## Unit III

Chapter 3

# Data Link Layer

**Contents**

**Data link services :** Framing, Flow control, Error control, **ARQ methods :** Transmission efficiency, Piggybacking.

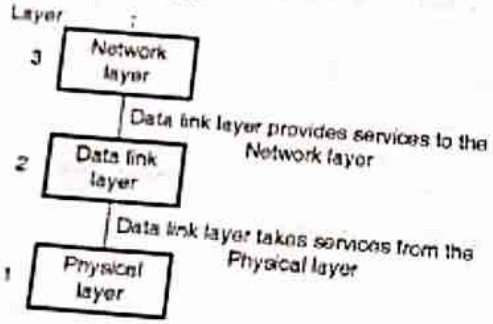
**Chapter Contents**

## 3.1 Introduction :

- The physical layer deals with the transmission of signals over different transmission media. A reliable and efficient communication between two adjacent machines can be achieved via the data link layer.
- This layer basically deals with frame formation, flow control, error control, addressing and link management.
- While sending data from source to destination errors may get introduced.
- The data communication circuits have only a finite data rate and there is non-zero propagation delay between the instant a bit is sent and the instant at which it is received.
- These limitations affect the efficiency of data transfer. The data link layer protocols used for communication take care of all these problems.
- Data link layer is the second layer in OSI reference model. It is above the physical layer.

### 3.1.1 Position of Data Link Layer :

- Fig. 3.1.1 shows the position of data link layer in the five layer Internet model.

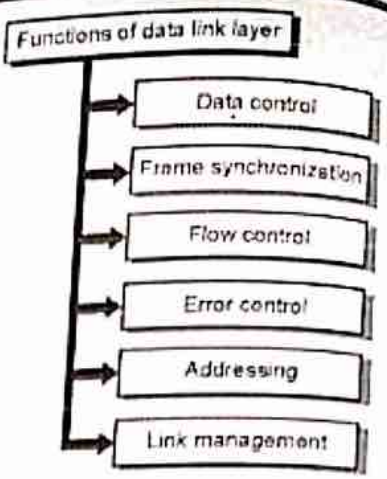



(L-663)Fig. 3.1.1 : Position of data link layer

- It is the second layer. It receives services from the physical layer and provides services to the network layer.

## 3.2 Data Link Layer Design Issues :

- The data link layer is supposed to carry out many specified functions.
- For effective data communication between two directly (physically) connected transmitting and receiving stations the data link layer has to carry out a number of specific functions as shown in Fig. 3.2.1.




(L-664)Fig. 3.2.1 : Functions of data link layer

1. **Services provided to the network layer :**
   - The data link layer provides a well defined service interface to the network layer.
   - The principle service is transferring data from the network layer on sending machine to the network layer on destination machine. This transfer always takes place via the DLL.
2. **Frame synchronization :**
   - The source machine sends data in the form of blocks called frames to the destination machine. The starting and ending of each frame should be identified so that the frames can be recognized by the destination machine.
3. **Flow control :**
   - The source machine must not send data frames at a rate faster than the capacity of destination machine to accept them.
4. **Error control :**
   - The errors introduced during transmission from source to destination machines must be detected and corrected at the destination machine.
5. **Addressing :**
   - When many machines are connected together (LAN), the identity of the individual machines must be specified while transmitting the data frames. This is known as addressing.
6. **Control and data on same link :**
   - The data and control information is combined in a frame and transmitted from the source to destination machine.

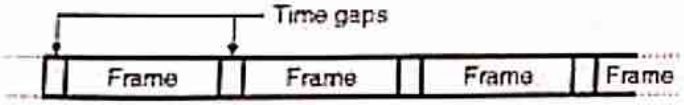
- The destination machine must be able to separate out the control information from the data being transmitted.

7. **Link management :**

- The communication link between the source and destination is required to be initiated, maintained and finally terminated for effective exchange of data.
- It requires co-ordination and co-operation among all the involved stations. Protocols or procedures are required to be designed for the link management.

## 3.3 Framing :

- The bits to be transmitted are first broken into discrete frames at the data link layer. In order to guarantee that the bit stream is error free, the checksum of each frame is computed.
- When a frame is received, the data link there, re-computes the checksum. If it is different from the checksum present in the frame, then the data link layer knows that an error has occurred.
- It then discards the bad frame and sends back a request for retransmission.
- Breaking the bit stream into frames is called as **framing.** One way of doing it is by inserting time gaps between frames as shown in Fig. 3.3.1.




(G-178) **Fig. 3.3.1 : Framing**

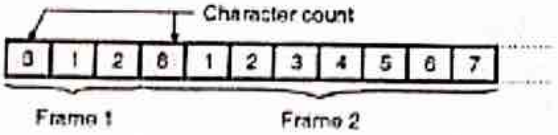
- But practically this framing technique does not work satisfactorily, because networks generally do not make any guarantees about the timing. So some other methods are derived.

### 3.3.1 Framing Methods :

- Following methods are used for carrying out framing :
  1. Character count method.
  2. Starting and ending characters, with character stuffing.
  3. Starting and ending flags with bit stuffing.
  4. Physical layer coding violations.

### 3.3.2 Character Count :

- In this method, a field in the header is used to specify the number of characters in the frame.
- This number helps the receiver to know the exact number of characters present in the frame following this count. The character count method is illustrated in Fig. 3.3.2.

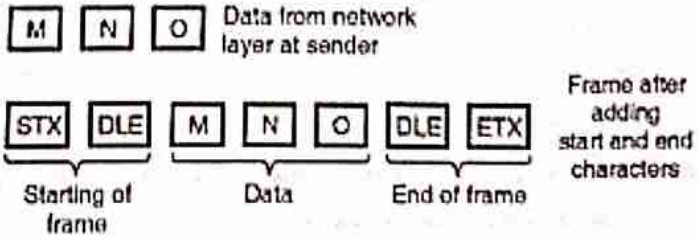



(L-668)**Fig. 3.3.2 : Character count method**

- The two frames shown in Fig. 3.3.2 contain 3 and 8 characters respectively and numbers 3 and 8 are inserted in the headers of the corresponding frames.
- The disadvantage of this method is that, an error can change the character count itself.
- If the wrong character count number is received due to error then the receiver will get out of synchronization and will not be able to locate the start of next frame.
- The character count method is rarely used in practice.

### 3.3.3 Starting and Ending Character with Charactor Stuffing :

- The problem of character count method is solved here by using a starting character before the starting of each frame and an ending character at the end of each frame.
- Each frame is preceded by the transmission of ASCII character sequence DLE STX.
- (DLE stands for data link escape and STX is start of Text).
- After each frame the ASCII character sequence DLE ETX is transmitted.
- Here DLE stands for Data Link Escape and ETX stands for End of Text.
- So if the receiver loses the synchronization, it just has to search for the DLE STX or DLE ETX characters to return back on track. This is shown in Fig. 3.3.3.

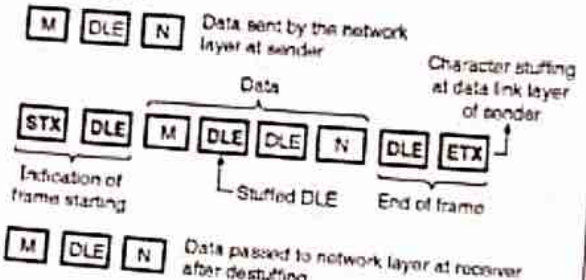



(L-669)**Fig. 3.3.3**

### 3.3.4 Character Stuffing :

- The problem with this system is that the characters DLE STX or DLE ETX can be a part of data as well. If so, they will be misinterpreted by the receiver as start or end of frame.

- This problem is solved by using a technique called **character stuffing** which is as follows.
- The data link layer at the sending end inserts an ASCII DLE character just before each accidental DLE character in the data being transmitted.
- The data link layer at the receiving end will remove these DLE characters before transferring the data to the network layer.
- Thus the DLE STX or DLE ETX used for framing purpose can be distinguished from the one in data because DLEs in the data always appear more than once.
- This is called character stuffing and it is shown in Fig. 3.3.4.




**(G-181) Fig. 3.3.4 : Character stuffing**

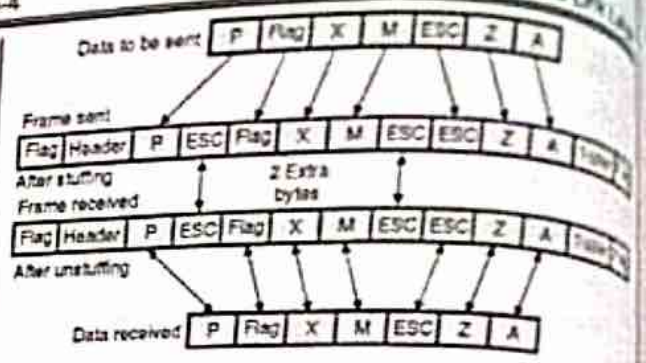
- Note that at the receiving end the de-stuffing is essential.
- De-stuffing process is exactly opposite to the character stuffing process.

**Disadvantages :**

- The main disadvantage of this framing method is that we have to use the 8 bit characters and ASCII code.
- This problem can be overcome by using the next framing technique.

**Byte stuffing :**

- In byte stuffing a special byte is added to the data section of the frame when there is a character with the same pattern as the flag.
- The data section is stuffed with an extra byte. This byte is called as the escape character (ESC).
- At the receiver these ESC bytes are removed from the data section and the next character is treated as data. Fig. 3.3.5 demonstrates the concept of byte stuffing.
- Byte stuffing by the escape character will allow the presence of the flag is the data section of the frame.




**(G-182) Fig. 3.3.5 : Byte-stuffing**

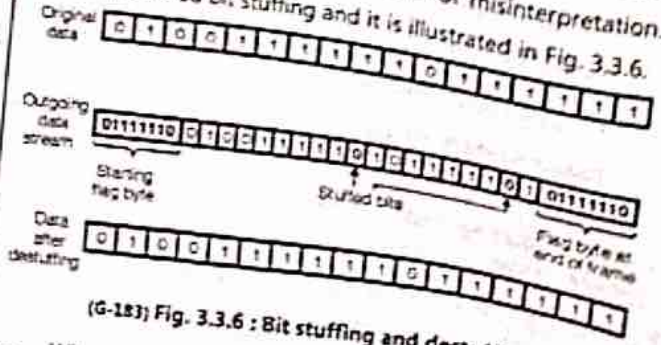
- But it has a problem. if the text contains one or more escape characters followed by a flag. Because then the receiver will remove the escape character but will keep the flag.
- This problem is solved by marking the escape characters that are a part of the text by another escape (ESC) character as shown in Fig. 3.3.5.

## 3.3.5 Starting and Ending Flags, with Bit Stuffing :

- In this framing techniques at the beginning and end of each frame, a specific bit pattern 0111 1110 called **flag byte** is transmitted by the sending station.
- Since there are six consecutive 1s in the flag byte a technique called **bit stuffing** which is similar to character stuffing is used. It is as explained below.

**Bit stuffing :**

- Whenever the sender data link layer detects the presence of five consecutive ones in the data stream, it automatically stuffs a 0 bit into the outgoing bit stream.
- Thus the six consecutive 1s will never appear in the data stream. Hence there is no chance of misinterpretation. This is called bit stuffing and it is illustrated in Fig. 3.3.6.



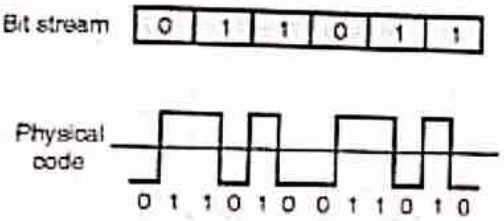

**(G-183) Fig. 3.3.6 : Bit stuffing and destuffing**

- When a receiver detects presence of five consecutive ones in the received bit stream, it automatically deletes the 0 bit following the five ones.

- This is called de-stuffing. It is shown in Fig. 3.3.6. Due to bit stuffing, the possible problem if the data contains the flag byte pattern (0111 1110) is eliminated.

### 3.3.6 Physical Layer Coding Violations :

- This method of framing is applicable only to those networks in which the encoding on the physical medium contains some redundancy.
- Some LANs encode each bit of data using two physical bits for example the use of the Manchester coding (Refer Fig. 3.3.7).
- The physical Manchester code makes a transition at the middle of the bit interval as shown. Therefore a 1 bit is encoded into a 10 pair and a 0 bit is encoded into a 01 pair as shown in Fig. 3.3.7.




(G-184) **Fig. 3.3.7**

- This helps in recognizing the boundaries of bits in a precise manner. This use of invalid physical code is a part of 802 LAN standards.

**Which method of framing is used practically ?**

- Many data link protocols use the combination of the character count technique with one of the other techniques so as to have an extra safety.

## 3.4 Error Control :

- The next problem to be dealt with is to make sure that all frames are eventually delivered to the network layer at the destination, in proper order.
- Generally the receiver sends back some feedback (positive or negative) to convey the information about whether it has received a frame or not.
- A positive acknowledgement (feedback) ACK indicates a successful and error free delivery of a frame.
- Whereas a negative acknowledgement (NAK) means that something has gone wrong and that particular frame needs to be retransmitted.
- Due to the presence of noise burst a frame may vanish completely.
- So the receiver does not receive anything and it does not react at all (no acknowledgement).
- This problem is overcome by introducing a timer in the data link layer. Its function of this timer is as follows.

### 3.4.1 Function of a Timer :

- As soon as a sender transmits a frame, it also starts the data link timer.
- The timer timing is set by taking into account the factors such as the time required for the frame to reach the destination, processing time at the destination and the time required for the acknowledgement to return back.
- Normally the frame is received correctly and the acknowledgement will return back to the sender before the timer runs out. This shows that a frame has been received and the timer is cancelled.
- But if a frame is lost or acknowledgement is lost, then the timer will go off. This will alert the sender that there is some problem. The solution to this problem is that the sender retransmits the same frame.
- But when a frame is transmitted multiple times, there is a possibility that the receiver will receive the same frame two or more times and pass it to the network layer more than once. This is called as duplication.
- To avoid this each outgoing frame is assigned a distinct sequence number. This will help the receiver to distinguish retransmission.

### 3.4.2 Cyclic Redundancy Check (CRC) :

**Definition :**

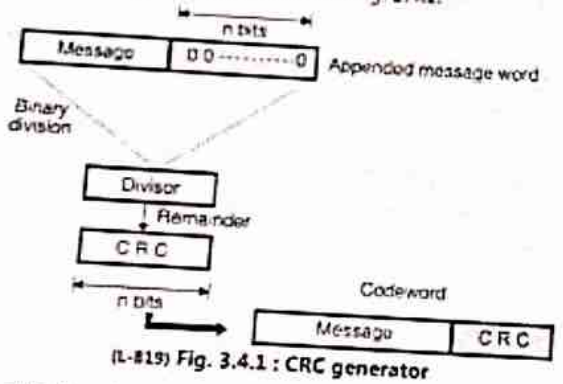
- CRC is an error detection code which is included in each transmitted codeword as shown in Fig. 3.4.1 and used by the receiver to detect the errors in the received codeword.
- This is a type of polynomial code in which a bit string is represented in the form of polynomials with coefficients of 0 and 1 only.
- Polynomial arithmetic uses a modulo-2 arithmetic i.e. addition and subtraction are identical to EXOR. For CRC code the sender and receiver should agree upon a generator polynomial G(x).
- A codeword can be generated for a given data word (message) polynomial M(x) with the help of long division. This technique is more powerful than the parity check and checksum error detection.

**Procedure of error detection :**

- CRC works on the principle of binary division. A sequence of redundant bits called CRC or CRC remainder is appended at the end of the message. We will call this word as appended message word.
- The appended word thus obtained becomes exactly divisible by the generator word corresponding to G (x).
- The sender appends the CRC to the message word to form a codeword. At the receiver, this codeword is divided by the same generator word which corresponds to G (x).
- There is no error if the remainder of this division is zero. But a non-zero remainder indicates presence of errors in the received codeword.
- Such an erroneous codeword is then rejected.

**CRC generator :**

- The CRC generator is shown in Fig. 3.4.1.




(L-815) **Fig. 3.4.1 : CRC generator**

- The stepwise procedure in CRC generation is as follows :

**Step 1 :** Append a train of n 0s to the message word where n is 1 less than the number of bits in the predecided divisor (i.e. generator word). If the divisor is 5-bit long then we have to append 4-zeros to the message.

**Step 2 :** Divide the newly generated data unit in step 1 by the divisor (generator). This is a binary division.

**Step 3 :** The remainder obtained after the division in step 2 is the n bit CRC.

**Step 4 :** This CRC will replace the n 0s appended to the data unit in step 1, to get the codeword to be transmitted as shown in Fig. 3.4.1.

**Generation of CRC code :**

- The generation of CRC code is clear after solving the following example.

**Ex. 3.4.1 :** Generate the CRC code for the data word to be transmitted 1 1 0 0 1 0 1 0 1. The divisor is 1 0 1 0 1.

**Soln. :**

**Given :** Data word : 1 1 0 0 1 0 1 0 1

Divisor : 1 0 1 0 1.

The number of data bits = m = 9

The number of bits in the codeword = n = 5

**Step 1 : Obtain the dividend :**

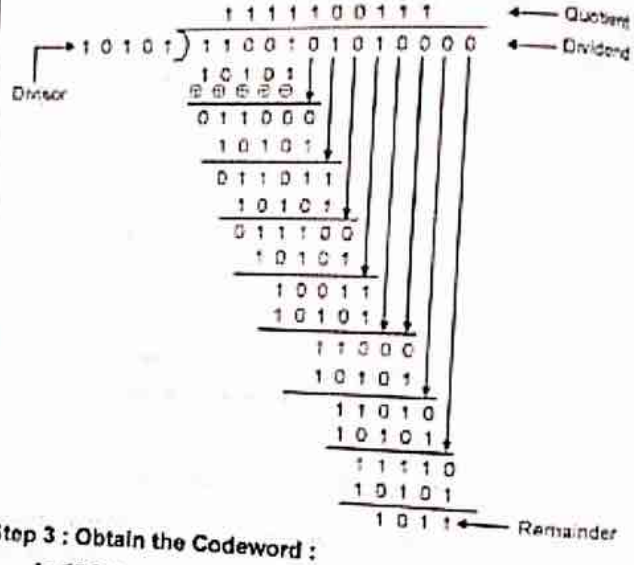
Dividend = Data word + (n – 1) zeros. (G-801(a))

Dividend = 1 1 0 0 1 0 1 0 1 | 0 0 0 0

Data word — 4 additional zeros

**Step 2 : Carry out the division :**

- Carry out the division as follows : (G-801(b))




**Step 3 : Obtain the Codeword :**

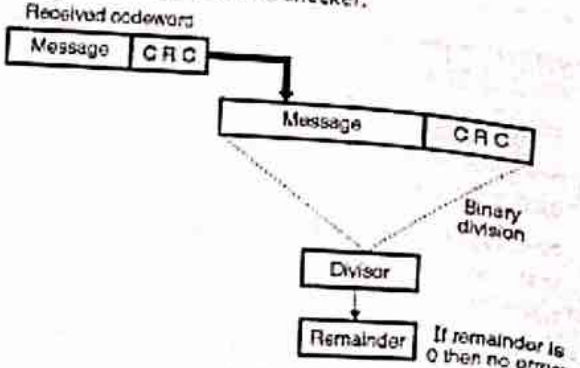
- In CRC the required codeword is obtained by writing the data word followed by the remainder.

∴ Codeword = 1 1 0 0 1 0 1 0 1 | 1 0 1 1

Data word — Remainder (G-1761)

## 3.4.3 CRC Checker :

- Fig. 3.4.2 shows the CRC checker.




(L-820) **Fig. 3.4.2 : CRC checker**

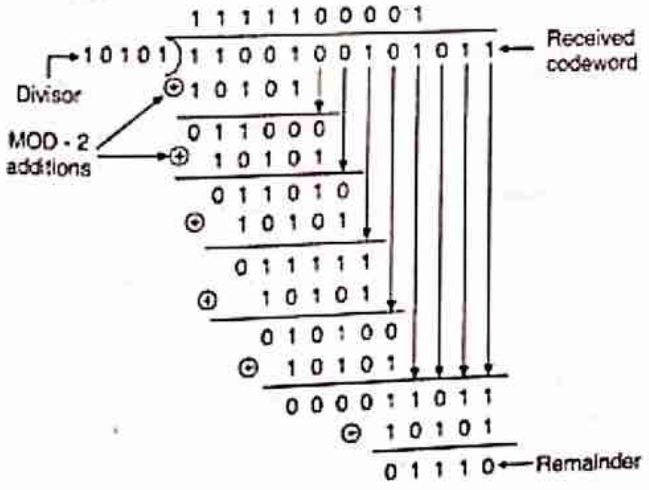
- The codeword received at the receiver consists of message and CRC. (Fig. 3.4.2)
- The receiver treats it as one unit and divides it by the same (n + 1) bit divisor (generator word) which was used at the transmitter.
- The remainder of this division is then checked. If the remainder is zero, then the received codeword is error free and hence should be accepted.
- But a non-zero remainder indicates presence of errors hence the corresponding codeword should be rejected.

