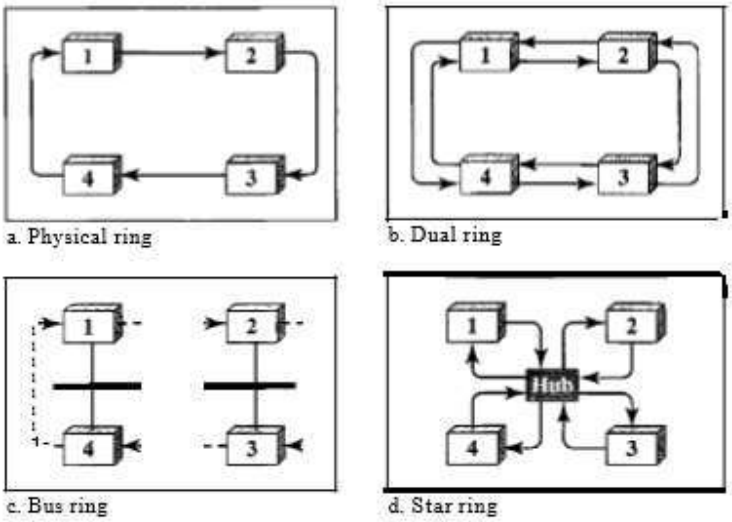


Figure S *Logical ring and physical topology in token-passing access method*

- Figure S show four different physical topologies that can create a logical ring.
  a. *Physical ring :* In the physical ring topology, when a station sends the token to its successor, the token cannot be seen by other stations; the successor is the next one in line. This means that the token does not have to have the address of the next successor. The problem with this topology is that if one of the links-the medium between two adjacent stations fails, the whole system fails.
  b. *Bus ring :* In the bus ring topology, also called a token bus, the stations are connected to a single cable called a bus. They, however, make a logical ring, because each station knows the address of its successor (and also predecessor for token management purposes). When a station has finished sending its data, it releases the token and inserts the address of its successor in the token. Only the station with the address matching the destination address of the token gets the token to access the shared media. The Token Bus LAN, standardized by IEEE, uses this topology.
  c. *Dual ring :* The dual ring topology uses a second (auxiliary) ring which operates in the reverse direction compared with the main ring. The second ring is for emergencies only (such as a spare tire for a car). If one of the links in the main ring fails, the system automatically combines the two rings to form a temporary ring. After the failed link is restored, the auxiliary ring becomes idle again. Note that for this topology to work, each station needs to have two transmitter ports and two receiver ports. The high-speed Token Ring networks called FDDI (Fiber Distributed Data Interface) and CDDI (Copper Distributed Data Interface) use this topology.
  d. *Star ring :* In a star ring topology, the physical topology is a star. There is a hub, however, that acts as the connector. The wiring inside the hub makes the ring; the stations are connected to this ring through the two wire connections. This topology makes the network less prone to failure because if a link goes down, it will be bypassed by the hub and the rest of the stations can operate. Also adding and removing stations from the ring is easier. This topology is still used in the Token Ring LAN designed by IBM

2.2. Ethernet (802.3)

The original Ethernet was created in 1976 at Xerox's Palo Alto Research Center (PARC).

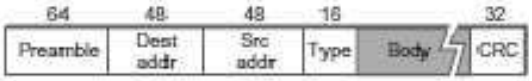


Figure 2.2 Ethernet Frame Format

There are ***four generations***:

1. Standard Ethernet (10 Mbps)
2. Fast Ethernet (100 Mbps)
3. Gigabit Ethernet (l Gbps)
4. Ten-Gigabit Ethernet (l0 Gbps).

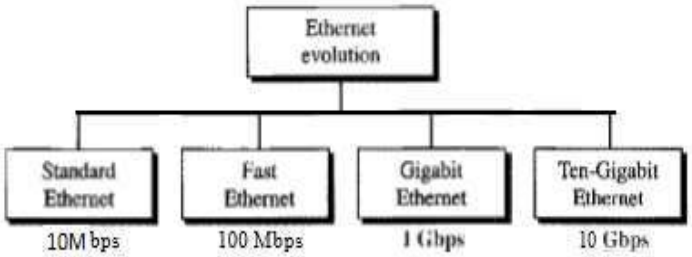


Figure 2.3 *Ethernet evolution through four generations*

2.2.1. STANDARD ETHERNET

Frame Format

- The Ethernet frame contains *seven fields*: preamble, SFD, DA, SA, length or type of protocol data unit (PDU), upper-layer data, and the CRC.
- Ethernet does not provide any mechanism for acknowledging received frames, making it what is known as an unreliable medium.
- Acknowledgments must be implemented at the higher layers.

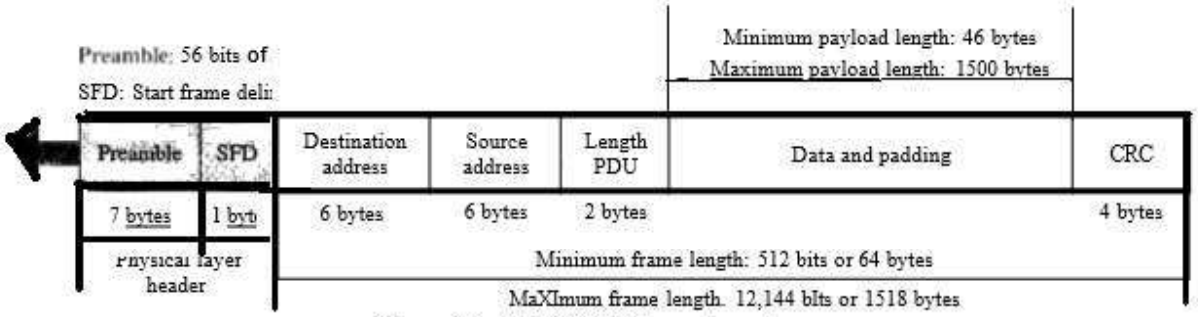


Figure 2.4. 802.3 MAC Frame Format

i. *Preamble.* The first field ofthe 802.3 frame contains 7 bytes (56 bits) of alternating O's and 1's that alerts the receiving system to the coming frame and enables it to synchronize its input timing. The pattern provides only an alert and a timing pulse. The 56-bit pattern allows the stations to miss some bits at the beginning of the frame. The preamble is actually added at the physical layer and is not (formally) part of the frame.

ii. *Start frame delimiter (SFD).* The second field (l byte: 10101011) signals the beginning of the frame. The SFD warns the station or stations that this is the last chance for synchronization. The last 2 bits is 11 and alerts the receiver that the next field is the destination address.

iii. *Destination Address(DA).* The DA field is 6 bytes and contains the physical address of the destination station or stations to receive the packet.

iv. *Source address (SA).* The SA field is also 6 bytes and contains the physical address of the sender of the packet.

v. *Length or type.* This field is defined as a type field or length field. The original Ethernet used this field as the type field to define the upper-layer protocol using the MAC frame. The IEEE standard used it as the length field to define the number of bytes in the data field.

vi. *Data.* This field carries data encapsulated from the upper-layer protocols. It is a minimum of 46 and a maximum of 1500 bytes.

vii. *CRC*. The last field contains error detection information

viii. ***Frame Length***

- An Ethernet frame needs to have a minimum length of 512 bits or 64 bytes. Part of this length is the header and the trailer.
- If we count 18 bytes of header and trailer (6 bytes of source address, 6 bytes of destination address, 2 bytes of length or type, and 4 bytes of CRC), then the minimum length of data from the upper layer is 64 - 18 = 46 bytes.
- If the upper-layer packet is less than 46 bytes, padding is added to make up the difference.
- The maximum length restriction has two historical reasons.
  - ✓ Memory was very expensive when Ethernet was designed: a maximum length restriction helped to reduce the size of the buffer.
  - ✓ The maximum length restriction prevents one station from monopolizing the shared medium, blocking other stations that have data to send.

Addressing

✓ Each station on an Ethernet network (such as a PC, workstation, or printer) has its own *network interface card (NIC)*. The NIC fits inside the station and provides the station with a 6-byte physical address.

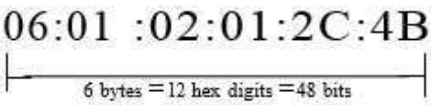


Figure 2.5 *Example of an Ethernet address in hexadecimal notation*

✓ As shown in Figure 2.5, the Ethernet address is 6 bytes (48 bits), normally written in hexadecimal notation, with a colon between the bytes.

✓ *Unicast, Multicast, and Broadcast Addresses A unicast destination* address defines only one recipient; the relationship between the sender and the receiver is one-to-one.

A multicast destination address defines a group of addresses; the relationship between the sender and the receivers is one-to-many.

The broadcast address is a special case of the multicast address; the recipients are all the stations on the LAN. A broadcast destination address is forty eight 1s.

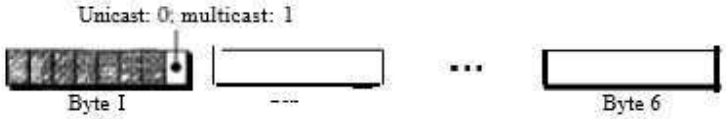


Figure 2.6 *Unicast and multicast addresses*

- Figure 2.6 shows how to distinguish a unicast address from a multicast address. The least significant bit of the first byte defines the type of address. If the bit is 0, the address is unicast; otherwise, it is multicast.

Example

Define the type of the following destination addresses:

a. 4A:30:10:21:1O:1A

b. 47:20:1B:2E:08:EE

c. FF:FF:FF:FF:FF:FF

Solution

To find the type of the address, we need to look at the second hexadecimal digit from the left. If it is even, the address is unicast. If it is odd, the address is multicast. If all digits are F's, the address is broadcast. Therefore, we have the following:

a. This is a unicast address because A in binary is 1010 (even).

b. This is a multicast address because 7 in binary is 0111 (odd).

c. This is a broadcast address because all digits are F's.

The way the addresses are sent out on line is different from the way they are written in hexadecimal notation. The transmission is left-to-right, byte by byte; however, for each byte, the least significant bit is sent first and the most significant bit is sent last. This means that the bit that defines an address as unicast or multicast arrives first at the receiver.

Example

Show how the address 47:20:1B:2E:08:EE is sent out on line.

Solution

The address is sent left-to-right, byte by byte; for each byte, it is sent right-to-left, bit by bit, as shown below:

11100010 00000100 11011000 01110100 00010000 01110111

Access Method: CSMA/CD

- Standard Ethernet uses 1-persistent CSMA/CD.

Slot Time

- In an Ethernet network, the round-trip time required for a frame to travel from one end of a maximum-length network to the other plus the time needed to send the jam sequence is called the slot time.

```
Slot time = round-trip time + time required to send the jam sequence
```

- The slot time in Ethernet is defined in bits. It is the time required for a station to send 512 bits. This means that the actual slot time depends on the data rate; for traditional 10-Mbps Ethernet it is 51.2μs.

Slot Time and Collision

- The choice of a 512-bit slot time was not accidental. It was chosen to allow the proper functioning of CSMA/CD. Two cases are used for this above statements.
  - ✓ In the first case, we assume that the sender sends a minimum-size packet of 512 bits. Before the sender can send the entire packet out, the signal travels through the network and reaches the end of the network. If there is another signal at the end of the network (worst case), a collision occurs. The sender has the opportunity to abort the sending of the frame and to send a jam sequence to inform other stations of the collision. The round-trip time plus the time required to send the jam sequence should be less than the time needed for the sender to send the minimum frame, 512 bits. The sender needs to be aware of the collision before it is too late, that is, before it has sent the entire frame.
  - ✓ In the second case, the sender sends a frame larger than the minimum size (between 512 and 1518 bits). In this case, if the station has sent out the first 512 bits and has not heard a collision, it is guaranteed that collision will never occur during the transmission of this frame. The reason is that the signal will reach the end of the network in less than one-half the slot time.

Slot Time and Maximum Network Length

There is a relationship between the slot time and the maximum length of the network (collision domain). It is dependent on the propagation speed of the signal in the particular medium. In most transmission media, the signal propagates at 2 x 10^8 rn/s (two-thirds of the rate for propagation in air). For traditional Ethernet, we calculate

MaxLength = (PropagationSpeedx(SlotTime/2))

MaxLength= ((2x 10^8) * (51.2 *10^{-6} /2)) = 5120m

The delay times in repeaters and interfaces, and the time required to send the jam sequence are reduce the maximum-length of a traditional Ethernet network to 2500 m, just 48 percent of the theoretical calculation.

MaxLength=2500m

Physical Layer

The Standard Ethernet defines several physical layer implementations; four of the most common, are shown in Figure 2.7.

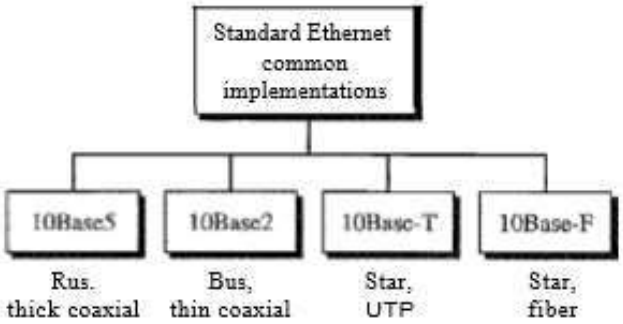


Figure 2.7 *Categories of Standard Ethernet*

Encoding and Decoding

All standard implementations use digital signaling (baseband) at 10 Mbps. At the sender, data are converted to a digital signal using the Manchester scheme; at the receiver, the received signal is interpreted as Manchester and decoded into data.

In Manchester encoding, the duration of the bit is divided into two halves. The voltage remains at one level during the first half and moves to the other level in the second half. The transition at the middle of the bit provides synchronization.

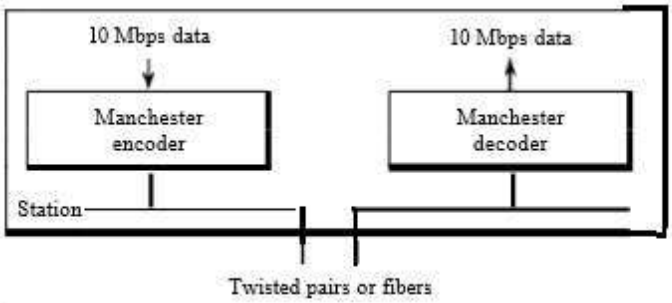


Figure 2.8 *Encoding in a Standard Ethernet implementation*

10Base5: Thick Ethernet

- 10Base5 (thick Ethernet or Thicknet) was the first Ethernet specification to use a bus topology with an external transceiver (transmitter/receiver) connected via a tap to a thick coaxial cable.
- The transceiver is responsible for transmitting, receiving, and detecting collisions.
- The transceiver is connected to the station via a transceiver cable that provides separate paths for sending and receiving. This means that collision can only happen in the coaxial cable.
- The maximum length of the coaxial cable must not exceed 500m, otherwise, there is excessive degradation of the signal.

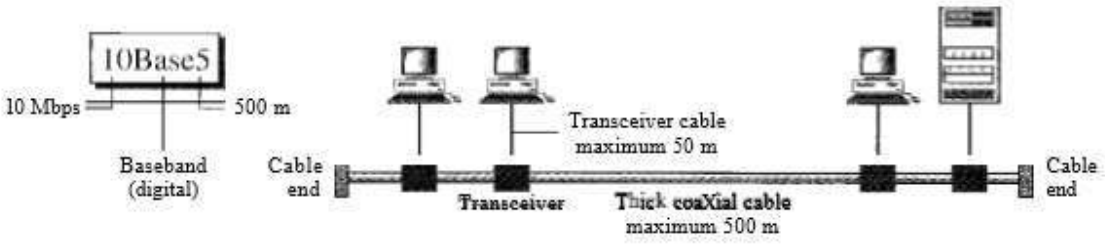


Figure 2.9 *IOBase5 implementation*

- If a length of more than 500 m is needed, up to five segments, each a maximum of 500m, can be connected using repeaters.

10Base2: Thin Ethernet

- The second implementation is called lOBase2, thin Ethernet, or Cheapernet.
- IOBase2 also uses a bus topology, but the cable is much thinner and more flexible.

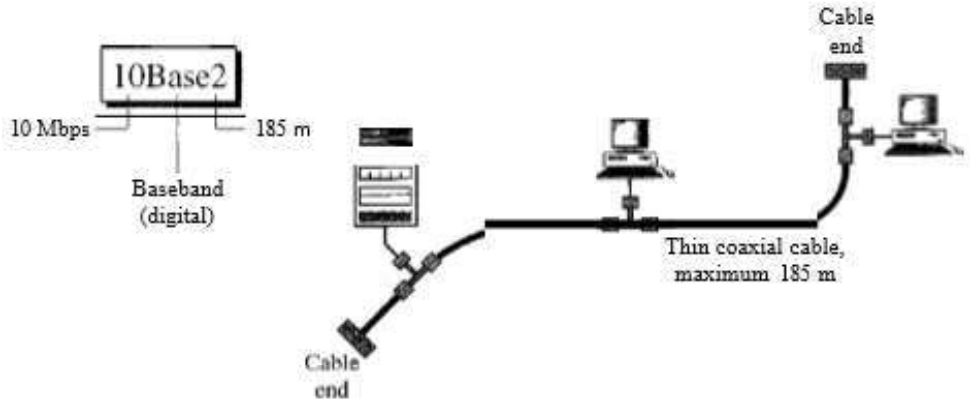


Figure 2.10 *lOBase2 implementation*

- The cable can be bent to pass very close to the stations. In this case, the transceiver is normally part of the network interface card (NIC), which is installed inside the station.
- The collision here occurs in the thin coaxial cable. This implementation is more cost effective than 10Base5 because thin coaxial cable is less expensive than thick coaxial and the tee connections are much cheaper than taps.
- Installation is simpler because the thin coaxial cable is very flexible. However, the length of each segment cannot exceed 185 m (close to 200 m) due to the high level of attenuation in thin coaxial cable.

10Base-T: Twisted-Pair Ethernet

- The third implementation is called 10Base-T or twisted-pair Ethernet.
- 10Base-T uses a physical star topology. The stations are connected to a hub via two pairs of twisted cable, as shown in Figure 2.11.
- Note that two pairs of twisted cable create two paths (one for sending and one for receiving) between the station and the hub. Any collision here happens in the hub.
- Compared to 10Base5 or lOBase2, the hub actually replaces the coaxial cable as far as a collision is concerned.
- The maximum length of the twisted cable here is defined as 100 m, to minimize the effect of attenuation in the twisted cable.

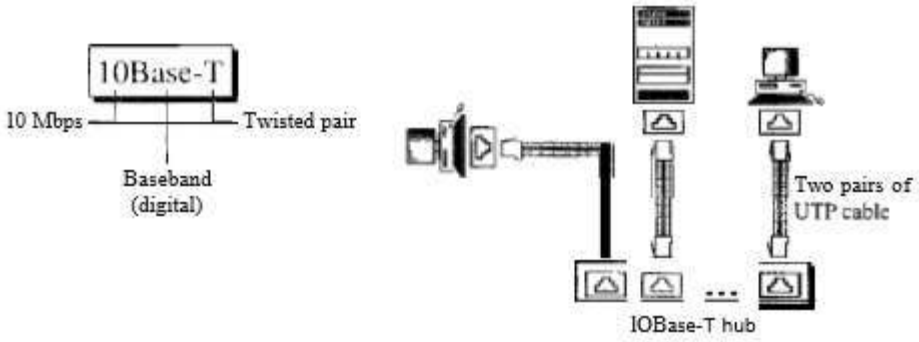


Figure 2.11 *IOBase-T implementation*

10Base-F: Fiber Ethernet

- There are several types of optical fiber 10Mbps Ethernet, the most common is called 10Base-F.
- 10Base-F uses a star topology to connect stations to a hub.
- The stations are connected to the hub using two fiber-optic cables, as shown in Figure 2.12

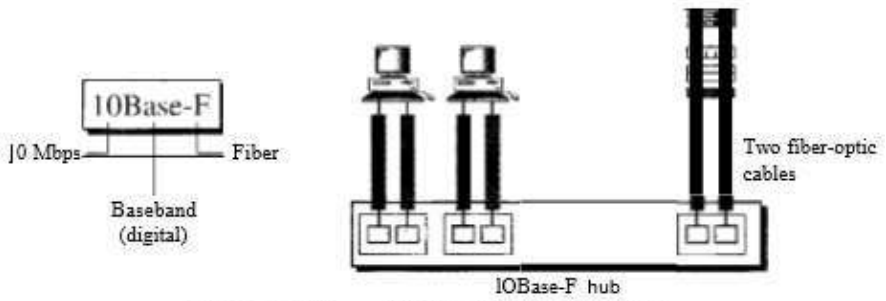


Figure 2.12 *10Base-F implementation*

Summary

Table *Summary of Standard Ethernet implementations*

Characteristics	*10Base5*	*10Base2*	*10Base-T*	*10Base-F*
Media	Thick coaxial cable	Thin coaxial cable	2UTP	2 Fiber
Maximum length	500m	185 m	100m	2000m
Line encoding	Manchester	Manchester	Manchester	Manchester

2.2.2. FAST ETHERNET

- Fast Ethernet was designed to compete with LAN protocols such as FDDI or Fiber Channel (or Fibre Channel, as it is sometimes spelled).
- IEEE created Fast Ethernet under the name 802.3u. Fast Ethernet is backward-compatible with Standard Ethernet, but it can transmit data 10 times faster at a rate of 100 Mbps.
- The goals of Fast Ethernet can be summarized as follows:
  1. Upgrade the data rate to 100 Mbps
  2. Make it compatible with Standard Ethernet.
  3. Keep the same 48-bit address.
  4. Keep the same frame format.
  5. Keep the same minimum and maximum frame lengths.

MAC Sublayer

- A main consideration in the evolution of Ethernet from 10 to 100 Mbps was to keep the MAC sublayer untouched.
- A decision was made to drop the bus topologies and keep only the star topology. For the star topology, there are two choices, *half duplex and full duplex*.
  1. In the half-duplex approach, the stations are connected via a hub; in the full-duplex approach, the connection is made via a switch with buffers at each port. The access method is the same (CSMA/CD) for the half-duplex approach;
  2. For full duplex Fast Ethernet, there is no need for CSM/CD. However, the implementations keep CSMA/CD for backward compatibility with Standard Ethernet.

Autonegotiation

- A new feature added to Fast Ethernet is called autonegotiation.

- It allows a station or a hub a range of capabilities.
- Autonegotiation allows two devices to negotiate the mode or data rate of operation. It was designed particularly for the following purposes:
  - To allow incompatible devices to connect to one another. For example, a device with a maximum capacity of 10 Mbps can communicate with a device with a 100 Mbps capacity (but can work at a lower rate).
  - To allow one device to have multiple capabilities. o To allow a station to check a hub's capabilities.

Physical Layer

- The physical layer in Fast Ethernet is more complicated than the one in Standard Ethernet.

Topology: Fast Ethernet is designed to connect two or more stations together. If there are only two stations, they can be connected point-to-point. Three or more stations need to be connected in a star topology with a hub or a switch at the center, as shown in Figure 2.13.

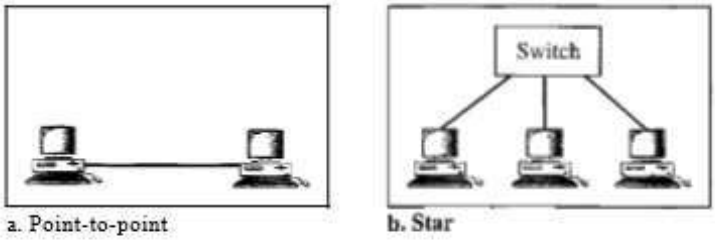


Figure 2.13 *Fast Ethernet topology*

Implementation

- Fast Ethernet implementation at the physical layer can be categorized as either two-wire or four-wire.
- The two-wire implementation can be either category 5 UTP (100Base-TX) or fiber-optic cable (100Base-FX). The four-wire implementation is designed only for category 3 UTP (l00Base-T4). (Figure 2.14)

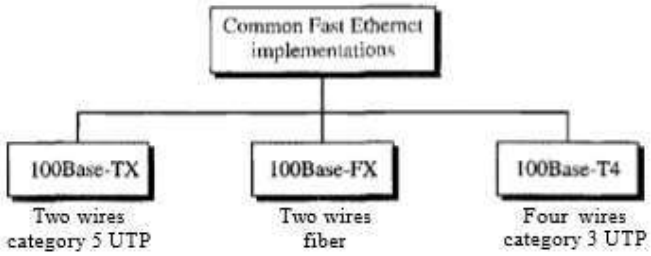


Figure 2.14 *Fast Ethernet implementations*

Encoding

- Manchester encoding needs a 200-Mbaud bandwidth for a data rate of 100 Mbps, which makes it unsuitable for a medium such as twisted-pair cable. For this reason, the Fast Ethernet designers sought some alternative encoding/decoding scheme.
- ***100Base-TX***
  - ✓ It uses two pairs of twisted-pair cable (either category 5 UTP or STP). For this implementation, the MLT-3 scheme was selected since it has good bandwidth performance. However, since MLT-3 is not a self-synchronous line coding scheme, 4B/5B block coding is used to provide bit synchronization by preventing the occurrence of a long sequence of 0s and 1s.
  - ✓ This creates a data rate of 125 Mbps, which is fed into MLT-3 for encoding.
- ***100Base-FX***
  - ✓ uses two pairs of fiber-optic cables.

- ✓ Optical fiber can easily handle high bandwidth requirements by using simple encoding schemes. The designers of 100Base-FX selected the NRZ-I encoding scheme for this implementation. However, NRZ-I has a bit synchronization problem for long sequences of 0s (or Is, based on the encoding).
- ✓ To overcome this problem, the designers used 4B/5B block encoding as we described for 100Base-TX. The block encoding increases the bit rate from 100 to 125 Mbps, which can easily be handled by fiber-optic cable.
- ✓ A 1OOBase-TX network can provide a data rate of100 Mbps, but it requires the use of category 5 UTP or STP cable. This is not cost-efficient for buildings that have already been wired for voice-grade twisted-pair (category 3). A new standard, called 100Base-T4, was designed to use category 3 or higher UTP.
- ✓ The implementation uses four pairs of UTP for transmitting 100 Mbps. Encoding/decoding in 100Base-T4 is more complicated. As this implementation uses category 3 UTP, each twisted-pair cannot easily handle more than 25 Mbaud. In this design, one pair switches between sending and receiving. Three pairs of UTP category 3, however, can handle only 75 Mbaud (25 Mbaud) each
- ✓ In 8B/6T, eight data elements are encoded as six signal elements. This means that 100 Mbps uses only (6/8) x 100 Mbps, or 75 Mbaud.

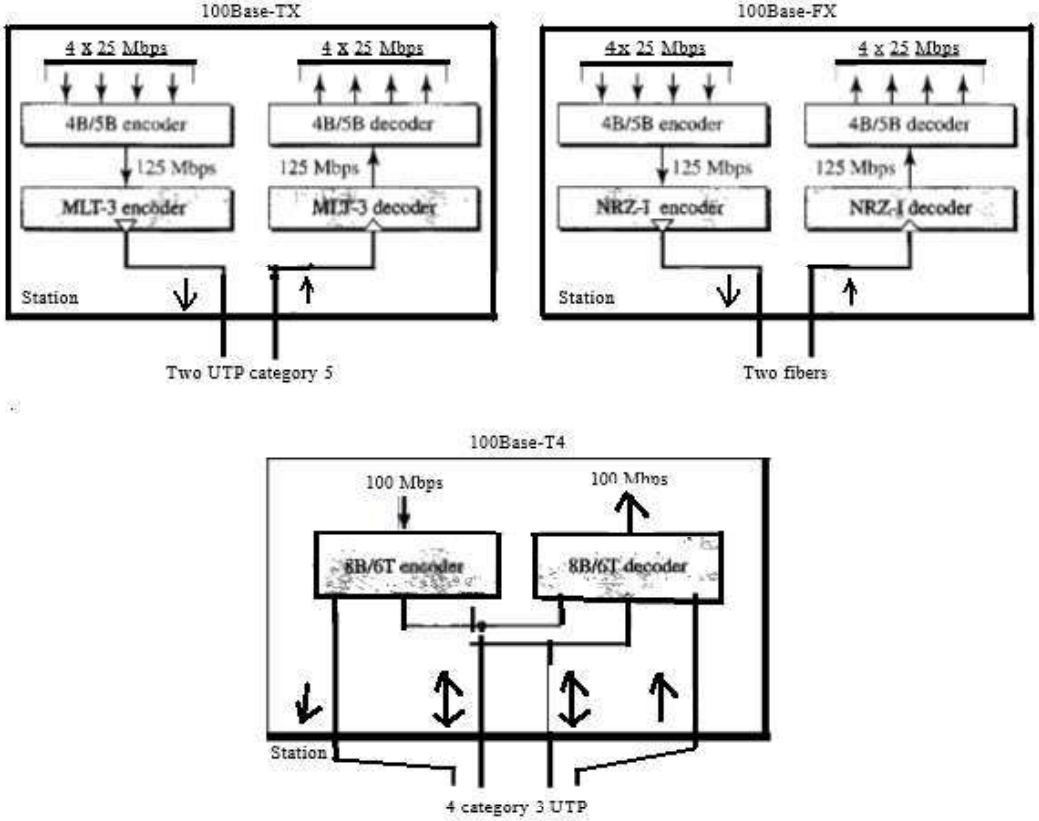


Figure 2.15 *Encoding for Fast Ethernet implementation*

Summary

Table *Summary of Fast Ethernet implementations*

Characteristics	*lOOBase-TX*	*lOOBase-FX*	*100Base-T4*
Media	Cat 5 UTP or STP	Fiber	Cat 4 UTP
Number of wires	2	2	4
Maximum length	100m	100m	100m
Block encoding	4B/5B	4B/5B	
Line encoding	MLT-3	NRZ-I	8B/6T

2.2.3. GIGABIT ETHERNET

- The need for an even higher data rate resulted in the design of the Gigabit Ethernet protocol (1000 Mbps).
- The IEEE committee calls the Standard 802.3z. The goals ofthe Gigabit Ethernet design can be summarized as follows:
  1. Upgrade the data rate to 1 Gbps
  2. Make it compatible with Standard or Fast Ethernet.
  3. Use the same 48-bit address.
  4. Use the same frame format.
  5. Keep the same minimum and maximum frame lengths.
  6. To support autonegotiation as defined in Fast Ethernet.