**Ex. 3.4.2 :** The codeword is received as 1100 1001 01011. Check whether there are errors in the received codeword, if the divisor is 10101. (The divisor corresponds to the generator polynomial).

**Soln. :**

- As we know the codeword is formed by adding the dividend and the remainder.
- This codeword will have an important property that it will be completely divisible by the divisor.
- Thus at the receiver we have to divide the received codeword by, the same divisor and check for the remainder.
- If there is no remainder then there are no errors.
- But if there is remainder after division, then there are errors in the received codeword.
- Let us use this technique and find if there are errors.

Data word : 1100 1001 01011
Divisor : 10101 (G-201(a))



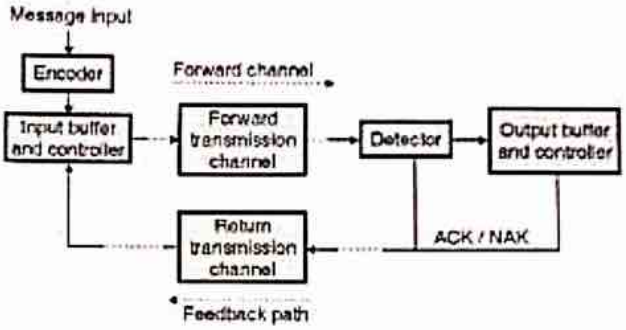

- The non zero remainder shows that there are errors in the received codeword.

## 3.4.4 ARQ Technique :

- There are two basic systems of error detection and correction FEC system and ARQ system.
- In the ARQ system of error control, when an error is detected, a request is made for the retransmission of that signal.
- Therefore a feedback channel is required for sending the request for retransmission.

**Basic ARQ system :**

- The block diagram of the basic ARQ system is as shown in Fig. 3.4.3.




(L-372) **Fig. 3.4.3 : Block diagram of the basic ARQ system**

**Operation of ARQ system :**

- The encoder produces code words for each message signal at its input.
- Each codeword at the encoder output is stored temporarily and transmitted over the forward transmission channel.
- At the destination a decoder will decode the code words and look for errors.
- The decoder will output a "positive acknowledgment" (ACK) if no errors are detected and it will output a negative acknowledgment (NAK) if errors are detected.
- On receiving a negative acknowledgment (NAK) signal via the return transmission path the "controller" will retransmit the appropriate word from the words stored by the input buffer.
- A particular word may be retransmitted only once or it may be retransmitted twice or more number of times.
- The output controller and buffer on the receiver side assemble the output bit stream from the code words accepted by the decoder.

**Types of ARQ system :**

- The three types of ARQ systems are :
  1. Stop-and-wait ARQ system
  2. Go back n ARQ and 3. Selective repeat ARQ

**Note :** Error control in the data link layer is based on the principle of request for automatic retransmission (ARQ) of the missing, lost or damaged frames.

## 3.5 Flow Control :

- This is another important design issue related to the data link layer
- In flow control the problem to be handled is what to do with the sender computer that wants to send data at a faster rate than the capacity of the receiver to receive them
- This happens when the sender is using a faster computer than the receiver. The data sent at a very fast rate will completely overwhelm the receiver.
- The receiver will keep losing some of the frames simply because they are arriving too quickly. The solution to this problem is to introduce the **flow control.**
- The flow control will control the rate of frame transmission to a value which can be handled by the receiver.
- It requires some kind of a feedback mechanism from the receiver to the sender, so as to adjust the sending rate automatically.
- We are going to discuss some flow control techniques based on this principle.
- It is a set of procedures that tells the sender how much data it can transmit before it must wait for an acknowledgement from the receiver, otherwise there will be overflow of data.
- The data flow should not be so fast that the receiver is over-whelmed.
- The speed of processing of any receiving device is a limited and it also has a limited amount of memory storage space, for storing the incoming data.
- There has to be some system, for reverse communication from the receiver to transmitter.
- The receiver can tell the transmitter about adjusting the data flow rate to suit its speed or even stop temporarily.
- As the rate of processing at the receiver is generally slower than the rate of transmission. Each receiver has a finite memory called **buffer.**
- The incoming data is first stored in the buffer and then sequentially processed.
- If the buffer begins to fill up, the receiver must be able to tell the sender to stop transmission until the buffer gets empty.
- Similarly the transmitter also has a buffer for storing the bits if the transmission is stopped.

**Note :** Flow control can be defined as a set of procedures which are used for limiting the amount of data a transmitter can send before waiting for acknowledgement.

## 3.6 Elementary Data Link Protocols :

- In this section we are going to discuss some elementary data link layer protocols.

### 3.6.1 An Unrestricted Simplex Protocol :

- This protocol is the simplest possible protocol. The transmission of data takes place in only one direction. So, it is a simplex (unidirectional) protocol.
- It is assumed that the network layers of sender and receiver are always ready. It is also assumed that we can ignore the processing time and the buffer space available infinite.
- The communication channel is imagined to be noise free so it does not damage or lose any frames.
- All this is highly unrealistic. This protocol is also called as "utopia". This protocol consists of two distinct procedures, namely a sender and a receiver.
- They run in the data link layers of their respective machines. No sequence numbers or acknowledgements are used.

### 3.6.2 A Simplex Stop and Wait Protocol :

- The most unrealistic restriction in the previous protocol is the assumption that the receiving network layer can process the data with zero processing time.
- In the simplex stop and wait protocol it is assumed that a finite processing time is essential.
- However like the first protocol, the communication channel is assumed to be noise free and the communication is simplex i.e. only in one direction at a given time.
- This protocol deals with an important problem i.e. how to prevent the sender from flooding the receiver due to the data rates faster than processing speed of the receiver.
- In this protocol, a small dummy frame is sent back from the receiver to the transmitter to indicate that it can send the next frame. The small dummy frame is called as acknowledgement.
- The transmitter sends one frame and then waits for the dummy frame called acknowledgement.

- Once the acknowledgement is received, it sends the next frame and waits for the acknowledgement. Hence, this protocol is known as **stop and wait** protocol.
- The best thing about this protocol is that the incoming frame is always an acknowledgement. It need not be even checked.

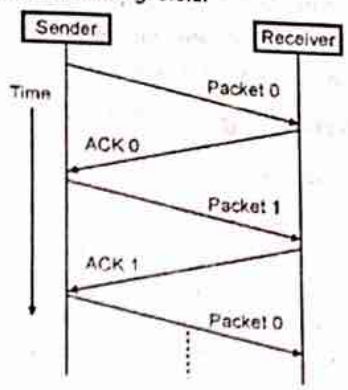
### 3.6.3 A Simplex Protocol for Noisy Channel :

- This is the third protocol in which we go one step ahead and assume that the communication channel is noisy and can introduce errors in the data travelling over it.
- The channel noise can either damage the frames or they may get lost completely. In this protocol, the sender waits for a positive acknowledgement before advancing to the next data item.
- There is a timer set at the sender when a frame is sent. If the sender times out it will resend the same frame again.
- So it is called as PAR (Positive acknowledgement with retransmission) or Automatic Repeat Request (ARQ) type protocol.
- If a frame is badly damaged or lost then the sender would retransmit it. Note that due to retransmission (time out or any other reason), there is always a possibility of duplication of frames at the receiver.
- To avoid this, the sender puts a sequence number in the header of each frame it sends. The receiver can check the sequence number of each arriving frame to check for the possible duplicate frame.
- If a frame is duplicated then receiver will discard it.
- The operation can be divided into two modes :

  1. Normal operation and 2. Time out.

**1. Normal operation :**

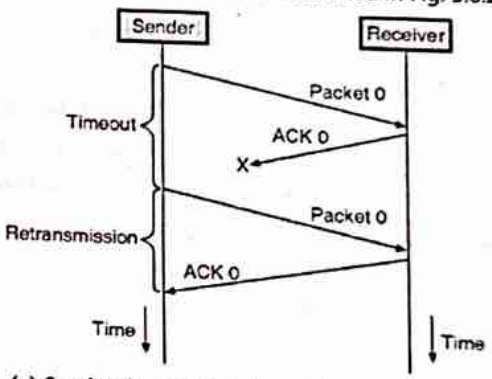
- After transmitting one frame, the sender waits for an acknowledgement (ACK) from the receiver before transmitting the next one.
- In this way, the sender can recognize that the previous packet is transmitted successfully and we could say "stop and wait" guarantees reliable transfer between nodes.
- To support this feature, the sender keeps a record of each frame it sends.
- Also, to avoid confusion caused by delayed or duplicated ACKs, "stop-and-wait" sends each packet with unique sequence numbers and receives that numbers in each ACKs.
- This is illustrated in Fig. 3.6.1.



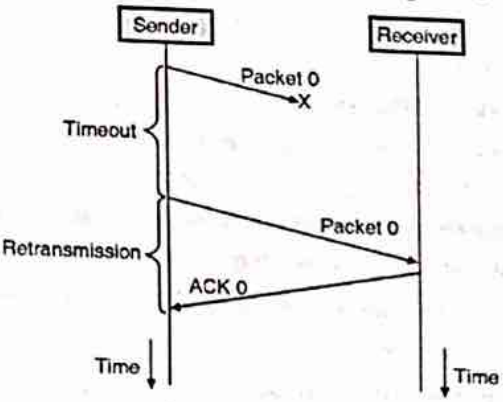

**(G-220)Fig. 3.6.1 : Positive acknowledgement with retransmission**

**2. Time out :**

- If the sender does not receive **ACK** for previous sent frame after a certain period of time, the sender times out and retransmit that frame again.
- There are two cases when the sender does not receive ACK; one is when the ACK is lost and the other is when the frame itself is not received i.e. it got lost.
- These two possible cases are illustrated in Fig. 3.6.2.




**(a) Sender does not receive acknowledgement**




**(b) Frame is lost**

**(G-221) Fig. 3.6.2 : Timeout and retransmission**

- To support this feature, the sender keeps timer for each frame. We have already discussed that a timer is introduced in the data link layer.

### 3.6.4 Piggybacking :

- In all the practical situations, the transmission of data needs to be bi-directional. This is called as full-duplex transmission.
- One way of achieving full duplex transmission is to have two separate channels one for forward data transmission and the other for reverse data transfer (for acknowledgements).
- But this will waste the bandwidth of the reverse channel almost entirely.
- A better solution would be to use each channel (forward and reverse) to transmit frames both ways, with both channels having the same capacity.
- Let A and B be the users. Then the data frames from A to B are intermixed with the acknowledgements from A to B.
- By checking the kind field in the header of the received frame the received frame can be identified as either data frame or acknowledgement.
- One more improvement can be made. When a data frame arrives, the receiver waits, does not send the control frame (acknowledgement) back immediately.
- The receiver waits until its network layer passes in the next data packet. The acknowledgement is then attached to this outgoing data frame.
- Thus the acknowledgement travels along with next data frame. This technique in which the outgoing acknowledgement is delayed temporarily is called as **piggybacking**.

**Advantage of piggybacking :**

- The major advantage of piggybacking is better use of available channel bandwidth. This happens because an acknowledgement frame need not be sent separately.

**Disadvantages of piggybacking :**

1. The disadvantage of piggybacking is the additional complexity.
2. If the data link layer waits too long before transmitting acknowledgement, then retransmission of frame would take place.

## 3.7 Sliding Window Protocols :

- The next three protocols are more robust and bi-directional protocols. All these protocols are special type of protocol called **Sliding Window Protocols.**
- They show a different performance in terms of their efficiency, complexity and buffer requirements.

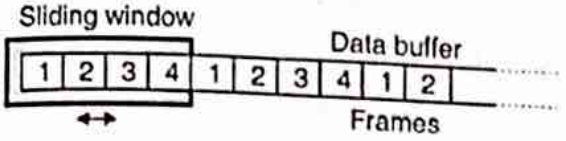
**Sequence number :**

- One of the important features of all the sliding window protocols is that each outbound frame contains a sequence number, ranging from 0 to some maximum value.
- The maximum value is generally equal to $(2^n - 1)$. The value of n can be arbitrary.

**Sliding windows :**

- Sliding windows are basically the imaginary boxes at the transmitter and receiver.
- This window holds the frames at the transmitting as well as receiving ends and provides the upper limit on the number of frames that can be transmitted before acknowledgement is obtained.
- So in short we can say that, at any instant of time, the sender maintains a set of sequence numbers corresponding to the frames it is permitted to send.
- These frames which are being permitted to sent are said to be residing inside the **sending window.**
- The receiver also maintains a **receiver window.** It corresponds to the set of frames that the receiver is permitted to accept. The sender and receiver windows can be of different sizes.
- The positive or negative acknowledgement (ACK or NAK) should be used after every frame.
- That means the sender sends frame, waits for the acknowledgement and sends the next frame or retransmits the original one, only after receiving either positive or negative acknowledgement from the receiver.
- In order to improve the efficiency, the sender sends multiple frames at time, the receiver checks the CRC of all the frames one by one and sends one acknowledgement for all the frames.
- This is the principle of operation of sliding window technique.
- In this technique, an imaginary window consisting of "n" number of data frames is defined.

- This means that upto n number of frames can be sent before receiving an acknowledgement.
- This is known as sliding window because this window can slide over the data buffer to be sent as shown in Fig. 3.7.1(a).

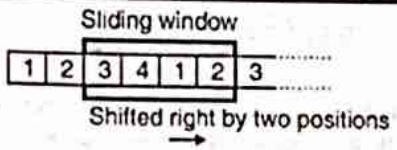



(G-222)**Fig. 3.7.1(a) : Sliding window**

- In Fig. 3.7.1(a) we have shown a sliding window of size n = 4. That means the sender can send four frames, at a time and then wait for the acknowledgement for the receiver.
- So there will be one acknowledgement corresponding to four sent frames. Note the numbering of frames in Fig. 3.7.1(a). As the window size is 4, the frame numbering is 1, 2, 3, 4 then again 1, 2, 3, ... the maximum frame number is restricted to n.

**Sender and receiver sliding windows :**

- The sender as well as the receiver maintains their own sliding windows.
- The sender sends the number of frames allowed by the size of its own sliding window and then waits for an acknowledgement from the receiver.
- The receiver sends an acknowledgement which includes the number of the next frame that the sender should send.
- For example if the sender has sent frames 1 and 2 to the receiver and if receiver receives them correctly, then the acknowledgement sent by the receiver will include number-3 indicating the sender to send frame number-3.
- Now if the sender transmits the first 4 frames as per the size of its window and receives an acknowledgement for the first two frames, then the sender will slide its window two frames to the right as shown in Fig. 3.7.1(b) and sends 5$^{th}$ and 6$^{th}$ frames (i.e. frames 1 and 2 of the next lot).
- The receiver now has four frames again, so it checks frames 3, 4, 1, 2 by checking their CRC.

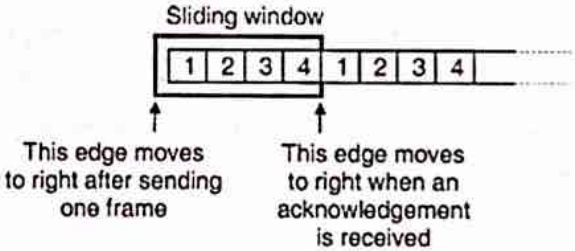



(G-223)**Fig. 3.7.1(b) : Illustration of sliding window mechanism**

- If it finds frame 3 faulty then it will send an acknowledgement which includes number 3.
- The sender will send 4-frames staring from frame-3 onwards.
- The sliding window mechanism thus uses two buffers and one window so as to exercise the flow control.
- The application program on the sender side will create the data to be transmitted and loads into the sender's buffer.
- Then the sender's sliding window is imposed on this buffer. These frames are then sent till all the frames have been sent.
- The receiver receives these data frames and carriers out checks such as CRC, missing or duplicate frames etc. and stores the correct frames in the receiver buffer. The application program at the receiver then the takes this data.

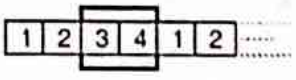
**Movement of sender's window :**

- Fig. 3.7.1(c) shows the sender's window.



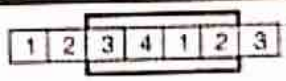

(G-224)**Fig. 3.7.1(c) : Sender sliding window**

- If the senders window size is 4 and frames 1 and 2 are sent but acknowledgement has not been received so far, then as shown in Fig. 3.7.1(d), the sender's windows will only contain two frames i.e. 3 and 4.




(G-225)**Fig. 3.7.1(d) : Sender's window after sending first two frames but no acknowledgement**

- Now if the sender receives acknowledgement bearing number 3 then it understands that the receiver has correctly received frames 1 and 2.
- The sender's window now expands and includes the next two frames as shown in Fig. 3.7.1(e).

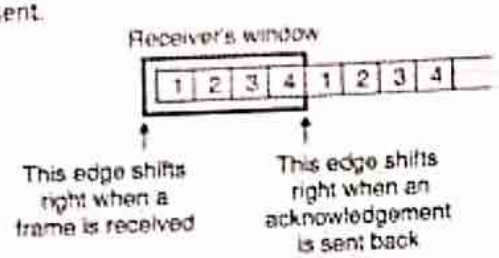



(G-226)**Fig. 3.7.1(e) : Sender's window after receiving acknowledgement bearing number-3**

- In this way the left edge of sender's window will shift right when the data frames are sent and the right edge of the senders window will shift right when the acknowledgement is received.

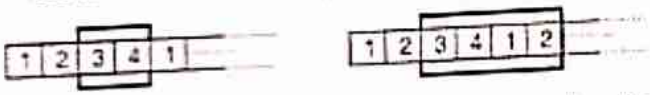
**Movement of receiver's windows :**

- Fig. 3.7.1(f) shows the receiver's window. Its left edge shifts right on receiving each data frame, where as its right edge shifts right when an acknowledgement is sent.




(G-227)**Fig. 3.7.1(f) : Receiver's sliding window**

- If we take the same example that we discussed for the sender's window then the position of receivers windows are as shown in Fig. 3.7.1(g) and (h).




**(g) Two frames (1 and 2) received but no acknowledgement sent** **(h) After sending the acknowledgement**

(G-228)**Fig. 3.7.1 : Movement of receiver window**

**Ex. 3.7.1 :** Two neighbouring nodes A and B uses sliding window protocol with 3 bit sequence number. As the ARQ mechanism Go back N is used with window size of 4. Assume A is transmitting and B is receiving show window position for the following events :

1. Before A send any frame.
2. After A send frame 0, 1, 2 and receive ACK (acknowledgement) from B for 0 and 1.

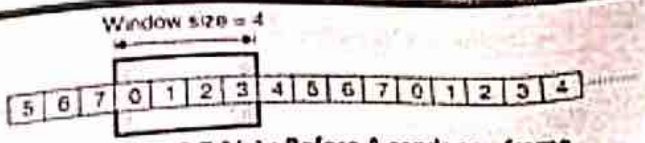
**Soln. :**

- The number of sequence number bits = m = 3.

  ∴ The sequence numbers will be 0, 1, 2, 3 ..., 6, 7. We can repeat these numbers. So the sequence will be,

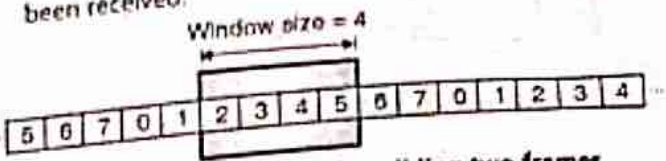
  0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, 3, 4, .....

- The size of the window is 4.
- Fig. P. 3.7.1(a) shows the sender window (at A) before sending any frame.




(G-229) **Fig. P. 3.7.1(a) : Before A sends any frame**

- Fig. P. 3.7.1(b) shows that the window slides 2 positions because acknowledgement for frames 0 and 1 have been received.




(G-230) **Fig. P. 3.7.1(b) : After sliding two frames**

## 3.7.1 A One Bit Sliding Window Protocol (Stop and Wait ARQ) :

- This protocol is called one bit protocol because the maximum window size here i.e. n is equal to 1. It uses the stop-and-wait technique which we have discussed earlier.
- The sender sends one frame and waits to get its acknowledgement. The sender transmits its next frame only after receiving the acknowledgement for the earlier frame.
- So one bit sliding window protocol is also called as **stop and wait protocol.**
- The sequence of events taking place when a frame is transmitted and received is as follows : (G-234)

1. The data link layer of the sending machine fetches the first packet from its network layer.
2. It builds the frame for it and sends it to receiver.
3. The receiver data link layer checks the received frame for duplication.
4. If ok, it passes the frame to its network layer.