MAC Sublayer

- Gigabit Ethernet has two distinctive approaches for medium access: half-duplex and full-duplex.

1. Full-Duplex Mode

- In full-duplex mode, there is a central switch connected to all computers or other switches.
- In this mode, each switch has buffers for each input port in which data are stored until they are transmitted.
- There is no collision in this mode.
- This means that CSMAlCD is not used. Lack ofcollision implies that the maximum length of the cable is determined by the signal attenuation in the cable, not by the collision detection process.
- In the full-duplex mode of Gigabit Ethernet, there is no collision; the maximum length of the cable is determined by the signal attenuation in the cable.

2. Half-Duplex Mode

- Gigabit Ethernet can also be used in half-duplex mode.
- In this case, a switch can be replaced by a hub, which acts as the common cable in which a collision might occur. The half-duplex approach uses CSMA/CD.
- The maximum length of the network in this approach is totally dependent on the minimum frame size. Three methods have been defined: *traditional, carrier extension, and frame bursting*.

1. **Traditional** : In the traditional approach, the minimum length of the frame as in traditional Ethernet (512 bits). However, because the length of a bit is 11100 shorter in Gigabit Ethernet than in 10-Mbps Ethernet, the slot time for Gigabit Ethernet is 512 bits x 111000 µs, which is equal to 0.512 µs.

   The reduced slot time means that collision is detected 100 times earlier. This means that the maximum length of the network is 25 m. This length may be suitable if all the stations are in one room, but it may not even be long enough to connect the computers in one single office.

2. **Carrier Extension** : To allow for a longer network, we increase the minimum frame length. The carrier extension approach defines the minimum length of a frame as 512 bytes (4096 bits). This means that the minimum length is 8 times longer.

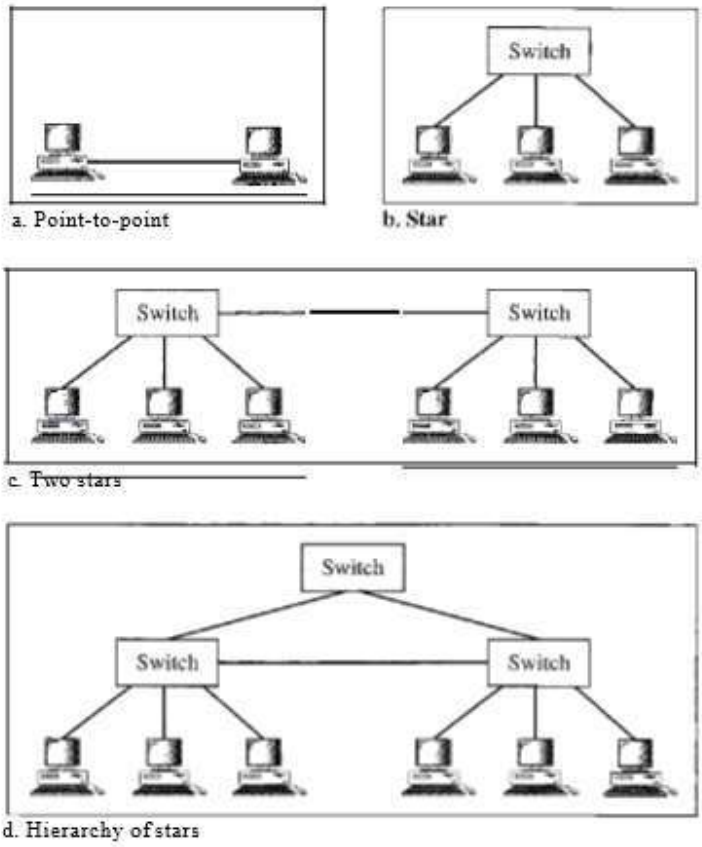


Figure 2.16 *Topologies of Gigabit Ethernet*

This method forces a station to add extension bits (padding) to any frame that is less than 4096 bits. In this way, the maximum length of the network can be increased 8 times to a length of 200 m. This allows a length of 100 m from the hub to the station.

3. **Frame Bursting** : Carrier extension is very inefficient if we have a series of short frames to send; each frame carries redundant data.

To improve efficiency, frame bursting was proposed.

Physical Layer

The physical layer in Gigabit Ethernet is more complicated than that in Standard or Fast Ethernet.

Topology

- Gigabit Ethernet is designed to connect two or more stations.
- If there are only two stations, they can be connected point-to-point. Three or more stations need to be connected in a star topology with a hub or a switch at the center.
- Another possible configuration is to connect several star topologies or let a star topology be part of another as shown in Figure 2.17.

Implementation

- Gigabit Ethernet can be categorized as either a two-wire or a four-wire implementation.
- The two-wire implementations use fiber-optic cable (1000Base-SX, short-wave, or 1000Base-LX, long-wave), or STP (1000Base-CX).
- The four-wire version uses cate- gory 5 twisted-pair cable (1000Base-T). In other words, we have four implementations, as shown in Figure 2.58.

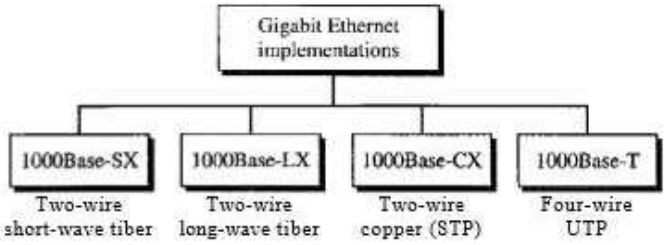


Figure 2.17 *Gigabit Ethernet implementations*

- 1000Base-T was designed in response to those users who had already installed this wiring for other purposes such as Fast Ethernet or telephone services.

Encoding

Figure 2.18 shows the encoding/decoding schemes for the four implementations.

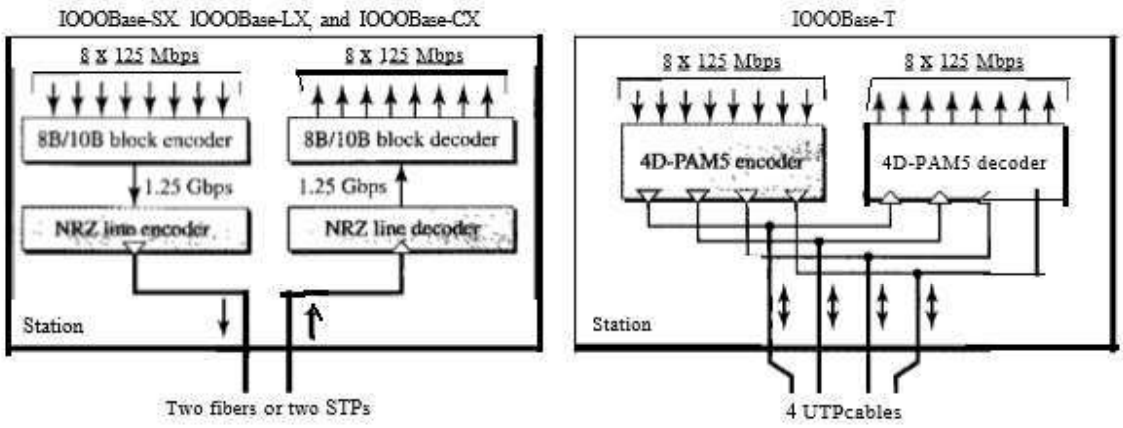


Figure 2.18 *Encoding in Gigabit Ethernet implementations*

- Gigabit Ethernet cannot use the Manchester encoding scheme because it involves a very high bandwidth (2 GBaud).
- The two-wire implementations use an NRZ scheme, but NRZ does not self-synchronize properly.
- To synchronize bits, particularly at this high data rate, 8B/10B block encoding is used.

Summary

Characteristics	*lOOOBase-SX*	*lOOOBase-LX*	*lOOOBase-CX*	*lOOOBase-T*
Media	Fiber short-wave	Fiber long-wave	STP	Cat 5 UTP
Number of wires	2	2	2	4
Maximum length	550m	5000m	25m	100m
Block encoding	8B/lOB	*8B/lOB*	8B/lOB	
Line encoding	NRZ	NRZ	NRZ	4D-PAM5

2.2.4. Ten-Gigabit Ethernet

- The IEEE committee created Ten-Gigabit Ethernet and called it Standard 802.3ae.
- The goals of the Ten-Gigabit Ethernet design can be summarized as follows:
  1. Upgrade the data rate to 10 Gbps.
  2. Make it compatible with Standard, Fast, and Gigabit Ethernet.
  3. Use the same 48-bit address.
  4. Use the same frame format.
  5. Keep the same minimum and maximum frame lengths.
  6. Allow the interconnection ofexisting LANs into a metropolitan area network (MAN) or a wide area network (WAN).
  7. Make Ethernet compatible with technologies such as Frame Relay and ATM.

MAC Sublayer

Ten-Gigabit Ethernet operates only in full duplex mode which means there is no need for contention; *CSMA/CD is not used in Ten-Gigabit Ethernet*.

Physical Layer

- The physical layer in Ten-Gigabit Ethernet is designed for using fiber-optic cable over long distances.
- Three implementations are the most common: 10GBase-S, 10GBase-L, and 10GBase-E.

Summary

Characteristics	*lOGBase-S*	*lOGBase-L*	*lOGBase-E*
Media	Short-wave S50-nrn rnultimode	Long-wave 131O-nm single mode	Extended 1550-mrn single mode
Maximum length	300m	lOkm	40km

2.3. Wireless LANs

- Wireless communication is one of the fastest-growing technologies without the use of cables is increasing everywhere. Wireless LANs can be found on college campuses, in office buildings, and in many public areas.
- Two promising wireless technologies for LANs:
  - ✓ IEEE 802.11 wireless LANs, sometimes called wireless Ethernet
  - ✓ Bluetooth

2.3.1. IEEE 802.11

Architecture

- The standard defines two kinds of services:
  - ✓ Basic service set (BSS)
  - ✓ Extended service set *(ESS)*.

Basic Service Set

- IEEE 802.11 defines the basic service set (BSS) as the building block of a wireless LAN.
- A basic service set is made of stationary or mobile wireless stations and an optional central base station, known as the access point (AP).
- Figure 2.19 shows two sets in this standard.

- The BSS without an AP is a stand-alone network and cannot send data to other BSSs. It is called an *ad hoc architecture*. In this architecture, stations can form a network without the need of an AP; they can locate one another and agree to be part of a BSS. A BSS with an AP is sometimes referred to as an infrastructure network.

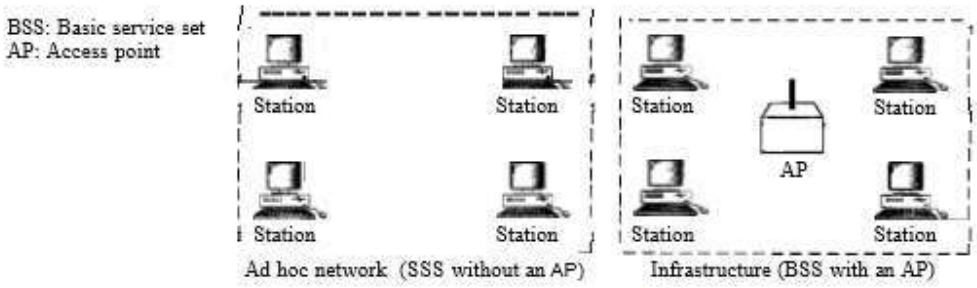


Figure 2.19 *Basic service sets (BSSs)*

Extended Service Set

- An extended service set (ESS) is made up of two or more BSSs with APs. In this case, the BSSs are connected through a distribution system, which is usually a wired LAN. The distribution system connects the APs in the BSSs.
- IEEE 802.11 does not restrict the distribution system; it can be any IEEE LAN such as an Ethernet.
- The extended service set uses two types of stations:
  - ✓ *Mobile Stations:* The *mobile stations* are normal stations inside a BSS.
  - ✓ *Stationary Stations* : The s*tationary stations* are AP stations that are part of a wired LAN.
- When BSSs are connected, the stations within reach of one another can communicate without the use of an AP.
- Example: cellular network if we consider each BSS to be a cell and each AP to be a base station.

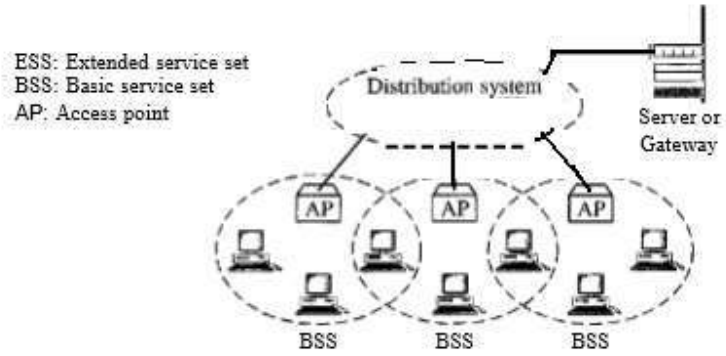


Figure 2.20 *Extended service sets (ESSs)*

Station Types

- IEEE 802.11 defines three types of stations based on their mobility in a wireless LAN: no-transition, BSS- transition, and ESS-transition mobility.
  - ✓ A *station with no-transition mobility* is either stationary (not moving) or moving only inside a BSS.
  - ✓ A *station with BSS-transition mobility* can move from one BSS to another, but the movement is con- fined inside one ESS.
  - ✓ A *station with ESS-transition mobility* can move from one ESS to another.

MAC Sublayer

✓ IEEE 802.11 defines two MAC sublayers:

1. The distributed coordination function (DCF)
2. Point coordination function (PCF)

✓ Figure 2.21 shows the relationship between the two MAC sublayers, the LLC sublayer, and the physical layer.

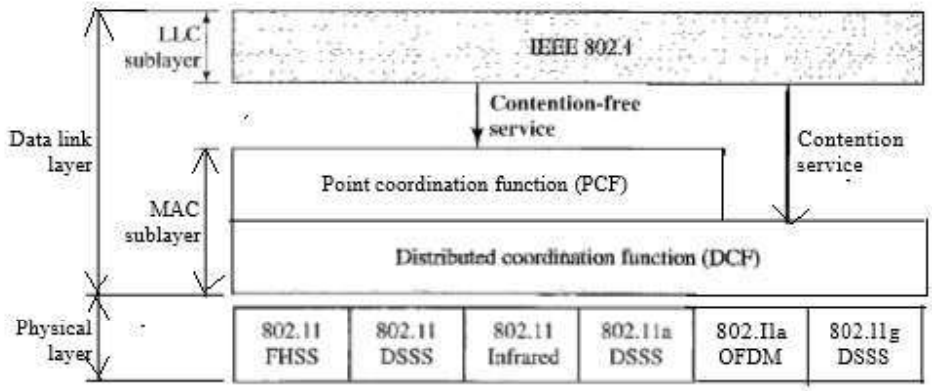


Figure 2.21 *MAC layers in IEEE 802.11 standard*

✓ ***Distributed Coordination Function***

- One of the two protocols defined by IEEE at the MAC sublayer is called the distributed coordination function (DCF).
- DCF uses CSMA/CA as the access method.
- Wireless LANs cannot implement CSMA/CD for three reasons:

1. For collision detection a station must be able to send data and receive collision signals at the same time. This can mean costly stations and increased bandwidth requirements.
2. Collision may not be detected because of the hidden station problem.
3. The distance between stations can be great. Signal fading could prevent a station at one end from hearing a collision at the other end.

- *Process Flowchart* : Figure 2.22 shows the process flowchart for CSMA/CA as used in wireless LANs.
- *Frame Exchange Time Line* : Figure 2.23 shows the exchange of data and control frames in time.
  - Before sending a frame, the source station senses the medium by checking the energy level at the carrier frequency.
    a. The channel uses a persistence strategy with back-off until the channel is idle.
    b. After the station is found to be idle, the station waits for a period of time called the distributed inter frame space (DIFS); then the station sends a control frame called the request to send (RTS).
  - After receiving the RTS and waiting a period of time called the short inter frame space (SIFS), the destination station sends a control frame, called the clear to send (CTS), to the source station. This control frame indicates that the destination station is ready to receive data.
  - The source station sends data after waiting an amount of time equal to SIFS.
  - The destination station, after waiting an amount of time equal to SIFS, sends an acknowledgment to show that the frame has been received. Acknowledgment is needed in this protocol because the station does not have any means to check for the successful arrival of its data at the destination. On the other hand, the lack of collision in CSMA/CD is a kind of indication to the source that data have arrived.
- *Network Allocation Vector* : How do other stations defer sending their data if one station acquires access? or how is the collision avoidance aspect of this protocol accomplished?

When a station sends an RTS frame, it includes the duration of time that it needs to occupy the channel. The stations that are affected by this transmission create a timer called a network allocation vector (NAV) that shows how much time must pass before these stations are allowed to check the channel for idleness.

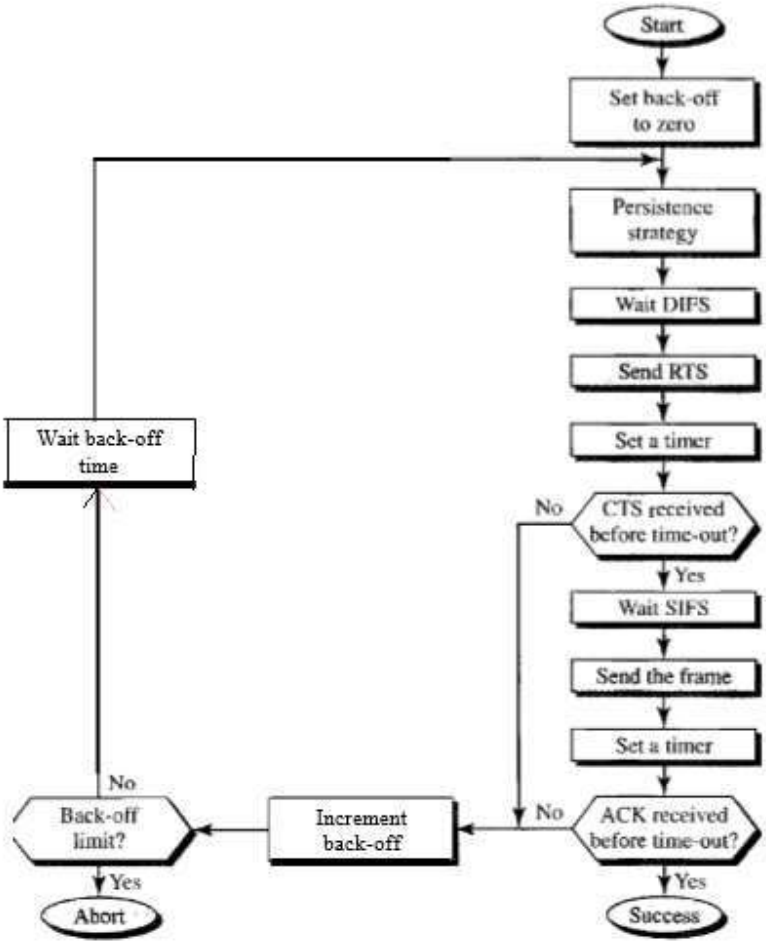


Figure 2.22 *CSMA/CA flowchart*

Each time a station accesses the system and sends an RTS frame, other stations start their NAV. In other words, each

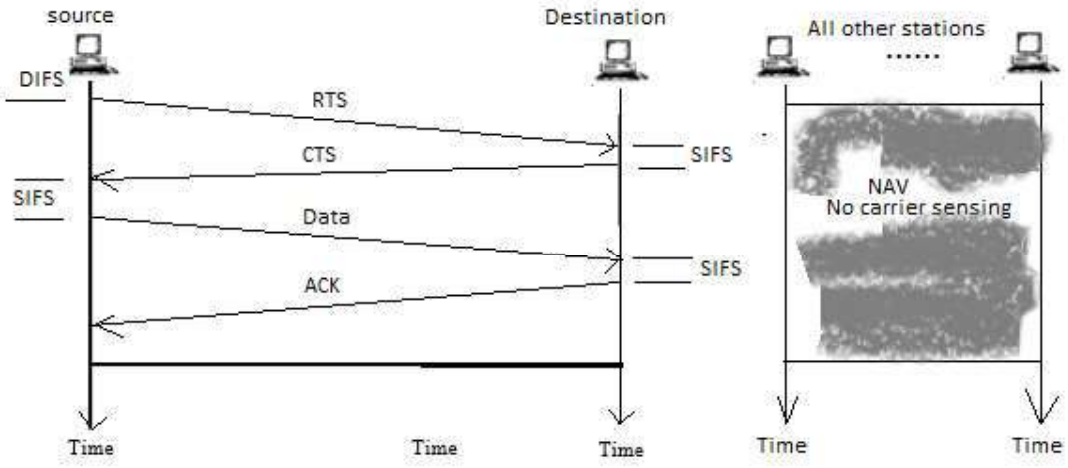


Fig 2.23 CSMA/ CA and NAV

station, before sensing the physical medium to see if it is idle, first checks its NAV to see if it has expired. Figure 2.23 shows the idea of NAV.

- *Collision During Handshaking:* What happens if there is collision during the time when RTS or CTS control frames are in transition, often called the handshaking period? Two or more stations may try to send RTS frames at the same time. These control frames may collide. However, because there is no mechanism for collision detection, the sender assumes there has been a collision if it has not received a CTS frame from the receiver. The back-off strategy is employed, and the sender tries again.

✓ ***Point Coordination Function (PCP)***

- The point coordination function (PCF) is an optional access method that can be implemented in an infrastructure network (not in an ad hoc network). It is implemented on top of the DCF and is used mostly for time-sensitive transmission.
- PCF has a centralized, contention-free polling access method. The AP performs polling for stations that are capable of being polled.
- The stations are polled one after another, sending any data they have to the AP. To give priority to PCF over DCF, another set of inter frame spaces has been defined: *PIFS and SIFS.*
  - The SIFS is the same as that in DCF, but the PIFS (PCF IFS) is shorter than the DIFS. This means that if, at the same time, a station wants to use only DCF and an AP wants to use PCF, the AP has priority.
  - *Repetition interval:* Due to the priority of PCF over DCF, stations that only use DCF may not gain access to the medium. To prevent this, a has been designed to cover both contention-free (PCF) and contention-based (DCF) traffic. The repetition interval, which is repeated continuously, starts with a special control frame, called a beacon frame. When the stations hear the beacon frame, they start their NAV for the duration of the contention-free period of the repetition interval. Figure 2.24 shows an example of a

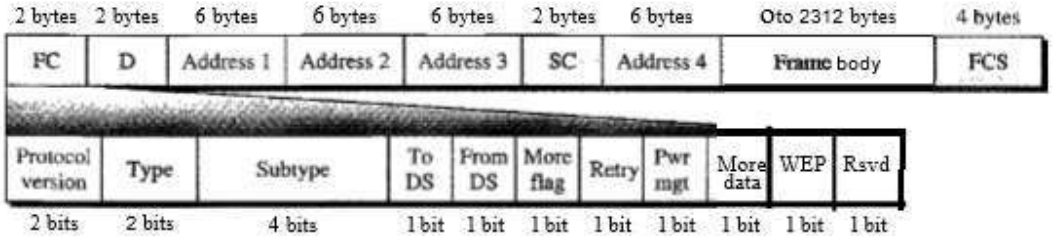


Figure 2.25 *Frameformat*

repetition interval.

- *Time :* During the repetition interval, the PC (point controller) can send a poll frame, receive data, send an ACK, receive an ACK, or do any combination of these (802.11 uses piggybacking). At the end of the contention-free period, the PC sends a CF end (contention-free end) frame to allow the contention-based stations to use the medium.
- *Fragmentation :* The wireless environment is very noisy; a corrupt frame has to be retransmitted. The protocol, therefore, recommends fragmentation-the division of a large frame into smaller ones. It is more efficient to resend a small frame than a large one.
- *Frame Format :* The MAC layer frame consists of nine fields, as shown in Figure 2.25.

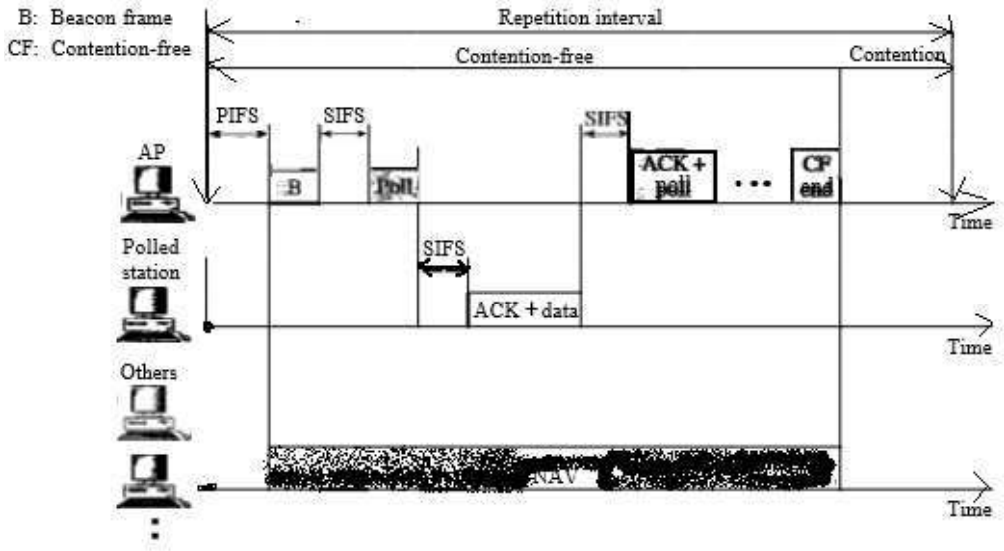


Figure 2.24 *Example of repetition interval*

i. **Frame Control (FC)**. The FC field is 2 bytes long and defines the type of frame and some control information. The following table describes the subfields of FC.

ii. **D**. In all frame types except one, this field defines the duration of the transmission that is used to set the value of NAV. In one control frame, this field defines the ID of the frame.

iii. **Addresses.** There are four address fields, each 6 bytes long. The meaning of each address field depends on the value of the To DS and From DS subfields.

iv. **Sequence control.** This field defines the sequence number of the frame to be used in flow control.

v. **Frame body.** This field, which can be between 0 and 2312 bytes, contains information based on the type and the subtype defined in the FC field.

vi. **FCS.** The FCS field is 4 bytes long and contains a CRC-32 error detection sequence.

- *Frame Types:* A wireless LAN defined by IEEE 802.11 has *three categories of frames*:
  - ✓ Management frames

Field	*Explanation*
Version	Current version is 0
Type	Type of information: management (00), control (01), or data (10)
Subtype	Subtype of each type
ToDS	Defined later
FromDS	Defined later
More flag	When set to 1, means more fragments
Retry	When set to 1, means retransmitted frame
Pwr mgt	When set to 1, means station is in power management mode
More data	When set to 1, means station has more data to send
WEP	Wired equivalent privacy (encryption implemented)
Rsvd	Reserved

  - ✓ Control frames

- ✓ Data frames
- ✓ Management Frames : Management frames are used for the initial communication between stations and access points.
- ✓ Control Frames : Control frames are used for accessing the channel and acknowledging frames. The

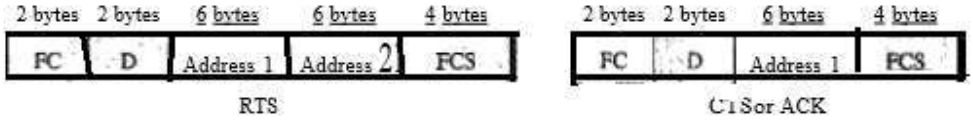


Figure 2.26 *Controlframes*

following table describes the subfields values of control frames

Subtype	*Meaning*
1011	Request to send (RTS)
1100	Clear to send (CTS)
1101	Acknowledgment (ACK)

- ✓ Data Frames: Data frames are used for carrying data and control information.
  - o *Addressing Mechanism*

    The IEEE 802.11 addressing mechanism specifies *four cases*, defined by the value of the two flags in the FC field, To DS and From DS. Each flag can be either 0 or I, resulting in four different situations. The interpretation of the four addresses (address1 to address 4) in the MAC frame depends on the value of these flags, as shown in following Table.