**The operation of protocol :**

- The operation of this protocol is based on the ARQ (automatic repeat request) principle. So the sliding window protocols are also called as ARQ protocols.
- In this method the transmitter transmits one frame of data and waits for an acknowledgement from the receiver.
- If it receives a positive acknowledgement (ACK) transmits the next frame. If it receives a negative acknowledgement (NAK) it retransmits the same frame.

**Features added for retransmission :**

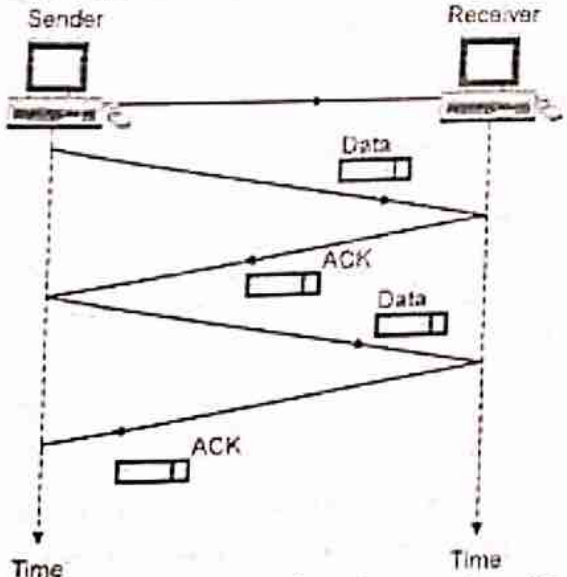
- For retransmission, four features are added to the basic flow control mechanism :
  1. The transmitter stores the copy of last frame transmitted until an acknowledgement for that frame is received from the destination.
  2. For distinctly identifying different types of frames both data and ACK frames are numbered alternately 0 and 1. The first data frame sent is numbered as 0. This frame is acknowledged by an ACK 1 frame. After receiving ACK1 the sender sends next data frame having a number 1.
  3. If an error occurs while transmission, the receiver sends a NAK frame back to the transmitter for retransmission of the corrupted frame. NAK frames which are not numbered tell the transmitter to retransmit the last frame transmitted.
  4. The transmitter has a timer to take care of the frame ACK which are lost. After a specified time if the transmitter does not receive a ACK or NAK frame it retransmits the last frame.

**When is the retransmission necessary ?**

- The retransmission of frame is essential under the following events :
  1. If the received frame is damaged.
  2. If the transmitted frame is lost.
  3. If the acknowledgement from the receiver is lost.
- Let us see the operation of the protocol under these circumstances one by one.

**Operation under normal condition :**

- Fig. 3.7.2 illustrates the protocol operation when everything is normal.

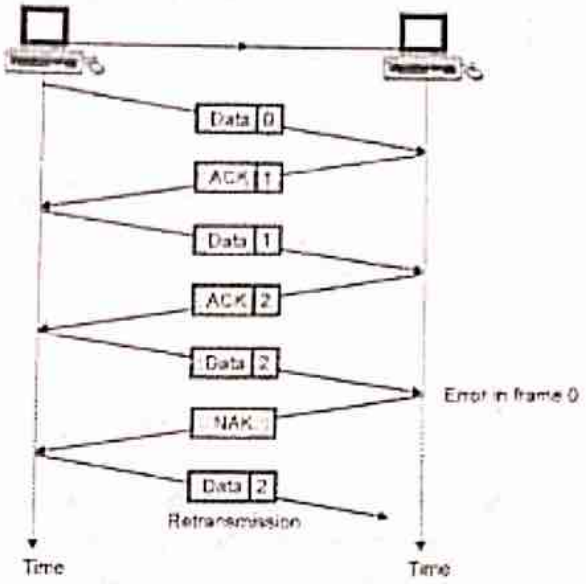



**(G-235) Fig. 3.7.2 : Stop and wait under normal condition**

- No frame is lost so retransmission is not necessary.

**Stop and wait ARQ for damaged frame :**

- As seen in Fig 3.7.3(a) the transmitter transmits data frame numbered 0.

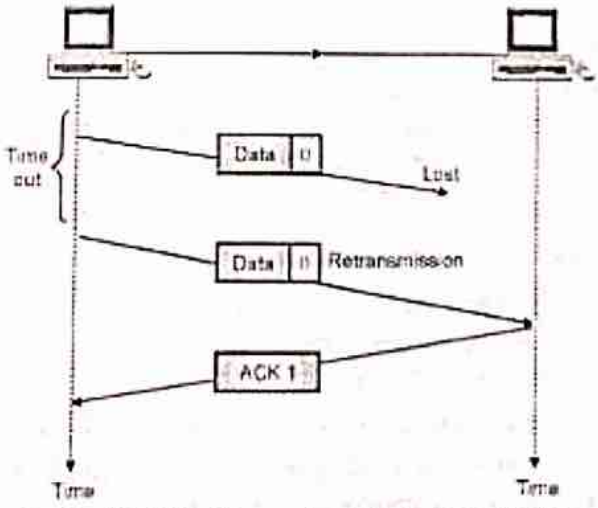



**(G-236)Fig. 3.7.3(a) : Stop and wait ARQ damaged frame**

- The receiver returns an ACK 1 indicating that the data frame numbered 0 is received without any error. The next data frame i.e. data 1 is sent. The corresponding acknowledgement ACK2 is received.
- The process goes on in this way, but if an error occurs the receiver sends a NAK requesting retransmission of the corrupted data frame (data 2). So the transmitter retransmits the data frame 2.

**Stop and wait ARQ for lost data frame :**

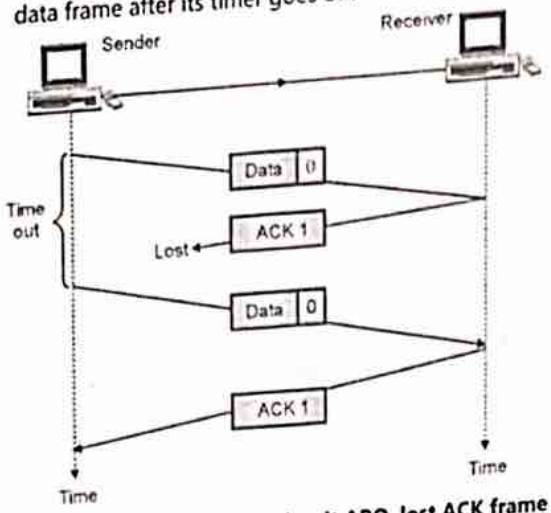
- Fig. 3.7.3(b) shows that if a data frame is lost and if the transmitter does not receive any type of acknowledgement from the receiver with a specified time it retransmits the same frame again.




**(G-237)Fig. 3.7.3(b) : Stop and wait ARQ, lost data frame**

**Stop and wait ARQ for lost acknowledgement :**

- Fig. 3.7.3(c) shows that if the acknowledgement sent by the receiver is lost, the transmitter retransmits the same data frame after its timer goes off.




(G-238) **Fig. 3.7.3(c) : Stop and wait ARQ, lost ACK frame**

- Stop and wait ARQ protocol becomes inefficient when the propagation delay is much greater than the time to transmit a frame. e.g. let us assume that we are transmitting frames that are 800 bits long over a channel that has a speed of 1 Mbps and let us also assume that the time taken for transmission of the frame and its acknowledgement is 30 ms.
- The number of bits that can be transmitted over this channel in 30 ms is equal to $30 \times 10^{-3} \times 1 \times 10^{6}$ = 30,000 bits. But in the stop-and-wait ARQ only 800 bits can be transmitted in this time period.
- This inefficiency is due to the fact that in stop and wait ARQ the transmitter waits, for an acknowledgement from the receiver before sending the next frame.
- The product of the bit rate and the delay that elapses before an action can take place is called the Delay-bandwidth product.
- The Delay-bandwidth product helps in measuring the lost opportunity in terms of transmitted bits.

**Note :** Stop-and Wait ARQ was used in IBM's Binary Synchronous Communications (Bisync) Protocol. It is also used in Xmodem, a popular file transfer protocol for modem.

**Disadvantages of stop and wait protocol :**

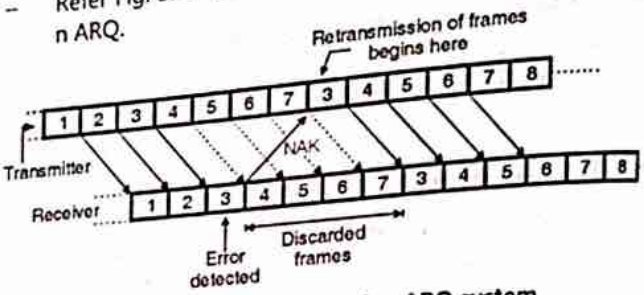
1. Problem with Stop-and-Wait protocol is that it is very inefficient. At any one moment, only one frame is in transition.
2. The sender will have to wait at least one round trip time before sending next. The waiting can be long for a slow network such as satellite link.

## 3.7.2 A Protocol using GO Back n :

- In this stop and wait protocol it was assumed that the transmission time required for a frame to arrive at the receiver plus the transmission time for the acknowledgement to come back is negligible.
- But in some practical situations, this assumption is not correct. In the systems like satellite system the round trip time can be as long as 500 ms (propagation delay).
- This will reduce the efficiency of the protocol.
- Therefore an improved protocol known as **GO-Back-n ARQ** has been developed.
- It is a method used to overcome the inefficiency of the stop and wait ARQ by allowing the transmitter to continue sending enough frames so that the channel is kept busy while the transmitter waits for acknowledgements.
- In this method if one frame is damaged or lost, all frames are sent since the last frame acknowledged are retransmitted.

**Principle of GO-back-n ARQ :**

- Refer Fig. 3.7.4 to understand the principle of GO-Back-n ARQ.




(G-239) **Fig. 3.7.4 : Go back n ARQ system**

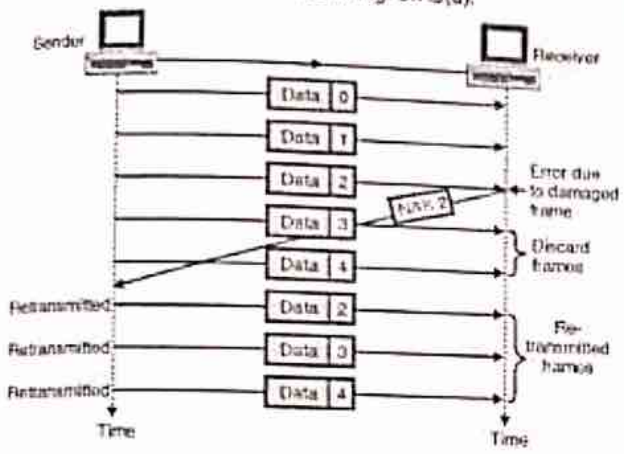
- The major difference between this and the previous system is that the sender does not wait for ACK signal for the transmission of next frame.
- It transmits the frames continuously as long as it does not receive the "NAK" signal. NAK is the negative acknowledgement signal sent by the receiver to the transmitter.
- When the receiver detects an error in the third frame as shown in Fig. 3.7.4, the receiver sends a NAK signal back to sender.
- But this signal takes some time to reach the transmitter. By that time the transmitter has transmitted frames up to frame 7.
- On reception of the NAK signal, the transmitter will retransmit all the frames from 3 onwards.
- The receiver discards all the frames it has received after 3 i.e. 3 to 7. It will then receive all the frames that are retransmitted by the transmitter.

**Sources of error :**

- The errors can get introduced, if the transmitted frames are damaged or lost or if the acknowledgement is lost.
- Let us consider the operation of this protocol under these conditions.

**Operation when the frame is lost :**

- This condition is illustrated in Fig. 3.7.5(a).

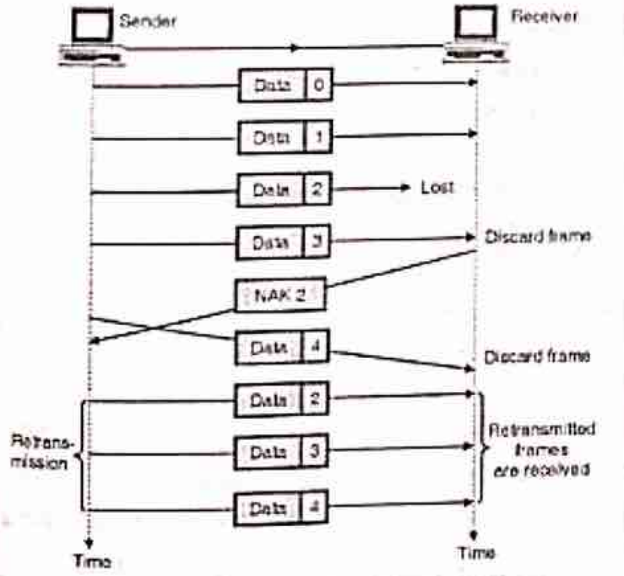



(G-240) **Fig. 3.7.5(a) : Go-back-n, damaged data frame**

- The second data frame is damaged, so the error is detected and receiver send NAK-2 signal back.
- On receiving this signal, the transmitter starts retransmission from frame 2. All the frames received after frame 2 are discarded by the receiver.

**Operation when a frame is lost :**

- As shown in Fig. 3.7.5(b) the case of lost frame is also treated in the same manner as that of the damaged frame.

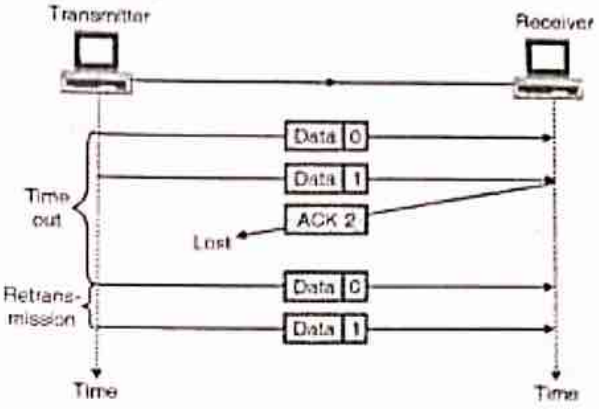



(G-241) **Fig. 3.7.5(b) : Go-back-n, lost data frame**

- The receiver, if it does not receive a particular data frame it sends a NAK to the transmitter and the transmitter retransmits all the frames sent since the last frame acknowledged.

**Operation when the acknowledgement is lost :**

- Fig. 3.7.5(c) shows the condition for lost acknowledgement.




(G-242) **Fig. 3.7.5(c) : Go-back-n, lost ACK frame**

- In case of go-back-n method the transmitter does not expect an acknowledgement after every data frame.
- It cannot use the absence of sequential ACK numbers to identify lost ACK or NAK frames, instead it uses a timer.
- The transmitter can send as many frames as the window allows before waiting for an acknowledgement.
- Once the limit has been reached or the transmitter has no more frames to transmit it must wait till the timer goes off and retransmit all the data frames again.
- The disadvantage of Go-back-n ARQ protocol is that in noisy channels it has poor efficiency because of the need to retransmit the frame in error and all the subsequent frames.

**Disadvantages of Go back n :**

1. It transmits all the frames if one frame is damaged or lost.
2. It transmits frames continuously as long as it does not receive the NAK signal.
3. The NAK signal takes some time to reach the sender. Till that time the sender has already sent some frames. All those will be retransmitted after receiving the NAK.
4. The error can get introduced if the NAK is lost.

## 3.7.3 Pipelining :

- In networking a new task is often started before the previous task has been completed. This is called pipelining.

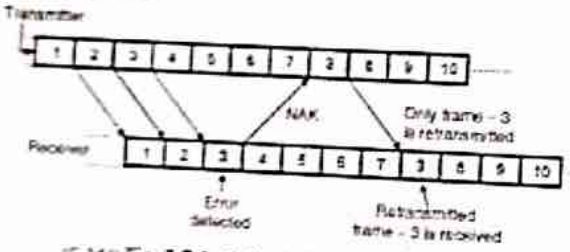
- The principle of pipelining is not used in stop-and-wait ARQ but it is used in GO-Back-n ARQ and the selective repeat ARQ. Pipelining improves the efficiency of transmission.

### 3.7.4 Selective Repeat ARQ :

- In this method only the specified damaged or lost frame is retransmitted.
- A selective repeat systems differs from the go-back-n method in the following ways :
  1. The receiver can do sorting of data frames and is also able to store frames received after it has sent the NAK until the damaged frame has been replaced.
  2. The transmitter has a searching mechanism that allows it to choose only those frames which are requested for retransmission.
  3. The window size in this method is less than or equal to (n + 1)/2, whereas in case of go-back-n it is n – 1.

**Principle :**

- The principle of operation of this protocol is illustrated in Fig. 3.7.6.




(G-24) **Fig. 3.7.6 : Selective repeat ARQ system**

- In this system as well, the transmitter does not wait for the ACK signal for the transmission of the next frame.
- It transmits the frames continuously till it receives the "NAK" signal from the receiver.
- The receiver sends the "NAK" signal back to the transmitter as soon as it detects an error in the received frame.
- For example the receiver detects an error in the third frame, as shown in Fig. 3.7.6. By the time this "NAK" signal reaches the transmitter, it had transmitted the frames up to 7 as shown in Fig. 3.7.6.
- On reception of "NAK" signal, the transmitter will retransmit only the frame-3 and then continues with the sequence 8, 9... as shown in Fig. 3.7.6.
- The frames 4, 5, 6 and 7 received by the receiver, which do not contain any error are not discarded by the receiver.
- The receiver receives the retransmitted frames in between the regular frames.
- Therefore the receiver will have to maintain the frames sequentially.

**Hence the selective repeat ARQ is the most efficient but the most complex protocol, of all the ARQ protocols.**

- Thus in selective repeat ARQ only the frame which is damaged or lost is retransmitted by the transmitter. The lost ACK or NAK frames are treated in the same manner as the go-back-n method.
- When the transmitter reaches either the capacity of its window [(n + 1)/2] or the end of its transmission it sets a timer.
- If no acknowledgement arrives in the allotted time, all the frames that remain unacknowledged are retransmitted.
- The disadvantage of this method is that because of the complexity of sorting and storage required by the receiver and the extra logic needed by the transmitter to select frames for retransmission, the system becomes more expensive.
- The advantage of this system is that it gives the best throughput efficiency. This is due to the use of pipelining in selective repeat ARQ.

## Review Questions

Q. 1 State the various design issues for the data link layer.

Q. 2 What are the different framing methods ?

Q. 3 Explain character stuffing.

Q. 4 What is bit stuffing ?

Q. 5 Write a note on error control.

Q. 6 Explain the simplex protocol for noisy c

Q. 7 What is piggybacking ?

Q. 8 Write a note on sliding window protoco

Q. 9 Explain the stop and wait protocol.

Q. 10 State drawbacks of stop and wait proto

Q. 11 Explain the Go back n protocol.

Q. 12 What is pipelining ?

Q. 13 Write a note on : Selective repeat AR

Unit III

Chapter 4

# Network Layer

## Syllabus

**Introduction :** Functions of network layer. **Switching Techniques :** Circuit switching, Message switching, Packet switching. **Network Routing and Algorithms :** Static routing, Dynamic routing, Distance vector routing, Link state routing, Path vector.

## Chapter Contents

## 4.1 Network Layer :

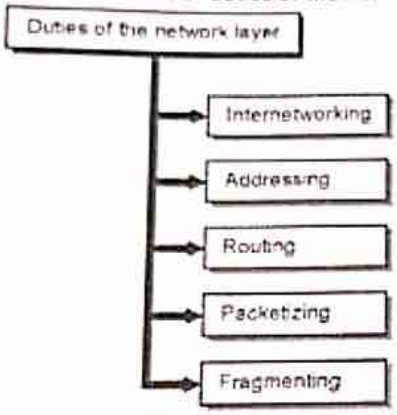
- The network layer is responsible for carrying the packet from the source all the way to destination. In short it is responsible for host-to-host delivery.
- The network layer has a higher responsibility than the data link layer, because the data link layer is only supposed to move the frames from one end of the wire to the other end.
- Thus network layer is the lowest layer that deals with the end to end transmission.

**Position of Network Layer :**

- Network layer is the third layer in the 5-layer internet model.
- It receives services from the data link layer and provides services to the transport layer.

### 4.1.1 Functions of Network Layer :

- Fig. 4.1.1 shows the set of duties of the network layer.




**(G-434) Fig. 4.1.1 : Duties of the network layer**

**1. Internetworking :**

- This is the main duty of network layer. It provides the logical connection between different types of networks.

**2. Addressing :**

- Addressing is necessary to identify each device on the Internet uniquely. This is similar to a telephone system.
- The addresses used in the network layer should be able to uniquely define the connection of a computer to the Internet universally.

**3. Routing :**

- In a network, there are multiple roots available from a source to a destination and one of them is to be chosen.
- The network layer decides which root is to be taken. This is called as routing and it depends on various criterions.