To DS	*From* DS	*Address* 1	*Address* 2	*Address* 3	*Address* 4
0	0	Destination	Source	BSS ID	N/A
0	1	Destination	SendingAP	Source	N/A
1	0	Receiving AP	Source	Destination	N/A
1	1	Receiving AP	SendingAP	Destination	Source

Note:

Address 1 is always the address of the next device.

Address 2 is always the address of the previous device.

Address 3 is the address of the final destination station if it is not defined by address I.

Address 4 is the address of the original source station if it is not the same as address 2.

- ❖ **Case 1:** 00 In this case, To DS = 0 and From DS = 0. This means that the frame is not going to a distribution system (To DS = 0) and is not coming from a distribution system (From DS=0). The frame is going from one station in a BSS to another without passing through the distribution system. The ACK frame should be sent to the original sender. The addresses are shown in Figure 2.27.
- ❖ **Case 2:** 01 In this case, To DS = 0 and From DS = 1. This means that the frame is coming from a distribution system (From DS = 1). The frame is coming from an AP and going to a station. The ACK should be sent to the AP. The addresses are shown in Figure 2.27. Note that address 3 contains the original sender of the frame (in another BSS).

- **Case 3:** 10 In this case, To DS =1 and From DS =O. This means that the frame is going to a distribution system (To DS= 1). The frame is going from a station to an AP. The ACK is sent to the original station. The addresses are shown in Figure 2.27. Note that address 3 contains the final destination of the frame
- **Case 4:**11 In this case, To DS=1and From DS=1. Thus is the case in which the distribution system is also wireless. The frame is going from one AP to another AP in a wire- less

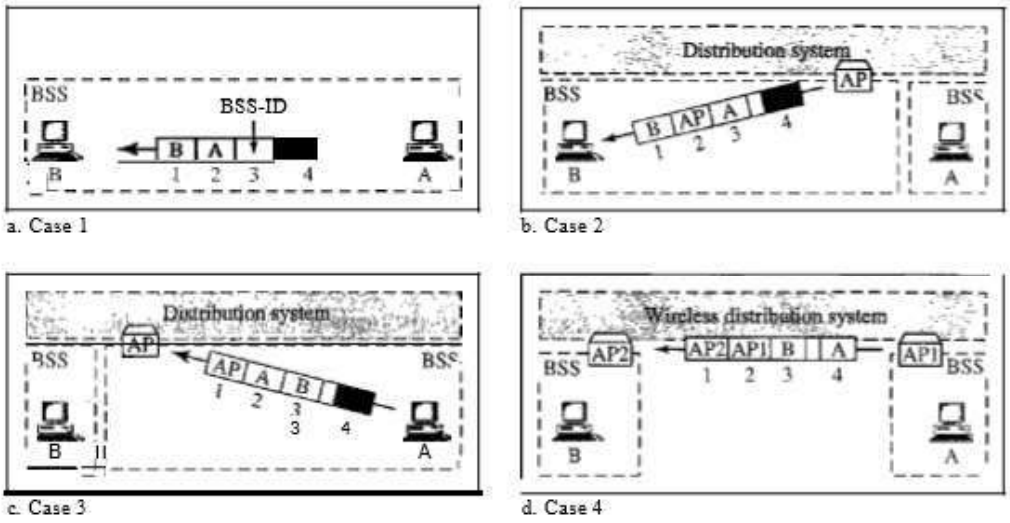


Figure 2.27 *Addressing mechanisms*

distribution system. We do not need to define addresses if the distribution system is a wired LAN because the frame in these cases has the format of a wired LAN frame (Ethernet, for example). Here, we need four addresses to define the original sender, the final destination, and two intermediate APs. Figure 2.27shows the situation.

Two problems in Wireless LAN

i. Hidden Station Problems

ii. Exposed Station Problems

i. Hidden Station Problems

- ✓ Figure 2.28 shows an example of the hidden station problem.
- ✓ Station B has a transmission range shown by the left oval (sphere in space); every station in this range can hear any signal transmitted by station B. Station C has a transmission range shown by the right oval (sphere in space); every station located in this range can hear any signal transmitted by C.
- ✓ Station C is outside the transmission range of B; likewise, station B is outside the transmission range of C. Station A, however, is in the area covered by both Band C; it can hear any signal transmitted by B or C.
- ✓ Assume that station B is sending data to station A. In the middle of this transmission, station C also has data to send to station A.
- ✓ Station C is out of B's range and transmissions from B cannot reach C. Therefore C thinks the medium is free. Station C sends its data to A, which results in a collision at A because this station is receiving data from both B and C.

- ✓ In this case, we say that stations B and C are hidden from each other with respect to A. Hidden stations can reduce the capacity of the network because of the possibility of collision.

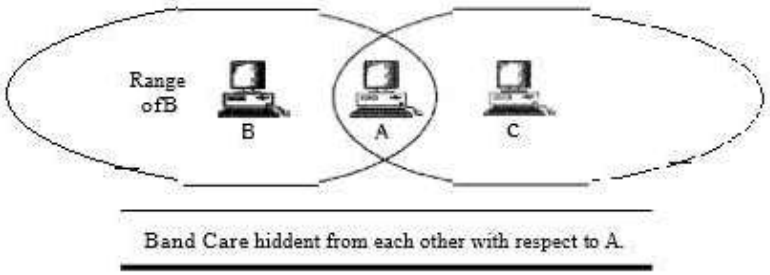


Figure 2.28 *Hidden station problem*

- ✓ The solution to the hidden station problem is the use of the handshake frames (RTS and CTS). Figure 2.29 shows that the RTS message from B reaches A, but not C. However, because both Band C are within the range of A, the CTS message, which contains the duration of data transmission from B to A reaches C. Station C knows that some hidden station is using the channel and refrains from transmitting until that duration is over.

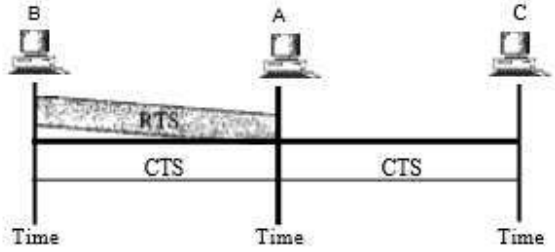


Figure 2.29 *Use of handshaking to prevent hidden station problem*

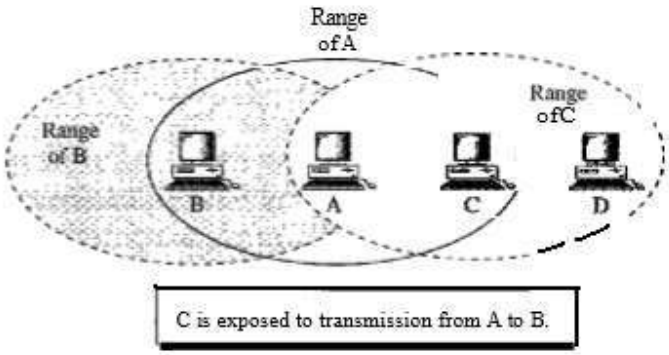


Figure 2.30 *Exposed station problem*

ii. Exposed Station Problem

- ✓ In this problem a station refrains from using a channel when it is, in fact, available. In Figure 2.30, station A is transmitting to station B. Station C has some data to send to station D, which can be sent without interfering with the transmission from A to B.
- ✓ However, station C is exposed to transmission from A; it hears what A is sending and thus refrains from sending.
- ✓ Station C hears the RTS from A, but does not hear the CTS from B.

- ✓ Station C, after hearing the RTS from A, can wait for a time so that the CTS from B reaches A; it then sends an RTS to D to show that it needs to communicate with D. Both stations B and A may hear this RTS, but station A is in the sending state, not the receiving state. Station B, however, responds with CTS. The problem is here.
- ✓ If station A has started sending its data, station C cannot hear the CTS from station D because of the collision; it cannot send its data to D. It remains exposed until A finishes sending its data as Figure 2.31 shows.

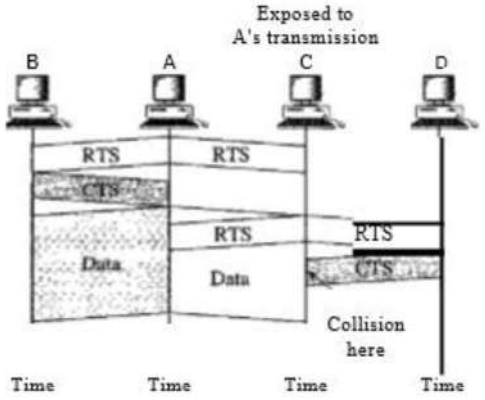


Figure 2.31 *Use of handshaking in exposed station problem*

Physical Layer

- Table shows six specifications . All implementations, except the infrared, operate in the industrial, scientific, and medical (ISM) band, which defines

three unlicensed bands in the three ranges 902-928 MHz, 2.400--4.835 GHz, and 5.725-5.850 GHz, as shown in Figure 2.32.

Table *Physical layers*

IEEE	*Technique*	*Band*	*Modulation*	*Rate (Mbps)*
802.11	FHSS	2.4 GHz	FSK	1 and 2
	DSSS	2.4 GHz	PSK	1 and 2
		Infrared	PPM	1 and 2
802.11a	OFDM	5.725 GHz	PSKorQAM	6 to 54
802.11b	DSSS	2.4 GHz	PSK	5.5 and 11
802.1Ig	OFDM	2.4 GHz	Different	22 and 54

Figure 2.32 *industrial, scientific, and medical (ISM) band*

- o *IEEE 802.11 FHSS*
  - ✓ IEEE 802.11 FHSS uses the frequency-hopping spread spectrum (FHSS) method. FHSS uses the 2.4-GHz ISM band.

- ✓ The band is divided into 79 sub bands of 1 MHz (and some guard bands).
- ✓ A pseudorandom number generator selects the hopping sequence.
- ✓ The modulation technique in this specification is either two-level FSK or four-level FSK with 1 or 2 bits/baud, which results in a data rate of 1 or 2 Mbps, as shown in Figure 2.33.

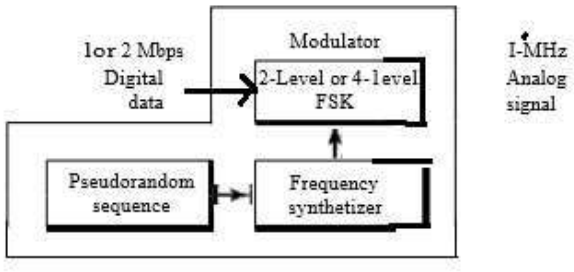


Figure 2.33 *Physical layer ofIEEE 802.11 FHSS*

- o *IEEE 802.11 DSSS*
  - ✓ IEEE 802.11 DSSS uses the direct sequence spread spectrum (DSSS) method. DSSS uses the 2.4-GHz ISM band.
  - ✓ The modulation technique in this specification is PSK at 1Mbaud/s.
  - ✓ The system allows 1or 2 bits/baud (BPSK or QPSK), which results in a data rate of 1 or 2 Mbps, as shown in Figure 2.34.

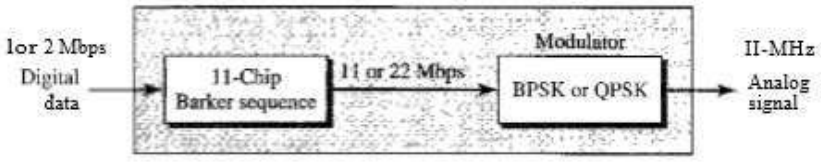


Figure 2.34 *Physical layer ofIEEE 802.11 FHSS*

- o *IEEE 802.11 Infrared*
  - ✓ IEEE 802.11 infrared uses infrared light in the range of 800 to 950 nm.
  - ✓ The modulation technique is called *pulse position modulation (PPM)*.
  - ✓ For a 1-Mbps data rate, a 4-bit sequence is first mapped into a 16-bit sequence in which only one bit is set to 1 and the rest are set to 0.
  - ✓ For a 2-Mbps data rate, a 2-bit sequence is first mapped into a 4-bit sequence in which only one bit is set to 1 and the rest are set to 0.
  - ✓ The mapped sequences are then converted to optical signals; the presence of light specifies 1, the absence of light specifies 0. See Figure 2.35.

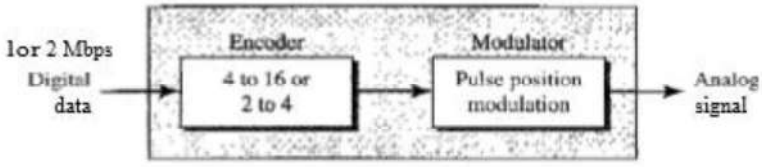


Figure 2.35 *Physical layer ofIEEE 802.11 infrared*

- o *IEEE 802.11a OFDM*
  - ✓ IEEE 802.1la OFDM describes the orthogonal frequency-division multiplexing (OFDM) method for signal generation in a 5-GHz ISM band.
  - ✓ OFDM is similar to FDM, with one major difference: All the sub bands are used by one source at a given time. Sources contend with one another at the data link layer for access.
  - ✓ The band is divided into 52 sub bands, with 48 sub bands for sending 48 groups of bits at a time and 4 sub bands for control information. Dividing the band into sub bands diminishes the effects of interference.
  - ✓ If the sub bands are used randomly, security can also be increased OFDM uses PSK and QAM for modulation. The common data rates are 18 Mbps (PSK) and 54 Mbps (QAM).
- o *IEEE 802.11b DSSS*

- ✓ IEEE 802.11b DSSS describes the high-rate direct sequence spread spectrum (HR- DSSS) method for signal generation in the 2.4-GHz ISM band.
- ✓ HR-DSSS is similar to DSSS except for the encoding method, which is called complementary code keying (CCK).
- ✓ CCK encodes 4 or 8 bits to one CCK symbol. To be backward compatible with DSSS, HR-DSSS defines four data rates: 1,2, 5.5, and 11 Mbps.
- ✓ The first two use the same modulation techniques as DSSS. The 5.5-Mbps version uses BPSK and transmits at 1.375 Mbaud/s with 4-bit CCK encoding.
- ✓ The II-Mbps version uses QPSK and transmits at 1.375 Mbps with 8-bit CCK encoding.
- ✓ Figure 2.36 shows the modulation technique for this standard. This new specification defines forward error correction and OFDM using the 2.4-GHz ISM band.

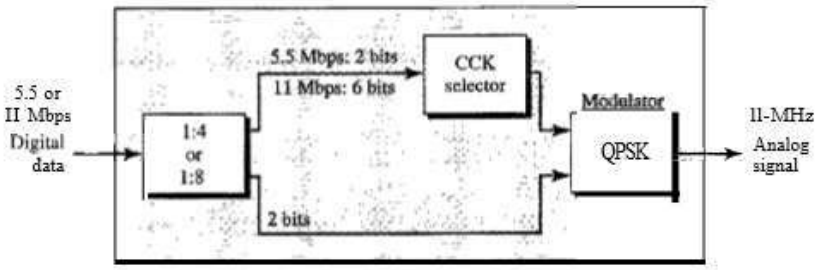


Figure 2.36 *Physical layer of IEEE 802.11b*

- ✓ The modulation technique achieves a 22- or 54-Mbps data rate. It is backward- compatible with 802.11b, but the modulation technique is OFDM.

2.3.2. BLUETOOTH - IEEE 802.15

- Bluetooth is a wireless LAN technology designed to connect devices of different functions such as telephones, notebooks, computers (desktop and laptop), cameras, printers, coffee makers, and so on.
- A Bluetooth LAN is an ad hoc network, which means that the network is formed spontaneously; the devices, sometimes called gadgets, find each other and make a network called *a piconet*.
- A Bluetooth LAN can even be connected to the Internet if one of the gadgets has this capability.
- Home security devices can use this technology to connect different sensors to the main security controller.

Architecture

Bluetooth defines two types of networks: *piconet and scatternet.*

Piconets

- A Bluetooth network is called a piconet, or a small net.
- A piconet can have up to eight stations, one of which is called the primary the rest are called secondary's.
- All the secondary stations synchronize their clocks and hopping sequence with the primary. Note that a piconet can have only one primary station.
- The communication between the primary and the secondary can be one-to-one or one-to-many. Figure 2.37 shows a piconet.

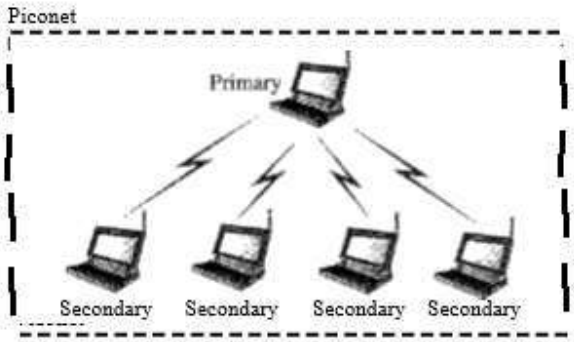


Figure 2.37 *Piconet*

- A piconet can have a maximum of *seven secondaries*, an *additional eight secondaries* can be in the parked state.
- A secondary in a parked state is synchronized with the primary, but cannot take part in communication until it is moved from the parked state. Because only eight stations can be active in a piconet, activating a station from the parked state means that an active station must go to the parked state.

Scatternet

- Piconets can be combined to form what is called a scatternet.
- A secondary station in one piconet can be the primary in another piconet. This station can receive messages. Figure 2.38 illustrates a scatternet.

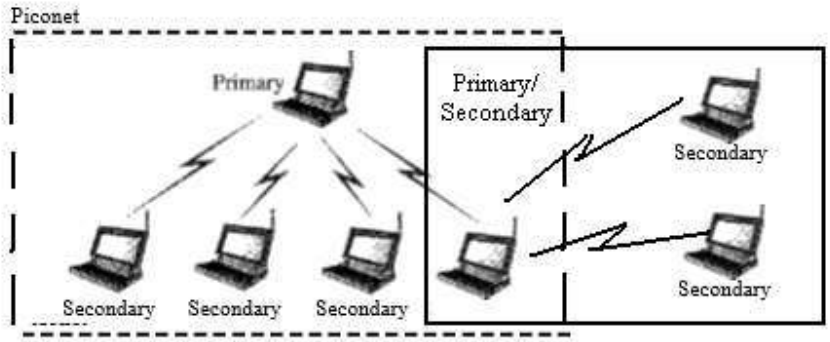


Figure 2.38 **Scatternet**

Bluetooth Devices

- A Bluetooth device has a built-in short-range radio transmitter.
- The current data rate is 1Mbps with a 2.4-GHz bandwidth. This means that there is a possibility of interference between the IEEE 802.11b wireless LANs and Bluetooth LANs.

Bluetooth Layers

Bluetooth uses several layers Figure 2.39 shows these layers.

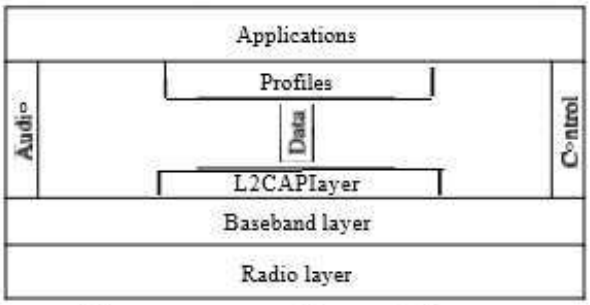


Figure 2.39 *Bluetooth layers*

Radio Layer

The radio layer is roughly equivalent to the physical layer of the Internet model. Bluetooth devices are low-power and have a range of 10 m.

✓ *Band :*Bluetooth uses a 2.4-GHz ISM band divided into 79 channels of 1 MHz each.

✓ *FHSS*

- Bluetooth uses the frequency-hopping spread spectrum (FHSS) method in the physical layer to avoid interference from other devices or other networks.
- Bluetooth hops 1600 times per second, which means that each device changes its modulation frequency 1600 times per second.
- A device uses a frequency for only 625µs (1/1600 s) before it hops to another frequency; the dwell time is 625µs.

✓ *Modulation*

- To transform bits to a signal, Bluetooth uses a sophisticated version of FSK, called GFSK (FSK with Gaussian bandwidth filtering.
- GFSK has a carrier frequency.
- Bit 1 is represented by a frequency deviation above the carrier; bit 1 is represented by a frequency deviation below the carrier.
- The frequencies, in megahertz, are defined according to the following formula for each channel:

$$f_c=2402+n \qquad n=0, 1,2,3, ..., 78$$

- For example, the first channel uses carrier frequency 2402 MHz (2.402 GHz), and the second channel uses carrier frequency 2403 MHz (2.403 GHz).

Baseband Layer

- The baseband layer is roughly equivalent to the MAC sublayer in LANs. The access method is TDMA.
- The primary and secondary communicate with each other using time slots. The length of a time slot is exactly the same as the dwell time, 625 µs. This means that during the time that one frequency is used, a sender sends a frame to a secondary, or a secondary sends a frame to the primary.

TDMA

- Bluetooth uses a form of TDMA that is called TDD-TDMA (time- division duplex TDMA).
- TDD-TDMA is a kind of half-duplex communication in which the secondary and receiver send and receive data, but not at the same time (half- duplex); however, the communication for each direction uses different hops.
- This is similar to walkie-talkies using different carrier frequencies.

Single-Secondary Communication

- If the piconet has only one secondary, the TDMA operation is very simple. The time is divided into slots of 625 µs.
- The primary uses even- numbered slots (0, 2, 4, ...); the secondary uses odd-numbered slots (1, 3, 5, ...).
- TDD-TDMA allows the primary and the secondary to communicate in half-duplex mode.
- In slot 0, the primary sends, and the secondary receives; in slot 1, the secondary sends, and the primary receives. The cycle is repeated. Figure 2.40shows the concept.

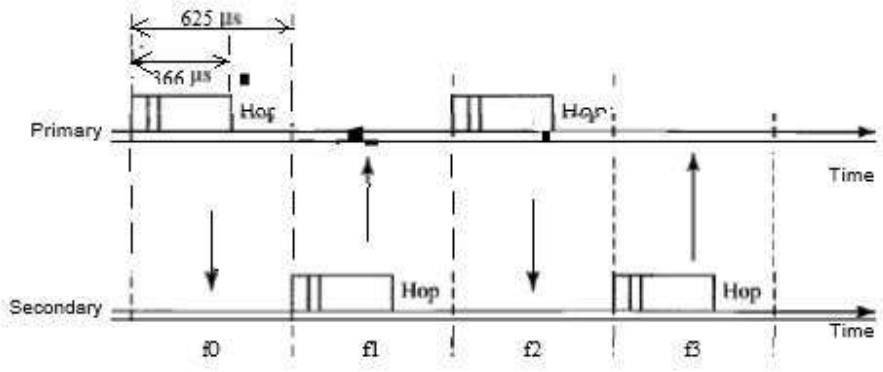


Figure 2.40 *Single-secondary communication*

Multiple-Secondary Communication

- The process is a little more involved if there is more than one secondary in the piconet. Again, the primary uses the even-numbered slots, but a secondary sends in the next odd-numbered slot if the packet in the previous slot was addressed to it.
- All secondaries listen on even-numbered slots, but only one secondary sends in any odd-numbered slot. Figure 2.41 shows a scenario.

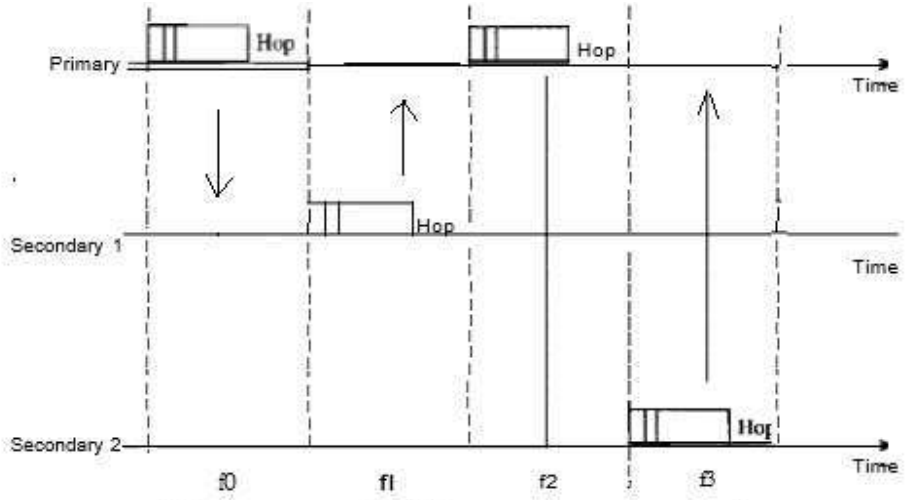


Figure 2.41 *Multiple-secondary communication*

- Let us elaborate on the figure.
  1. In slot 0, the primary sends a frame to secondary 1. 2. In slot 1, only secondary I sends a frame to the primary because the previous frame was addressed to secondary 1; other secondaries are silent.
  2. In slot 2, the primary sends a frame to secondary 2. 4. In slot 3, only secondary 2 sends a frame to the primary because the previous frame was addressed to secondary 2; other secondaries are silent. 5. The cycle continues.

Physical Links

❖ Two types of links can be created between a primary and a secondary: *SCO links and ACL links*.

➢ *SCO A synchronous connection-oriented (SCO) link* is used when avoiding latency (delay in data delivery) is more important than integrity (error-free delivery).

- ✓ In an SCO link, a physical link is created between the primary and a secondary by reserving specific slots at regular intervals.
- ✓ The basic unit of connection is two slots, one for each direction.
- ✓ If a packet is damaged, it is never retransmitted. SCO is used for real-time audio where avoiding delay is all-important.
- ✓ A secondary can create up to three SCO links with the primary, sending digitized audio (PCM) at 64 kbps in each link.

➢ *ACL An asynchronous connectionless link (ACL)* is used when data integrity is more important than avoiding latency.

- ✓ In this type of link, if a payload encapsulated in the frame is corrupted, it is retransmitted.
- ✓ A secondary returns an ACL frame in the available odd-numbered slot if and only if the previous slot has been addressed to it. ACL can use one, three, or more slots and can achieve a maximum data rate of 721 kbps.

Frame Format

A frame in the baseband layer can be one of three types: *one-slot, three-slot, or five-slot*. A slot, is 625µs. In a one-slot frame exchange, 259µs is needed for hopping and control mechanisms. This means that a one-slot frame can last only 625 - 259, or 366µs.

- ✓ With a I-MHz bandwidth and 1 bit/Hz, the size of a one-slot frame is 366 bits. A three-slot frame occupies three slots. However, since 259µs is used for hopping, the length of the frame is 3 x 625 - 259 = 1616µs or 1616 bits.
- ✓ A device that uses a three-slot frame remains at the same hop (at the same carrier frequency) for three slots.
- A five-slot frame also uses 259 bits for hopping, which means that the length of the frame is 5 x 625 - 259 =2866 bits. Figure 2.42 shows the format of the three frame types. The following describes each field:
  - *Access code.* This 72-bit field normally contains synchronization bits and the identifier of the primary to distinguish the frame of one piconet from another.
  - *Header.* This 54-bit field is a repeated I8-bit pattern. Each pattern has the following subfields:
    1. *Address.* The 3-bit address subfield can define up to seven secondaries (1 to 7). If the address is zero, it is used for broadcast communication from the primary to all secondaries.
    2. *Type.* The 4-bit type subfield defines the type of data coming from the upper layers. We discuss these types later.
    3. *F.* This I-bit subfield is for flow control. When set (1), it indicates that the device is unable to receive more frames (buffer is full).
    4. *A.* This I-bit subfield is for acknowledgment. Bluetooth uses Stop-and-Wait ARQ; I bit is sufficient for acknowledgment.
    5. *S.* This I-bit subfield holds a sequence number. Bluetooth uses Stop-and-Wait ARQ; I bit is sufficient for sequence numbering.
    6. *HEC.* The 8-bit header error correction subfield is a checksum to detect errors in each 18-bit header section.