**4. Packetizing :**

- As discussed earlier, the network layer receives the packets from upper layer protocol and encapsulates them to form new packets.
- This is called as packetizing. A network layer protocol called IP (Internetworking Protocol), does the job of packetizing.

**5. Fragmenting :**

- The sent datagram can travel through different networks. Each router decapsulates the IP datagram from the received frame. Then the datagram is processed and encapsulated in another frame.

**Other Issues :**

- The other issues which are not directly related to the duties of network layer but need to be discussed are
  1. Address resolution.
  2. Multicasting.
  3. Routing protocols.

**Other supporting protocols :**

- The Internetworking Protocol (IP) needs the support of another protocol ICMP or ARP etc. in the network layer.

**How to achieve the goals ?**

- In order to achieve the goals, the network layer must know about the topology of the communication subnet i.e. the set of all routers. It also should choose appropriate paths for communication.
- The routes should be chosen in such a way that overloading of some routers and idle operation of others should be avoided.

## 4.2 Network Layer Design Issues :

- The important network layer design issues include the service provided to the transport layer and the internal design of subnet.
- The network layer has been designed with the following goals :
  1. The services provided should be independent of the underlying technology.
  2. Users of the service need not know about the physical implementation of the network.
- This design goal has great importance because there is a great variety of networks in operation.
- The design of the layer must not disable us from connecting to networks of different technologies.

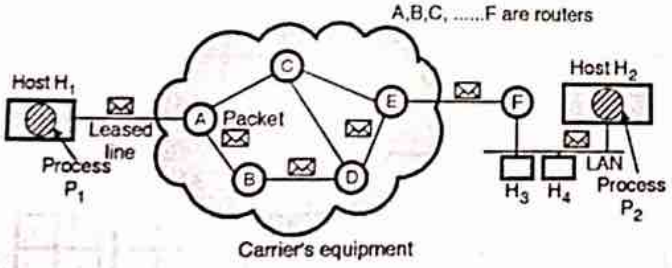
- The transport layer (that is the host computer) should be shielded from the number, type and different topologies of the subnets user uses.
- That is, all that transport layer wants is a communication link, it need not know how that link is established.
- Finally, there is a need for some uniform addressing scheme for network addresses.

**Types of services :**

- With these goals in mind, two different types of services emerged :
  1. Connection oriented Network Services
  2. Connectionless Network Services
- **A connection-oriented service** is one in which the user is given a "reliable" end to end connection.
- To communicate, the user first makes a request for connection, then uses the connection to communicate his content, and then closes the connection.
- A telephone call is the classic example of a connection oriented service.
- In **a connectionless service,** the user simply puts his information into bundles called packets, puts an address on it, and then sends it for the destination.
- There is no guarantee that the bundle will reach the destination. So a connectionless service is one which is similar to the postal system.

### 4.2.1 Store and Forward Packet Switching :

- Refer Fig. 4.2.1 which demonstrates the environment of the network layer protocols.




**(G-435) Fig. 4.2.1 : The environment of the network layer protocols**

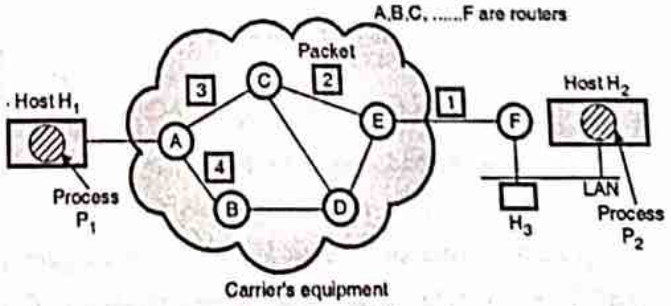
- This system of Fig. 4.2.1 is made up of following components :
  1. Carrier equipments (routers and transmission lines).
  2. Customer's equipments.
- $H_1$ is host – 1 and it is directly connected to router A via a leased line. Host $H_2$ is on a LAN which is connected to router F.
- Host $H_1$ wants to send a packet. So it communicates with its nearest router (A). Router A will store the packet until it has fully arrived so that the checksum can be verified.
- Then the packet is forwarded to the next router (B). This process continues till it reaches the destination host $H_2$.This mechanism is called as the store and forward packet switching.

### 4.2.2 Services Provided to the Transport Layer :

- The network layer services are designed to achieve the following goals :
  1. The services should not be dependent on the subnet technology.
  2. Transport layer should not be exposed to the number, type and topology of the subnet.
  3. The network address which is made available to the transport layer must use a uniform numbering plan.
- The network service can be connectionless or connection oriented.
- The Internet has a connectionless network layer whereas the ATM networks have a connection oriented network layer.
- The connection oriented and connectionless services both have their own sets of advantages and disadvantages. Finally we can say that the network layer should provide a raw means to send packets from a to b and that is all.

### 4.2.3 Implementation of Connectionless Service :

- In the connectionless service, the packets from sending host $H_1$ are injected into the subnet individually and each packet is routed independently as shown in Fig. 4.2.2.

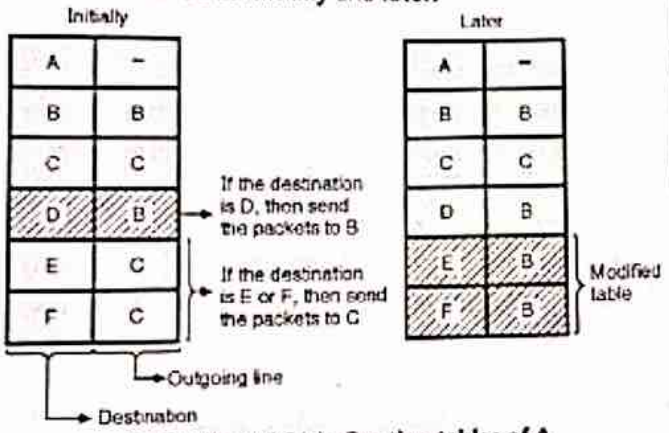



**(G-436) Fig. 4.2.2 : Routing within a datagram subnet**

- No advanced connection establishment is required. The packets are called as **datagrams** and the subnet is called as **datagram subnet.**

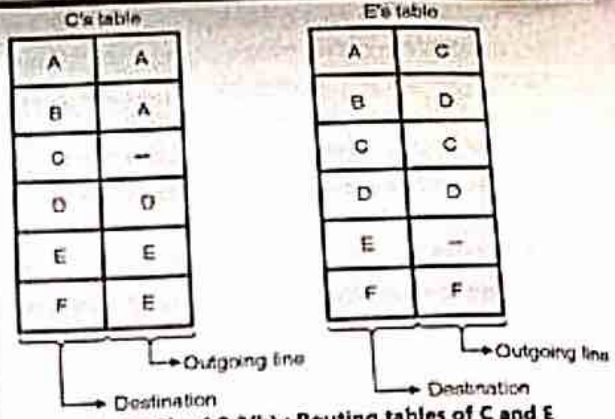
**Working :**

- Process $P_1$ on host $H_1$ wants to send a long message to process $P_2$ on host $H_2$.
- Let this message be broken into four packets 1, 2, 3 and 4 at the network layer.
- Then all these packets are sent to router A. Every router has its internal table which tells it where to send packets for each possible destination.
- Each entry in the router's table is a pair that consists of a destination and the outgoing line to be used to send the packet for that destination. In Fig. 4.2.2, C has two outgoing lines E and D.
- So every packet coming to router C should be sent to either D or E, even if the ultimate destination is F. Fig. 4.2.3(a) shows the routing table of A. It has two tables named as **initially** and **later.**




(G-437) **Fig. 4.2.3(a) : Routing tables of A**

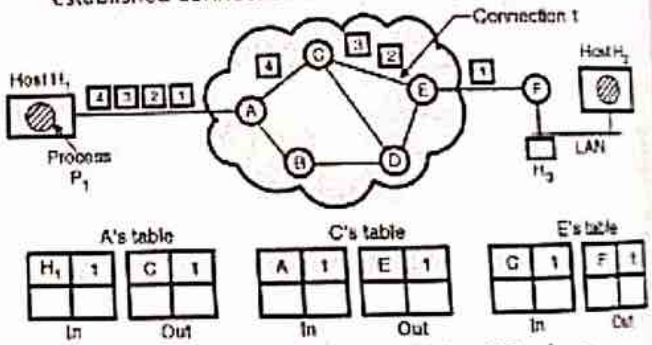
- As per the **initial** routing table of A, since the destination is F the packets 1, 2 and 3 were first sent to C, then to E and finally to F.
- But when packet 4 arrived at the input of A, even though the destination was F, the packet was not sent to C instead it was sent to B.
- The reason can be a traffic jam along the ACE path.
- As soon as A learned about the traffic jam along the ACE path it modified its routing table as shown in Fig. 4.2.3(a) as later and routed the 4th packet via path ABDEF. Fig. 4.2.3(b) shows the routing tables for C and E.




(G-438) **Fig. 4.2.3(b) : Routing tables of C and E**

## 4.2.4 Implementation of Connection-Oriented Service :

- For the connection oriented service, a path from source to destination needs to be established before sending any data packet.
- This connection is called as **Virtual Circuit (VC)** and the subnet is called as the **Virtual Circuit Subnet.** Here all the packets will follow the same path which was established before communication.
- When the connection is opened, the virtual circuit is also terminated. In the connection oriented service each packet carries an identifier.
- This identifier can tell us about the virtual circuit (VC) that this packet belongs to. Refer Fig. 4.2.4. Host $H_1$ has established connection 1 with host $H_2$.




(G-439) **Fig. 4.2.4 : Routing within a VC subnet**

- This connection is remembered as the first entry in each routing table.
- As shown in Fig. 4.2.4, the first line of A's table shows that if a packet having connection identifier 1 arrives from $H_1$, it should be routed to C and a connection identifier 1 should be given to it.
- Similarly the first line of C's table shows that it routes the packets to E with an identifier 1.

### 4.2.5 Internal Organization of the Network Layer :

- Basically there are two philosophies for organizing the subnet :
  1. To use connection oriented service.
  2. To use connectionless service.
- In the connection oriented service, a connection is called as **virtual circuit.** It is similar to a physical connection between the sender host and the destination host.
- In the connectionless organization, the independent packets are called as **datagrams.** They are analogous to telegrams.

**Virtual circuits :**

- The principle behind the virtual circuits is to choose only one route from source to destination.
- When a connection is established, it is used for sending all the traffic over this connection. When the connection is released, the virtual circuit is terminated.

**Datagram :**

- With a datagram, the routes from source to destination are not decided in advance. Each packet sent is routed independently.
- Different packets of the same message can follow different routes.
- The datagram subnets have to do more work but they are more robust and deal with failures and congestion more easily as compared to virtual circuit subnets.

**Features of virtual circuits :**

- In virtual circuits every router will have to maintain and update a table. Each packet must have a virtual circuit number field in its header in addition to sequence number checksum etc.
- It is necessary to setup a VC before communication. The users are charged for connect time as well as for the amount of data transported.

**Features of a datagram :**

- The routers do not have to maintain any tables. Each datagram must contain full destination address. These addresses can be very long.
- When a packet comes in, the router finds an available outgoing line and sends the packet out on that line, so that it can reach the destination.

### 4.2.6 Comparison of Virtual Circuit and Datagram Subnets :

- Table 4.2.1 shows the comparison of VC subnet and datagram subnets.

**Table 4.2.1 : Comparison of VC and Datagram Subnets**

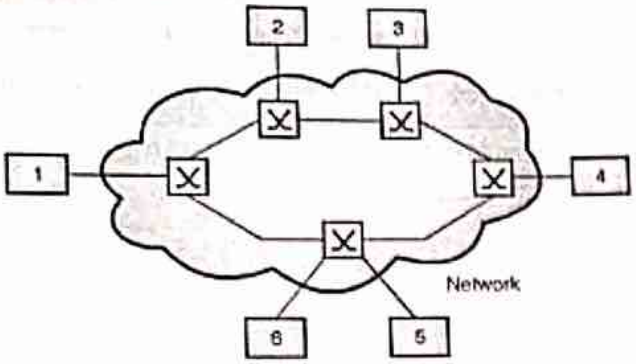
| Sr. No. | Parameter | VC subnet | Datagram subnet |
|---|---|---|---|
| 1. | Connection set up | Required | Not required. |
| 2. | Addressing | Each packet contains a short VC number | Each packet contains the source as well as destination address. |
| 3. | Repairs | Harder to repair | Easy to repair. |
| 4. | State information | A table is needed to hold the state information. | Subnet does not hold state information. |
| 5. | Routing | Route chosen is fixed. All packets follow this route. This is static routing. | Each packet is routed independently. This is dynamic routing. |
| 6. | Congestion control | Easy | Difficult. |
| 7. | Effect of router failure. | All VCs which passed through failed router are terminated. | No other effect except for the packets lost at the time of crash. |

## 4.3 Switching :

- A network consists of many switching devices. In order to connect multiple devices, one solution could be to have a point to point connection between each pair of devices. But this increases the number of connections.
- The other solution could be to have a central device and connect every device to each other via the central device (Star topology).
- Both these methods are wasteful and impractical for very large networks. The other topologies also cannot be used.
- Hence a better solution is **switching.** A switched network is made of a series of interconnected nodes called switches.

**Definition of a switch :**

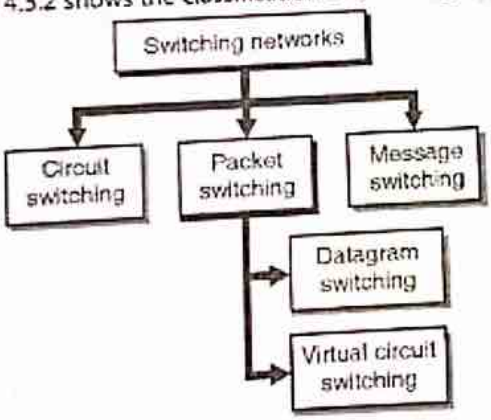
- Switch is a device that creates temporary connections between two or more devices. Fig. 4.3.1 shows a switched network.




**(L-616) Fig. 4.3.1 : Switched network**

## 4.3.1 Switching Methods :

- The three basic methods of switching are :
  1. Circuit switching
  2. Packet switching
  3. Message switching
- Out of these, the circuit and packet switching are commonly used today but the message switching has been phased out in general communication but is still used in the networking applications.
- Fig. 4.3.2 shows the classification of switching methods.




**(L-617) Fig. 4.3.2 : Classification of switching methods**

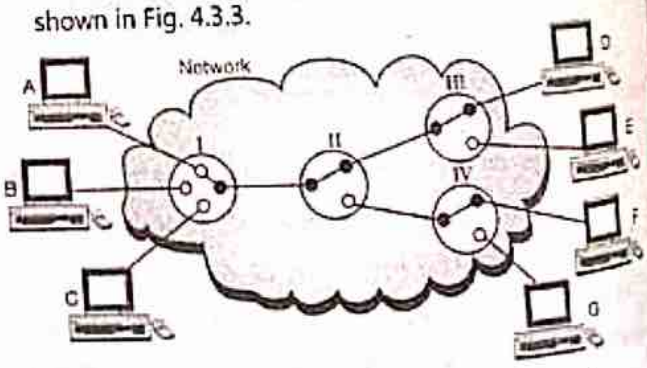
## 4.3.2 Circuit Switching :

**Concept :**

- Circuit switching is a method of implementing a telecommunications network in which two network nodes establish a dedicated communication channel (Circuit) before the nodes communicate with each other.

**Example :**

- The simplest and the oldest telephone network is a circuit switched network.
- In a telephone network, when a call is made, from one telephone to the other, the switches within the telephone exchanges creates a continuous circuit (wired) between the two telephones as long as the call lasts.
- Circuit switching is used in public telephone networks. It was developed to handle voice traffic but it can also handle digital data.
- However circuit switching cannot handle digital data efficiently. Using the circuit switching, a dedicated path is established between two stations for communication.
- The telephone network provide telephone service which involves the two way, real-time transmission of voice signals across a network.
- The network connection allows electrical current and the associated voice signal to flow between the two users. The end to end connection is maintained for the duration of the call.
- The telephone networks are connection oriented because they require the setting up of a connection before the actual transfer of information can take place.
- The transfer mode of a network that involves setting up a dedicated end to end connection is called circuit switching.
- In circuit switching the routing decision is made when the path is set up across the network.
- After the link has been set between the sender and receiver, the information is forwarded continuously over the link.
- After the link has been set up no additional address information about the receiver or destination machine is required.
- In circuit switching a dedicated path is established between the sender and the receiver which is maintained for the entire duration of conversation, as shown in Fig. 4.3.3.




**(L-618) Fig. 4.3.3 : Circuit-switched network**

- In telephone systems circuit switching is used. If circuit switching is used in computer networks the sending machine has to first establishes a link with the receiving machine.
- After the link is established the data is transmitted from the sender to the receiver. After the data flow stops, the link is released.

**Block diagram :**

- In Fig. 4.3.3, I, II, III and IV are called as the switching nodes. They are used to connect one user to the other.
- The circuit switched networks operate in three phases :
  1. Set up phase.
  2. Data transfer phase.
  3. Tear down phase.
- The circuit switching corresponds to the physical layer.
- Before starting communication in the setup phase the resources are reserved during communication. Some of these resources are channels, switch buffers, input / output ports etc.
- Data transferred between two stations is not in the packet form instead the data gets transferred continuously.
- No addressing is involved during the data transfer as the dedicated connection is established between the sender and receiver.
- The switches route the data on the basis of the allotted frequency band (FDM) or allotted time slot (TDM).

**Three Phases :**

- Communication via circuit switching takes place over three phases of operation as follows :
  1. Circuit establishment
  2. Data transfer
  3. Circuit disconnect (tear down).

**1. Circuit establishment :**

- In a circuit switching network, before any signal is transmitted, it is necessary to establish an end-to-end (station to station) link.
- For example, in Fig. 4.3.3, if the communication is to be between A and D, then the path from A to node I to node II to node III and D has to be established first.
- The node to node links are usually multiplexed. They either use FDM or TDM.

**2. Data transfer :**

- The information can now be transferred from A to D through the network. The data can be analog or digital depending on the nature of network.
- Generally all the internal connections are duplex.

**3. Circuit disconnect (tear down phase) :**

- After some time the connection between two users is terminated usually by the action of one or two stations. Circuit switching is inefficient in most of the applications.
- The entire channel capacity is dedicated for the duration of connection, even if the data is not being transferred. Once the circuit is established, the network is effectively transparent to the users with no delays involved.

**Efficiency :**

- In circuit switching the resources remain dedicated as long as a connection is alive.
- Due to the allocation of resources during the entire duration of the connection, the efficiency of circuit switched networks is lower than the other two types of switching.

**Delay :**

- Even though the efficiency is low, the delay in this type of networks is very small.

**Features :**

- Some of the important feature of circuit switched networks are as follows :
  1. Two nodes are connected to each other over a dedicated communication channel (circuit).
  2. Switches are used to make or break the dedicated circuit.
  3. Information or data is transferred continuously.
  4. Circuit switching is preferred for voice communication.
  5. No addressing needed during the data transfer phase.
  6. Efficiency is low.
  7. Delays are small.

**Application :**

- The circuit switching is used in the telephone networks.

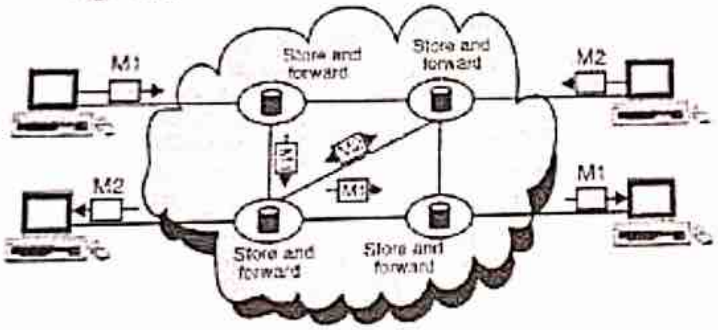
## 4.3.3 Circuit Switched Technology in Telephone Networks :

- The telephone companies previously used the circuit switching technology for switching and routing a call. This was a physical layer technology.

- However, today the tendency is to use other switching techniques.
- For example the telephone number is used as the global address and a signalling system (called SS7) is used for creating and disconnecting the connections.

## 4.4 Message Switching :

- In telegraphy the text message is encoded using the Morse code into sequences of dots and dashes. Each dot or dash is communicated by transmitting short and long pulses of electrical current over a copper wire.
- In telegraph networks the text message is transmitted from the source telegraph office to the telegraph switching station. At this switching station an operator takes the decision of routing the message based on the destination address information. The operator will either forward the message if a communication line to the destination is free or store the message still the communication line becomes free.
- Message switching does not establish a dedicated path between two communicating devices. In message switching, each message is treated as an independent unit and includes its own destination and source address.
- Each complete message is then transmitted from device to device through the internetwork as shown in Fig. 4.4.1.




(L-620) **Fig. 4.4.1 : Message switching**

- Each intermediate device receives the message, stores it, until the next device is ready to receive it and then forwards it to the next device. For this reason, a message switching network is sometimes called as a store and forward network.
- Messages switches can be programmed with information about the most efficient routes as well as information regarding neighbouring switches that can be used to forward messages to their ultimate destination.

### 4.4.1 Advantages :

1. It provides efficient traffic management by assigning priorities to the messages to be switched.
2. It reduces network traffic congestion because it is able to store message until a communication channel becomes available.
3. With message switching, the network devices share the data channels.
4. It provides asynchronous communication across time zones.

### 4.4.2 Disadvantages :

1. The storing and forwarding introduces delay hence cannot be used for real time applications like voice and video.
2. The intermediate devices require a large storing capacity since it has to store the message unless a free path is available.

## 4.5 Packet Switching :

**Definition :**

- Packet switching is a method of switching in which the data (to be sent) is transmitted over a digital network in the form of **packets.**

**Packet :**

- A packet is made of two parts : header and payload or data as shown in Fig. 4.5.1.

| Header | Payload (Data) |
|---|---|

(L-913) **Fig. 4.5.1 : A packet**

- The networking hardware uses the header contents to direct the packet to its destination. Packet switching is extensively used for data communication in computer networks.

**Principle :**

- In packet switching, messages are broken up into packets.
- Each packets has a header with source, destination and intermediate node address information.
- The other part of the packet includes data load.
- Individual packets can take different routes to reach the destination. Independent routing of packets gives two advantages :
  1. Bandwidth is reduced due to splitting data onto different routes in a busy circuit.

2. If a certain link in the network goes down during the transmission, the remaining packets can be sent through another route.

- The packets can arrive out of order at the receiver and have to be reassembled in proper sequence.
- In packet switching, the packet length is restricted to a certain maximum length.
- This length is short enough to allow the switching devices to store the packet data in memory.
- There are two methods of packet switching :
  1. Datagram packet switching
  2. Virtual circuit packet switching.

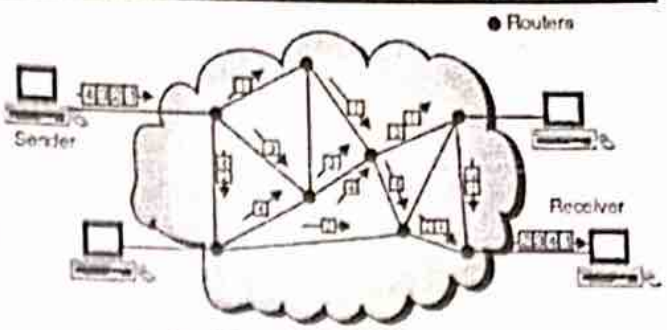
### 4.5.1 Datagram Packet Switching :

**Principle :**

- In this method a message is divided into a stream of packets. Each packet has its individually included address and treated as an independent unit with its own control instructors.
- The switching devices would route each packet independently through the network. Each intermediate node will determine the packet's next route segment.
- Before transmission starts, the sequence of packets and their destinations are communicated by exchanging control information between the sending terminal, the network and the receiving terminal.
- In packet switching, the resources are not allocated for any packet so there is no reserved bandwidth and no scheduled processing time allotted for each packet.
- No dedicated connection is established between the sender and receiver.
- The resource allocation is on demand and on the first come first serve basis.
- When a switch receives a packet, it has to wait if there are any other packets being processed. This will increase the delay.
- The datagram packet switching generally corresponds to the network layer. The packet are called as **datagrams.**

**Schematic diagram :**

- Datagram packet switching is shown in Fig. 4.5.2.
- In this circuit, four packets are to be delivered from the sender to receiver. The switches in the datagram network are called as **routers.**




**(L-623) Fig. 4.5.2 : Datagram packet switching**

- All the four packets (datagrams) belong to the same message in this circuit however actually they can get originated from any computer.
- The four datagrams, as shown in Fig. 4.5.2 may travel different paths to reach the destination.
- Due to this the packets may arrive out of order at the destination.
- The delay associated with each packet will be different as a result of the different paths followed by them. The datagrams may get lost or dropped out due to lack of resources.
- The upper layer protocols are supposed to reorder the received datagrams or ask for the lost ones before passing them on the application.
- The datagram networks, are called as the **connectionless** networks. This is because the switch (packet switch) does not keep any information about the connection state.
- There are no connection set up or tear down processes in the packet switching networks.

### 4.5.2 Efficiency :

- As the resources are allocated only when the packets are to be transferred, the efficiency of datagram network is **higher** than that of the circuit switched network.

### 4.5.3 Delay :

- There are no set up or tear down phases in datagram circuit switching but each packet may have to wait at a switch before getting forwarded.
- All the packets in a message take different paths. Hence the delay associated with each packet is different.

### 4.5.4 Advantages of Packet Switching :

1. Greater line utilization efficiency, as a single node-to-node link can be dynamically shared by many packets over time.

2. A packet switching network can perform data-rate conversion.
3. When traffic become heavy on circuit switching network, some calls are blocked. On a packet switching network, packets are still accepted, but delivery delay increases.
4. Priorities can be used.
5. Each terminal in group sharing the same physical circuit may be connected to a totally different destination. This versatility is one of the major strengths of packet switching.
6. No single user or large data block can tie up circuit or node resources indefinitely, making it well suited for interactive traffic.
7. Data protection against corruption or loss, errors are corrected by retransmission.
8. Users can select different destinations for each virtual call, overcoming the inflexibility of point to point dedicated networks.
9. Simultaneous calls allow PC users to access multiple windows to different remote applications.
10. Since many users can share transmission resources efficiently, the cost of intermittent data communication is reduced.
11. New calls can be added and old ones disconnected without affecting other users.

### 4.5.5 Disadvantages of Packet Switching :

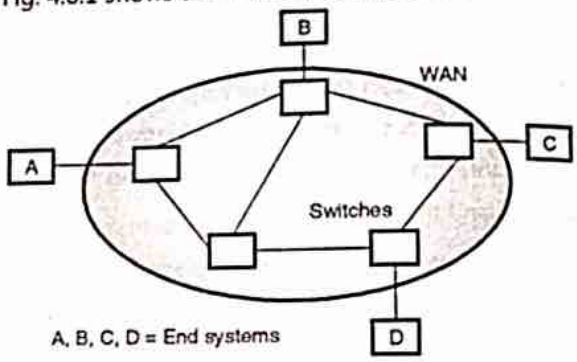
1. Delays are longer than those in circuit switching.
2. Header overhead reduces capacity to carry user data.
3. More processing is required at node.

### 4.5.6 Datagram Networks in Internet :

- The internet uses the datagram approach to switching at the Network layer.
- The routing of packets in Internet takes place on the basis of the universal addresses defined in the network layer.

## 4.6 Virtual Circuit Packet Switching :

- It establishes a logical connection between the sending and receiving devices called virtual circuit.
- The sending device and receiving device agree upon some important communication parameters, such as maximum message size and the network path to be taken.
- Once this virtual circuit is established the two devices use it for the rest of the conversation.
- In virtual circuit packet switching, all the packets travel through the virtual circuit established between the sending device and the receiving device.
- Virtual circuit switching has some characteristics of both circuit switched network and a datagram network.
- Similar to circuit switched network, there are setup and tear down phases along with the data transfer phase.
- It is possible to allocate the resources either in the set up phase similar to the circuit switched networks or as per requirement similar to the datagram networks.
- Similar to datagram networks, the data is sent in the form of packets. Each packet carries the address of the next switch and not the final destination address.
- Similar to circuit switching networks all the packets follow the same path established during the set up phase. That means packets don't take different paths to arrive at the destination.
- Virtual circuit corresponds to the data link layer. Fig. 4.6.1 shows the virtual circuit network.




(L-626) **Fig. 4.6.1 : Virtual circuit network**

- The network consists of switches which route the traffic from source to destination.

### 4.6.1 Three Phases of Communication :

- A source and destination have to undergo three phases to communicate between each other.
- The three phases are :

  1. Set up 2. Data transfer 3. Teardown

**Set up phase :**

- In this phase the source and destination use their global addresses. This will help switches to make table entries for the connection.

**Data transfer :**

- The data transfer is the second phase in which the frames are transferred from source to destination.

**Teardown :**

- In the teardown phase both source and destination will communicate the switches to erase the corresponding entry.

### 4.6.2 Efficiency :

- In the virtual circuit networks, the resources can be either allocated during the set up phase or they can be allocated on demand during the data transfer phase.
- Even though resources are allocated on demand, it is possible for the source to check the availability of resources, without actually reserving them.
- This is a big advantage as it saves a lot of time and effect. This increases the efficiency of the virtual circuit network.

### 4.6.3 Delay :

- There are different delays present in virtual circuit networks.
- One component of delay is the time delay for set up phase. The other component is the time delay corresponding to the tear down phase.
- If the resources are allocated during the set up phase, then there is no waiting delays for individual packets.
- The virtual circuit networks are used in switched WANs such as Frame Relay and ATM networks.

### 4.6.4 Advantages of Virtual Circuit Packet Switching :

1. Virtual circuit packet switching uses abbreviated headers and hardware based table lookup, which allows fast processing and forwarding of packets.
2. In the virtual circuit packet switching, resources can be allocated during connection setup.
3. The number of bit required in the header is much smaller than the number required to provide full destination network addresses. This reduces the wastage of transmission bandwidth.
4. Virtual circuit packet switching uses Virtual-Circuit Identifier (VCI) which uses to identify connection and to specify the type of priority given to the packet by scheduler that controls the transmissions in next output port.
5. The efficiency of virtual circuit packet switching is high.

### 4.6.5 Disadvantages of Virtual Circuit Packet Switching :

1. The switches in the network need to maintain information about the flows they are handling. The amount of required 'state' information grows very quickly with number of flows.
2. In the case of fault occurs in the network, all affected connections must be set up again.
3. Connection setup is not possible, if the switch is unable to handle the volume of traffic allowed or link utilization exceeds certain thresholds.

### 4.6.6 Comparison of Switching Techniques :

**Table 4.6.1 : Comparison of Switching Techniques**

| Parameter | Message switching | Circuit switching | Packet switching |
|---|---|---|---|
| Application | Telegraph network for transmission of telegrams. | Telephone network for bi-directional, real time transfer of voice signals. | Internet for datagram and reliable stream service between computers. |
| End terminal | Telegraph, teletype. | Telephone, modem. | Computer |
| Information type | Data in the form of Morse, Baudot, ASCII codes. | Analog voice or PCM digital voice | Binary information |
| Transmission system | Digital data over different transmission media | Analog and digital data over different transmission media | Digital data over different transmission media. |
| Addressing scheme | Geographical addresses | Hierarchical numbering plan | Hierarchical address space |
| Routing scheme | Manual | Route selected during call setup. | Each packet is routed independently. |
| Multiplexing scheme | Character or message multiplexing | Circuit multiplexing. | Packet multiplexing shared media access networks. |

## Routing in Packet Switched Network :

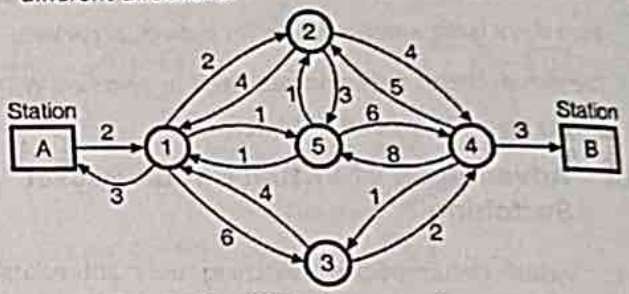
- Routing is one of the most complex and important design aspects of switched data networks.
- Some of the important characteristics that are used for classification of routing strategies are as given below :

### 4.7.1 Characteristics :

- A packet switched network has a prime function of accepting packets from a source and deliver them to its destination.
- To accomplish this, a path or route through the network should be determined.
- Generally more than one paths are available. Hence the **routing function** needs to be performed.
- The requirements of the **routing function** are as follows :
  1. Correctness
  2. Simplicity
  3. Robustness
  4. Efficiency
  5. Fairness
  6. Stability
  7. Optimality
- Correctness and simplicity are self explanatory and do not need any explanation.
- **Robustness** is defined as the ability of the network to deliver a packet via some alternative route if a route fails due to local reasons, or network overloads.
- In order to ensure **robustness,** the network must react properly to any such situation in time.
- Unfortunately networks are sometimes too slow to respond and sometimes too quick to respond posing the **stability** problem.
- Therefore a **trade off** exists between the robustness and stability.
- Similarly a **trade off** exists between fairness and optimality.
- If higher priority is given to exchange of packets between nearby stations then it may maximize the throughput but will be unfair to the stations that are involved in a more distant communication.
- Due to processing overheads at node alongwith a transmission overhead, the **efficiency** is reduced.
- Keeping all these requirements in mind, we can now access various design elements that contribute to a **routing strategy.**

### 4.7.2 Performance Criteria :

- Some performance criteria required for the selection of a route are as follows :
  1. Number of hops
  2. Cost
  3. Delay
  4. Throughput
- The simplest criteria is to choose a path with **minimum number of hops** i.e. the route that passes through the least number of nodes.
- The minimum hop criteria minimizes the consumption of network resources and it is generalized as **Least Cost Routing.**
- A **cost** is attached to each link as shown in the example network of Fig. 4.7.1. In the least cost routing a path between two communicating stations that has the **smallest** cost is selected.
- The **cost** associated with a link can be different for different directions.




(G-2504) **Fig. 4.7.1 : An example network**

- The **shortest path** i.e. the path with minimum hops is (A-1-2-4-B) or (A-1-5-4-B) or (A-1-3-4-B) but the **least cost path** is (A-1-2-4-B) with a cost of (2 + 2 + 4 + 3 = 11).
- The **cost** of a link depends on :
  1. The data rate supported by it.
  2. The delay introduced by it.
- A link with higher data rate or lower delay is said to have a **lower cost.**
- Many least cost algorithms are available. We have discussed some of them in this chapter.

## 4.8 Routing :

- Routing is a very important issue in the network layer.

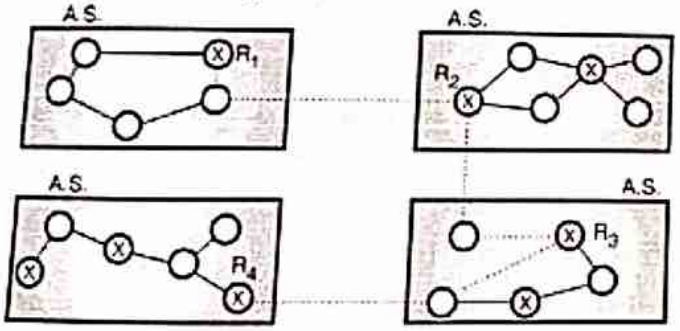
- A router creates its routing table so as to help forwarding a datagram in the connectionless services. It also helps in creating a virtual circuit in the connection oriented service.
- In the following sections we are going to discuss about the types of routing and different routing algorithms such as distance vector routing, link state routing and hierarchical routing.

### 4.8.1 Types of Routing :

- Routing can be broadly classified into three types :
  1. Unicast routing.
  2. Broadcast routing
  3. Multicast routing.
- We can also classify the routing into two types as follows :
  1. Intradomain routing.
  2. Interdomain routing.

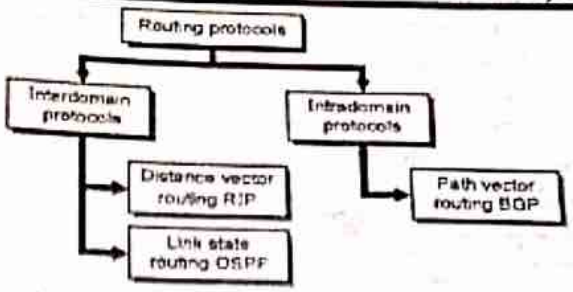
### 4.8.2 Intra and Interdomain Routing :

- Today the size of the Internet is so big that one routing protocol cannot handle the task of updating the routing tables of all the routers.
- Hence an internet is divided into **Autonomous Systems** (AS).
- An Autonomous System (AS) is a group of networks and routers which is controlled by a single administrator. An AS is shown in Fig. 4.8.1.




(G-1292) **Fig. 4.8.1 : Autonomous systems**

- The **Intradomain routing** is defined as the routing inside an autonomous system whereas the routing between autonomous system is known as the **Interdomain routing.**
- Several intradomain and interdomain protocols are used. They are as shown in Fig. 4.8.2.




(G-1291) **Fig. 4.8.2 : Classification of routing protocols**

- The examples of interdomain routing protocols are :
  1. Distance vector routing 2. Link state routing.
- An example of intradomain routing protocol is path vector routing. Each A.S. is allowed to choose one or more intradomain routing protocols in order to handle the routing inside the A.S.
- But only one interdomain routing protocol will handle routing between autonomous systems. The Routing Information Protocol (RIP) is an implementation of distance vector routing.
- Whereas the OSPF is an implementation of link state protocol. The BGP is an implementation of the path vector protocol.

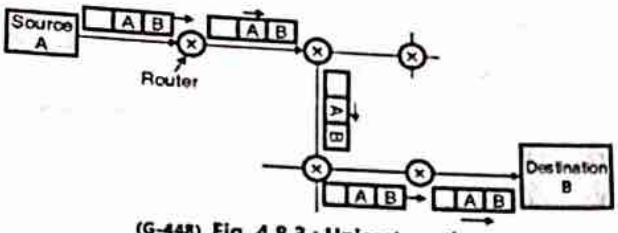
**Difference between Intra and Interdomain routing :**

- In routing, the problems of scale and administrative autonomy can be solved by organizing routers into Automatic Systems (AS).
- Each AS consists of a group of routers that are under the same administrative control. Routers within the same AS run the same routing algorithm such as Link state or Distance Vector algorithms.
- The routing algorithm running within an AS is called as intra-autonomous system protocol or intra-domain routing protocol. There can be a large number of ASs connected to each other.
- The task of inter-connecting them is handled by the inter-autonomous system protocol or inter domain routing protocol.
- The example of intradomain routing protocol is RIP. Refer section 5.21 for RIP. The example of interdomain routing protocol is OSPF. Refer section 5.22 for OSPF.

### 4.8.3 Unicast Routing :

- In unicast routing there is a one to one relation between the source and the destination.

- That means only one source sends packets to only one destination.
- The type of source and destination addresses included in the IP datagram are unicast addresses assigned to the hosts. The concept of unicast routing is illustrated in Fig. 4.8.3.




(G-448) **Fig. 4.8.3 : Unicast routing**

- In unicast routing when a router receives a packet, it forwards that packet through only one of its ports which corresponds to the optimum path.
- The router can discard the packet if it cannot find the destination address.

**Metric :**

- A metric is defined as the cost assigned for passing through a network.
- The metric assigned to each network depends on the type of protocol.

**Interior and exterior routing :**

- An Internet is so large that for one routing protocol it is impossible to handle the task of updating the routing tables of all the routers.
- So an Internet is divided into a number of Autonomous Systems (AS). An AS is group of networks and routers.

**Interior routing :**

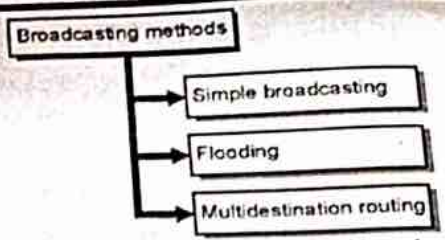
- The routing that takes place inside an AS is called as interior routing.

**Exterior routing :**

- The routing that takes place among various autonomous systems is called as exterior routing.

## 4.8.4 Broadcast Routing :

- In certain applications, the host has to send packets to many or all other hosts. If the sender sends a packet to all destinations simultaneously then it is called as **broadcasting.**
- Various methods of broadcasting are as shown in Fig. 4.8.4.




(G-449) **Fig. 4.8.4 : Various methods of broadcasting**

1. **Simple broadcasting :**

- In this method the source will simply send a distinct (a separate) packet to each destination. This method has two drawbacks :
  1. A lot of bandwidth is wasted.
  2. The source has to have a complete list of all destinations.

2. **Flooding :**

- Flooding is another method used for broadcasting. The problem with flooding is that it has a point to point routing algorithm. So it consumes a lot of bandwidth and generates too many packets.

3. **Multi destination routing :**

- This is the third algorithm used for broadcasting. In this algorithm each packet will contain a list of destinations or a bit map which indicates the desired destination.
- When such a packet arrives at a router, the router first checks all the destinations. Then it decides the set of output lines that will be required based on the destination addresses.
- The router then generates a new copy of the received packet for each output line to be used. It includes a list of only those destinations that are to use the line in each packet going out on that line.
- This will save bandwidth to a great extent. Also generation of too many packets right from the sending end will also be avoided.