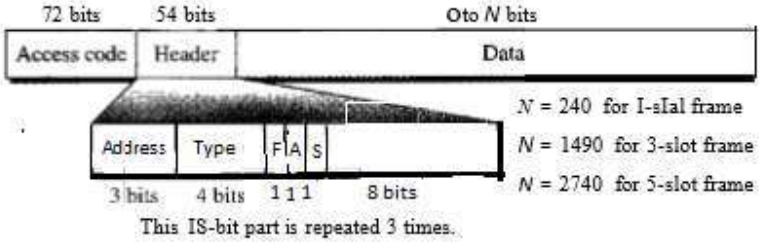


Figure 2.42 *Framefannat types*

The header has three identical 18-bit sections. The receiver compares these three sections, *bit by bit*. If each of the corresponding bits is the same, the bit is accepted; if not, the majority opinion rules. This is a form of forward error correction (for the header only).

  - *Payload.* This subfield can be 0 to 2740 bits long. It contains data or control information corning from the upper layers.

L2CAP

- The Logical Link Control and Adaptation Protocol, or L2CAP (L2 here means LL), is used for data exchange on an ACL link; SCO channels do not use L2CAP.
- Figure 2.43 shows the format of the data packet at this level.
  - ✓ The I6-bit length field defines the size of the data, in bytes, coming from the upper layers.

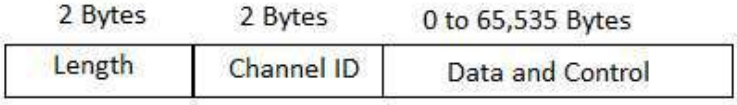


Figure 2.43 L2CAP data packet

- ✓ Data can be up to 65,535 bytes.
- ✓ The channel ID (CID) defines a unique identifier for the virtual channel created at this level.
- ✓ The L2CAP has specific duties: multiplexing, segmentation and reassembly, quality of service (QoS), and group management.

- *Multiplexing*
  - The L2CAP can do multiplexing.
  - At the sender site, it accepts data from one of the upper-layer protocols, frames them, and delivers them to the baseband layer.
  - At the receiver site, it accepts a frame from the baseband layer, extracts the data, and delivers them to the appropriate protocol layer.
  - It creates a kind of virtual channel that we will discuss in later chapters on higher-level protocols.
- *Segmentation and Reassembly*
  - The maximum size of the payload field in the baseband layer is 2774 bits, or 343 bytes. This includes 4 bytes to define the packet and packet length. Therefore, the size of the packet that can arrive from an upper layer can only be 339 bytes.
  - The L2CAP divides these large packets into segments and adds extra information to define the location of the segments in the original packet. The L2CAP segments the packet at the source and reassembles them at the destination.
- *QoS*
  - Bluetooth allows the stations to define a quality-of-service level.
- *Group Management*
  - L2CAP is to allow devices to create a type of logical addressing between themselves.
  - This is similar to multicasting. For example, two or three secondary devices can be part of a multicast group to receive data from the primary.

Other Upper Layers Bluetooth defines several protocols for the upper layers that use the services of L2CAP.

2.4. SWITCHING AND BRIDGING

❖ A switch is a mechanism that allows us to interconnect links to form a larger network. A switch is a multi-input, multi-output device that transfers packets from an input to one or more outputs. Thus, a switch adds the star topology to the point-to-point link, bus (Ethernet), and ring topologies.

❖ A star topology has several attractive properties:

- Even though a switch has a fixed number of inputs and outputs, which limits the number of hosts that can be connected to a single switch, large networks can be built by interconnecting a number of switches can connect switches to each other and to hosts using point-to-point links
- Adding a new host to the network by connecting it to a switch does not necessarily reduce the

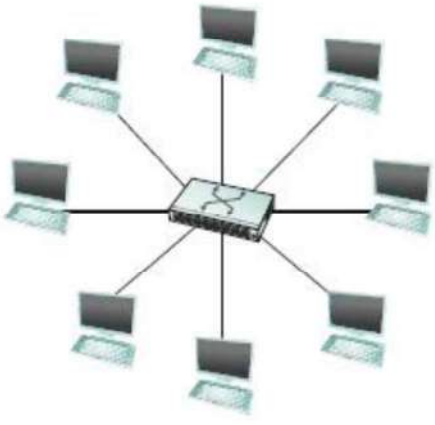


Figure 1.44 A switch provides a star topology

performance of the network for other hosts already connected.

- For example, it is impossible for two hosts on the same 10-Mbps Ethernet segment to transmit continuously at 10 Mbps because they share the same transmission medium.
- Every host on a switched network has its own link to the switch, so it may be entirely possible for many hosts to transmit at the full link speed (bandwidth), provided that the switch is designed with enough aggregate capacity.
- Goal: Providing high aggregate throughput is one of the design for a switch.
- Switched networks are considered more *scalable* (i.e., more capable of growing to large numbers of nodes)at full speed.
- A switch is connected to a set of links and, for each of these links, runs the appropriate data link protocol to communicate with the node at the other end of the link.
- A switch's primary job is to receive incoming packets on one of its links and to transmit them on some other link. This function is sometimes referred to as either *switching* or *forwarding,* and in the Open Systems Interconnection (OSI) architecture, it is the main function of the network layer.
- The header of the packet is used to make the decision.
- Two common approaches
  - ✓ D*atagram* or *connectionless* approach.
  - ✓ V*irtual circuit* or *connection-oriented* approach.
  - ✓ *Source routing*
- All networks have a way to identify the end nodes. Such identifiers are usually called *addresses*.
- All Ethernet cards are assigned a *globally unique* identifier.
- There are at least two sensible ways to identify ports: One is *to number each port*, and the other *is to identify the port by the name of the node* (switch or host) to which it leads.

2.4.1. Datagrams

- Here, every packet contains the complete destination address.
- Consider the example network illustrated in Figure 2.45, in which the hosts have addresses A, B, C, and so on.
  - To decide how to forward a packet, a switch consults a *forwarding table* (sometimes called a *routing table*), an example of which is depicted in Table 3.1.
  - This particular table shows the forwarding information that switch2 needs to forward datagrams in the example network.
- *Routing* is as a process that takes place in the background so that, when a data packet turns up, we will have the right information in the forwarding table to be able to forward, or switch, the packet.
- Datagram networks have the following characteristics:
  - ✓ A host can send a packet anywhere at any time, since any packet that turns up at a switch can be immediately forwarded (assuming a correctly populated forwarding table). For this reason, datagram networks are often called *connectionless*; this contrasts with the *connection-oriented* networks, in which some *connection state* needs to be established before the first data packet is sent.
  - ✓ When a host sends a packet, it has no way of knowing if the network is capable of delivering it or if the destination host is even up and running.

- ✓ Each packet is forwarded independently of previous packets that might have been sent to the same destination. Thus, two successive packets from host A to host B may follow completely different paths (perhaps because of a change in the forwarding table at some switch in the network).
- ✓ A switch or link failure might not have any serious effect on communication if it is possible to find an alternate route around the failure and to update the forwarding table accordingly.

2.4.2. Virtual Circuit Switching

- ❖ It uses the concept of a *virtual circuit* (VC).
- ❖ This approach, which is also referred to as a *connection-oriented model*, requires setting up a virtual connection from the source host to the destination host before any data is sent.
- ❖ Consider Figure 2.46, where host A again wants to send packets to host B. This is a two-stage process.
  1. connection setup phase
  2. Data Transfer phase

Connection setup

- It is necessary to establish a "connection state" in each of the switches between the source and destination hosts.
  - o The connection state for a single connection consists of an entry in a "VC table" in each switch through which the connection passes.
  - o One entry in the VC table on a single switch contains:

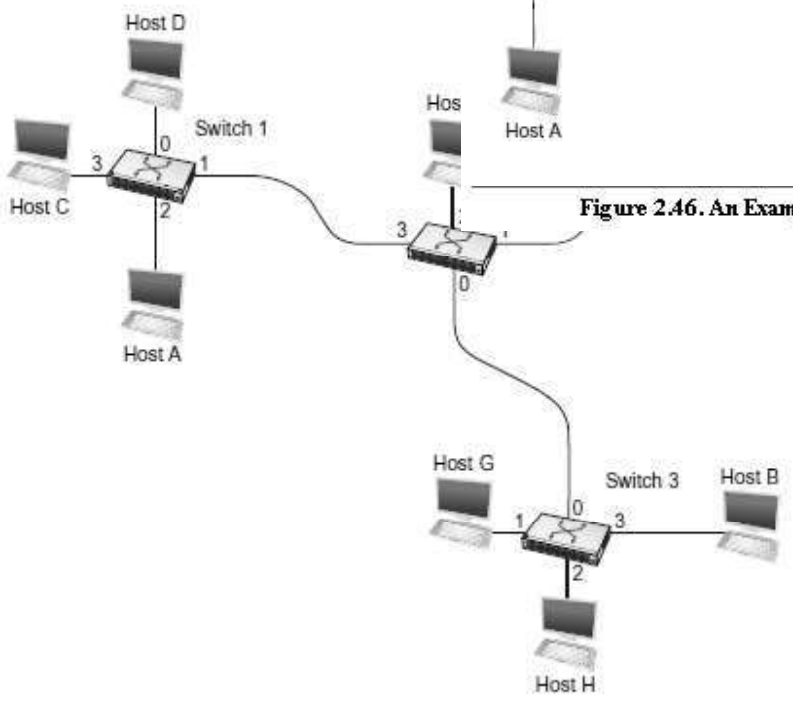


Fig 2.45 Datagram Forwarding: an example network

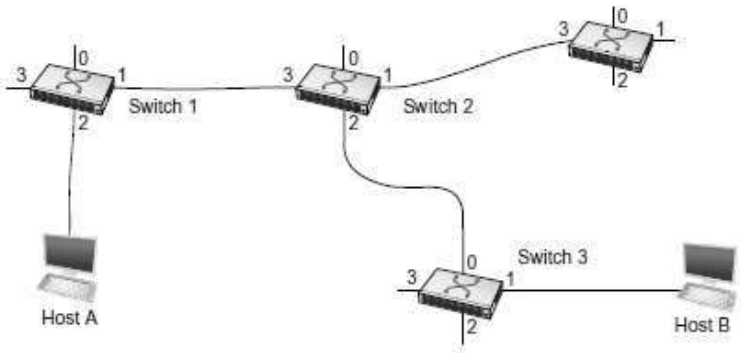


Figure 2.46. An Example of a virtual circuit networks

Destination	Port
A	3
B	0
C	3
D	3
E	2
F	1
G	0
H	0

- A *virtual circuit identifier* (VCI) that uniquely identifies the connection at this switch and which will be carried inside the header of the packets that belong to this connection
- An incoming interface on which packets for this VC arrive at the switch
- An outgoing interface in which packets for this VC leave the switch n A potentially different VCI that will be used for outgoing packets
- The semantics of one such entry is as follows:
  - ✓ If a packet arrives on the designated incoming interface and that packet contains the designated VCI value in its header, then that packet should be sent out the specified outgoing interface with the specified outgoing VCI value having been first placed in its header.
- o Whenever a new connection is created, we need to assign a new VCI for that connection on each link that the connection will traverse.
- o There are two broad approaches to establishing connection state.
  - One is to have a network administrator configure the state, in which case the virtual circuit is "permanent".
  - A host can send messages into the network to cause the state to be established. This is referred to as *signalling*, and the resulting virtual circuits are said to be *switched*.

Permanent Approach

- Let's assume that a network administrator wants to manually create a new virtual connection from host A to host B.2 First, the administrator needs to identify a path through the network from A to B. In the example network of Figure 2.46, there is only one such path, but in general this may not be the case.

Table Virtual Circuit Table Entry for Switch 1

Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
2	5	1	11

- The administrator then picks a VCI value that is currently unused on each link for the connection. For the purposes of our example, let's suppose that the VCI value 5 is chosen for the link from host A to switch 1, and that 11 is chosen for the link from switch 1 to switch 2. In that case, switch 1 needs to have an entry in its VC table configured as shown in Table (switch 1).

Table Virtual Circuit Table Entries for Switches 2 and 3

VC Table Entry at Switch 2

Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
3	11	2	7

VC Table Entry at Switch 3

Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
0	7	1	4

- Similarly, suppose that the VCI of 7 is chosen to identify this connection on the link from switch 2 to switch 3 and that a VCI of 4 is chosen for the link from switch 3 to host B. In that case, switches 2 and 3 need to

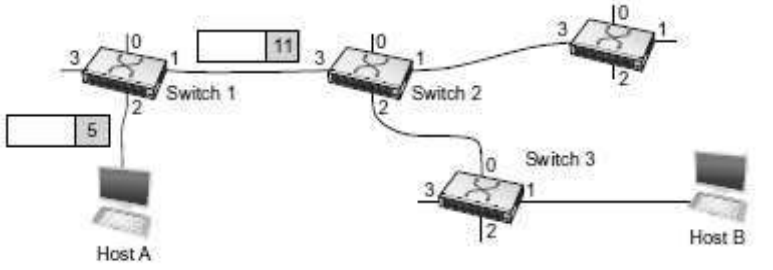


Figure 2.47. A packet is sent into a virtual circuit network

be configured with VC table entries as shown in Table (switches 2 & 3). Note that the "outgoing" VCI value at one switch is the "incoming" VCI value at the next switch.

- Once the VC tables have been set up, the data transfer phase can proceed, as illustrated in Figure 2.47. For any packet that it wants to send to host B, A puts the VCI value of 5 in the header of the packet and sends it to switch 1. Switch 1 receives any such packet on interface 2, and it uses the combination of the interface and the VCI in the packet header to find the appropriate VC table entry. As shown in Table (switch 1), the table entry in this case tells switch 1 to forward the packet out of interface 1 and to put the VCI value 11 in the header when the packet is sent.

  Thus, the packet will arrive at switch 2 on interface 3 bearing VCI 11. Switch 2 looks up interface 3 and VCI 11 in its VC table (as shown in Table (switch 2 & 3)) and sends the packet on to switch 3 after updating the VCI value in the packet header appropriately, as shown in Figure 2.48. This process continues until it arrives at host B with the VCI value of 4 in the packet. To host B, this identifies the packet as having come from host A.

Signalling Approach

- To start the signalling process, host A sends a setup message into the network—that is, to switch 1.
- The setup message contains, among other things, the complete destination address of host B.

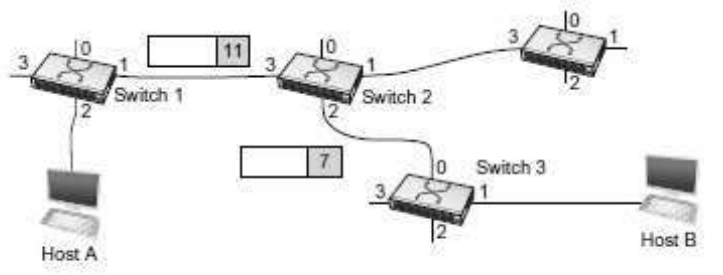


Figure 2.48 A packet makes its way through a virtual circuit network

- The setup message needs to get all the way to B to create the necessary connection state in every switch along the way.
- The setup message to B is a lot like getting a datagram to B, in that the switches have to know which output to send the setup message to so that it eventually reaches B. For now, let's just assume that the switches know enough about the network topology to figure out how to do that, so that the setup message flows on to switches 2 and 3 before finally reaching host B.
- When switch 1 receives the connection request, in addition to sending it on to switch 2, it creates a new entry in its virtual circuit table for this new connection. This entry is exactly the same as shown previously in Table (switch 1). The main difference is that now the task of assigning an unused VCI value on the interface is performed by the switch for that port.
- In this example, the switch picks the value 5. The virtual circuit table now has the following information: "When packets arrive on port 2 with identifier 5, send them out on port 1." Another issue is that, somehow, host A will need to learn that it should put the VCI value of 5 in packets that it wants to send to B; we will see how that happens below.
  - ✓ When switch 2 receives the setup message, it performs a similar process; in this example, it picks the value 11 as the incoming VCI value.

- ✓ Similarly, switch 3 picks 7 as the value for its incoming VCI. Each switch can pick any number it likes, as long as that number is not currently in use for some other connection on that port of that switch. As noted above, VCIs have link-local scope; that is, they have no global significance.
- ✓ Finally, the setup message arrives as host B. Assuming that B is healthy and willing to accept a connection from host A, it too allocates an incoming VCI value, in this case 4. This VCI value can be used by B to identify all packets coming from host A.
- ✓ Now, to complete the connection, everyone needs to be told what their downstream neighbor is using as the VCI for this connection.
- ✓ Host B sends an acknowledgment of the connection setup to switch 3 and includes in that message the VCI that it chose (4).
- ✓ Now switch 3 can complete the virtual circuit table entry for this connection, since it knows the outgoing value must be 4. Switch 3 sends the acknowledgment on to switch 2, specifying a VCI of 7. Switch 2 sends the message on to switch 1, specifying a VCI of 11.
- ✓ Finally, switch 1 passes the acknowledgment on to host A, telling it to use the VCI of 5 for this connection.
- ✓ At this point, everyone knows all that is necessary to allow traffic to flow from host A to host B. Each switch has a complete virtual circuit table entry for the connection. Furthermore, host A has a firm acknowledgment that everything is in place all the way to host B.
- ✓ At this point, the connection table entries are in place in all three switches just as in the administratively configured example above, but the whole process happened automatically in response to the signalling message sent from A.

Data transfer phase

- ➢ When host A no longer wants to send data to host B, it tears down the connection by sending a teardown message to switch 1. The switch removes the relevant entry from its table and forwards the message on to the other switches in the path, which similarly delete the appropriate table entries.
- ➢ At this point, if host A were to send a packet with a VCI of 5 to switch 1, it would be dropped as if the connection had never existed.
- ➢ There are several things to note about virtual circuit switching:
  - ✓ Since host A has to wait for the connection request to reach the far side of the network and return before it can send its first data packet, there is at least one round-trip time (RTT) of delay before data is sent 3
  - ✓ While the connection request contains the full address for host B (which might be quite large, being a global identifier on the network), each data packet contains only a small identifier, which is only unique on one link. Thus, the per-packet overhead caused by the header is reduced relative to the datagram model.
  - ✓ If a switch or a link in a connection fails, the connection is broken and a new one will need to be established. Also, the old one needs to be torn down to free up table storage space in the switches.

- ✓ The issue of how a switch decides which link to forward the connection request on has been glossed over. In essence, this is the same problem as building up the forwarding table for datagram forwarding, which requires some sort of *routing algorithm*.

❖ One of the nice aspects of virtual circuits is that by the time the host gets the go-ahead to send data, it knows quite a lot about the network - for example, that there really is a route to the receiver and that the receiver is willing and able to receive data. It is also possible to allocate resources to the virtual circuit at the time it is established.

❖ For example, X.25 was an early (and now largely obsolete) virtual-circuit-based networking technology.

❖ X.25 networks employ the following three-part strategy:

1. Buffers are allocated to each virtual circuit when the circuit is initialized.
2. The sliding window protocol (Section 2.5) is run between each pair of nodes along the virtual circuit, and this protocol is augmented with flow control to keep the sending node from over-running the buffers allocated at the receiving node.
3. The circuit is rejected by a given node if not enough buffers are available at that node when the connection request message isprocessed.

❖ each node is ensured of having the buffers it needs to queue the packets that arrive on that circuit. This basic strategy is usually called *hop-by-hop flow control.*

❖ *Comparison*

1. In A datagram network has no connection establishment phase, and each switch processes each packet independently.
   - Each arriving packet competes with all other packets for buffer space. If there are no free buffers, the incoming packet must be discarded.
2. In the virtual circuit model, each circuit with a different *quality of service* (QoS). The term *quality of service* is usually taken to mean that the network gives the user some kind of performance-related guarantee, which in turn implies that switches set aside the resources they need to meet this guarantee.
   - For example, the switches along a given virtual circuit might allocate a percentage of each outgoing link's bandwidth to that circuit.
   - As another example, a sequence of switches might ensure that packets belonging to a particular circuit not be delayed (queued) for more than a certain amount of time.
   - Eg. of virtual circuit technologies are X.25, Frame Relay, and Asynchronous Transfer Mode (ATM).
   - One of the most common applications of virtual circuits was the construction of *virtual private networks* (VPNs).

2.4.2.1. ATM

Asynchronous Transfer Mode(ATM) is the cell relay protocol designed by the ATM Forum and adopted by the ITU-T.

A. Six Design Goals

1. Foremost is the need for a transmission system to optimize the use of high-data-rate transmission media, in particular optical fiber. In addition to offering large bandwidths, newer transmission media and equipment are dramatically less susceptible to noise degradation. A technology is needed to take advantage of both factors and thereby maximize data rates.

2. The system must interface with existing systems and provide wide-area interconnectivity between them without lowering their effectiveness or requiring their replacement.
3. The design must be implemented inexpensively so that cost would not be a barrier to adoption. If ATM is to become the backbone of international communications, as intended, it must be available at low cost to every user who wants it.
4. The new system must be able to work with and support the existing telecommunications hierarchies (local loops, local providers, long-distance carriers, and so on).
5. The new system must be connection-oriented to ensure accurate and predictable delivery.
6. Last but not least, one objective is to move as many of the functions to hardware as possible (for speed) and eliminate as many software functions as possible (again for speed).

B. Architecture

- ATM is a cell-switched network. The user access devices, called the endpoints, are connected through a user-to-network interface (UNI) to the switches inside the network.
- The switches are connected through network-to-network interfaces (NNIs).

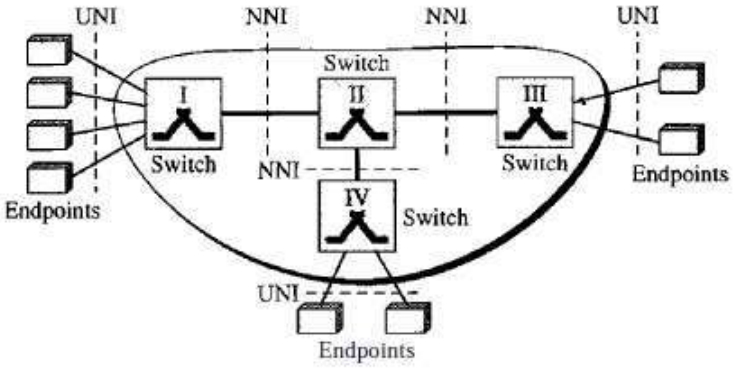


Figure 2.49 *Architecture of an ATM network*

Virtual Connection

- Connection between two endpoints is accomplished through transmission paths (TPs), virtual paths (YPs), and virtual circuits (YCs).
- A transmission path (TP) is the physical connection (wire, cable, satellite, and so on) between an endpoint and a switch or between two switches.

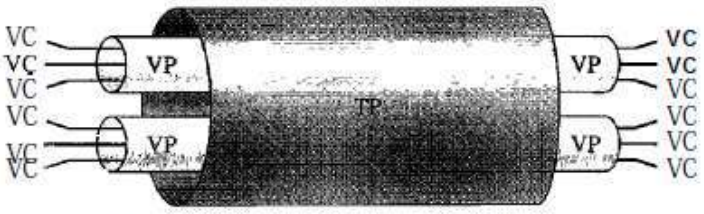


Figure 2.50 *Tp, VPs, and VCs*

- A transmission path is divided into several virtual paths. A virtual path (VP) provides a connection or a set of connections between two switches. Think of a virtual path as a highway that connects two cities. Each highway is a virtual path; the set of all highways is the transmission path.
- Cell networks are based on virtual circuits (VCs). All cells belonging to a single message follow the same virtual circuit and remain in their original order until they reach their destination. Think of a virtual circuit as the lanes of a highway (virtual path).
- Figure 2.50 shows the relationship between a transmission path (a physical connection), virtual paths (a combination of virtual circuits that are bundled together because parts of their paths are the same), and virtual circuits that logically connect two points.
- ***Identifiers***
  - ✓ In a virtual circuit network, to route data from one endpoint to another, the virtual connections need to be identified. Two levels of Identifiers:

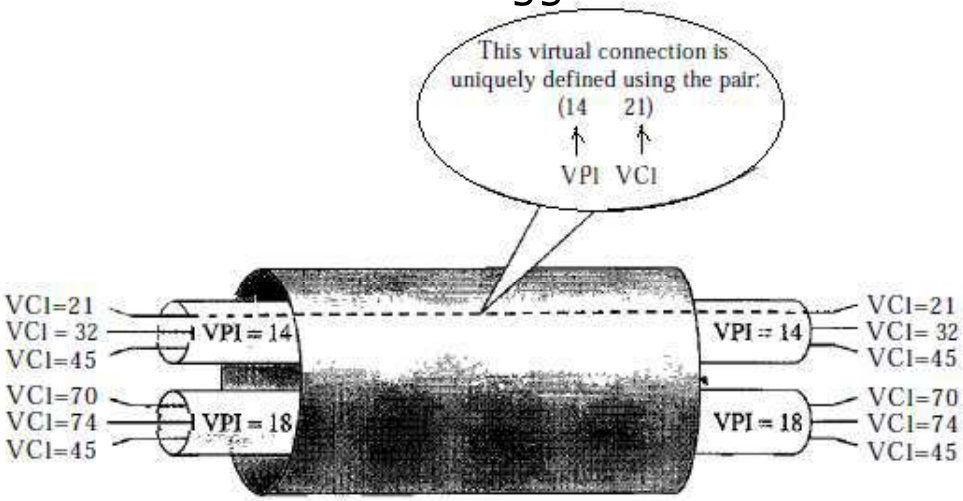


Figure 2.51 *Connection identifiers*

i. *Virtual path identifier (VPI)* : The VPI defines the specific VP. The VPI is the same for all virtual connections that are bundled (logically) into one VP.

ii. *Virtual-circuit identifier (VCI) :* The VCI defines a particular VC inside the VP..

✓ Figure 2.51 shows the VPIs and VCIs for a transmission path. The rationale for dividing an identifier into two parts will become clear when we discuss routing in an ATM network.

- The lengths of the VPIs for UNIs and NNIs are different. In a UNI, the VPI is 8 bits, whereas in an NNI, the VPI is 12 bits. The length of the VCI is the same in both interfaces (16 bits). We therefore can say that a virtual connection is identified by 24 bits in a UNI and by 28 bits in an NNI (see Figure 2.52) .

- ***Cells***

✓ The basic data unit in an ATM network is called a cell.

✓ A cell is only 53 bytes long with 5 bytes allocated to the header and 48 bytes carrying the payload than 48 bytes).

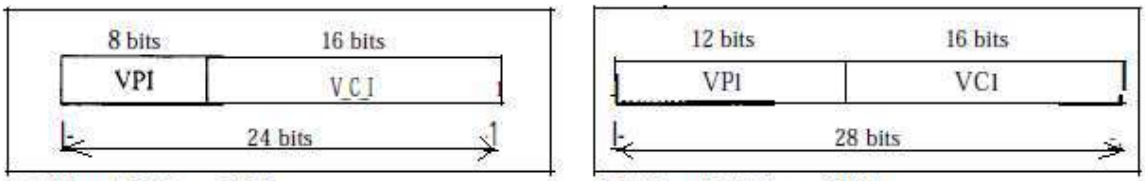


Figure 2.52 *Virtual connection identifiers in UNIs and NNIs*

- ***Connection Establishment and Release***

❖ ATM uses two types of connections:

i. *PVC* : A permanent virtual-circuit connection is established between two endpoints by the network provider. The VPIs and VCIs are defined for the permanent connections, and the values are entered for the tables of each switch.

ii. *SVC* : In a switched virtual-circuit

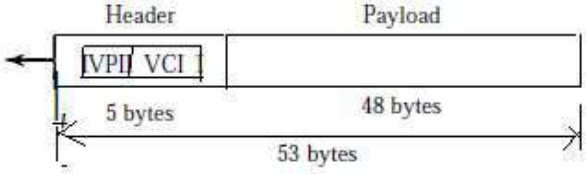


Figure 2.53 *An ATM cell*

connection, each time an endpoint wants to make a connection with another endpoint, a new virtual circuit must be established. The signaling mechanism of this other protocol makes a connection request by using the network layer addresses of the two endpoints. The actual mechanism depends on the network layer protocol.

C. *Switching*

- ATM uses switches to route the cell from a source endpoint to the destination endpoint.
- A switch routes the cell using both the VPIs and the VCIs. The routing requires the whole identifier.
- Figure 2.54 shows how a VPC switch routes the cell. A cell with a VPI of 153 and VCI of 67 arrives at switch interface (port) 1.
  - The switch checks its switching table, which stores six pieces of information per row: arrival interface number, incoming VPI, incoming VCI, corresponding outgoing interface number, the new VPI, and the new VCL The switch finds the entry with the interface 1, VPI 153, and VCI 67 and discovers that the combination corresponds to output interface 3, VPI 140, and VCI 92. It changes the VPI and VCI in the header to 140 and 92, respectively, and sends the cell out through interface 3.

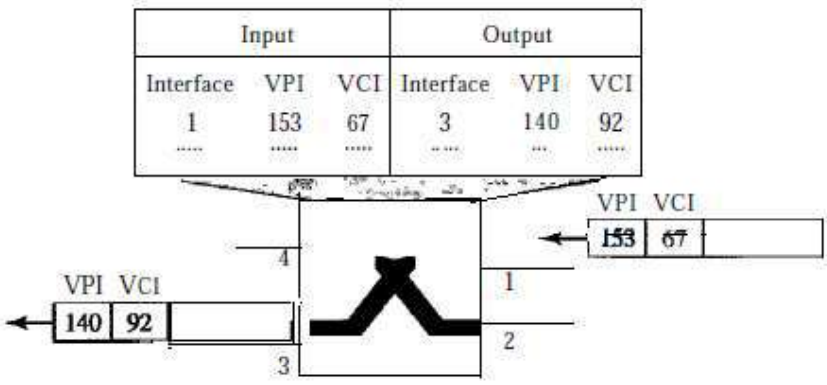


Figure 2.54 *Routing with a switch*

D. *ATM Layer*

- ✓ The ATM layer provides routing, traffic management, switching, and multiplexing services.
- ✓ It processes outgoing traffic by accepting 48-byte segments from the AAL sublayers and transforming them into 53-byte cells by the addition of a 5-byte header (see Figure 2.55).

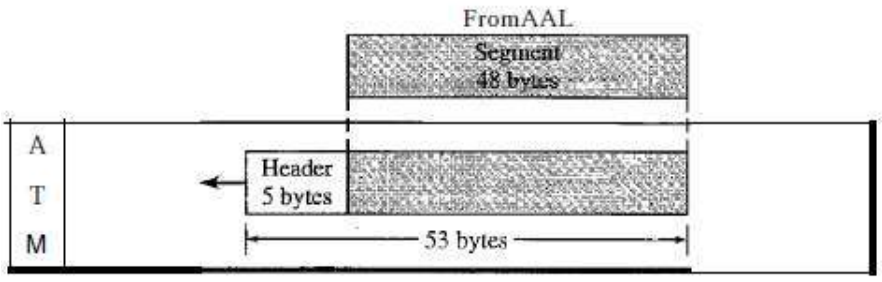


Figure 2.55 *ATM layer*

- ✓ *Header Format*
  - ATM uses two formats for this header, one for user-to-network interface (UNI) cells and another for network-to-network interface (NNI) cells.

- Figure 2.56 shows these headers in the byte-by-byte format preferred by the ITU-T (each row represents a byte).

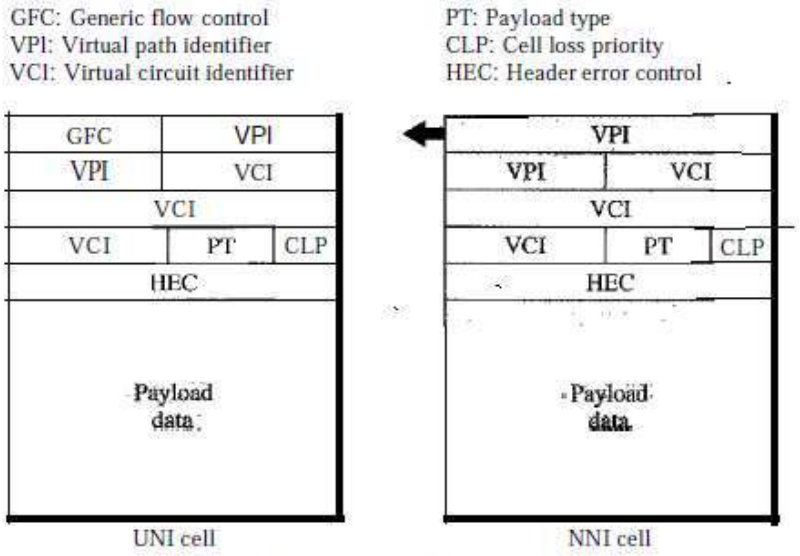


Figure 2.56 *ATM headers*

- *Generic flow control (GFC).* The 4-bit GFC field provides flow control at the UNI level.
  - *Virtual path identifier (VPI).* The VPI is an 8-bit field in a UNI cell and a 12-bit field in an NNI cell.
  - *Virtual circuit identifier (VCI).* The VCI is a 16-bit field in both frames.
  - *Payload type (PT).* In the 3-bit PT field, the first bit defines the payload as user data or managerial information. The interpretation of the last 2 bits depends on the first bit.
  - *Cell loss priority (CLP).* The 1-bit CLP field is provided for congestion control. A cell with its CLP bit set to I must be retained as long as there are cells with a CLP of O.
  - *Header error correction (HEC).* The HEC is a code computed for the first 4 bytes of the header. It is a CRC with the divisor $x8 + x2 + x + 1$ that is used to correct single-bit errors and a large class of multiple-bit errors.

E. Application Adaptation Layer

❖ The application adaptation layer (AAL) was designed to enable two ATM concepts.

✓ First, ATM must accept any type of payload, both data frames and streams of bits.

- A data frame can come from an upper-layer protocol that creates a clearly defined frame to be sent to a carrier network such as ATM.
- ATM must also carry multimedia payload. It can accept continuous bit streams and break them into chunks to be encapsulated into a cell at the ATM layer. AAL uses two sublayers to accomplish these tasks.

✓ Whether the data are a data frame or a stream of bits, the payload must be segmented into 48-byte segments to be carried by a cell. At the destination, these segments need to be reassembled to recreate the original payload.

✓ The AAL defines a sublayer, called a segmentation and reassembly (SAR) sublayer, to do so. Segmentation is at the source; reassembly, at the destination.

- ✓ Before data are segmented by SAR, they must be prepared to guarantee the integrity of the data. This is done by a sublayer called the convergence sublayer (CS).

i. AAL1:

- AAL1 supports applications that transfer information at constant bit rates, such as video and voice. It allows ATM to connect existing digital telephone networks such as voice channels and T lines.
- Figure 2.57 shows how a bit stream of data is chopped into 47-byte chunks and encapsulated in cells.
- The CS sublayer divides the bit stream into 47-byte segments and passes them to the SAR sublayer below.
- The SAR sublayer adds 1 byte of header and passes the 48-byte segment to the ATM layer. The header has two fields:

1. *Sequence number (SN).* This 4-bit field defines a sequence number to order the bits. The first bit is sometimes used for timing, which leaves 3 bits for sequencing (modulo 8).
2. *Sequence number protection (SNP).* The second 4-bit field protects the first field. The first 3 bits automatically correct the SN field. The last bit is a parity bit that detects error over all 8 bits.

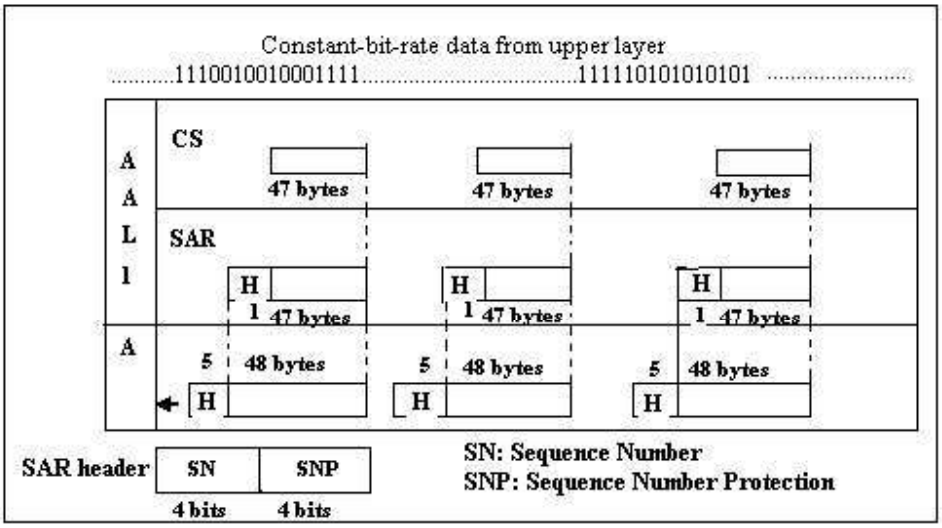


Figure 2.57. AAL1

ii. AAL2:

- AAL2 was intended to support a variable-data-rate bit stream, but it has been redesigned. It is now used for low-bit-rate traffic and short-frame traffic such as audio (compressed or uncompressed), video, or fax.
- A good example of AAL2 use is in mobile telephony.
- AAL2 allows the multiplexing of short frames into one cell.
- Figure 2.58. shows the process of encapsulating a short frame from the same source (the same user of a mobile phone) or from several sources (several users of mobile telephones) into one cell.
- **PPT:** Packet payload type
  - ✓ The CS layer overhead consists of five fields:
    - *Channel identifier (CID).* The 8-bit CID field defines the channel (user) of the short packet.
    - *Length indicator (LI).* The 6-bit LI field indicates how much of the final packet is data.
    - *Packet payload type (PPT).* The PPT field defines the type of packet.
    - *User-to-user indicator (UUI).* The UUI field can be used by end-to-end users.

- *Header error control (HEC).* The last 5 bits is used to correct errors in the header.

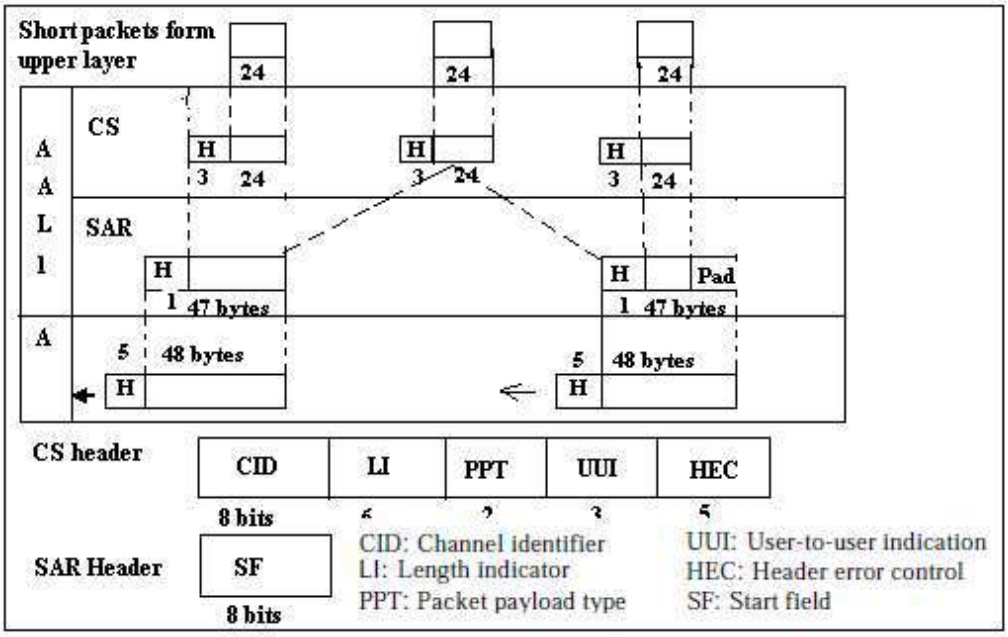


Figure 2.58 AAL 2

- The only overhead at the SAR layer is the start field (SF) that defines the offset from the beginning of the packet.

3. **AAL3/4**

- AAL3/4 provides comprehensive sequencing and error control mechanisms.
- Figure 2.59 shows the AAL3/4 sublayer.
  - The CS layer header and trailer consist of six field so Common part identifier (CPI). The CPI defines how the subsequent fields are to be interpreted. The value at present is O.
  - *Begin tag (Btag).* The value of this field is repeated in each ceU to identify all the cells belonging to the same packet. The value is the same as the Etag (see below).
  - *Bufer allocation size (BAsize).* The 2-byte BA field tells the receiver what size buffer is needed for the coming data.
  - *Alignment (AL).* The I-byte AL field is included to make the rest of the trailer 4 bytes long.
  - *Ending tag (Etag).* The I-byte ET field serves as an ending flag. Its value is the same as that of the beginning tag.
  - *Length (L).* The 2-byte L field indicates the length of the data unit.
  - The SAR header and trailer consist of five fields:
    - *Segment type (ST).* The 2-bit ST identifier specifies the position of the segment in the message: beginning (00), middle (01), or end (10). A single-segment message has an ST of 11.
    - *Sequence number (SN).* This field is the same as defined previously.
    - *Multiplexing identifier (MID).* The 10-bit MID field identifies cells coming from different data flows and multiplexed on the same virtual connection.
    - *Length indicator (LI).* This field defines how much of the packet is data, not padding.

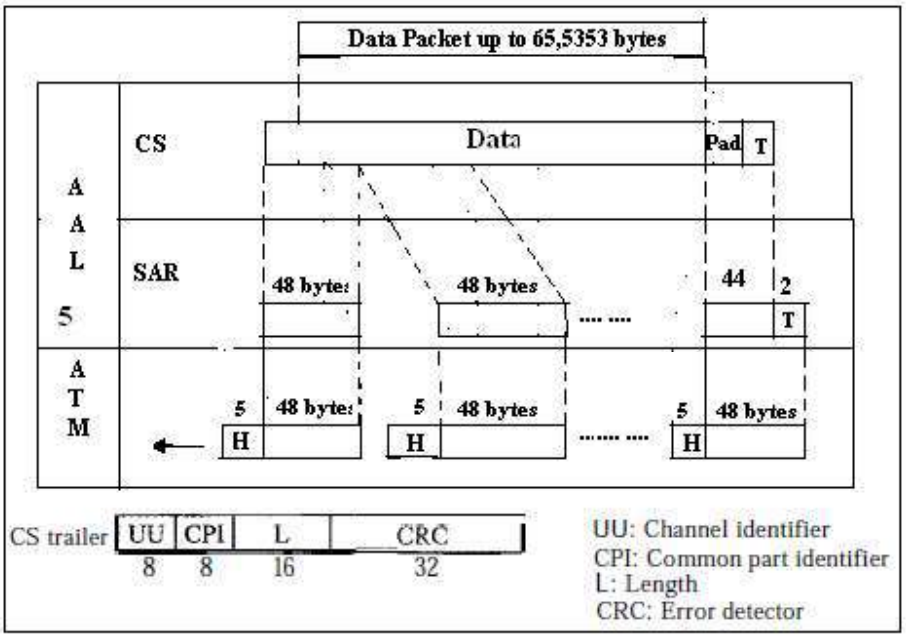


Figure 2.60 *AAL5*

- *CRC.* The last 10 bits of the trailer is a CRC for the entire data unit.

4. **AAL5**

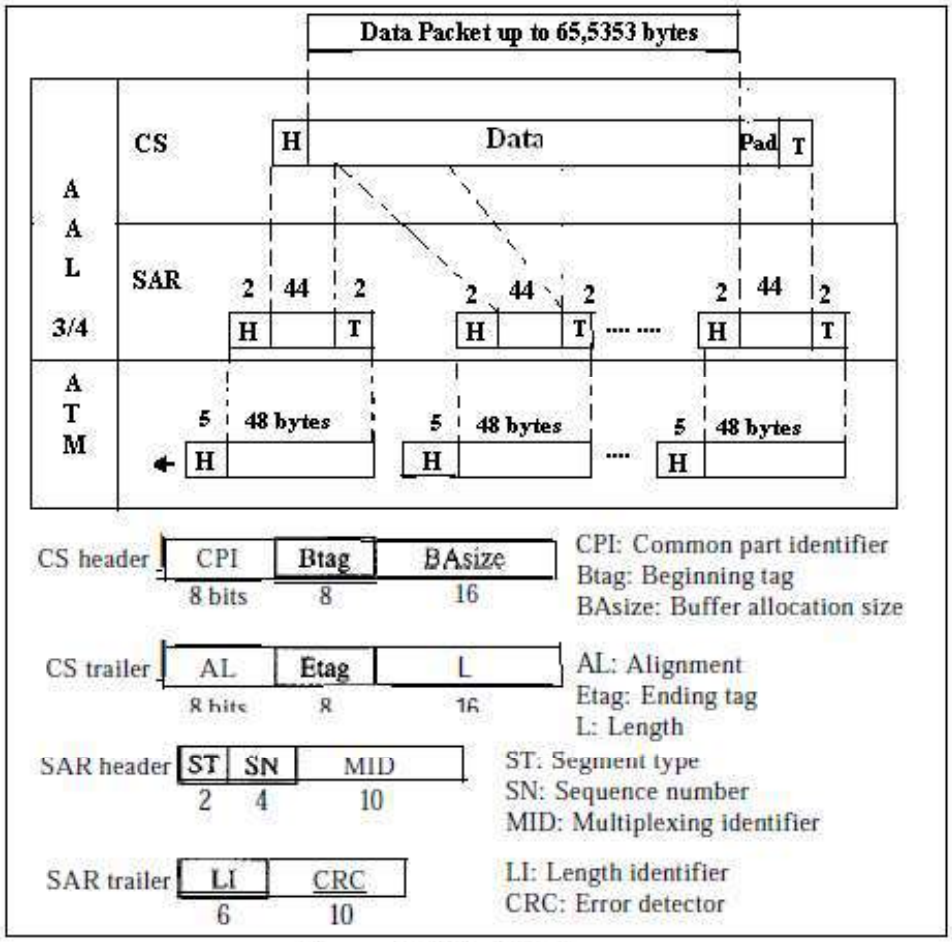


Figure 2.59 AAL 3/4

- Fifth AAL sublayer, called the simple and efficient adaptation layer (SEAL).
- AALS assumes that all cells belonging to a single message travel sequentially and that control functions are included in the upper layers of the sending application.
- Figure 2.225 shows the AAL5 sublayer.
- The four trailer fields in the CS layer are
  - *User-to-user (UU)*. This field is used by end users
  - *Common part identifier (CPI)*. This field is the same as defined previously.
  - *Length (L)*. The 2-byte L field indicates the length of the original data.
  - *CRC*. The last 4 bytes is for error control on the entire data unit.
  - *Congestion Control and Quality of Service :* ATM has a very developed congestion control and quality of service.

2.4.3. Source Routing

- A third approach to switching that uses neither virtual circuits nor conventional datagram is known as *source routing*.
- The name derives from the fact that all the information about network topology that is required to switch a packet across the network is provided by the source host.
- There are various ways to implement source routing.
  - One would be to assign a number to each output of each switch and to place that number in the header of the packet. The switching function is then very simple: For each packet that arrives on an input, the switch would read the port number in the header and transmit the packet on that output.
  - One way to do this would be to put an ordered list of switch ports in the header and to rotate the list so that the next switch in the path is always at the front of the list.

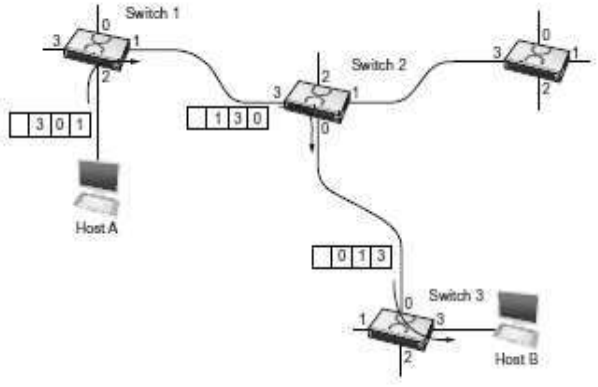


Figure 2.61 Source Routing in a switched network (where the switch reads the right most number)

- Figure 2.61 illustrates this idea. In this example, the packet needs to traverse three switches to get from host A to host B. At switch 1, it needs to exit on port 1, at the next switch it needs to exit at port 0, and at the third switch it needs to exit at port 3.
- Thus, the original header when the packet leaves host A contains the list of ports (3, 0, 1), where we assume that each switch reads the rightmost element of the list. To make sure that the next switch gets the appropriate information, each switch rotates the list after it has read its own entry.

- Thus, the packet header as it leaves switch 1 en route to switch 2 is now (1, 3, 0); switch 2 performs another rotation and sends out a packet with (0, 1, 3) in the header. Although not shown, switch 3 performs yet another rotation, restoring the header to what it was when host A sent it.
- There are several things to note about this approach.
  - First, it assumes that host A knows enough about the topology of the network to form a header that has all the right directions in it for every switch in the path.
  - Second, observe that we cannot predict how big the header needs to be, since it must be able to hold one word of information for every switch on the path.
  - Third, there are some variations on this approach. For example, rather than rotate the header, each switch could just strip the first element as it uses it. Rotation has an advantage over stripping, however: Host B gets a copy of the complete header, which may help it figure out how to get back to host A. Yet another alternative is to have the header carry a pointer to the current "next port" entry, so that each switch just updates the pointer rather than rotating the header; this may be more efficient to implement.
  - These three approaches in Figure 2.62. In each case, the entry that this switch needs to read is A, and the entry that the next switch needs to read is B.

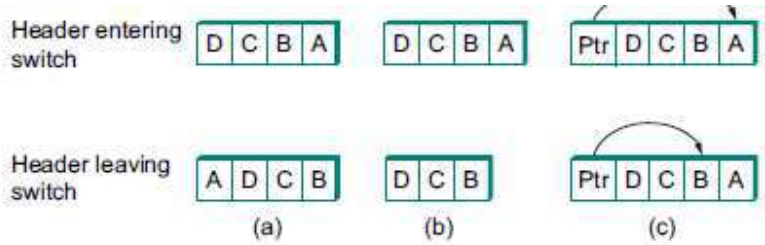


Figure 2.62. Three ways to handle headers for source routing (a)rotation (b)stripping (c) pointer. The tables are read from right to left

- ❖ Source routing is used in Datagram network and packet switched network. For example, the Internet Protocol, which is a datagram protocol, includes a source route option that allows selected packets to be source routed, while the majority are switched as conventional datagram.
- ❖ Source routing is also used in some virtual circuit networks as the means to get the initial setup request along the path from source to destination.
- ❖ Source routes are sometimes categorized as

  i. Strict. In a strict source route, every node along the path must be specified

  ii. Loose.
  - A loose source route only specifies a set of nodes to be traversed, without saying exactly how to get from one node to the next.
  - A loose source route can be thought of as a set of way points rather than a completely specified route.
  - The loose option can be helpful to limit the amount of information that a source must obtain to create a source route.

2.4.4. Bridges and LAN Switches

- ❖ A bridge is a multi-input, multi-output device, which transfers packets from an input to one or more outputs.

- This provides a way to increase the total bandwidth of a network. For example, while a single Ethernet segment might carry only 100 Mbps of total traffic, an Ethernet bridge can carry as much as 100n Mbps, where n is the number of ports (inputs and outputs) on the bridge.
- A class of switch that is used to forward packets between LANs (local area networks) such as Ethernets are sometimes known as LAN switches; also been referred to as *bridges*, and they are very widely used in campus and enterprise networks.
- Pair of Ethernets is interconnected using a *repeater*. Limitation is that no more than two repeaters between any pair of hosts and no more than a total of 2500 m in length are allowed.
- A collection of LANs connected by one or more bridges is usually said to form an *extended LAN*.

2.4.4.1. Learning Bridges

- Consider the bridge in Figure 2.63.
- Whenever a frame from host A that is addressed to host B arrives on port 1, there is no need for the bridge to forward the frame out over port 2.
- How does a bridge come to learn on which port the various hosts reside?
  - One option would be to have a human download a table into the bridge similar to the one given in Table. Then, whenever the bridge receives a frame on port 1 that is addressed to host A, it would not forward the frame out on port 2; there would be no need because host A would have already directly received the frame on the LAN connected to port 1.
  - Anytime a frame addressed to host A was received on port 2, the bridge would forward the frame out on port 1.
  - No one actually builds bridges in which the table is configured by hand.

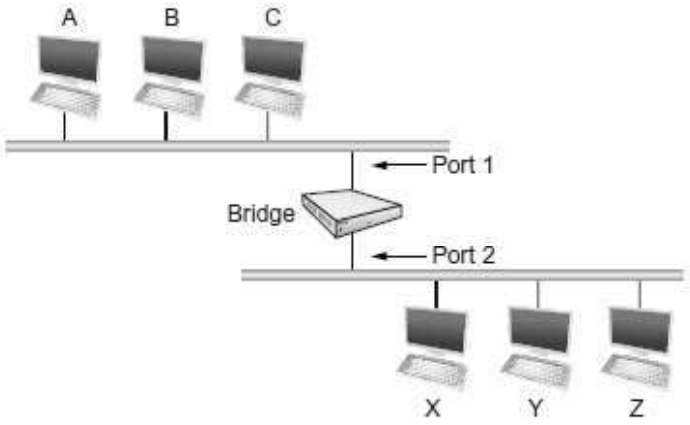


Figure 2.63 Illustration of a learning bridge

Table Forwarding Table Maintained by a Bridge

Host	Port
A	1
B	1
C	1
X	2
Y	2
Z	2

  - Having a human maintain this table is too burdensome, and there is a simple trick by which a bridge can learn this information for itself.
  - The idea is for each bridge to inspect the *source* address in all the frames it receives. Thus, when host A sends a frame to a host on either side of the bridge, the bridge receives this frame and records the fact that a frame from host A was just received on port 1.

- Each packet carries a global address, and the bridge decides which output to send a packet on by looking up that address in a table.
- When a bridge first boots, this table is empty; entries are added over time. Also, a timeout is associated with each entry, and the bridge discards the entry after a specified period of time. This is to protect against the situation in which a host—and, as a consequence, its LAN address- is moved from one network to another.
- This table is simply an optimization that filters out some frames; it is not required for correctness.

Implementation

❖ The code that implements the learning bridge algorithm

✓ Structure BridgeEntry defines a single entry in the bridge's forwarding table; these are stored in a Map structure (which supports mapCreate, mapBind, and mapResolve operations) to enable entries to be efficiently located when packets arrive from sources already in the table.

✓ The constant MAX TTL specifies how long an entry is kept in the table before it is discarded.

```
#define BRIDGE_TAB_SIZE 1024 /* max. size of bridging table */
#define MAX_TTL 120 /* time (in seconds) before an entry is flushed */
typedef struct
{
      MacAddr destination; /* MAC address of a node */
      int ifnumber; /* interface to reach it */
      u_short TTL; /* time to live */
      Binding binding; /* binding in the Map */
} BridgeEntry;
int numEntries = 0;
Map bridgeMap = mapCreate(BRIDGE_TAB_SIZE,sizeof(BridgeEntry));
```

✓ The routine that updates the forwarding table when a new packet arrives is given by updateTable.

✓ The arguments passed are the source media access control (MAC) address contained in the packet and the interface number on which it was received.

```
void updateTable (MacAddr src, int inif)
{
      BridgeEntry *b;
      if (mapResolve(bridgeMap, &src, (void **)&b) == FALSE )
      {
            /* this address is not in the table, so try to add it */
            if (numEntries < BRIDGE_TAB_SIZE)
            {
                  b = NEW(BridgeEntry);
                  b->binding = mapBind( bridgeMap, &src, b);
                  /* use source address of packet as dest. address in
                  table */
                  b->destination = src;
                  numEntries++;
```

```
                }
                else
                {
                        /* can't fit this address in the table now, so give up
                        */
                        return;
                }
        }
        /* reset TTL and use most recent input interface */
        b->TTL = MAX_TTL;
        b->ifnumber = inif;
}
```

- ✓ This implementation adopts a simple strategy in the case where the bridge table has become full to capacity—it simply fails to add the new address.
- ✓ If there is some entry in the table that is not currently being used, it will eventually time out and be removed, creating space for a new entry.

Spanning Tree Algorithm

- ❖ Example in Figure 2.64,
  - ➢ where, for example, bridges B1, B4, and B6 form a loop. Suppose that a packet enters bridge B4 from Ethernet J and that the destination address is one not yet in any bridge's forwarding table: B4 sends a copy of the packet out to Ethernets H and I. Now bridge B6 forwards the packet to Ethernet G, where B1 would see it and forward it back to Ethernet H; B4 still doesn't have this destination in its table, so it forwards the packet back to Ethernets I and J.
  - ➢ Packets looping in both directions among B1, B4, and B6.
  - ➢ Why would an extended LAN come to have a loop in it?
    - ✓ One possibility is that the network is managed by more than one administrator, for example, because it spans multiple departments in an organization.