## 4.8.5 Multicast Routing :

- In multicasting a message from a sender is to be sent to a group of destinations but not all the destinations in a network. A process has to send a message to all other processes in the group.
- For a small group it is possible to send a point-to-point message. But this is expensive if the group is large. So we have to send messages to a well defined groups which are small compared to the network size.

- Sending message to such a group is called **multicasting** and the routing algorithm used for multicasting is **multicast routing.**
- Multicast routing is a special class of broadcast routing.

## 4.9 Routing Algorithms :

- One of the important functions of the network layer is to route the packets from the source machine to the destination machine.
- The major area of network layer design includes the algorithms which choose the routes and the data structures which are used.
- **Routing algorithm** is a part of network layer software. It is responsible for deciding the output line over which a packet is to be sent.
- Such a decision is dependent on whether the subnet is a virtual circuit or it is datagram switching.

**Design goals for routing algorithms :**

- Various routing algorithms are designed for one or more of the following design goals :
  1. Optimality.
  2. Simplicity and low overheads.
  3. Robustness and stability.
  4. Rapid convergence.
  5. Flexibility.
- Let discuss these design goals one by one.

**1. Optimality :**

- We may define the optimality as the capability of a routing algorithm to select the best possible route, which depends on the metrics and metric weights used to make the calculations.

**2. Simplicity :**

- Routing algorithms are designed to be as simple as possible. That means the routing algorithm should work properly and efficiently with a minimum software and utilization overheads.

**3. Robustness and stability :**

- Routing algorithms should be designed for robustness. That means they should be able to perform correctly in all the unusual or unforeseen circumstances.
- The routing protocols are also supposed to withstand the test of time and prove stable under a variety of network conditions.

**4. Rapid convergence :**

- In addition, routing protocols should converge rapidly. Convergence can be defined as the process of agreement by all the routers, on optimal routes.
- That means in response to the routing update messages, the recalculation of optimal routes should be carried out quickly by all the routers. Routing algorithms that converge slowly can cause rooting loops or network outage.

**5. Flexibility :**

- Routing algorithms should also be flexible. That means they should adapt to different network circumstances quickly and accurately.
- That means in the event of failure of a network segment, different routers should quickly select the next best path for all routes which are using the failed segment.
- It is possible to program the routing algorithms to adapt to the changes in network bandwidth, router queue size and network delays.

### 4.9.1 Desired Properties of a Routing Algorithm :

- There are certain desirable properties of a routing algorithm as follows :
  1. Correctness 2. Robustness
  3. Stability 4. Fairness and 5. Optimality.

### 4.9.2 Types of Routing Algorithms :

- Routing algorithms can be divided into two groups :
  1. Non-adaptive algorithms. 2. Adaptive algorithms.

**1. Non-adaptive algorithms :**

- For this type of algorithms, the routing decision is not based on the measurement or estimation of current traffic and topology.
- However the choice of the route is done in advance, off-line and it is downloaded to the routers. This is called as static routing.

**2. Adaptive algorithms :**

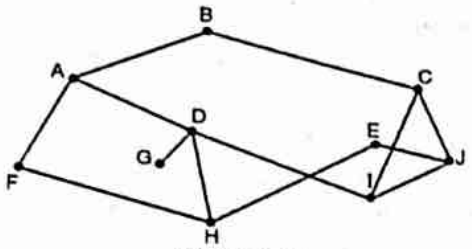
- For these algorithms the routing decision can be changed if there are any changes in topology or traffic etc.
- This is called as dynamic routing. In the following sections we are going to discuss various static and dynamic algorithms.

### 4.9.3 Optimality Principle :

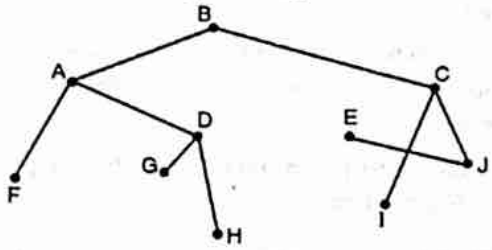
- A general statement about optimality is called as optimality principle. It states that if router J is on the optimal path from router I to router K, then the optimal path from J to K will also be along the same route.

**Sink tree :**

- A set of optimal routes from all the sources to a given destination form a tree called sink tree and it is shown in Fig. 4.9.1.




**(a) A subnet**




**(b) A sink tree for router B**

**(G-450) Fig. 4.9.1**

- The root of the sink tree is at the destination. Note that a sink tree need not be unique. Other trees with the same path lengths may also exist.
- All the routing algorithms are supposed to discover and use the sink trees for all routers.
- In the sink tree of Fig. 4.9.1, the distance metric is the number of hops. In Fig. 4.9.1(b) a sink tree for router B has been shown. The paths from B to every router with minimum number of hops.

## 4.10 Static Routing :

- The examples of static algorithms are :
  1. Shortest path routing.
  2. Flooding.
  3. Flow based routing.

### 4.10.1 Shortest Path Routing :

- This algorithm is based on the simplest and most widely used principle. Here a graph of subnet is prepared in which each node represents either a host or a router and each arc represents a communication link.
- So as to choose a path between any two routers, this algorithm simply finds the shortest path between them.

**How to decide the shortest path ?**

- One way of measuring the path length is the number of hops. Another way (metric) is the geographical distance in kilometres. Some other metrics are also possible.
- For example we can label each arc (link) with the mean queuing and transmission delay and obtain the shortest path as the fastest path.

**Labels on the arcs :**

- The labels on the arcs can be computed as a function of distance bandwidth, average traffic, mean queue length, cost of communication, measured delay etc.
- The algorithm compares various parameters and calculates the shortest path, on the basis of any one or combination of criterions stated above.

**Various shortest path algorithms :**

- There are many algorithms for computing the shortest path between two nodes. One of them is Dijkstra algorithm. The other one is Bellman-Ford algorithm.

### 4.10.2 Flooding :

- This is another static algorithm. In this algorithm every incoming packet is sent out on every outgoing line except the line on which it has arrived.
- That is why the name flooding. Each line except the incoming lines are flooded with the copies of the same packet.
- One disadvantage of flooding is that it generate a large number of duplicate packets. In fact it produces infinite number of duplicate packets unless we somehow stop the process.
- There are various damping techniques such as :
  1. Using a hop counter.
  2. To keep a track of which packets have been flooded.
  3. Selective flooding.
- To prevent endless copies of packets circulating for very long time through the network a hop count may be used to suppress onwards transmission of packets after a number of hops which exceed the network "diameter".
- The other problem is that destination must be prepared to receive multiple copies of an incoming packet. Flooding has two interesting characteristics that arise from the fact that all possible routes are tried :

1. As long as there is a route from source to destination the packet will be definitely delivered to the destination.
2. One copy of the packet will reach the destination via the quickest possible route.

**Selective flooding :**

- This is slightly more practical type of flooding principle. In this algorithm every incoming packet is not sent out on every output line.
- Instead packet is sent only on those lines which are likely to go in the desired direction.

**Applications of flooding :**

- Flooding does not have many practical applications. But it is useful in military applications where a large number of routers are blown into pieces (damaged) at any instant.
- So placing a packet on every outgoing line really makes sense. In such applications robustness of flooding is very much desirable.
- Second application is in the distributed database applications. Flooding always chooses the shortest path so it produces the shortest possible delay.

## 4.11 Dynamic Routing :

- The modern computer networks normally use the dynamic routing algorithms. Two dynamic routing algorithms namely distance vector routing and link state routing are used popularly.
- Both these algorithms are suitable for the packet switched networks. Both these algorithms assume that a router knows the address of each neighbouring router and the cost of reaching each neighbour.
- In the distance vector routing, each node tells its neighbours about its distance to every other node in the network.
- In the link state routing, a node tells every other node in the network the distance to its neighbours.
- So both these routing algorithms are distributed type and so they are suitable for large internetworks.

## 4.12 Distance Vector Routing :

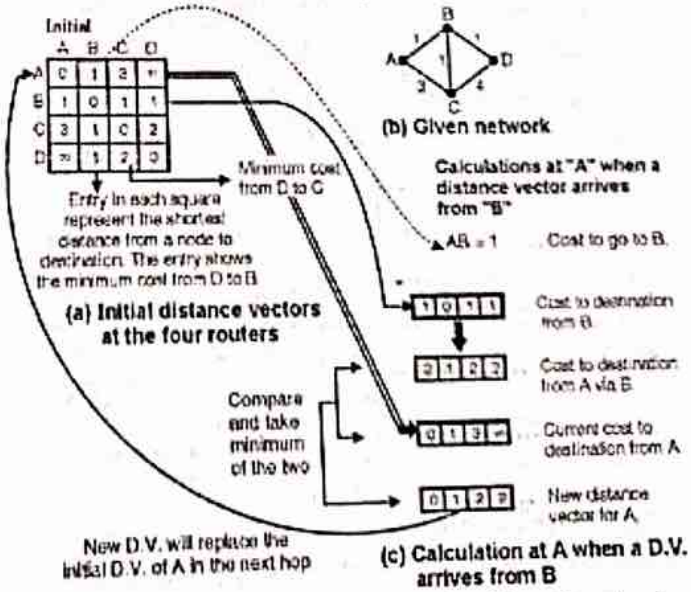
- In this algorithm, each router maintains a table called vector, such a table gives the best known distance to each destination and the information about which line to be used to reach there.
- This algorithm is sometimes called by other names such as :
  1. Distributed Bellman-Ford routing algorithm.
  2. Ford-Fulkerson algorithm
- In distance vector routing, each router maintains a routing table. It contains one entry for each router in the subnet.
- This entry has two parts :
  1. The first part shows the preferred outgoing line to be used to reach the specific destination.
  2. Second part gives an estimate of the time or distance to that destination.

**Distance vector :**

- In distance vector routing, we assume that each router knows the identity of every other router in the network, but the shortest part to each router is not known.
- A **distance vector** is defined as the list of <destination, cost> tuples, one tuple per destination.
- Each router maintains a distance vector. The cost in each tuple is equal the sum of costs on the shortest path to the destination.

**Updation of router tables :**

- A router periodically sends a copy of its distance vector to all its neighbours.
- When a router receives a distance vector from its neighbour, it tries to find out whether its cost to reach any destination would decrease if it routed packets to that destination through that particular neighbouring router. This is illustrated in Fig. 4.12.1.

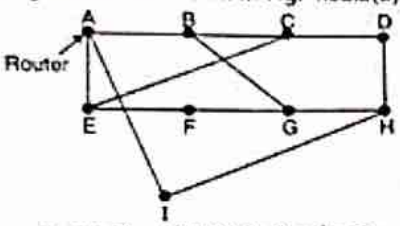



(G-463) **Fig. 4.12.1 : Distance vector algorithm at router A**

- Fig. 4.12.1 shows how the D.V. at A is automatically modified when a D.V. is received from B. A similar calculation takes place at the other routers as well.
- So the entries at every router can change. In Fig. 4.12.1 the initial distance vector is shown.
- The entries indicate to the costs corresponding to the shortest distance between the routers indicate to that square.
- For example, AC = 3 indicates the cost corresponding to the shortest path in terms of number of hops from A to C.
- Even if nodes asynchronously update their distance vectors the routing tables eventually converge. The well known example of distance vector routing is the Bellman-Ford algorithm.

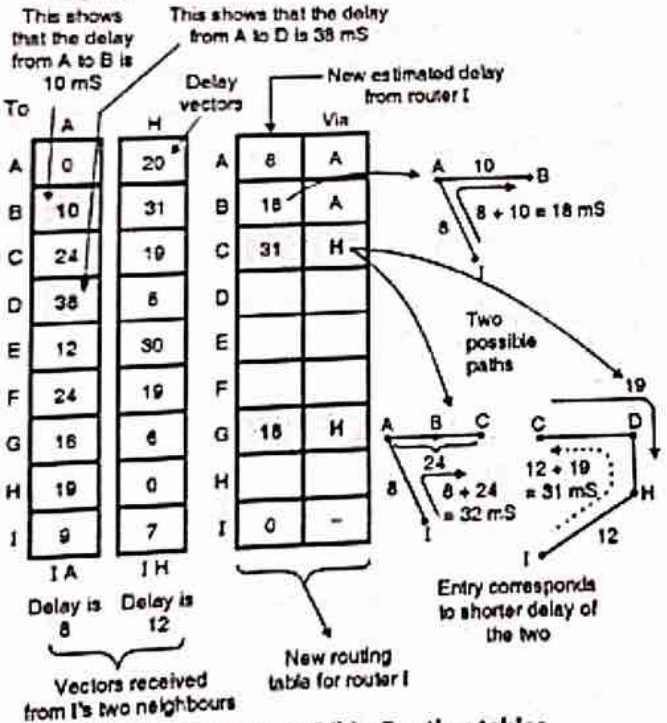
**Routing procedure in distance vector routing :**

- The example of a subnet is shown in Fig. 4.12.2(a) and the routing tables are shown in Fig. 4.12.2(b).



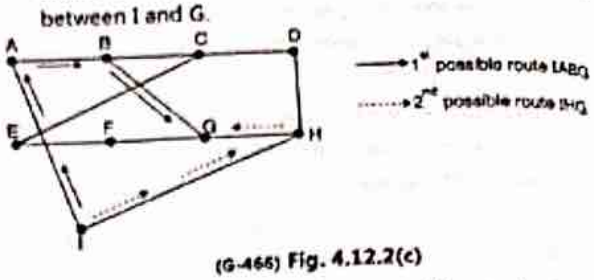

(G-464) **Fig. 4.12.2(a) : A subnet**

- The entries in router tables of Fig. 4.12.2(b) are the delay vectors.




(G-465) **Fig. 4.12.2(b) : Routing tables**

- For example consider the shaded boxes of Fig. 4.12.2(b).
- The entry in the first shaded box shows that the delay from A to B is 10 msec, whereas the entry in the other shaded box indicates that the delay from A to D is 38 msec.
- Consider how router I computes its new route to router G. Fig. 4.12.2(c) shows the two possible routes between I and G.




(G-466) **Fig. 4.12.2(c)**

- I knows that the reach G via A, the delay required is :

I to A Delay = 8mS
A to G Delay = 16mS
$\therefore$ I to G Delay = 8 + 16 = 24 msec

(L-891)

Whereas the delay between I and G via H (route IHG) is

I to H Delay = 12mS
H to G Delay = 6mS
$\therefore$ I to G Delay = 12 + 6 = 18 msec

(L-892)

- The best of these values is 18 msec corresponding to the path IHG. Hence it makes an entry in its routing table (I's table) that the delay to G is 18 msec and that the route to use it is via H.
- The new routing table for router I is shown in Fig. 4.12.2(b). Similarly we can calculate the delays, from I to different destinations from A to I and enter the minimum possible delay into the I's router table.

## 4.12.1 Disadvantages :

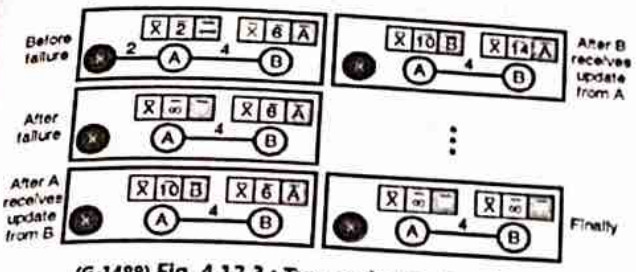
1. The distance vector routing takes a long time in converging to the correct answer. This is due to a problem called count-to-infinity problem. The problem can be solved by using the split horizon algorithm.
2. Another problem is that this algorithm does not take the line bandwidth into consideration when choosing a root. This is a serious problem due to which this algorithm was replaced by the Link State Routing algorithm.

## 4.12.2 Looping in Distance Vector Routing Protocol :

- A problem in distance vector rooting is its instability. A network using this protocol can become unstable.

**Two node loop instability :**

- A network with three nodes has been shown in Fig. 4.12.3.

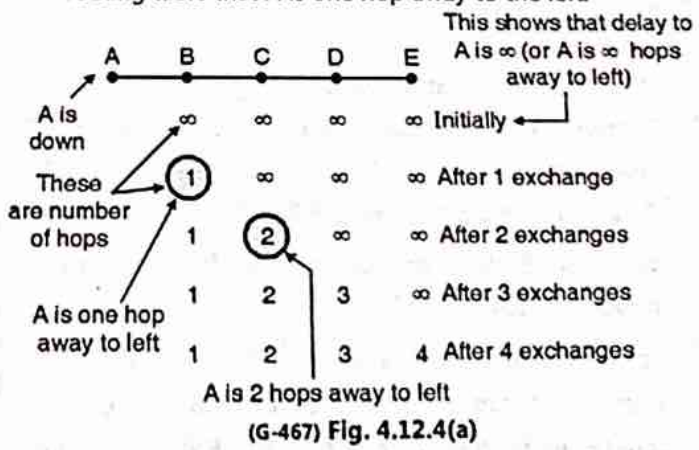



(G-1499) **Fig. 4.12.3 : Two node loop instability**

- Note that the routing tables are shown partially for discussion. At the beginning both nodes A and B know how to reach node X.
- But the link joining A and X fails suddenly. So node A changes its table. If A could send its changed routing table to B immediately, everything is okay. No problem will occur.
- But the system becomes unstable if B sends its routing table to A before receiving A's routing table.
- This is because node A receives the updated B's routing table and assumes that B has found a new path to reach node X.
- So A immediately updates its routing table (which is incorrect). Based on this update now A sends its new update to B. Now B thinks that something has changed around A and so it updates its routing table.
- Due to this process, the cost of reaching X increases gradually and finally becomes infinite. At this moment both A and B understand that now it is impossible to reach X.
- Note that during this entire time the system is unstable. A thinks that the route to X goes via B whereas B thinks that the route is via node A.
- So if A receives a packet for X, it goes to B and then again returns back to A. Similarly if B receives a packet destined for X, it goes to A and returns back to B.
- This bouncing of packets between nodes A and B is known as the **two-node loop problem.**
- This problem can be solved by using one of the following strategies :
  1. Defining infinity
  2. Split horizon
  3. Split horizon and poison reverse.
- There is a similar problem called three node loop problem present in the system using distance vector routing.

### 4.12.3 Count to Infinity Problem :

- Theoretically the distance vector routing works properly but practically it has a serious problem. The problem is that we get a correct answer but we get it slowly.
- In other words it reacts quickly to good news but it reacts too slowly to bad news. Consider a router whose best route to destination X is large.
- If on the next exchange neighbour A suddenly reports a short delay to X, the router will switch over and start using the line to A for sending the traffic to destination X.
- Thus in one vector exchange, the good news is processed. Let us see how fast does a good news propagate.
- Consider a linear subnet of Fig. 4.12.4 which has five nodes.
- The delay metric used is the number of hops. Assume that A is initially down and that all the other routers know this.
- So all the routers have recorded that the delay to A is infinity. When A becomes OK, the other routers come to know about it via the vector exchanges.
- Then suddenly a vector exchange at all the routers will take place simultaneously.
- At the time of first vector exchange, B comes to know that its left neighbour has a zero delay to A.
- So as shown in Fig. 4.12.4(a). B makes an entry in its routing table that A is one hop away to the left.

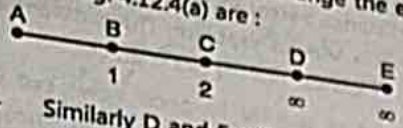



(G-467) **Fig. 4.12.4(a)**

- All the other routers still think that A is down. So in the second row of Fig. 4.12.4(a), the entries below C D E are ∞.

... hop length.

... not change their table entries. So after the second vector exchange the entries in the third row of Fig. 4.12.4(a) are :



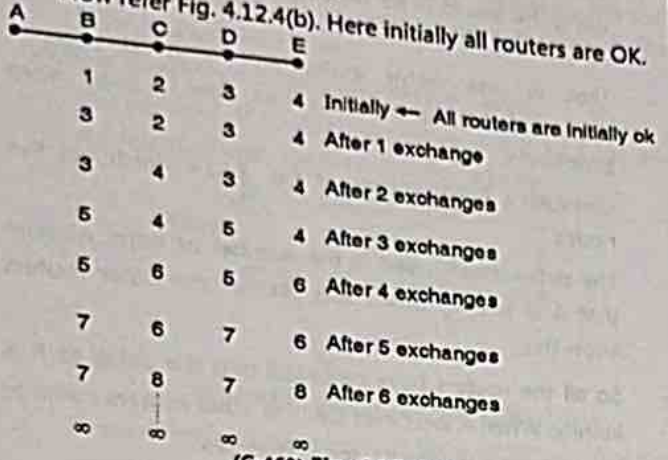

(G-468(a))

After 2 exchanges

- Similarly D and E will update their routing tables after 3 and 4 exchanges respectively. So we conclude that the good news of A has recovered has spread at a rate of one hop per exchange.