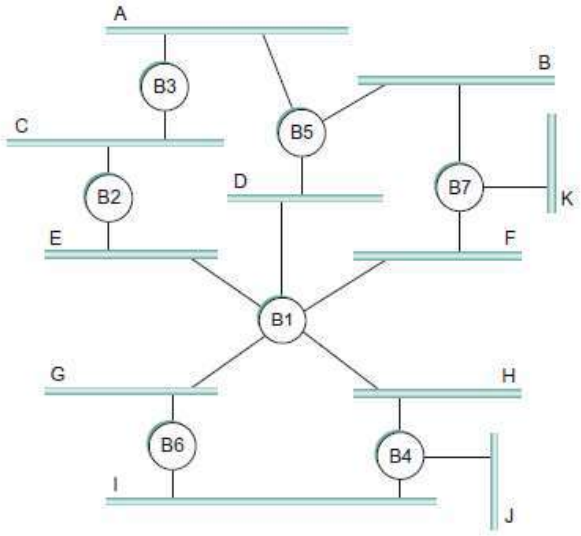


Figure 2.64 Extended LAN with Loop

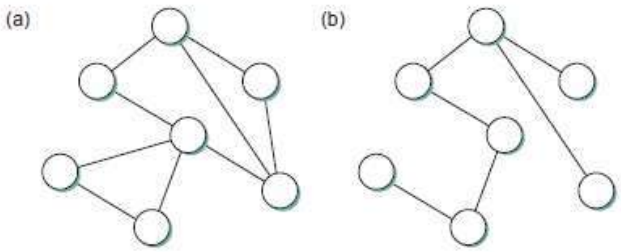


Figure 2.65. Example of (a) a cyclic graph; (b)a corresponding spanning tree

- ✓ A second, more likely scenario is that loops are built into the network on purpose—to provide redundancy in case of failure.
- ✓ A network with no loops needs only one link failure to become split into two separate partitions.

❖ A spanning tree keeps all of the vertices of the original graph but throws out some of the edges.

❖ For example, Figure 2.65 shows a cyclic graph on the left and one of possibly many spanning trees on the right.

❖ Idea of Spanning Tree

- ✓ It's a subset of the actual network topology that has no loops and that reaches all the LANs in the extended LAN.
- ✓ The hard part is how all of the bridges coordinate their decisions to arrive at a single view of the spanning tree.
- ✓ After all, one topology is typically able to be covered by multiple spanning trees.

❖ The spanning tree algorithm, which was developed by Radia Perlman at the Digital Equipment Corporation, is a protocol used by a set of bridges to agree upon a spanning tree for a particular extended LAN. (The IEEE 802.1 specification for LAN bridges is based on this algorithm.)

❖ This means that each bridge decides the ports over which it is and is not willing to forward frames. In a sense, it is by removing ports from the topology that the extended LAN is reduced to an acyclic tree.

❖ It is even possible that an entire bridge will not participate in forwarding frames, which seems kind of strange at first glance.

❖ The algorithm is dynamic, however, meaning that the bridges are always prepared to reconfigure themselves into a new spanning tree should some bridge fail, and so those unused ports and bridges provide the redundant capacity needed to recover from failures.

❖ The algorithm selects ports as follows.

- ✓ Each bridge has a unique identifier; for our purposes, we use the labels B1, B2, B3, and so on.
- ✓ The algorithm first elects the bridge with the smallest ID as the root of the spanning tree. This election takes place is described below.
  - o The root bridge always forwards frames out over all of its ports.
  - o Next, each bridge computes the shortest path to the root and notes which of its ports is on this path. This port is also selected as the bridge's preferred path to the root.
  - o Finally, all the bridges connected to a given LAN elect a single *designated* bridge that will be responsible for forwarding frames toward the root bridge.
    - Each LAN's designated bridge is the one that is closest to the root.
    - If two or more bridges are equally close to the root, then the bridges' identifiers are used to break ties, and the smallest ID wins.
    - Each bridge is connected to more than one LAN, so it participates in the election of a designated bridge for each LAN it is connected to.
    - The bridge forwards frames over those ports for which it is the designated bridge.

❖ Figure 2.66 shows the spanning tree that corresponds to the extended LAN shown in Figure 2.64.

- In this example, B1 is the root bridge, since it has the smallest ID. Notice that both B3 and B5 are connected to LAN A, but B5 is the designated bridge since it is closer to the root.

- Similarly, both B5 and B7 are connected to LAN B, but in this case B5 is the designated bridge since it has the smaller ID; both are an equal distance from B1.
- While it is possible for a human to look at the extended LAN given in Figure 2.64 and to compute the spanning tree given in Figure 2.66 according to the rules given above, the bridges in an extended LAN do not have the luxury of being able to see the topology of the entire network, let alone peek inside other bridges to see their ID.
- Specifically, the configuration messages contain three pieces of information:
    1. The ID for the bridge that is sending the message
    2. The ID for what the sending bridge believes to be the root bridge
    3. The distance, measured in hops, from the sending bridge to the root bridge
- Each bridge records the current *best* configuration message it has seen on each of its ports ("best" is defined below), including both messages it has received from other bridges and messages that it has itself transmitted.
    - Initially, each bridge thinks it is the root, and so it sends a configuration message out on each of its ports identifying itself as the root and giving a distance to the root of 0.
    - Upon receiving a configuration message over a particular port, the bridge checks to see if that new message is better than the current best configuration message recorded for that port.
    - The new configuration message is considered *better* than the currently recorded information if any of the following is true:
        - It identifies a root with a smaller ID.
        - It identifies a root with an equal ID but with a shorter distance.
        - The root ID and distance are equal, but the sending bridge has a smaller ID
    - If the new message is better than the currently recorded information, the bridge discards the old information and saves the new information.

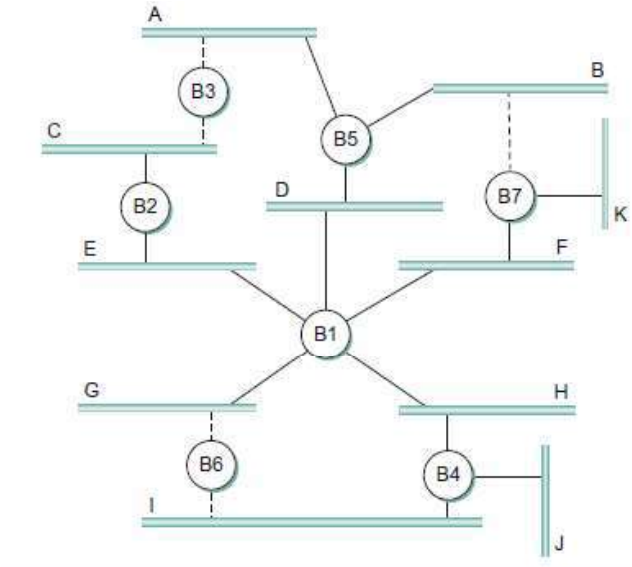


Figure 2.66 Spanning tree with some ports not selected

- In Figure 2.66 if the power had just been restored to the building housing this network, so that all the bridges boot at about the same time. All the bridges would start off by claiming to be the root. We denote a configuration message from node X in which it claims to be distance d from root node Y as (Y, d,X). Focusing on the activity at node B3, a sequence of events would unfold as follows:
  1. B3 receives (B2, 0, B2).
  2. Since 2 < 3, B3 accepts B2 as root.
  3. B3 adds one to the distance advertised by B2 (0) and thus sends (B2, 1, B3) toward B5.
  4. Meanwhile, B2 accepts B1 as root because it has the lower ID, and it sends (B1, 1, B2) toward B3.
  5. B5 accepts B1 as root and sends (B1, 1, B5) toward B3.
  6. B3 accepts B1 as root, and it notes that both B2 and B5 are closer to the root than it is; thus, B3 stops forwarding messages on both its interfaces.
- The goal of a bridge is to transparently extend a LAN across multiple networks, and since most LANs support both broadcast and multicast, then bridges must also support these two features.
  - ✓ Broadcast is simple—each bridge forwards a frame with a destination broadcast address out on each active (selected) port other than the one on which the frame was received.
  - ✓ Multicast can be implemented in exactly the same way, with each host deciding for itself whether or not to accept the message.

Limitations of Bridges

- The bridge-based solution just described is meant to be used in only a fairly limited setting—to connect a handful of similar LANs.
- Scale
  - One reason for this is that the spanning tree algorithm scales linearly; that is, there is no provision for imposing a hierarchy on the extended LAN.
  - A second reason is that bridges forward all broadcast frames. While it is reasonable for all hosts within a limited setting (say, a department) to see each other's broadcast messages, it is unlikely that all the hosts in a larger environment (say, a large company or university) would want to have to be bothered by each other's broadcast messages.
  - Broadcast does not scale, and as a consequence extended LANs do not scale.
  - One approach to increasing the scalability of extended LANs is the *virtual LAN* (VLAN). VLANs allow a single extended LAN to be partitioned into several seemingly separate LANs. Each virtual LAN is assigned an identifier (sometimes called a *color*), and packets can only travel from one segment to another if both segments have the same identifier.
    - Example Figure 2.67 shows four hosts on four different LAN segments. In the absence of VLANs, any broadcast

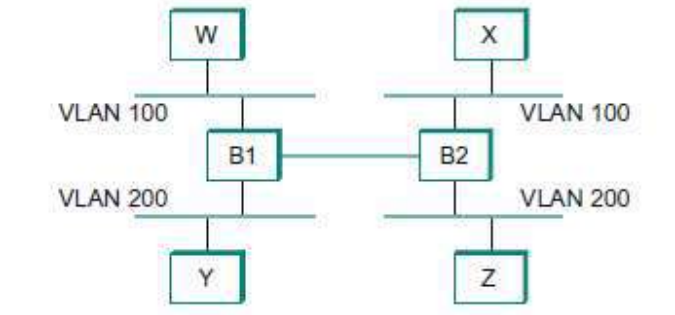


Figure 2.67 Two VLAN share a common backbone

packet from any host will reach all the other hosts.

- The segments that connect to hosts Y and Z as being in VLAN 200.

- To configure this with a VLAN ID on each port of bridges B1 and B2.
- The link between B1 and B2 is considered to be in both VLANs.
- When a packet sent by host X arrives at bridge B2, the bridge observes that it came in a port that was configured as being in VLAN 100.
- It inserts a VLAN header between the Ethernet header and its payload. The interesting part of the VLAN header is the VLAN ID; in this case, that ID is set to 100.
- A broadcast packet—be sent out the interface to host Z, which is in VLAN 200.
- The packet is forwarded on to bridge B1, which follows the same rules and thus may forward the packet to host W but not to host Y.

- Heterogeneity
  - ✓ Bridges make use of the network's frame header and so can support only networks that have exactly the same format for addresses.
  - ✓ Bridges can be used to connect Ethernets to Ethernets, token rings to token rings, and one 802.11 network to another. It's also possible to put a bridge between, say, an Ethernet and an 802.11 network, since both networks support the same 48-bit address format.

Aadvantage

- ✓ Their main is that they allow multiple LANs to be transparently connected; that is, the networks can be connected without the end hosts having to run any additional protocols (or even be aware, for that matter).

2.5. Basic Internetworking

2.5.1. IPv4 ADDRESSES

- An IPv4 address is a 32-bit address that *uniquely* and *universally* defines the connection of a device (for example, a computer or a router) to the Internet.
- If a device operating at the network layer has *m* connections to the Internet, it needs to have *m* addresses.
- IPv4 addresses are universal in the sense that the addressing system must be accepted by any host that wants to be connected to the Internet.
- IPv4 is an unreliable and connectionless datagram protocol-a best-effort delivery service.
- The term *best-effort* means that IPv4 provides no error control or flow control (except for error detection on the header).
- ***Address Space***
  - ✓ An address space is the total number of addresses used by the protocol.
  - ✓ If a protocol uses *N* bits to define an address, the address space is *2N* because each bit can have two different values (0 or 1) and *N* bits can have *2N* values.
  - ✓ The address space of IPv4 is 232 or 4,294,967,296.
    - ***Notations***
  - ✓ There are two prevalent notations to show an IPv4 address: binary notation and dotted decimal notation.
  - ✓ *Binary Notation*

- Each octet is often referred to as a byte.

- An IPv4 address referred to as a 32-bit address or a 4-byte address.
- The following is an example of an IPv4 address in binary notation:

01110101 10010101 00011101 00000010

✓ *Dotted-Decimal Notation*

- In decimal form with a decimal point (dot) separating the bytes.
- The following is the dotted decimal notation of the above address:

117.149.29.2

- Figure 2.68 shows an IPv4 address in both binary and dotted-decimal notation.

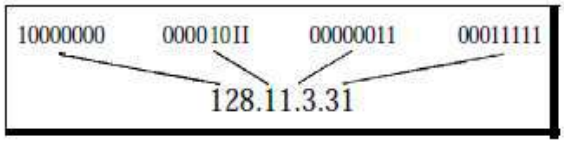


Figure 2.68 *Dotted-decimal notation and binary notation for an IPv4 address*

- Each byte (octet) is 8 bits, each number in dotted-decimal notation is a value ranging from 0 to 255.

➢ ***Classful Addressing***

✓ In classful addressing, the address space is divided into five classes: A, B, C, D and E.

✓ Each class occupies some part of the address space.

✓ If the address is given in binary notation, the first few bits can immediately tell us the class of the address.

✓ If the address is given in decimal-dotted notation, the first byte defines the class. Both methods are shown in Figure 2.69.

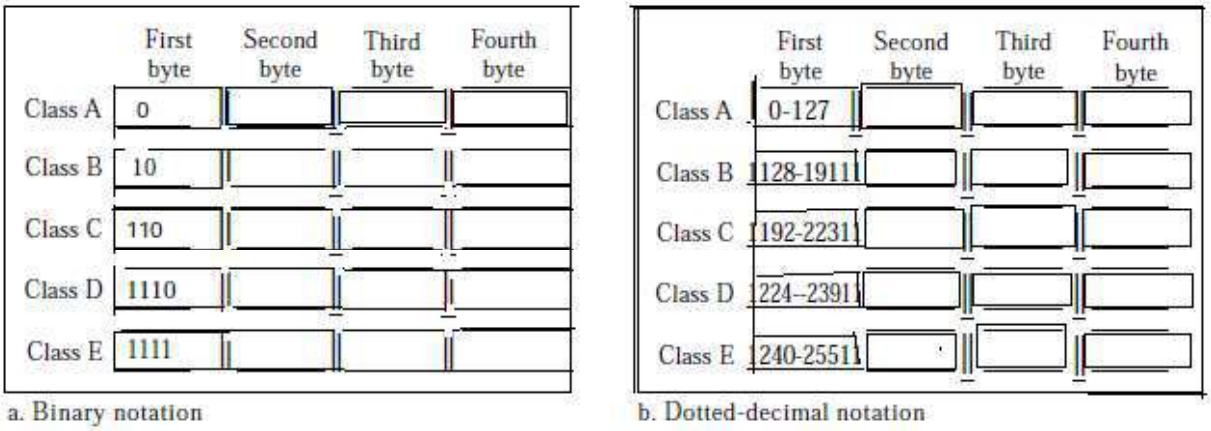


Figure 2.69 *Finding the classes in binary and dotted-decimal notation*

Example

Find the class of each address.

a. 00000001 00001011 00001011 11101111

b. 11000001 10000011 00011011 11111111

c. 14.23.120.8

d. 252.5.15.111

Solution

a. The first bit is O. This is a class A address.

b. The first 2 bits are 1; the third bit is O. This is a class C address.

c. The first byte is 14 (between 0 and 127); the class is A.

d. The first byte is 252 (between 240 and 255); the class is E.

- ***Classes and Blocks***
  - ✓ One problem with classful addressing is that each class is divided into a fixed number of blocks with each block having a fixed size as shown in Table.
  - ✓ Let us examine the table. Previously, when an organization requested a block of addresses, it was granted

Class	*Number ofBlocks*	*Block Size*	*Application*
A	128	16,777,216	Unicast
B	16,384	65,536	Unicast
C	2,097,152	256	Unicast
D	1	268,435,456	Multicast
E	1	268,435,456	Reserved

one in class A, B, or C.

  - ✓ Class A addresses were designed for large organizations with a large number of attached hosts or routers.
  - ✓ Class B addresses were designed for midsize organizations with tens of thousands of attached hosts or routers.
  - ✓ Class C addresses were designed for small organizations with a small number of attached hosts or routers.
  - ✓ A block in class A address is too large. This means most of the addresses in class A were wasted and were not used.
  - ✓ A block in class B is also very large, probably too large for many of the organizations that received a class B block.
  - ✓ A block in class C is probably too small for many organizations. Class D addresses were designed for multicasting. .
  - ✓ The class E addresses were reserved for future use; only a few were used, resulting in another waste of addresses.
- ***Netid and Hostid***
  - ✓ In classful addressing, an IP address in class A, B, or C is divided into netid and hostid.
  - ✓ Figure 2.70 shows some netid and hostid bytes. The netid is in color, the hostid is in white. This concept does not apply to classes D and E.
  - ✓ In class A, one byte defines the netid and three bytes define the hostid.
  - ✓ In class B, two bytes define the netid and two bytes define the hostid.
  - ✓ In class C, three bytes define the netid and one byte defines the hostid.
- ***Mask/ Classless Interdomain Routing (CIDR)***
  - ✓ Length of the netid and hostid (in bits) is predetermined in classful addressing, it can also use a mask , a 32-bit number made of contiguous 1s followed by contiguous 0s.
  - ✓ The masks for classes A, B, and C are shown in Table.

- ✓ The mask can help us to find the netid and the hostid. For example, the mask for a class A address has eight 1s, which means the first 8 bits of any address in class A define the netid; the next 24 bits define the hostid.
- ✓ The last column of Table shows the mask in the form */n* where *n* can be 8, 16, or 24 in classful addressing. This notation is also called *slash notation* or *Classless Interdomain Routing (CIDR) notation*.

Class	*Binary*	*Dotted-Decimal*	*CIDR*
A	11111111 00000000 00000000 00000000	255.0.0.0	*18*
B	11111111 11111111 00000000 00000000	255.255.0.0	*116*
C	11111111 11111111 11111111 00000000	255.255.255.0	*124*

- ***Subnetting***
  - ✓ If an organization was granted a large block in class A or B, it could divide the addresses into several contiguous groups and assign each group to smaller networks (called subnets).
  - ✓ Subnetting increases the number of 1s in the mask. Refer Figure 2.71.
- ***Supernetting***
  - ✓ In supernetting, several networks are combined to create a supernetwork or a supemet.
  - ✓ For example, an organization that needs 1000 addresses can be granted four contiguous class C blocks. Supernetting decreases the number of 1s in the mask.
  - ✓ For example, if an organization is given four class C addresses, the mask changes from /24 to /22.
- ***Address Depletion***
  - ✓ The number of devices on the Internet is much less than the 232 address space.
  - ➢ ***Classless Addressing***
  - ✓ To give more organizations access to the Internet, classless addressing was designed and implemented.
  - ✓ In this scheme, there are no classes, but the addresses are still granted in blocks.
- ***Address Blocks***
  - ✓ In classless addressing, when an entity, small or large, needs to be connected to the Internet, it is granted a block (range) of addresses.
  - ✓ The size of the block (the number of addresses) varies based on the nature and size of the entity.
  - ✓ For example, a household may be given only two addresses; a large organization may be given thousands of addresses.
  - ✓ An ISP, as the Internet service provider, may be given thousands or hundreds of thousands based on the number of customers it may serve.

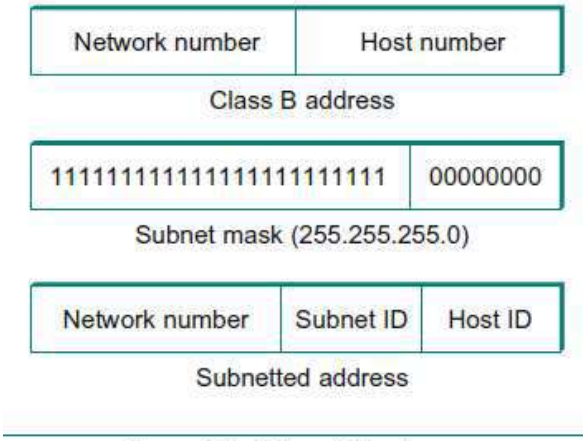


Figure 2.71. Subnet Addressing

- ✓ *Restriction* : To simplify the handling of addresses, the Internet authorities impose three restrictions on classless address blocks:
  1. The addresses in a block must be contiguous, one after another.
  2. The number of addresses in a block must be a power of 2 (I, 2, 4, 8, ...).
  3. The first address must be evenly divisible by the number of addresses.
- ✓ ***Example***

- Figure 2.70 shows a block of addresses, in both binary and dotted-decimal notation, granted to a small business that needs 16 addresses.

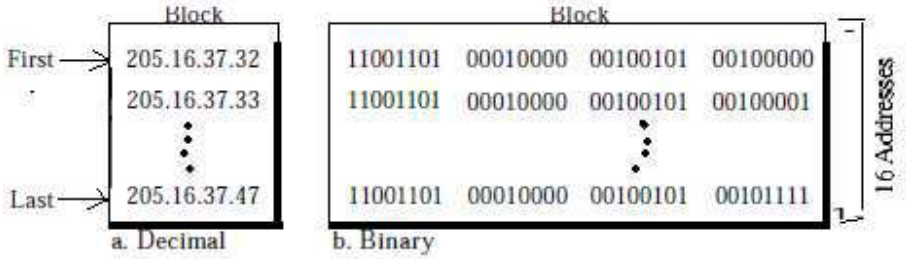


Figure 2.70 *A block of 16 addresses granted to a small organization*

- The addresses are contiguous. The number of addresses is a power of 2 ($16 = 2^4$), and the first address is divisible by 16. The first address, when converted to a decimal number, is 3,440,387,360, which when divided by 16 results in 215,024,210. In Appendix B, we show how to find the decimal value of an IP address.

- *Mask*
  - ✓ To define a block of addresses is to select any address in the block and the mask.
  - ✓ A mask is a 32-bit number in which the *n* leftmost bits are 1s and the 32-*n* rightmost bits are 0s.
  - ✓ A block can take any value from 0 to 32.
  - ✓ In 1Pv4 addressing, a block of addresses can be defined as x.y.z.t/n in which x.y.z.t defines one of the addresses and the */n* defines the mask. The address and the */n* notation completely define the whole block (the first address, the last address, and the number of addresses).
  - ✓ *First Address:* The first address in the block can be found by setting the 32 - *n* rightmost bits in the binary notation of the address to 0s.
  - ✓ The last address in the block can be found by setting the rightmost 32 - *n* bits to 1s.
  - ✓ *Number of Addresses:* The number of addresses in the block is the difference between the last and first address. It can easily be found using the formula 2^{32}- *n.*

  Example

  Find the number of addresses for 205.16.37.39/28

  Solution

  The value of *n* is 28, which means that number of addresses is 2^{32-28} or 16.

  Example for Subnetting

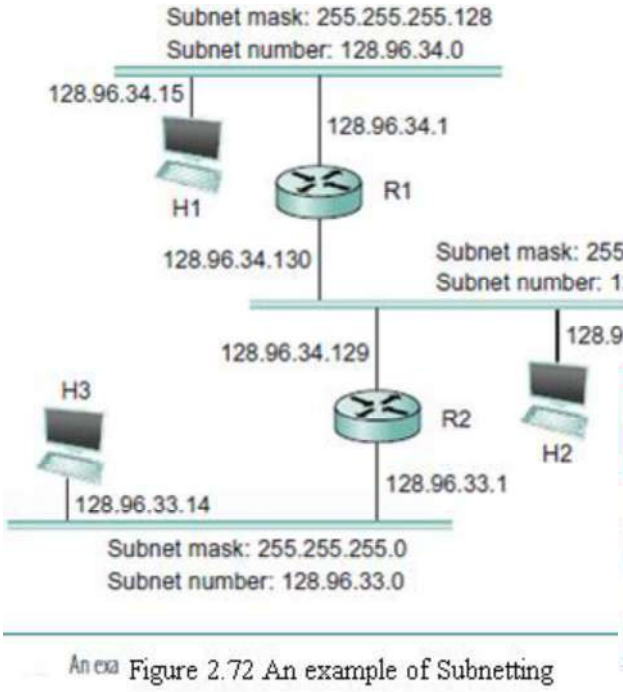


Figure 2.72 An example of Subnetting

Table Example Forwarding Table with Subnetting for Figure 2.72

SubnetNumber	SubnetMask	NextHop
128.96.34.0	255.255.255.128	Interface 0
128.96.34.128	255.255.255.128	Interface 1
128.96.33.0	255.255.255.0	R2

- ✓ In the example network of Figure 2.72, router R1 would have the entries shown in Table. Continuing with the example of a datagram from H1 being sent to H2,R1 would AND H2's address (128.96.34.139) with the subnet mask of the first entry (255.255.255.128) and compare the result (128.96.34.128) with the network number for that entry (128.96.34.0).
- ✓ Since this is not a match, it proceeds to the next entry. This time a match does occur, so R1 delivers the datagram to H2 using interface 1, which is the interface connected to the same network as H2.
- ✓ Describe the datagram forwarding algorithm in the following way:

```
D = destination IP address
   for each forwarding table entry hSubnetNumber, SubnetMask, NextHop
         D1 = SubnetMask & D
if D1 = SubnetNumber
   if NextHop is an interface
         deliver datagram directly to destination
   else
         deliver datagram to NextHop (a router)
```

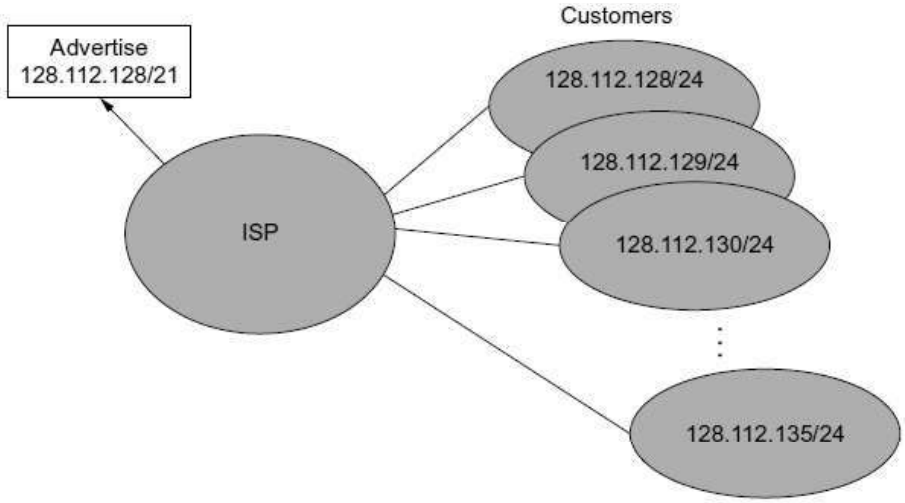


Figure 2.73 Route Aggregation with CIDR

➢ ***Datagram Format***

✓ Packets in the IPv4 layer are called datagrams. Figure 2.74 shows the IPv4 datagram format.

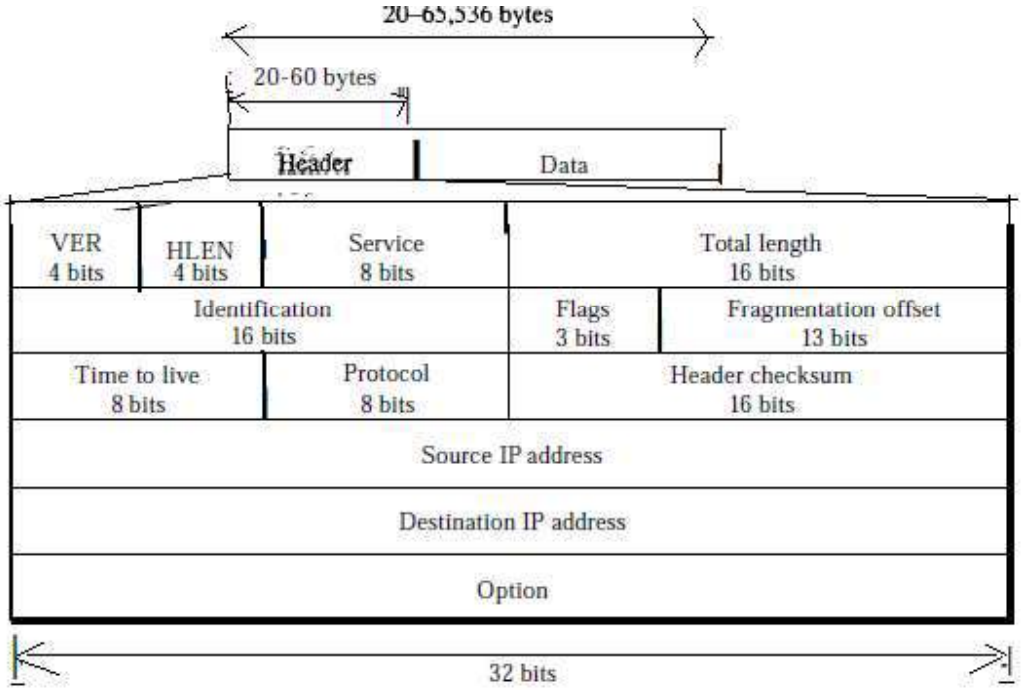


Figure 2.74 IPv4 Datagram Format

✓ A datagram is a variable-length packet consisting of two parts: *header and data*.

i. *Header:* It is 20 to 60 bytes in length and contains information essential to routing and delivery. A brief description of each field is in order.

- *Version (VER).* This 4-bit field defines the version of the IPv4 protocol.
  - Currently the version is 4. However, version 6 (or IPng) may totally replace version 4 in the future.

- This field tells the IPv4 software running in the processing machine that the datagram has the format of version 4.
- All fields must be interpreted as specified in the fourth version of the protocol.

- *Header length (HLEN).*
  - This 4-bit field defines the total length of the datagram header in 4-byte words.
  - This field is needed because the length of the header is variable (between 20 and 60 bytes).
  - When there are no options, the header length is 20 bytes, and the value of this field is 5 (5 x 4 = 20).
  - When the option field is at its maximum size, the value of this field is 15 (15 x 4 = 60).
- *Services.*
  - IETF has changed the interpretation and name of this 8-bit field. This field, previously called service type, is now called differentiated services.

1. *Service Type :* In this interpretation, the first 3 bits are called precedence bits. The next 4 bits are called type of service (TOS) bits, and the last bit is not used.

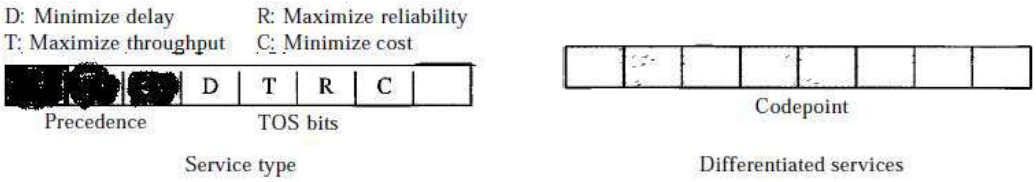


Figure 2.75 *Service type or differentiated services*

   a. Precedence is a 3-bit subfield ranging from 0 (000 in binary) to 7 (111 in binary).The precedence defines the priority of the datagram in issues such as congestion. If a router is congested and needs to discard some datagrams, those datagrams with lowest precedence are discarded first..
   b. TOS bits is a 4-bit subfield with each bit having a special meaning. Although a bit can be either 0 or 1, one and only one of the bits can have the value of 1 in each datagram. The bit patterns and their interpretations are given in the following Table. With only 1 bit set at a time, we can have five different types of services.

pplication programs can request a specific type of service. The defaults for some applications are shown in above Table.

2. *Differentiated Services :*In this interpretation, the first 6 bits make up the codepoint subfield, and the last 2 bits are not used. The codepoint subfield can be used in two different ways.
   a. When the 3 rightmost bits are 0s, the 3 leftmost bits are interpreted the same as the precedence bits in the service type interpretation. In other words, it is compatible with the old interpretation.

When the 3 rightmost bits are not all Os, the 6 bits define 64 services based on the priority assignment by the Internet or local authorities according to the below Table The first category contains 32 service types; the second and the third each contain 16. The first category (numbers 0, 2,4, ... ,62) is assigned by the Internet authorities (IETF). The second category (3, 7, 11, 15, , 63) can be used by local authorities (organizations). The third category (1, 5, 9, ,61) is temporary and can be used for experimental purposes.

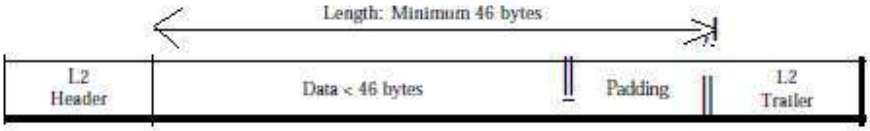


Figure 2.76 Encapsulation of a small datagram in an Ethernet frame

Default types of service

Protocol	*TOS Bits*	*Description*
ICMP	0000	Normal
BOOTP	0000	Normal
NNTP	0001	Minimize cost
IGP	0010	Maximize reliability
SNMP	0010	Maximize reliability
TELNET	1000	Minimize delay
FTP (data)	0100	Maximize throughput
FTP (control)	1000	Minimize delay
TFTP	1000	Minimize delay
SMTP (command)	1000	Minimize delay
SMTP (data)	0100	Maximize throughput
DNS (UDP query)	1000	Minimize delay
DNS (TCP query)	0000	Normal
DNS (zone)	0100	Maximize throughput

Types of service

TOS Bits	*Description*
0000	Normal (default)
0001	Minimize cost
0010	Maximize reliability
0100	Maximize throughput
1000	Minimize delay

Values for codepoints

Category	*Codepoint*	*Assigning Authority*
1	XXXXXO	Internet
2	XXXX11	Local
3	XXXXOI	Temporary or experimental

- *Total length.* To find the length of the data coming from the upper layer, subtract the header length from the total length. The header length can be found by multiplying the value in the HLEN field by 4.

Length of data =total length - header length

Since the field length is 16 bits, the total length of the IPv4 datagram is limited to 65,535 (2^{16} - 1) bytes, of which 20 to 60 bytes are the header and the rest is data from the upper layer.

- *Identification.* This field is used in fragmentation.
- *Flags.* This field is used in fragmentation.
- *Fragmentation offset.* This field is used in fragmentation.
- *Time to live.* A datagram has a limited lifetime in its travel through an internet. This field was originally designed to hold a timestamp, which was decremented by each visited router. The datagram was discarded when the value became zero.
- *Protocol.* This 8-bit field defines the higher-level protocol that uses the services of the IPv4 layer. An IPv4 datagram can encapsulate data from several higher-level protocols such as TCP, UDP, ICMP, and IGMP. This field specifies the final destination protocol to which the IPv4 datagram is delivered.

  The value of this field for each higher-level protocol is shown in Table.

Protocol values

Value	*Protocol*
1	ICMP
2	IGMP
6	TCP
17	UDP
89	OSPF

- *Checksum.* Error detect and correct
- *Source address*. This 32-bit field defines the IPv4 address of the source. This field must remain unchanged during the time the IPv4 datagram travels from the source host to the destination host.
- Destination address. This 32-bit field defines the IPv4 address of the destination. This field must remain unchanged during the time the IPv4 datagram travels from the source host to the destination host.
- **Fragmentation**
  - The format and size of the sent frame depend on the protocol used by the physical network through which the frame is going to travel.
  - For example, if a router connects a LAN to a WAN, it receives a frame in the LAN format and sends a frame in the WAN format.

- ***Maximum Transfer Unit (MTU)***

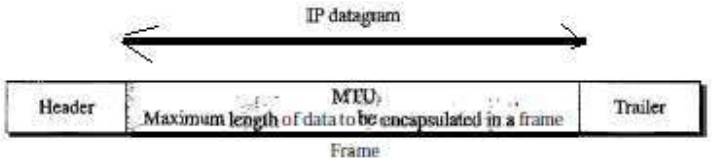


Figure 2.77 *Maximum transfer unit (MTU)*

  - Each data link layer protocol has its own frame format in most protocols. One of the fields defined in the format is the maximum size of the data field. The value of the MTU depends on the physical network protocol. The following Table shows the values for some protocols.

- ***Fields Related to Fragmentation***

  The fields that are related to fragmentation and reassembly of an IPv4 datagram are the identification, flags, and fragmentation offset fields.

  - *Identification.* This 16-bit field identifies a datagram originating from the source host. The combination of the identification and source IPv4 address must uniquely define a datagram as it leaves the source host. When the IPv4 protocol sends a datagram, it copies the current value of the counter to the identification field and increments the counter by 1. When a datagram is fragmented, the value in the identification field is copied to all fragments.

MTUs for some networks

Protocol	*MTU*
Hyperchannel	65,535
Token Ring (16 Mbps)	17,914
Token Ring (4 Mbps)	4,464
FDDI	4,352
Ethernet	1,500
X.25	576
PPP	296

- *Flags.* This is a 3-bit field. The first bit is reserved. The second bit is called the *do notfragment* bit. If its value is 1, the machine must not fragment the datagram. If it cannot pass the datagram through any available physical network, it discards the datagram and sends an ICMP error message to the source host. If its value is 0, the datagram can be fragmented if necessary. The third bit is called the *more fragment* bit. If its value is 1, it means the datagram is not the last fragment; there are more fragments after this one. If its value is 0, it means this is the last or only fragment.

Figure 2.78 *Flags used in fragmentation*

- *Fragmentation offset*. This 13-bit field shows the relative position of this fragment with respect to the whole datagram. It is the offset of the data in the original datagram measured in units of 8 bytes. Figure 2.79 shows a datagram with a data size of 4000 bytes fragmented into three fragments. The bytes in the original datagram are numbered 0 to 3999. The first fragment carries bytes 0 to 1399. The offset for this datagram is 0/8 =0. The second fragment carries bytes 1400 to 2799; the offset value for this fragment is 1400/8 = 175. Finally, the third fragment carries bytes 2800 to 3999. The offset value for this fragment is 2800/8 =350.

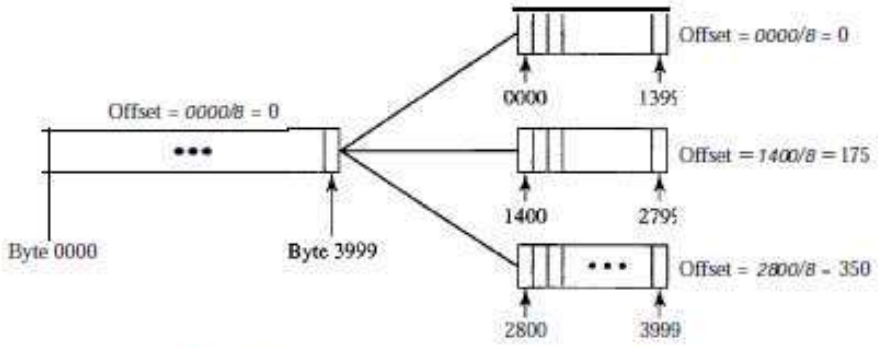


Figure 2.79 *Fragmentation example*

- Figure 2.80 shows an expanded view of the fragments in Figure 2.79.

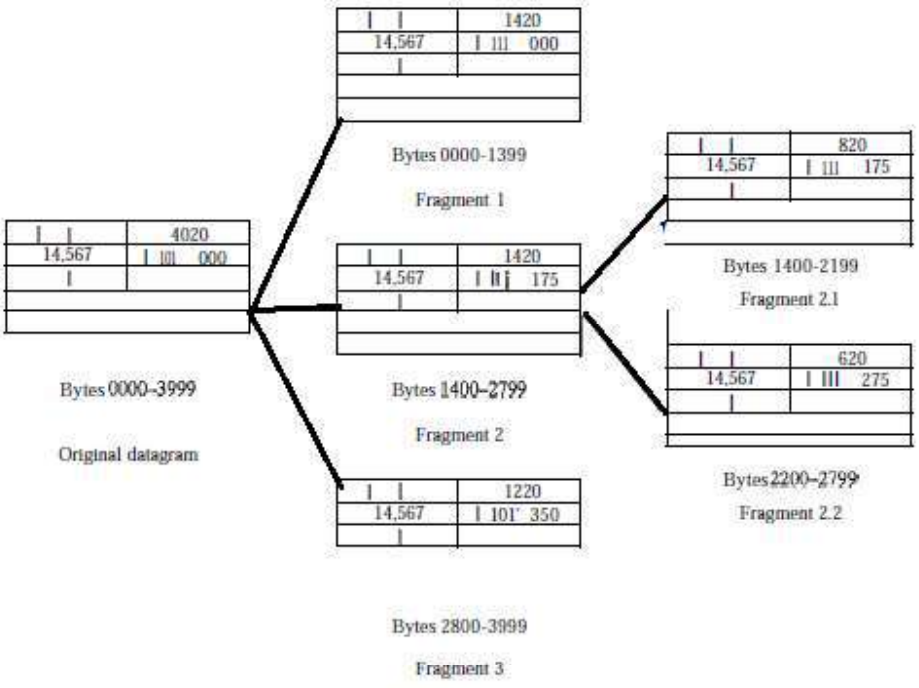


Figure 2.80 *Detailedfragmentation example*

- Fragmentation and Reassembly Strategy:

1. The first fragment has an offset field value of zero.
2. Divide the length of the first fragment by 8. The second fragment has an offset value equal to that result.
3. Divide the total length of the first and second fragments by 8. The third fragment has an offset value equal to that result.
4. Continue the process. The last fragment has a *more* bit value of O.

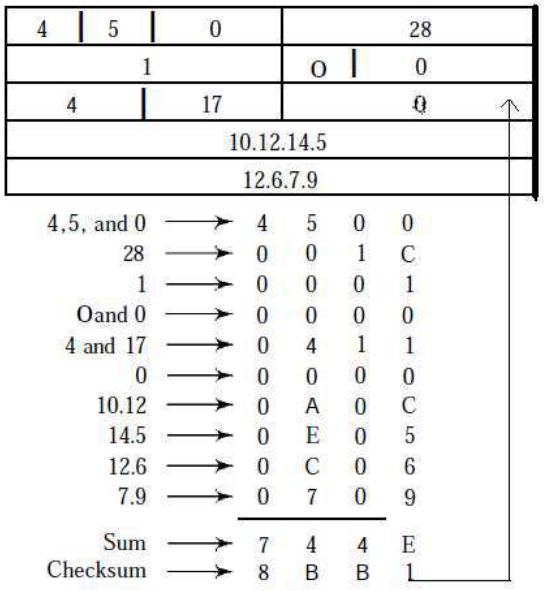


Figure 2.81 *Example ofchecksum calculation in IPv4*

- **Checksum :** Discussed in previous topic
- **Options**
  - The header of the IPv4 datagram is made of two parts: *a fixed part and a variable part*. The fixed part is 20 bytes long

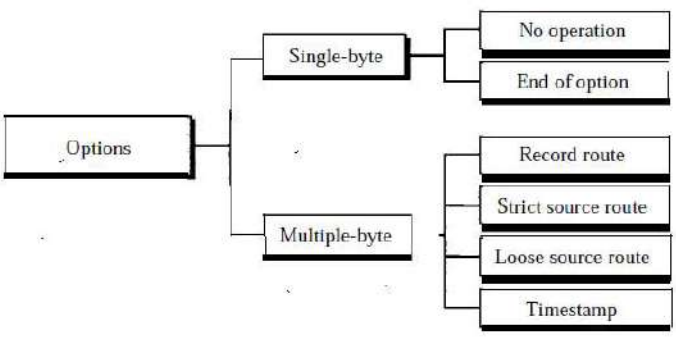


Figure 2.82 *options in IPv4*

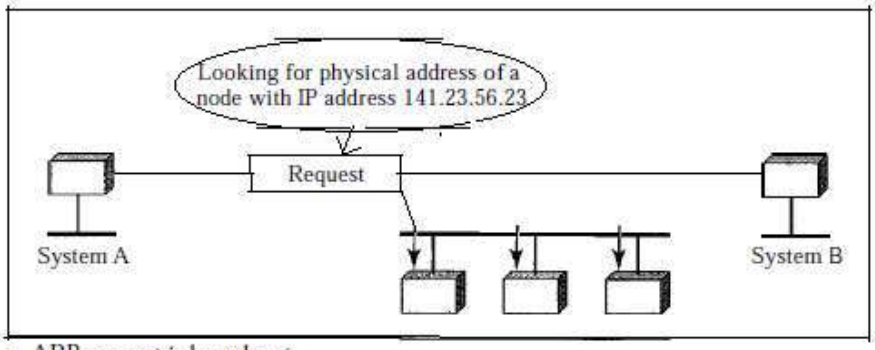
- The variable part comprises the options that can be a maximum of 40 bytes.
- *No Operation:* A *no-operation option* is a 1-byte option used as a filler between options.
- *End of Option:* An end-of-option option is a I-byte option used for padding at the end of the option field.
- *Record Route :* A record route option is used to record the Internet routers that handle the datagram.It can list up to nine router addresses. It can be used for debugging and management purposes.
- *Strict Source Route:* A strict source route option is used by the source to predetermine a route for the datagram as it travels through the Internet. The sender can choose a route with a specific type of service, such as minimum delay or maximum throughput. Alternatively, it may choose a route that is safer or more reliable for the sender's purpose. For example, a sender can choose a route so that its datagram does not travel through a competitor's network.

  If a datagram specifies a strict source route, all the routers defined in the option must be visited by the datagram. A router must not be visited if its IPv4 address is not listed in the datagram. If the datagram visits a router that is not on the list, the datagram is discarded and an error message is issued. If the datagram arrives at the destination and some of the entries were not visited, it will also be discarded and an error message issued.
- *Loose Source Route :* A loose source route option is similar to the strict source route, but it is less rigid. Each router in the list must be visited, but the datagram can visit other routers as well.
- *Timestamp :* A timestamp option is used to record the time of datagram processing by a router. The time is expressed in milliseconds from midnight, Universal time or Greenwich mean time.

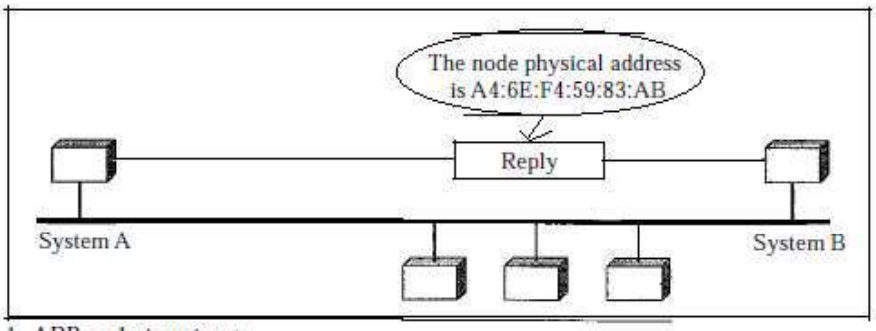
2.5.2. Mapping Logical to Physical Address: ARP

- The logical (IP) address is obtained from the DNS if the sender is the host or it is found in a routing table if the sender is a router. But the IP datagram must be encapsulated in a frame to be able to pass through the physical network. This means that the sender needs the physical address of the receiver.
- The host or the router sends an ARP query packet. The packet includes the physical and IP addresses of the sender and the IP address of the receiver. Because the sender does not know the physical address of the receiver, the query is broadcast over the network (see Figure 2.83).
- In Figure 3.43a, the system on the left (A) has a packet that needs to be delivered to another system (B) with IP address 141.23.56.23. System A needs to pass the packet to its data link layer for the actual delivery, but it does not know the physical address of the recipient. It uses the services of ARP by asking the ARP protocol to send a broadcast ARP request packet to ask for the physical address of a system with an IP address of 141.23.56.23.
- This packet is received by every system on the physical network, but only system B will answer it, as shown in Figure 3.43 b. System B sends an ARP reply packet that includes its physical address. Now system A can send all the packets it has for this destination by using the physical address it received.
- ***Cache Memory***
  - ✓ Using ARP is inefficient if system A needs to broadcast an ARP request for each IP packet it needs to send to system B. It could have broadcast the IP packet itself.
  - ✓ ARP can be useful if the ARP reply is cached (kept in cache memory for a while) because a system normally sends several packets to the same destination.

- ✓ A system that receives an ARP reply stores the mapping in the cache memory and keeps it for 20 to 30 minutes unless the space in the cache is exhausted.
- ✓ Before sending an ARP request, the system first checks its cache to see if it can find the mapping.



a. ARP request is broadcast



b. ARP reply is unicast

Figure 2.83 *ARP operation*

➢ ***Packet Format***

- ✓ The fields are as follows:
  - o *Hardware type.* This is a 16-bit field defining the type of the network on which ARP is running. Each LAN has been assigned an integer based on its type. For example, Ethernet is given type 1. ARP can be used on any physical network.
  - o *Protocol type*. This is a 16-bit field defining the protocol. For example, the value of this field for the IPv4 protocol is 080016, ARP can be used with any higher-level protocol.
  - o *Hardware length.* This is an 8-bit field defining the length of the physical address in bytes. For example, for Ethernet the value is 6.

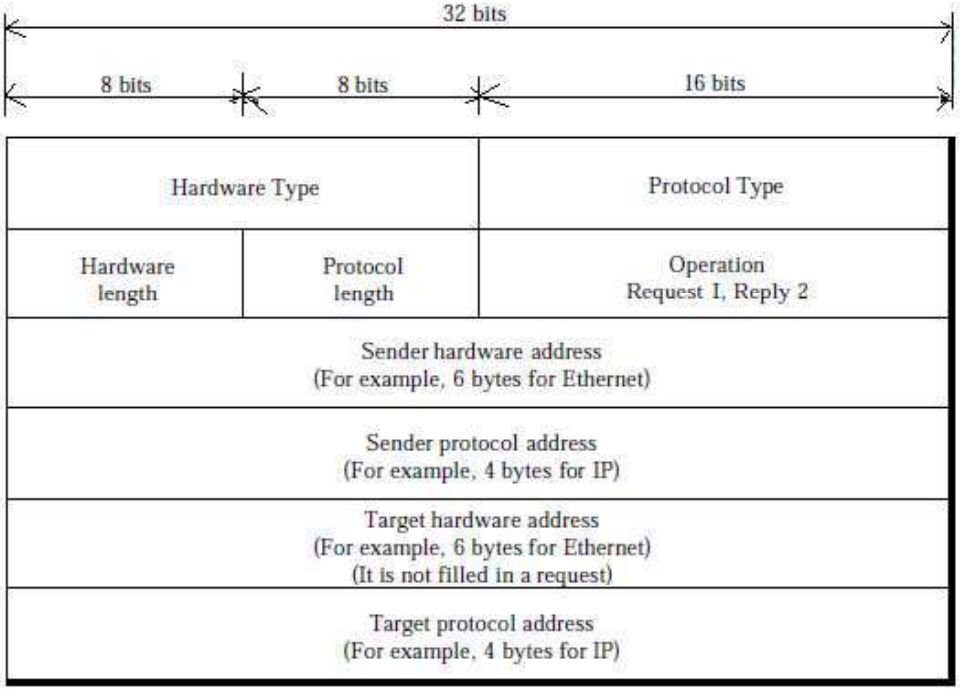


Figure 2.84 *ARP packet*

- *Protocol length.* This is an 8-bit field defining the length of the logical address in bytes. For example, for the IPv4 protocol the value is 4.
- *Operation.* This is a 16-bit field defining the type of packet. Two packet types are defined: ARP request (1) andARP reply (2).
- *Sender hardware address*. This is a variable-length field defining the physical address of the sender. For example, for Ethernet this field is 6 bytes long.
- *Sender protocol address*. This is a variable-length field defining the logical (for example, IP) address of the sender. For the IP protocol, this field is 4 bytes long.
- *Target hardware address*. This is a variable-length field defining the physical address of the target. For example, for Ethernet this field is 6 bytes long. For an ARP request message, this field is all 0s because the sender does not know the physical address of the target.
- *Target protocol address*. This is a variable-length field defining the logical (for example, IP) address of the target. For the IPv4 protocol, this field is 4 bytes long.

✓ ***Encapsulation***

- An ARP packet is encapsulated directly into a data link frame. For example, in Figure 2.85 an ARP packet is encapsulated in an Ethernet frame.
- The type field indicates that the data carried by the frame are an ARP packet.

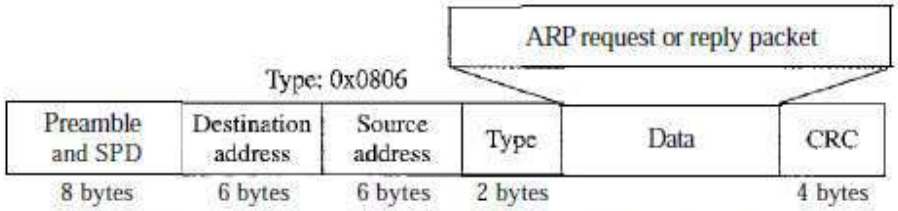


Figure 2.85 *Encapsulation ofARP packet*

- ***Operation***

The steps involved in an ARP process:

1. The sender knows the IP address of the target. We will see how the sender obtains this shortly.
2. IP asks ARP to create an ARP request message, filling in the sender physical address, the sender IP address, and the target IP address. The target physical address field is filled with 0s.
3. The message is passed to the data link layer where it is encapsulated in a frame by using the physical address of the sender as the source address and the physical broadcast address as the destination address.
4. Every host or router receives the frame. Because the frame contains a broadcast destination address, all stations remove the message and pass it to ARP. All machines except the one targeted drop the packet. The target machine recognizes its IP address.
5. The target machine replies with an ARP reply message that contains its physical address. The message is unicast.
6. The sender receives the reply message. It now knows the physical address of the target machine.
7. The IP datagram, which carries data for the target machine, is now encapsulated in a frame and is unicast to the destination.

- ***Four Different Cases***

The following are four different cases in which the services of ARP can be used.

1. The sender is a host and wants to send a packet to another host on the same network. In this case, the logical address that must be mapped to a physical address is the destination IP address in the datagram header.
2. The sender is a host and wants to send a packet to another host on another network. In this case, the host looks at its routing table and finds the IP address of the next hop (router) for this destination. If it does not have a routing table, it looks for the IP address of the default router. The IP address of the router becomes the logical address that must be mapped to a physical address.
3. The sender is a router that has received a datagram destined for a host on another network. It checks its routing table and finds the IP address of the next router. The IP address of the next router becomes the logical address that must be mapped to a physical address.
4. The sender is a router that has received a datagram destined for a host on the same network. The destination IP address of the datagram becomes the logical address that must be mapped to a physical address.

An ARP request is broadcast; an ARP reply is unicast.

Example

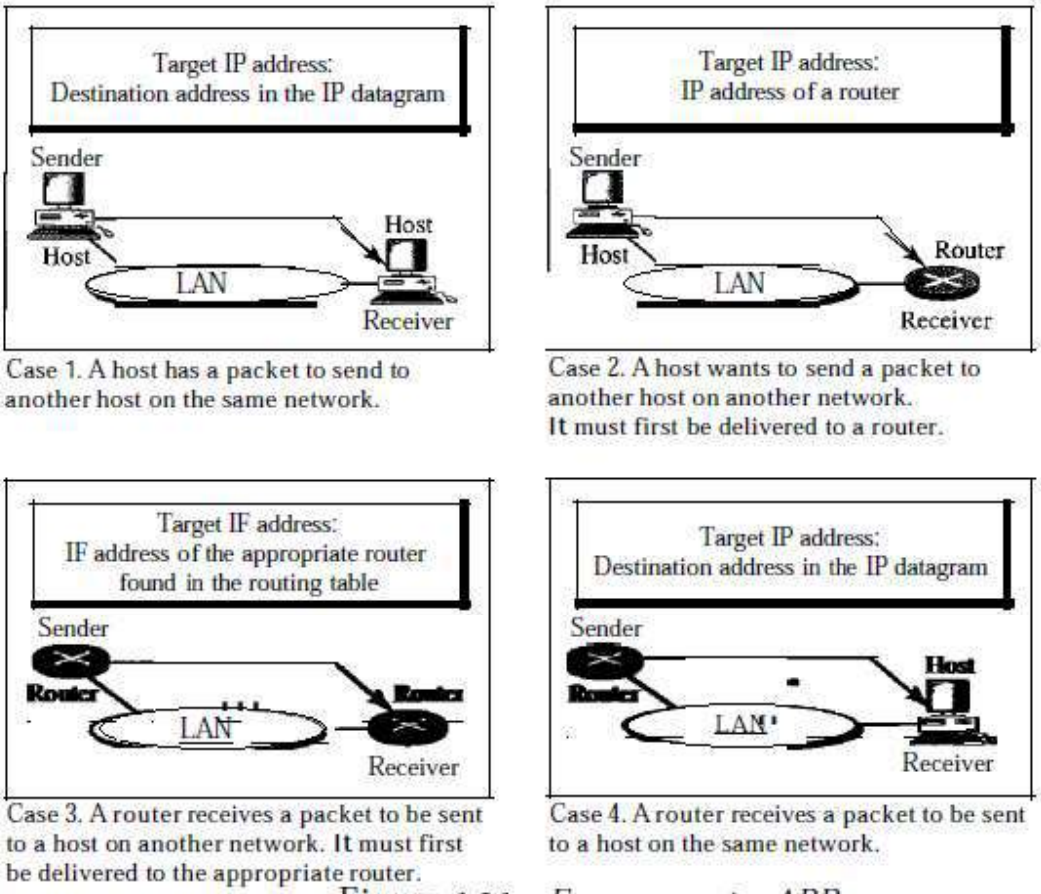
A host with IP address 130.23.43.20 and physical address B2:34:55: 10:22: 10 has a packet to send to another host with IP address 130.23.43.25 and physical address A4:6E:F4:59:83:AB (which is unknown to the first host). The two hosts are on the same Ethernet network. Show the ARP request and reply packets encapsulated in Ethernet frames.

Solution

Figure 3.47 shows the ARP request and reply packets. Note that the ARP data field in this case is 28 bytes, and that the individual addresses do not fit in the 4-byte boundary. That is why we do not show the regular 4-byte boundaries for these addresses.