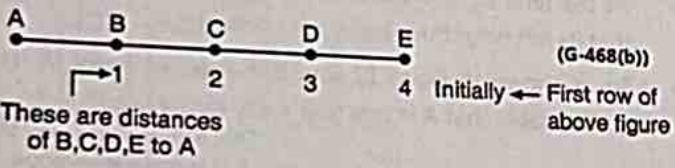
**Explanation of Fig. 4.12.4(b) :**

- Now refer Fig. 4.12.4(b). Here initially all routers are OK.



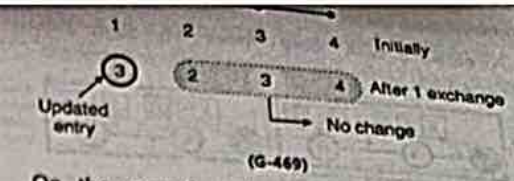

(G-468) **Fig. 4.12.4(b)**

- The routers B, C, D and E have distances of 1, 2, 3 and 4 respectively to A. So the first row of Fig. 4.12.4(b) is as follows :



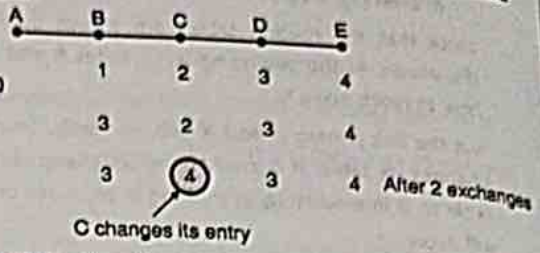

(G-468(b))

- Now imagine that suddenly A goes down or line between A and B is cut. At the first packet exchange B does not hear anything from A (because A is down). But C says " I have a path of length 2 to A".
- But poor B does not understand that this path is through B itself.
- So B thinks that it can reach A via C with a path length 3. (B to C 1 hop and C to A 2 hops) so it accordingly updates its routing table. But D and E do not update their entries.
- So the second row of Fig. 4.12.4(b) looks as follows :




(G-469)

- On the second exchange C realizes that both its neighbours (B and D) claim to have a path of length 3 to A. So it picks one of them at random and makes its new distance to A as 4.
- This is shown in row 3 of Fig. 4.12.4(b). It is repeated below.




(G-470)

- Similarly the other routers keep updating their tables after every exchange. It is expected that finally we should get ∞ in the router tables of B, C, D and E indicating that A is down.
- We do reach this state at the end in Fig. 4.12.4(b) but after a very long time. The conclusion is bad news propagates slowly. This problem is called as **count-to-infinity** problem.
- The solution to this problem is to use the split horizon algorithm.

## 4.12.4 Split Horizon Algorithm :

- To avoid the count to infinity problem, several changes in the algorithm have been suggested. But none of them work satisfactorily in all situations.
- One particular method which is widely implemented, is called as the **split horizon algorithm.**
- In this algorithm, the minimum cost to a given destination is not sent to a neighbour if the neighbour is the next node along the shortest path.
- For example if node A thinks that the best route to node B is via node C, then node A should not send the corresponding minimum cost to node C.

# 4.13 Link State Routing :

- Distance vector routing was used in ARPANET up to 1979. After that it was replaced by the link state routing.

- Variants of this algorithm are now widely used. The link state routing is simple and each router has to perform the following five operations.

**Router operations :**

1. Each router should discover its neighbours and obtain their network addresses.
2. Then it should measure the delay or cost to each of these neighbours.
3. It should construct a packet containing the network addresses and the delays of all the neighbours.
4. Send this packet to all other routers.
5. Compute the shortest path to every other router.

- The complete topology and all the delays are experimentally measured and this information is conveyed to each and every router.
- Then a shortest path algorithm such as Dijkshtra's algorithm can be used to find the shortest path to every other router.

**Protocols :**

- Link state routing is popularly used in practice. The OSPF protocol which is used in the Internet uses the link state algorithm.

IS-IS i.e. Intermediate system – Intermediate system is the other protocol which uses the link state algorithm. IS-IS is used in Internet backbones and in some digital cellular systems such as CDPD.

**Building a routing table in link state routing :**

Now we will discuss the development of routing table in link state routing. Here the term **link state** is used for defining the characteristic of a link or edge, which represents a network in the Internet.

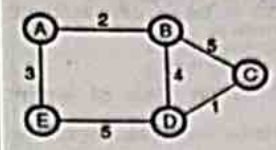
The cost associated with each link is important. The links having lower costs are preferred to the links having higher costs.

A nonexisting or broken link is indicated by an ∞ cost. In this method, each node must have a complete map of the network.

That means each node should have complete information about the state of each link. The collection of states of all the links in an Internet is called as **Link-State Database (LSDB).**

For the entire Internet, there is only one LSDB and its copy is available with each node. Each node uses it to create the least cost tree.

- The example of LSDB is as shown in Fig. 4.13.1(b) for the Internet shown in Fig. 4.13.1(a).



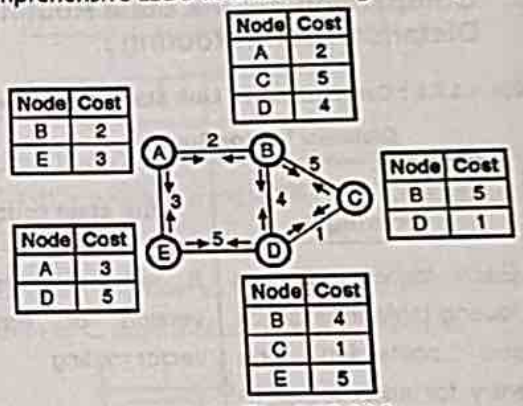

(a) Internetwork

| | A | B | C | D | E |
|---|---|---|---|---|---|
| A | 0 | 2 | ∞ | ∞ | 3 |
| B | 2 | 0 | 5 | 4 | ∞ |
| C | ∞ | 5 | 0 | 1 | ∞ |
| D | ∞ | 4 | 1 | 0 | 5 |
| E | 3 | ∞ | ∞ | 5 | 0 |

(b) Link state database (LSDB)

(G-2201) Fig. 4.13.1

- The next step is creation of LSDB (which contains all the information about the Internet) at each node. This can be achieved by a process called **flooding.**
- Each node sends a greeting message to all its immediate neighbours, so as to collect two important pieces of information as follows :
  1. The identity of the neighbouring node.
  2. Cost of the link.
- The packet containing this information is called as **LS Packet** (LSP), which is sent out of each interface. After receiving all the new LSPs each node will create the comprehensive LSDB as shown in Fig. 4.13.1(c).




(G-2202) Fig. 4.13.1(c)

- This LSDB is same for each node which shows the whole map of the internet.
- That means a node can use the LSDB to make the whole map of the Internet.

## 4.13.1 Advantage of LSR :

- The link state routing is advantageous than distance vector routing due to the following reasons :
  1. Link state routing is advanced version of distance vector routing (DVR).
  2. Link state routing is a faster algorithm than the DVR algorithm.
  3. For LSR a wider bandwidth is available.

4. In LSR all the delays are measured and distributed to every router.
5. In LSR, the line bandwidth is taken into account when choosing the routes.
6. LSR algorithm gets a common view of entire network whereas DVR views network topology from neighbour's perspective.
7. The LSR algorithm calculates the shortest path to the other routers whereas the DVR algorithm adds distance vectors from router to router.
8. The LSR sends only the event triggered updates due to which it converges fast whereas DVR sends frequent and periodic updates due to which it converges slowly.
9. In LSR, the link state routing updates are sent to the other routers while in DVR copies of routing tables are passed on to the neighbour routers.

### 4.13.2 Comparison of Link State Routing and Distance Vector Routing :

**Table 4.13.1 : Comparison of Link State Routing and Distance Vector Routing**

| Sr. No. | Distance vector routing | Link state routing |
|---|---|---|
| 1. | Each router maintains routing table indexed by and containing one entry for each router in the subnet. | It is the advanced version of distance vector routing |
| 2. | Algorithm took too long to converge. | Algorithm is faster. |
| 3. | Bandwidth is less. | Wide bandwidth is available. |
| 4. | Router measure delay directly with special ECHO packets. | All delays measured and distributed to every router. |
| 5. | It doesn't take line bandwidth into account when choosing the routes. | It considers the line bandwidth into account when choosing the routes. |

### 4.13.3 Advantages and Disadvantages of Dynamic Routing :

**Advantages :**

1. It is very simple
2. The complete topology and all the delays are experimentally measured and this information is conveyed to each and every router.

**Disadvantage :**

- For these algorithms the routing decision can be changed if there are any changes in topology traffic etc.

## 4.14 Least Cost Algorithms :

- We have already defined the term **cost** associated with a link and the factors affecting / deciding its value.
- These link costs or hop costs are used as inputs to a least cost routing algorithm.

**Principle :**

- The principle of least cost routing algorithms is as follows :
- If there is a network of nodes connected by bidirectional links, where each link as a cost associated with it in each direction, the cost of a path between two nodes is defined as the sum of cost of links traversed. For each pair of nodes find the path with least cost.

**Examples :**

- The well known examples of least cost routing algorithms are :
  1. Dijkstra's algorithm and
  2. Bellman-Ford algorithm.

### 4.14.1 Dijkstra's Algorithm :

- Dijkstra's algorithm is used for computing the shortest path from the **root** node to every other node in the network.
- The root node is defined as the node corresponding to the router where the algorithm is being run. The total number of nodes are divided into two groups namely the P group and T group.
- In the P group we have those nodes for which the shortest path has already been found.
- In T group the remaining nodes are placed. The path to every node in the T group should be computed from a node which is already present in group P.

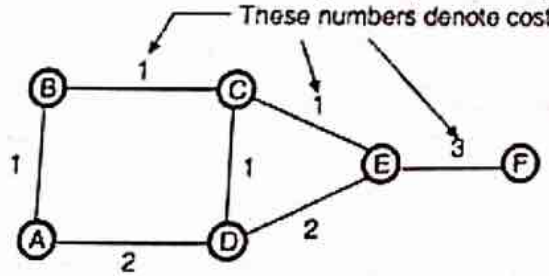
- We should find out every possible way to reach an outside node by a one hop path from a node which is already present in P and choose the shortest of these paths as the path to the desired node.
- As stated earlier we define two sets P (permanent) and T (temporary) of the nodes. In set P we have nodes to which the shortest path has already been found and in set T we have nodes to which we are considering the shortest paths.
- At the time of starting, P is initialized to the current node and T is initialized to null. The algorithm then repeats the following steps :

1. Start from the desired node say p. Write p in the P set.
2. For this node p, add each of its neighbours n to T set. The addition of these nodes in T will have to satisfy the following conditions :
   1. If the neighbouring node (say n) is not there in T then add it annotating it with the cost to reach it through p and p's ID.
   2. If n is already present in T and the path to n through p has a lower cost, then remove the earlier instance of n and add the new instance annotated with the cost to reach it through p and p's ID.
   3. Pick up the neighbour n which has the smallest cost in T, and if it is not present in P then add it to P. Use its annotation to determine the router p to use to react n.
3. Stop when T is empty.

- This algorithm will be clear after solving the following example.

---

**Ex. 4.14.1 :** For the network shown in Fig. P. 4.14.1(a), show the computations at node A using the Dijkshtra's algorithm.




**(G-451) Fig. P. 4.14.1(a) : Given network**

**Soln. :**

**Step 1 :**

- Since the computations are to be done at node A, the starting node will be A. We enter this node into group P as shown in Table P. 4.14.1(a).
- We add the neighbouring nodes B and D in group T along with the costs to reach them through A as shown in Table P. 4.14.1(a).

**(G-451(a)) Table P. 4.14.1(a)**

| Permanent (P) | Temporary (T) |
|---|---|
| A | B(A,1) D(A,2) |

A•

**Note :** B(A,1) means B is reached by A, and the cost is 1. Similarly D(A,2) means D is reached by A and the cost is 2.

**Step 2 :**

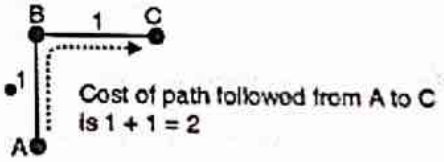
- Now pick up the neighbour with the smallest cost and add it to P set. Here the neighbour with smallest cost is B.
- So let us add B(A,1) to P group as shown in Table P. 4.14.1(b).
- As B is added to P group, we have to add its neighbour i.e. C to the T group, as shown in Table P. 4.14.1(b).

**(G-452) Table P. 4.14.1(b)**

| Permanent (P) | Temporary (T) |
|---|---|
| A | B(A,1) D(A,2) |
| A,B(A,1) | D(A,2),C(B,2) |

•A
A 1 B

- Note that D(A,2) has remained in T group as it is but C(B,2) is a new entry. C(B,2) means C is reached by A via B with a cost of 2.
- The cost is 2 due to the path followed from A to B and then to C, as illustrated in Fig. P. 4.14.1(b).



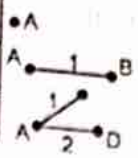

**(G-453) Fig. P. 4.14.1(b)**

**Step 3 :**

- Now pick up the neighbour in T set with the smallest cost in Table P. 4.14.1(b) and add it to the P set. Here we choose neighbour D because it is the immediate neighbour of A.
- Since D is added to P group, we have to add its neighbours i.e. C and E to the T group as shown in Table P. 4.14.1(c).
- Note that C(B,2) goes as it is, and E(D,4) is a new entry to Table P. 4.14.1(c).

(G-454) Table P. 4.14.1(c)

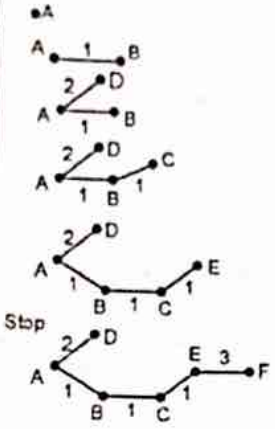
| Permanent (P) | Temporary (T) |
|---|---|
| A | B(A,1), D(A,2) |
| A,B(A,1) | D(A,2), C(B,2) |
| A,B(A,1), D(A,2) | E(D,4), C(B,2) |



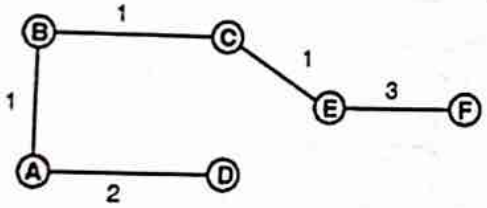

- But C(D,3) cannot be entered because its cost is 3.
- Where E(D,4) means E is reached by A via D and the cost is 4.
- Similarly we can proceed further. The final table is as shown in Table P. 4.14.1(d).

(G-455) Table P. 4.14.1(d) : Final table

| Permanent (P) | Temporary (T) |
|---|---|
| A | B(A,1) D(A,2) |
| A,B(A,1) | D(A,2) C(B,2) |
| A,B(A,1) D(A,2) | E(D,4) C(B,2) |
| A,B(A,1) D(A,2) C(B,2) | E(C,3) E(D,4) can not be included |
| A,B(A,1) D(A,2),C(B,2) E(C,3) | F(E,6) F(E,7) can not be included |
| A,B(A,1) D(A,2),C(B,2) E(C,3),F(E,6) | Empty (NULL) |



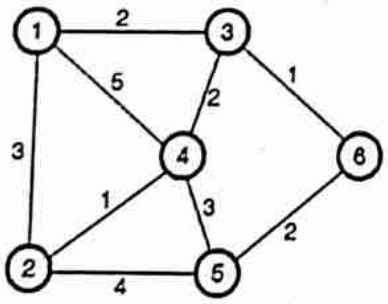

- The shortest paths from A to all other nodes are as shown in Fig. P. 4.14.1(c).




(G-456) Fig. P. 4.14.1(c) : Shortest paths from A to all other nodes

---

**Ex. 4.14.2 :** Write Dijkstra's algorithm. Find shortest path Fig. P. 4.14.2(a) to destination node 6.




(G-1383) Fig. P. 4.14.2(a)

**Soln. :**

- For Dijkstra's algorithm refer section 4.14.1. Let Node 1 → A, 2 → B, 3 → C, 4 → D, 5 → E, 6 → F

**Step 1 :**

- The starting node is A. Enter it into group P as shown in Table P. 4.14.2(a). Add neighbours B, C and D to the temporary group T.

(G-2880) Table P. 4.14.2(a)

| Permanent (P) | Temporary (T) |
|---|---|
| A | B(A, 3), C(A, 2) |
| | D(A, 5) |

**Step 2 :**

- Now pick up the neighbour with smallest cost i.e. c and add it to group P. As C is added to P group, we have to add neighbours of C to T group as shown in Table P. 4.14.2(b).

(G-2303) Table P. 4.14.2(b)

| Permanent (P) | Temporary (T) |
|---|---|
| A | B(A, 3), C(A, 2), D(A, 5) |
| A, C(A, 2) | B(A, 3), D(A, 5), D(C, 4), F(C, 3) |

- D(C, 4) is another entry in T group which shows that D is approached by A via C and the cost is 4.

**Step 3 :**

- Now move B(A, 3) from T to P and add neighbours D and E to T group as shown in Table P. 4.14.2(c).

(G-2881) Table P. 4.14.2(c)

| Permanent (P) | Temporary (T) |
|---|---|
| A | B(A, 3), C(A, 2), D(A, 5) |
| A, C(A, 2) | B(A, 3), D(C, 4), F(C, 3) |
| A, C(A, 2), B(A, 3) | D(C, 4), F(C, 3), D(B, 4), E(B, 7) |

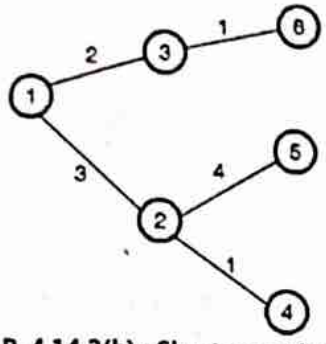
**Step 4 :**

- Now continue in the same manner to get the final table as shown in Table P. 4.14.2(d).

(G-2882)Table P. 4.14.2(d) : Final table

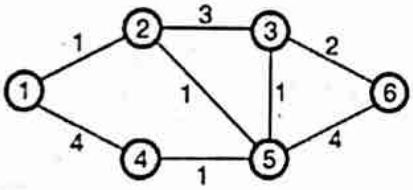
| Permanent (P) | Temporary (T) |
|---|---|
| A | B(A, 3), C(A, 2), D(A, 5) |
| A, C(A, 2) | B(A, 3), D(C, 4), F(C, 3) |
| A, C(A, 2), B(A, 3) | D(C, 4), F(C, 3), D(B, 4), E(B, 7) |
| A, C(A, 2), B(A, 3), D(B, 4) | F(C, 3), E(B, 7) |
| A, C(A, 2), B(A, 3), D(B, 4), E(B, 7) | F(C, 3), F(E, 9) |
| A, C(A, 2), B(A, 3), D(B, 4), E(B, 7), F(C, 3) | Null (stop) |

- Shortest path from node 1 to other nodes is shown in Fig. P. 4.14.2(b).




(G-1384) **Fig. P. 4.14.2(b) : Shortest path from 1 to all other nodes**

**Ex. 4.14.3 :** For the graph shown in Fig. P. 4.14.3 show the successive iterations of the Dijkstra's method of shortest path algorithm. Take node 1 as the root node.




(G-1287) **Fig. P. 4.14.3**

**Soln. :**

**Let node :** 1 → A, 2 → B, 3 → C, 4 → D, 5 → E, 6 → F.

**Step 1 :**

- Starting node is A. Enter it into group P as shown in Table P. 4.14.3(a). Add neighbours and D to temporary group T.

(G-2883)Table P. 4.14.3(a)

| Permanent (P) | Temporary (T) |
|---|---|
| A | B(A, 1), D(A, 4) |

**Step 2 :**

- Now pick up neighbour with smallest cost. i.e. B and add it to group P. As B is added to P group, we have to add neighbours of B to T group as shown in Table P. 4.14.3(b).

(G-1288) Table P. 4.14.3(b)

| Permanent (P) | Temporary (T) |
|---|---|
| A | B(A, 1), D(A, 4) |
| A, B(A, 1) | D(A, 4), C(B, 4), E(B, 2) |

**Step 3 :**

- Now move E(B, 2) from T to P and add neighbours C and F to T group as shown in Table P. 4.14.3(c).