- ***ProxyARP***

- A technique called *proxy* ARP is used to create a subnetting effect. A proxy ARP is an ARP that acts on behalf of a set of hosts.



Case 1. A host has a packet to send to another host on the same network.

Case 2. A host wants to send a packet to another host on another network. It must first be delivered to a router.

Case 3. A router receives a packet to be sent to a host on another network. It must first be delivered to the appropriate router.

Case 4. A router receives a packet to be sent to a host on the same network.

Figure 2.86 *Four cases using ARP*

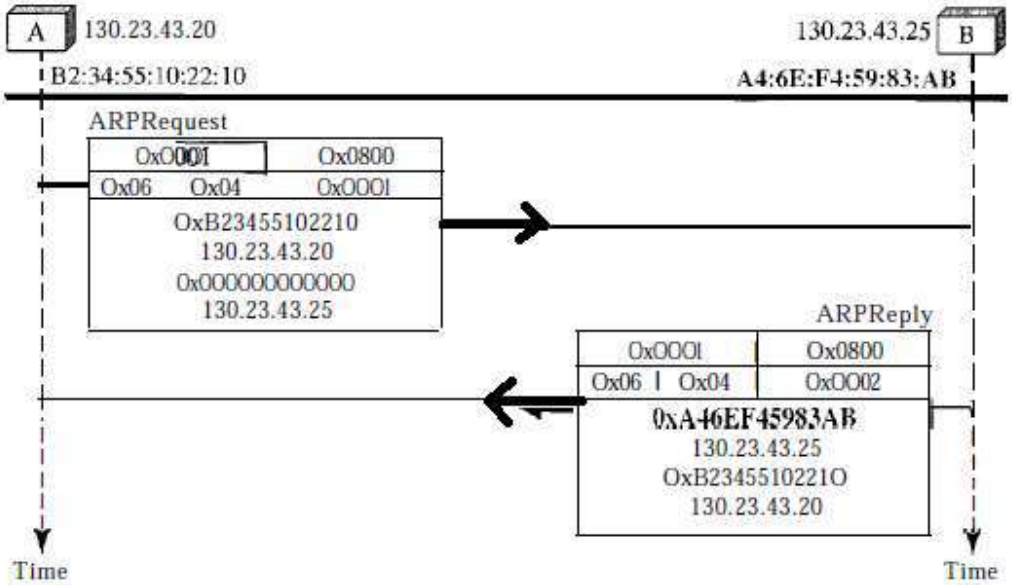


Figure 2.87 *anARP request and reply*

- Whenever a router running a proxy ARP receives an ARP request looking for the IP address of one of these hosts, the router sends an ARP reply announcing its own hardware (physical) address. After the router receives the actual IP packet, it sends the packet to the appropriate host or router.
- Let us give an example. In Figure 2.88 the ARP installed on the right-hand host will answer only to an ARP request with a target IP address of 141.23.56.23.
- However, the administrator may need to create a subnet without changing the whole system to recognize subnetted addresses. One solution is to add a router running a proxy ARP. In this case, the router acts on behalf of all the hosts installed on the subnet. When it receives an ARP request with a target IP address that matches the address of one of its proteges (141.23.56.21, 141.23.56.22, or 141.23.56.23), it sends an ARP reply and announces its hardware address as the target hardware address. When the router receives the IP packet, it sends the packet to the appropriate host.

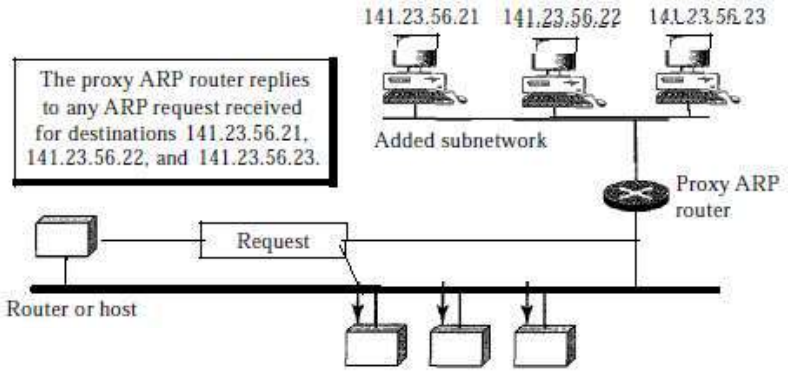


Figure 2.88 *Proxy ARP*

2.5.3. Mapping Physical to Logical Address: RARP

- A host knows its physical address, but needs to know its logical address. This may happen in two cases:

1. A diskless station is just booted. The station can find its physical address by checking its interface, but it does not know its IP address.
2. An organization does not have enough IP addresses to assign to each station; it needs to assign IP addresses on demand. The station can send its physical address and ask for a short time lease.

RARP

- Reverse Address Resolution Protocol (RARP) finds the logical address for a machine that knows only its physical address. Each host or router is assigned one or more logical (IP) addresses, which are unique and independent of the physical (hardware) address of the machine.
- To create an IP datagram, a host or a router needs to know its own IP address or addresses. The IP address of a machine is usually read from its configuration file stored on a disk file.
- The machine can get its physical address (by reading its NIC, for example), which is unique locally. It can then use the physical address to get the logical address by using the RARP protocol.
- A RARP request is created and broadcast on the local network. Another machine on the local network that knows all the IP addresses will respond with a RARP reply. The requesting machine must be running a RARP client program; the responding machine must be running a RARP server program.
- There is a serious *problem with RARP*:

- Broadcasting is done at the data link layer.
- The physical broadcast address, all 1s in the case of Ethernet, does not pass the boundaries of a network. This means that if an administrator has several networks or several subnets, it needs to assign a RARP server for each network or subnet.

2.4.4. DHCP

- Goal: DHCP is to minimize the amount of manual configuration required for a host to function; it would rather defeat the purpose if each host had to be configured with the address of a DHCP server.
- The *Dynamic Host Configuration Protocol (DHCP)* provides *static* and *dynamic address allocation* that can be *manual* or *automatic*.
  - *Static Address Allocation*
    - It is backward compatible with BOOTP, which means a host running the BOOTP client can request a static address from a DHCP server.
    - A DHCP server has a database that statically binds physical addresses to IP addresses.
  - *Dynamic Address Allocation*
    - When a DHCP client requests a temporary IP address, the DHCP server goes to the pool of available (unused) IP addresses and assigns an IP address for a negotiable period of time.
    - When a DHCP client sends a request to a DHCP server, the server first checks its static database. If an entry with the requested physical address exists in the static database, the permanent IP address of the client is returned.
    - The dynamic aspect of DHCP is needed when a host moves from network to network or is connected and disconnected from a network (as is a subscriber to a service provider).
    - DHCP provides temporary IP addresses for a limited time.
    - The addresses assigned from the pool are temporary addresses.
    - The DHCP server issues a lease for a specific time. When the lease expires, the client must either stop using the IP address or renew the lease.
    - The server has the option to agree or disagree with the renewal.
      - If the server disagrees, the client stops using the address.
  - *Manual and Automatic Configuration*
    - One major problem with the BOOTP protocol is that the table mapping the IP addresses to physical addresses needs to be manually configured. This means that every time there is a change in a physical or IP address, the administrator needs to manually enter the changes.
    - DHCP, allows both manual and automatic configurations.
    - Static addresses are created manually dynamic addresses are created automatically.

- ***Operations***
  - ✓ To contact a DHCP server, a newly booted or attached host sends a DHCPDISCOVER message to a special IP address (255.255.255.255) that is an IP broadcast address. This means it will be received by all hosts and routers on that network.
  - ✓ The server would then reply to the host that generated the discovery message
  - ✓ DHCP uses the concept of a *relay agent*.

- There is at least one relay agent on each network, and it is configured with just one piece of information: the IP address of the DHCP server.
- When a relay agent receives a DHCPDISCOVER message, it uncast it to the DHCP server and awaits the response, which it will then send back to the requesting client. The process of relaying a message from a host to a remote DHCP server is shown in Figure 2.89.

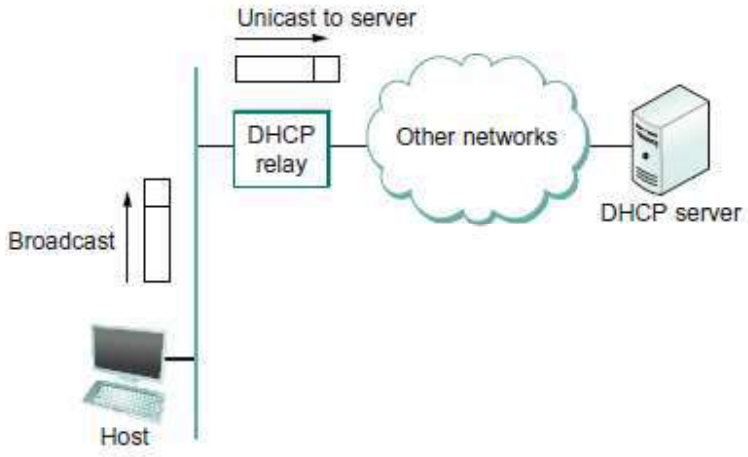


Figure 2.89 A DHCP relay agent receives a broadcast DHCPDISCOVER message from a host and sends a unicast DHCPDISCOVER to the DHCP server

- ***Format of DHCP message***.
  - ✓ The message is actually sent using a protocol called the *User Datagram Protocol* (UDP) that runs over IP.
  - ✓ This context is to provide a demultiplexing key that says, "This is a DHCP packet."
  - ✓ DHCP is derived from an earlier protocol called BOOTP, and some of the packet fields are thus not strictly relevant to host configuration.
  - ✓ When trying to obtain configuration information, the client puts its hardware address (e.g., its Ethernet address) in the chaddr field.
  - ✓ The DHCP server replies by filling in the yiaddr ("your" IP address) field and sending it to the client.
  - ✓ Other information such as the default router to be used by this client can be included in the options field.
  - ✓ In the case where DHCP dynamically assigns IP addresses to hosts, it is clear that hosts cannot keep addresses indefinitely, as this would eventually cause the server to exhaust its address pool.
  - ✓ At the same time, a host cannot be depended upon to give back its address,

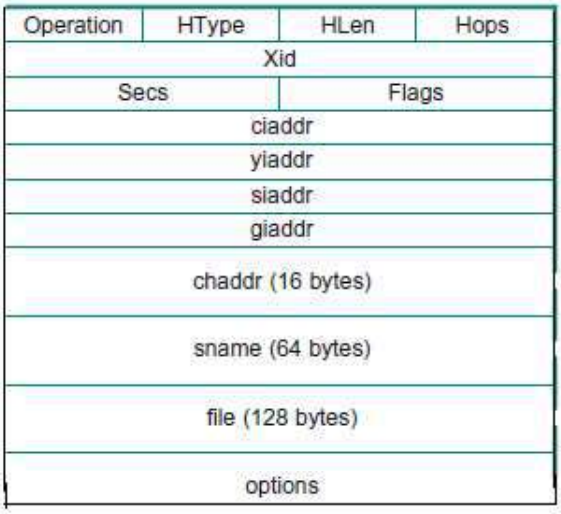


Figure 2.90 DHCP Packet format

since it might have crashed, been unplugged from the network, or been turned off.

2.4.5. ICMP

Need for ICMP : A host sometimes needs to determine if a router or another host is alive. And sometimes a network administrator needs information from another host or router. It is a companion to the IP protoco1.

Types of Messages

- ✓ ICMP messages are divided into two broad categories:

1. *error-reporting messages:* report problems that a router or a host (destination) may encounter when it processes an IP packet.
2. *query messages.* which occur in pairs, help a host or a network manager get specific information from a router or another host. For example, nodes can discover their neighbors. Also, hosts can discover and learn about routers on their network, and routers can help a node redirect its messages.

Message Format

- ✓ An ICMP message has an 8-byte header and a variable-size data section.
  - ➢ *ICMP type*: defines the type of the message.
  - ➢ The *code* field specifies the reason for the particular message type.
  - ➢ *header* is specific for each message type.
  - ➢ The *data section* in error messages carries information for finding the original packet that had the error. In query messages, the data section carries extra information based on the type of the query.

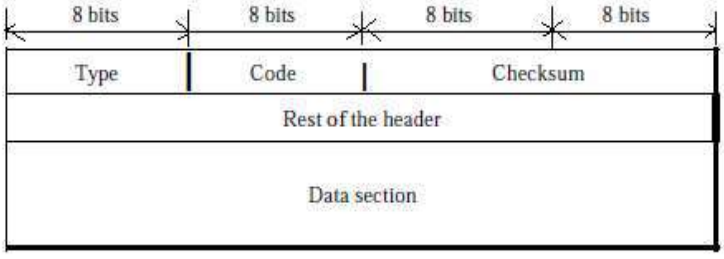


Figure 2.91 *Generalformat of[CM]? messages*

Error Reporting

- One of the main responsibilities of ICMP is to report errors.
- Error messages are always sent to the original source because the only information available in the datagram about the route is the source and destination IP addresses.
- ICMP uses the source IP address to send the error message to the source (originator) of the datagram.
- Five types of errors are handled: *destination unreachable, source quench, time exceeded, parameter problems, and redirection*.

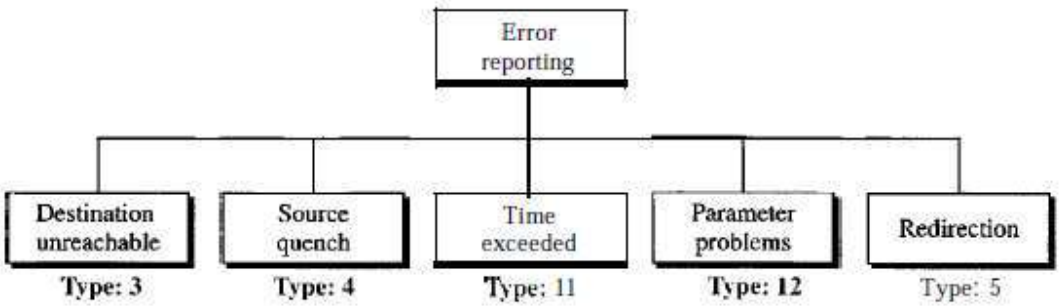


Figure 2.92 *Error-reporting messages*

- The following are important points about ICMP error messages:
  - No ICMP error message will be generated in response to a datagram carrying an ICMP error message.
  - No ICMP error message will be generated for a fragmented datagram that is not the first fragment.
  - No ICMP error message will be generated for a datagram having a multicast address.
  - No ICMP error message will be generated for a datagram having a special address such as 127.0.0.0 or 0.0.0.0.
- *Destination Unreachable*
  - When a router cannot route a datagram or a host cannot deliver a datagram, the datagram is discarded and the router or the host sends a destination-unreachable message back to the source host that initiated the datagram.
  - Destination-unreachable messages can be created by either a router or the destination host.
- *Source Quench :*
  - The IP protocol is a connectionless protocol. There is no communication between the source host, which produces the datagram, the routers, which forward it, and the destination host, which processes it.
  - IP does not have a flow control mechanism embedded in the protocol.
  - The source-quench message in ICMP was designed to add a kind of flow control to the IP.
  - When a router or host discards a datagram due to congestion, it sends a source-quench message to the sender of the datagram.
  - This message has two purposes.
    - First, it informs the source that the datagram has been discarded.
    - Second, it warns the source that there is congestion somewhere in the path and that the source should slow down (quench) the sending process.
- *Time Exceeded*
  - The time-exceeded message is generated in two cases: routers use routing tables to find the next hop (next router) that must receive the packet.
  - If there are errors in one or more routing tables, a packet can travel in a loop or a cycle, going from one router to the next or visiting a series of routers endlessly.
  - When a datagram visits a router, the value of this field is decremented by 1. When the time-to-live value reaches 0, after decrementing, the router discards the datagram.
  - When the datagram is discarded, a time-exceeded message must be sent by the router to the original source. Second, a time-exceeded message is also generated when not all fragments that make up a message arrive at the destination host within a certain time limit.
- *Parameter Problem*
  - If a router or the destination host discovers an ambiguous or missing value in any field of the datagram, it discards the datagram and sends a parameter-problem message back to the source.
- *Redirection*
  - When a router needs to send a packet destined for another network, it must know the IP address of the next appropriate router.

- The same is true if the sender is a host. Both routers and hosts, then, must have a routing table to find the address of the router or the next router.
- Routers take part in the routing update process, and are supposed to be updated constantly. Routing is dynamic.
- When a host comes up, its routing table has a limited number of entries. It usually knows the IP address of only one router, the default router.
- This concept of redirection is shown in Figure 2.93.

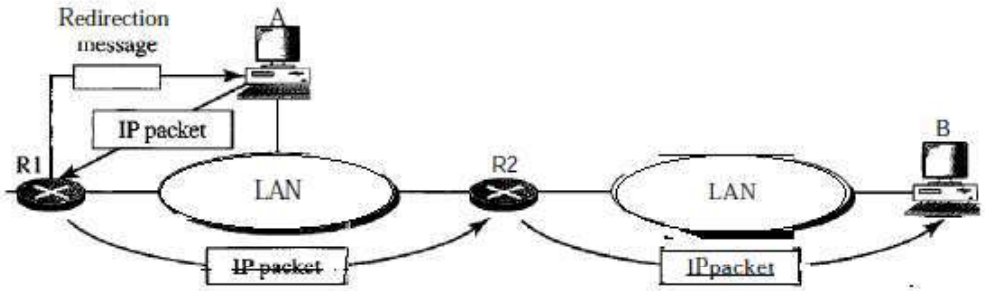


Figure 2.93 *Redirection concept*

- Host A wants to send a datagram to host B.
- Router R2 is obviously the most efficient routing choice, but host A did not choose router R2.
- The datagram goes to R1 instead. Router R1, after consulting its table, finds that the packet should have gone to R2.
- It sends the packet to R2 and, at the same time, sends a redirection message to host A. Host A's routing table can now be updated.

- *Query*
  - ✓ ICMP can diagnose some network problems.
  - ✓ This is accomplished through the query messages, a group of four different pairs of messages, as shown in Figure 2.94.

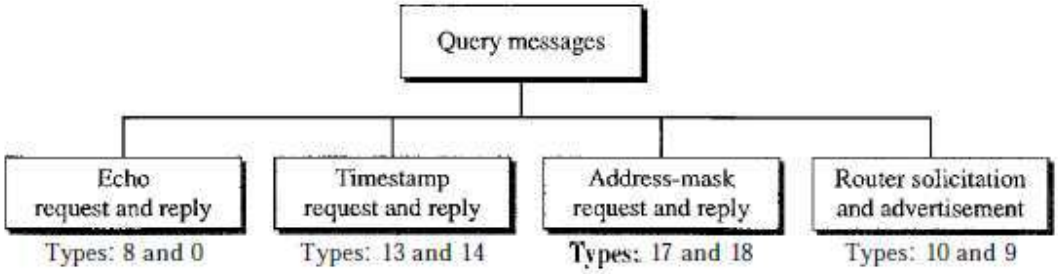


Figure 2.94 *Query messages*

  - ✓ In this type of ICMP message, a node sends a message that is answered in a specific format by the destination node.
  - ✓ *Echo Request and Reply*
    - The echo-request and echo-reply messages are designed for diagnostic purposes.
    - Network managers and users utilize this pair of messages to identify network problems.

- The combination of echo-request and echo-reply messages determines whether two systems (hosts or routers) can communicate with each other.

✓ *Timestamp Request and Reply*

- Two machines (hosts or routers) can use the timestamp request and timestamp reply messages to determine the round-trip time needed for an IP datagram to travel between them.
- It can also be used to synchronize the clocks in two machines.

✓ *Address-Mask Request and Reply*

- A host may know its IP address, but it may not know the corresponding mask.
- For example, a host may know its IP address as 159.31.17.24, but it may not know that the corresponding mask is /24. To obtain its mask, a host sends an address-mask-request message to a router on the LAN.
- If the host knows the address of the router, it sends the request directly to the router. If it does not know, it broadcasts the message.
- The router receiving the address-mask-request message responds with an address-mask-reply message, providing the necessary mask for the host. This can be applied to its full IP address to get its subnet address.

✓ *Router Solicitation and Advertisement*

- A host that wants to send data to a host on another network needs to know the address of routers connected to its own network.
- The host must know if the routers are alive and functioning.
- A host can broadcast (or multicast) a router-solicitation message.
- The router or routers that receive the solicitation message broadcast their routing information using the router-advertisement message.
- A router can also periodically send router-advertisement messages even if no host has solicited.

✓ *Checksum*

- In ICMP the checksum is calculated over the entire message (header and data).
- *Example*

Figure 3.57 shows an example of checksum calculation for a simple echo-request message. We randomly chose the identifier to be 1 and the sequence number to be 9. The message is divided into 16-bit (2-byte) words. The words are added and the sum is complemented. Now the sender can put this value in the checksum field.

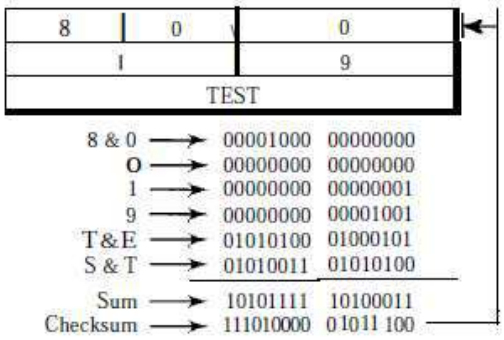


Figure 2.95 *Example ofchecksum calculation*

Debugging Tools

✓ There are several tools that can be used in the Internet for debugging.

✓ Two tools that use ICMP for debugging: *ping* and *traceroute.*

Ping

✓ The *ping* program to find if a host is alive and responding.

✓ The source host sends ICMP echo-request messages (type: 8, code: 0); the destination, if alive, responds with ICMP echo-reply messages.

✓ The *ping* program sets the identifier field in the echo-request and echo-reply message and starts the sequence number from 0; this number is incremented by 1 each time a new message is sent.

✓ The *ping* can calculate the round-trip time. It inserts the sending time in the data section of the message.

✓ When the packet arrives, it subtracts the arrival time from the departure time to get the round-trip time (RTT).

Example

We use the *ping* program to test the server fhda.edu. The result is shown below:

✓ The *ping* program sends messages with sequence numbers starting from 0.

✓ For each probe it gives us the RTT time. The TTL (time to live) field in the IP datagram that encapsulates an

```
$ ping thda.edu
PING thda.edu (153.18.8.1)   56 (84)  bytes of data.
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=O    ttl=62    time=1.91 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=1    ttl=62    time=2.04 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=2    ttl=62    time=1.90 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=3    ttl=62    time=1.97 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=4    ttl=62    time=1.93 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=5    ttl=62    time=2.00 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=6    ttl=62    time=1.94 ms
64 bytes from tiptoe.:fhda.edu (153.18.8.1): icmp_seq=7   ttl=62    time=1.94 ms
64 bytes from tiptoe.:fhda.edu (153.18.8.1): icmp_seq=8   ttl=62    time=1.97 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=9    ttl=62    time=1.89 ms
64 bytes from tiptoe.:fhda.edu (153.18.8.1): icmp_seq=lO   ttl==62   time=1.98 ms

--- thda.edu ping statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 10103ms
    rtt minJavg/max = 1.899/1.955/2.041 ms
```

ICMP message has been set to 62, which means the packet cannot travel mare than 62 hops.

✓ At the beginning, *ping* defines the number of data bytes as 56 and the total number of bytes as 84. It is obvious that if we add 8 bytes of ICMP header and 20 bytes of IP header to 56, the result is 84.

✓ In each probe *ping* defines the number of bytes as 64. This is the total number of bytes in the ICMP packet (56 + 8).

✓ The *ping* program continues to send messages, if we do not stop it by using the interrupt key (ctrl + c, for example). After it is interrupted, it prints the statistics of the probes. It tells us the number of packets sent, the number of packets received, the total time, and the RTT minimum, maximum, and average. Some systems may print more information.

Traceroute

✓ The *traceroute* program in UNIX or *tracert* in Windows can be used to trace the route of a packet from the source to the destination.

✓ The program uses two ICMP messages, time exceeded and destination unreachable, to find the route of a packet.

✓ This is a program at the application level that uses the services of UDP (see Chapter 23). Let us show the idea of the *traceroute* program by using Figure 2.96.

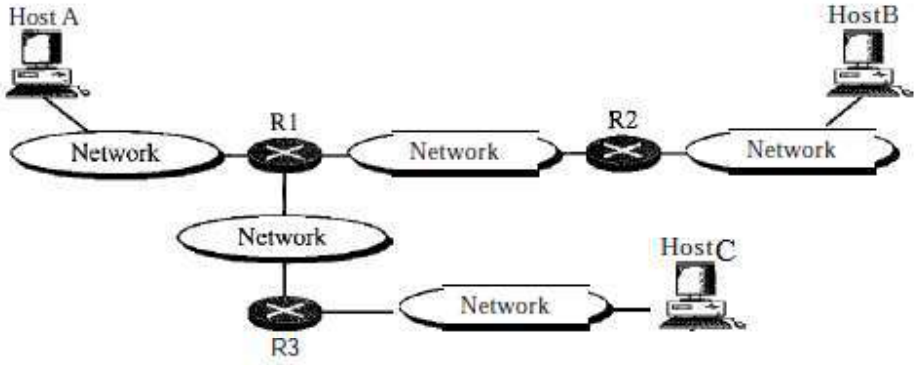


Figure 2.96 *The traceroute program operation*

✓ Given the topology, a packet from host A to host B travels through routers Rl and R2. There could be several routes from A to B.

✓ The *traceroute* program uses the ICMP messages and the TTL (time to live) field in the IP packet to find the route.

1. The *traceroute* program uses the following steps to find the address of the router Rl and the round-trip time between host A and router Rl.
   a. The *traceroute* application at hostA sends a packet to destination Busing UDP; the message is encapsulated in an IP packet with a TTL value of 1. The program notes the time the packet is sent.
   b. Router Rl receives the packet and decrements the value of TTL to 0. It then discards the packet (because TTL is 0). The router, however, sends a time-exceeded ICMP message (type: 11, code: 0) to show that the TTL value is 0 and the packet was discarded.
   c. The *traceroute* program receives the ICMP messages and uses the destination address of the IP packet encapsulating ICMP to find the IP address of router Rl. The program also makes note of the time the packet has arrived. The difference between this time and the time at step a is the round-trip time.

   The *traceroute* program repeats steps a to c three times to get a better average round-trip time. The first trip time may be much longer than the second or third because it takes time for the ARP program to find the physical address of router RI. For the second and third trips, ARP has the address in its cache.
2. The *traceroute* program repeats the previous steps to find the address of router R2 and the round-trip time between host A and router R2. However, in this step, the value of TTL is set to 2. So router RI forwards the message, while router R2 discards it and sends a time-exceeded ICMP message.
3. The *traceroute* program repeats step 2 to find the address of host B and the round-trip time between host A and host B. When host B receives the packet, it decrements the value of TIL, but it does not discard the message since it has reached its final destination.

✓ How can an ICMP message be sent back to host A?

- The destination port of the UDP packet is set to one that is not supported by the UDP protocol.
- When host B receives the packet, it cannot find an application program to accept the delivery.
- It discards the packet and sends an ICMP destination-unreachable message (type: 3, code: 3) to host A.

- Note that this situation does not happen at router RI or R2 because a router does not check the UDP header.
- The *traceroute* program records the destination address of the arrived IP datagram and makes note of the round-trip time.
- Receiving the destination unreachable message with a code value 3 is an indication that the whole route has been found and there is no need to send more packets.

Example

We use the *traceroute* program to find the route from the computer voyager.deanza.edu to the

```
$ traceroute fbda.edu
traceroute to fbda.edu   (153.18.8.1),30 hops max, 38 byte packets
 1  Dcore.fhda.edu       (153.18.31.254)   0.995 ms   0.899 ms   0.878 ms
 2  Dbackup.fhda.edu     (153.18.251.4)    1.039 fiS  1.064 fiS  1.083 fiS
 3  tiptoe.fhda.edu      (153.18.8.1)      1.797 illS 1.642 ms   1.757 fiS
```

- The unnumbered line after the command shows that the destination is 153.18.8.1.
- The TIL value is 30 hops.
- The packet contains 38 bytes: 20 bytes of IP header, 8 bytes of UDP header, and 10 bytes of application data.
- The application data are used by *traceroute* to keep track of the packets.
- The first line shows the first router visited. The router is named Dcore.fhda.edu with IP address 153.18.31.254.
- The first round-trip time was 0.995 ms, the second was 0.899 ms, and the third was 0.878 ms.
- The second line shows the second router visited. The router is named Dbackup.fhda.edu with IP address 153.18.251.4. The three round-trip times are also shown.
- The third line shows the destination host. We know that this is the destination host because there are no more lines.
- The destination host is the server thda.edu, but it is named tiptoe.fhda.edu with the IP address 153.18.8.1. The three round-trip times are also shown.