(G-1289) Table P. 4.14.3(c)

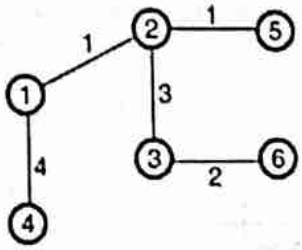
| Permanent (P) | Temporary (T) |
|---|---|
| A | B(A, 1), D(A, 4) |
| A, B(A, 1) | D(A, 4), C(B, 4), E(B, 2) |
| A, B(A, 1), E(B, 2) | D(A, 4), C(B, 4), F(E, 6) |

**Step 4 : Now continue in the same manner to get final table as shown in Table P. 4.14.3(d) :**

(G-2884)Table P. 4.14.3(d)

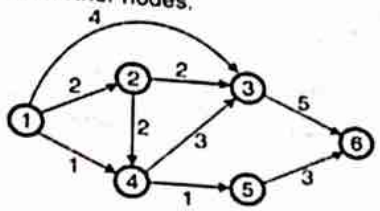
| Permanent (P) | Temporary (T) |
|---|---|
| A | B(A, 1), D(A, 4) |
| A, B(A, 1) | D(A, 4), C(B, 4), E(B, 2) |
| A, B(A, 1), E(B, 2) | D(A, 4), C(B, 4), F(E, 6) |
| A, B(A, 1), E(B, 2), C(B, 4) | D(A, 4), F(E, 6), E(C, 5), F(C, 6) |
| A, B(A, 1), E(B, 2), C(B, 4), D(A, 4) | F(C, 6), E(D, 5) |
| A, B(A, 1), E(B, 2), C(B, 4), D(A, 4), D(A, 4), F(C, 6) | Null (stop) |

- Shortest path from node 1 to other nodes is shown in Fig. P. 4.14.3(a).




(G-1290) **Fig. P. 4.14.3(a) : Shortest path from 1 to all other nodes**

**Ex. 4.14.4 :** Find the shortest path between the source node 1 to all other nodes for the network shown in Fig. P. 4.14.4 using Dijkstra's algorithm. Also draw the shortest path tree from node 1 to all other nodes.




(G-1491) Fig. P. 4.14.4

**Soln. :**

Let node 1 → A, 2 → B, 3 → C, 4 → D, 5 → E, 6 → F

**Step 1 :**

- Starting node is A. Enter it into group P as shown in Table P. 4.14.4(a). Add neighbours B and D to temporary group T.

(G-2885)**Table P. 4.14.4(a)**

| Permanent (P) | Temporary (T) |
|---|---|
| A | B(A, 2), D(A, 1), C(A, 4) |

**Step 2 :**

- Now pick up neighbour with smallest cost i.e. D and add it to group P.
- As D is added to P group we have to add neighbours of D to T group as shown in Table P. 4.14.4(b).

(G-1492) **Table P. 4.14.4(b)**

| Permanent (P) | Temporary (T) |
|---|---|
| A | B(A, 2), D(A, 1), C(A, 4) |
| A, D(A, 1) | B(A, 2), C(D, 4), E(D, 2), C(A, 4) |

**Step 3 :**

- Now move E (D, 1) from T to P and add neighbour F to T group as shown in Table P. 4.14.4(c).

(G-2886)**Table P. 4.14.4(c)**

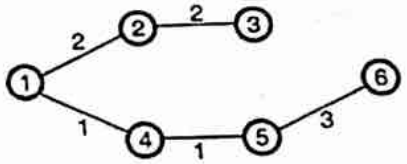
| Permanent (P) | Temporary (T) |
|---|---|
| A | B(A, 2), D(A, 1), C(A, 4) |
| A, D(A, 1) | B(A, 2), C(D, 4), C(A, 4), E(D, 2) |
| A, D(A, 1), E(D, 2) | B(A, 2), C(D, 4), F(E, 5) |

**Step 4 : Now continue in the same manner to get final table as shown in Table P. 4.14.4(d) :**

(G-2887)**Table P. 4.14.4(d)**

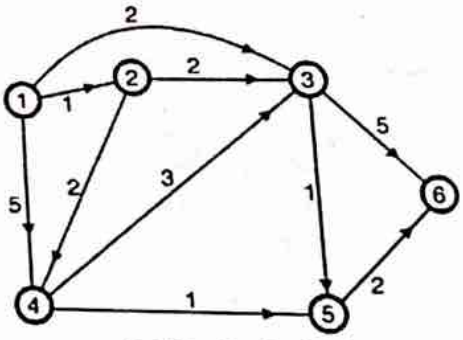
| Permanent (P) | Temporary (T) |
|---|---|
| A | B(A, 2), D(A, 1), C(A, 4) |
| A, D(A, 1) | B(A, 2), C(D, 4), C(A, 4), E(D, 2) |
| A, D(A, 1), E(D, 2) | B(A, 2), C(D, 4), F(E, 5) |
| A, D(A, 1), E(D, 2), B(A, 2) | C(D, 4), F(E, 5), C(B, 4), D(B, 4) |
| A, D(A, 1), E(D, 2), B(A, 2), C(B, 4) | F(E, 5), D(B, 4), F(C, 9) |
| A, D(A, 1), E(D, 2), B(A, 2), C(B, 4), F(E, 5) | Null (stop) |

- Shortest path from node 1 to all other nodes is shown in Fig. P. 4.14.4(a).




(G-1493) **Fig. P. 4.14.4(a) : Shortest path from node 1 to all other nodes**

**Ex. 4.14.5 :** Find the shortest path between the source node 1 to all other nodes for the network shown in Fig. P. 4.14.5 using Dijkstra's algorithm. Also draw the shortest path tree from node 1 to all other nodes.




(G-1489) **Fig. P. 4.14.5**

**Soln. :**

Let node 1 → A, 2 → B, 3 → C, 4 → D, 5 → E, 6 → F.

**Step 1 :**

- Starting node is A. Enter it into group P as shown in Table P. 4.14.5(a). And neighbours B, C and D to temporary group T.

(G-2888)Table P. 4.14.5(a)

| Permanent (P) | Temporary (T) |
|---|---|
| A | B(A, 1), D(A, 5), C(A, 2) |

**Step 2 :**

- Now pick up neighbour with smallest cost i.e. B and add it to group P.
- As B is added to P group we have to add neighbours of B to T group as shown in Table P. 4.14.5(b).

(G-1496) Table P. 4.14.5(b)

| Permanent (P) | Temporary (T) |
|---|---|
| A | B(A, 1), D(A, 5), C(A, 2) |
| A, B(A, 1) | D(A, 5), C(A, 2), D(A, 3), C(A, 3) |

**Step 3 :**

- Now move C (A, 2) from T to P and add neighbour E and E to T group as shown in Table P. 4.14.5(c).

(G-2889)Table P. 4.14.5(c)

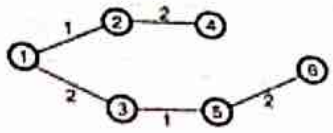
| Permanent (P) | Temporary (T) |
|---|---|
| A | B(A, 1), D(A, 5), C(A, 2) |
| A, B(A, 1) | D(A, 5), C(A, 2), D(A, 3), C(A, 3) |
| A, B(A, 1), C(A, 2) | D(A, 5), D(A, 3), C(A, 3), E(C, 3), F(C, 7) |

**Step 4 : Now continue in the same manner to get final table as shown in Table P. 4.14.5(d) :**

(G-2890)Table P. 4.14.5(d)

| Permanent (P) | Temporary (T) |
|---|---|
| A | B(A, 1), D(A, 5), C(A, 2) |
| A, B(A, 1) | D(A, 5), C(A, 2), D(A, 3), C(A, 3) |
| A, B(A, 1), C(D, 2) | D(A, 5), D(A, 3), C(A, 3), F(C, 3), F(C, 7) |
| A, B(A, 1), C(A, 2), D(A, 3) | C(A, 3), E(C, 3), F(C, 7), C(D, 6), E(D, 4) |
| A, B(A, 1), C(A, 2), D(A, 3), E(C, 3) | F(C, 7), F(E, 5) |
| A, B(A, 1), C(A, 2), D(A, 3), E(C, 3), F(E, 5) | Null (stop) |

- Shortest path from node 1 to all other nodes is shown in Fig. P. 4.14.5(a).




(G-1497) Fig. P. 4.14.5(a) : Shortest path from node 1 to all other nodes

## 4.15 Bellman-Ford Algorithm :

- Let us suppose that node 1 is the "destination" node and consider the problem of finding a shortest path from every node to node 1.
- We assume that there exists at least one path from every node to the destination.

1. To simplify the presentation, let us denote $d_{ij} = \infty$ if (i, j) is not an arc of the graph. Using the convention we can assume without loss of generality that there is an arc between every pair of nodes, since walks and paths consisting of true network arcs are the only ones with less than $\infty$.
2. A shortest walk from a given node i to node 1, subject to constraint that the walk contains at most 'h' arcs and goes through node 1 only once, is referred to as shortest ($\leq$ h) walk and its length is denoted by $D_i^h$.

   Note that such a walk may not be a path, that is, it may contain repeated nodes. We will later give conditions under which this is not possible.
3. By convention, we take

   $$D_i^h = 0, \quad \text{for all } h$$

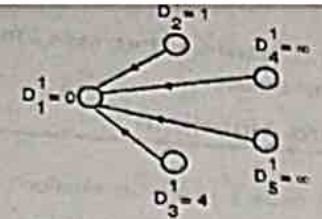
- We will prove that $D_i^h$ can be generated by the iteration.

$$D_i^{h+1} = \min_j [d_{ij} + D_j^h], \quad \text{for all } i \neq 1 \qquad ...(4.15.1)$$

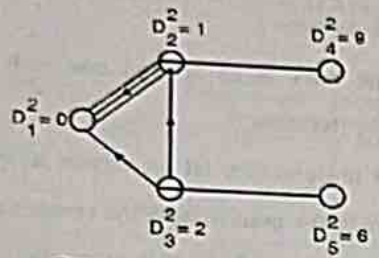
- Starting from the initial conditions,

$$D_i^0 = \infty \quad \text{for all } i \neq 1 \qquad ...(4.15.2)$$

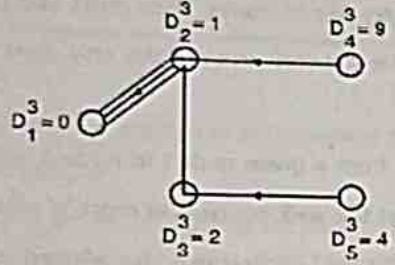
- This is the Bellman Ford algorithm illustrated in Fig. 4.15.1.



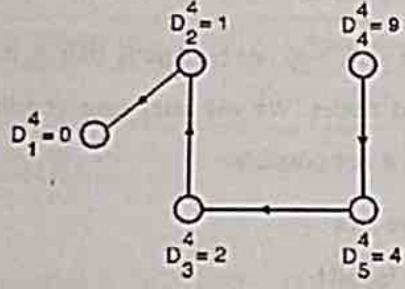

Shortest paths using at most 1 arc




Shortest paths using at most 2 arcs




Shortest paths using at most 3 arcs




Final tree of shortest paths

**(G-1376) Fig. 4.15.1 : Successive iterations of the Bellman-Ford method**

4. Thus Bellman Ford algorithm first finds the one-arc shortest walk lengths, then find the two-arc shortest path lengths and so forth.

   In this example, the shortest ($\leq$ h) walks are paths because all arc lengths are positive and therefore all cycles have positive length.

additional assumption that all cycles not containing node 1 have non-negative length. We say that the algorithm terminates after 'h' iterations if,

$$D_i^h = D_i^{h-1}, \text{ for all } i$$

- The following proposition provides the main result.

6. **Proposition :**

   Consider the Bellman-Ford algorithm Equation (4.15.1) with initial conditions $D_i^0 = \infty$ for all $i \neq 1$. Then

   (a) The scalars $D_i^h$ generated by the algorithm are equal to the shortest ($\leq$ h) walk lengths from node i to node 1.

   (b) The algorithm terminates after a finite number of iterations if and only if all cycles not containing node 1 have non-negative length. Furthermore, if the algorithm terminates, it does so after at most $h \leq N$ iterations and at termination, $D_i^h$ is the shortest path length from i to 1.

**Proof :**

(a) We argue by induction. From Equation (4.15.1) and (4.15.2) we have,

$$D_i^1 = d_{i1}, \text{ for all } i \neq 1$$

- So $D_i^1$ is indeed equal to shortest ($\leq$ 1) walk length from i to 1.
- Suppose that $D_i^k$ is equal to shortest ($\leq$ k) walk length from i to 1 for all $k \leq h$.
- We will show that $D_i^{h+1}$ is the shortest ($\leq$ h + 1) walk length from i to 1.
- Indeed, a shortest ($\leq$ h + 1) walk from i to 1 either consists of less than h + 1 arcs, in which case its length is equal to $D_i^h$, or else it consists of h + 1 arcs with the first arc being (i, j) for some $j \neq 1$, followed by an h - arc walk from j to 1 in which node 1 is not repeated.
- The latter walk must be a shortest ($\leq$ h) walk from j to 1 [otherwise by concatenating arc (i, j) and a shorter ($\leq$ h) walk from j to 1, we would obtain a shorter ($\leq$ h + 1) walk from i to 1] we thus conclude that,

$$\text{Shortest } (\leq h+1) \text{ walk length} = \min\{D_i^h, \min_{j \neq 1}[d_{ij} + D_j^h]\} \quad \ldots(4.15.3)$$

- Using the induction hypothesis, we have $D_j^k \le D_j^{k-1}$ for all $k \le h$ [since the set of ($\le k$) walks from node j to 1 contains the corresponding set of ($\le k-1$) walks]
- Therefore,

$$D_i^{h+1} = \min_j [d_{ij} + D_j^h] \le \min_j [d_{ij} + D_j^{h-1}] = D_i^h \quad ...(4.15.4)$$

- Furthermore, we have $D_i^h \le D_i^1 = d_{i1} = d_{i1} + D_1^h$ so from Equation (C) we obtain,

Shortest ($\le h+1$) walk length $= \min \{D_i^h, \min [d_{ij} + D_j^h]\}$

$= \min \{D_i^h, D_i^{h+1}\}$

In view of $D_i^{h+1} < D_i^h$ [Cf. Equation (D)] this yields

- Shortest ($\le h+1$) walk length $= D_i^{h+1}$
- Completing the induction proof.

b) If the Bellman-Ford algorithm terminates after 'h' iterations, we must have

$$D_i^k = D_i^h, \text{ for all i and } k \ge h \quad ... (4.15.5)$$

So we cannot reduce the lengths of the shortest walks by allowing more arcs in these walks.

It follows that there cannot exist a negative-length cycle not containing node 1, since such a cycle could be repeated an arbitrarily large number of times in walks from some nodes to node 1, thereby making their length arbitrarily small and contradicting Equation (4.15.5).

Conversely, suppose that all cycles not containing node 1 have non-negative length.

Then by deleting all such cycles from shortest ($\le h$) walks, we obtain paths of less or equal length.

Therefore for every i and h, there exists a path that is a shortest ($\le h$) walk from i to 1 and the corresponding shortest path length is equal to $D_i^h$.

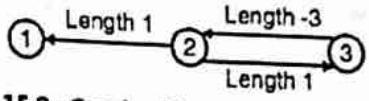
Since paths have no cycles, then it can contain at most N – 1 arcs. It follows that,

$$D_i^N = D_i^{N-1}, \text{ for all i}$$

Implying that the algorithm terminates after at most N iterations.

Note that the preceding proposition is valid even if there is no path from some nodes i to node 1 in the original network. Upon termination, we will simply have for those nodes $D_i^h = \infty$.

7. To estimate computation required to find the shortest path lengths, we note that in the worst case, the algorithm must be iterated N times, each iteration must be done for N – 1 nodes and for each node the minimization must be taken over no more than N – 1 alternatives. Thus the amount of computation grows at worst like $N^3$, which is written as $O(N^3)$.
8. Generally, the notation O (P(N)), where P(N) is a polynomial in N, is used to indicate a number depending on N that is smaller than Cp (N) for all N, where C is some constant independent of N. Actually, a more careful accounting shows that the amount of computation is O (mA), where A is the number of arcs and m is the number of iterations required for termination (m is also the maximum number of arcs contained in a shortest path).
9. The example in Fig. 4.15.2 shows the effect of negative length cycles not involving node 1 and illustrates that one can test for existence of such cycles simply by comparing $D_i^N$ with $D_i^{N-1}$ for each i.




**(G-1377) Fig. 4.15.2 : Graph with a negative cycle. The shortest path length from 2 to 1 is 1**

- The Bellman-Ford algorithm gives $D_2^2 = -1$ and $D_2^3 = -1$, indicating the existence of a negative length cycle.
- As implied by part (b) of the preceding proposition, there exists such a negative length cycle if and only if $D_i^N < D_i^{N-1}$ for some i.

**Bellman's equation and shortest path construction :**

1. Assume that all cycles not containing node 1 have non-negative length and denote by $D_i$ the shortest path length from node i to 1. Then upon termination of Bellman-Ford algorithm, we obtain

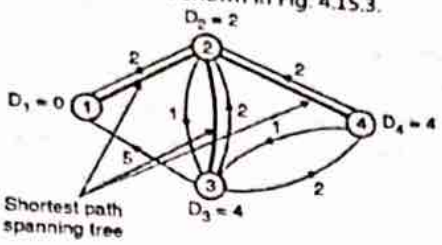
$$D_i = \min_j [d_{ij} + D_j], \quad \text{for all } i \ne 1 \quad ...(4.15.6)$$

$$D_1 = 0 \quad ...(4.15.7)$$

This is called Bellman's equation and expresses that the shortest path length from node i to 1 is the sum of the length of the arc to the node following i on the shortest path plus the shortest path length from that node to node 1.

2. From this equation it is easy to find the shortest paths (as opposed to the shortest path lengths) if all cycles not including node 1 have a positive length (as opposed to zero length). To do this, select for each $i \neq 1$, one arc $(i, j)$ that attains the minimum in the equation.

$D_i = \min_j [d_{ij} + D_j]$ and consider the subgraph consisting of these N – 1 arcs as shown in Fig. 4.15.3.




(G-1378) Fig. 4.15.3

3. To find the shortest path from any node i, start at i and follow the corresponding arcs of subgraph until node 1 is reached. Note that the same node cannot be reached twice before reaching node 1, since a cycle would be formed that (on the basis of equation $D_i = \min_j [d_{ij} + D_j]$ would have zero length

let $(i_1, i_2, ..., i_k, i_1)$ be the cycle and add the equations.

$$D_{i_1} = d_{i_1 i_2} + D_{i_2}$$
$$D_{i_{k-1}} = d_{i_{k-1} i_k} + D_{i_k}$$
$$D_{i_k} = d_{i_k i_1} + D_{i_1}$$

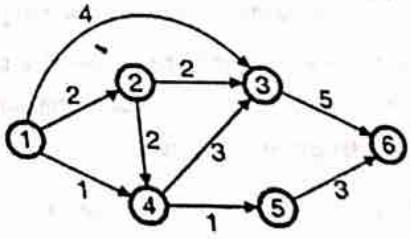
Obtaining $[d_{i_1 i_2} + ... + d_{i_k i_1} = 0]$

4. Since the subgraph connects every node to node 1 and has N – 1 arcs, it must be a spanning tree. We call this subgraph the shortest path spanning tree and not that it has a special structure of having a root (node 1), with every arc of the tree directed toward the root.

5. Using the preceding construction, it can be shown that if there are no zero (or negative) length cycles, then Bellman's Equation (4.15.6) and (4.15.7) (viewed as a system of N equations with N unknowns) has a unique solution. This fact is useful when we consider the Bellman-Ford algorithm starting from initial conditions other than ∞ [Cf Equation (B)].

- For a proof we suppose that $\tilde{D}_i$, i = 1, ..., N, are another solution of Bellman's Equation (4.15.6) and (4.15.2) with $\tilde{D}_1 = 0$ and we show that $\tilde{D}_i$ are equal to the shortest path lengths $D_i$.
- Let us repeat the path construction of the preceding paragraph with $\tilde{D}_i$ replacing $D_i$. Then $\tilde{D}_i$ is the length of the corresponding path from node i to node 1, showing that $\tilde{D}_i \geq D_i$. To show the reverse inequality, consider the Bellman-Ford algorithm with two different initial conditions.
- The first initial condition is $D_i^0 = \infty$, for $i \neq 1$ and $D_1^0 = 0$, in which case the true shortest path lengths $D_i$ are obtained after at most N – 1 iterations, as shown earlier.
- The second initial condition is $D_i^0 = \tilde{D}_i$, for all i, in which case $\tilde{D}_i$ is obtained after every iteration (since the $\tilde{D}_i$ solve Bellman's equation).
- Since the second initial condition is, for every i, less than or equal to the first, it seen from the Bellman-Ford iteration $D_i^{h+1} = \min_j [d_{ij} + D_j^h]$ that $\tilde{D}_i \leq D_i$ for all i.
- Therefore $\tilde{D}_i = D_i$ and the only solution of Bellman's equation is the set of the true shortest path lengths $D_i$.
- It is also possible to show that if there are zero length cycles not involving node 1, the Bellman's equation has a nonunique solution.
- It turns out that the Bellman-Ford algorithm works correctly even if the initial conditions $D_i^0$ for $i \neq 1$ are arbitrary numbers and the iterations are done in parallel for different nodes in virtually any order.

**Ex. 4.15.1 :** Apply Dijkstra and Bellman Ford algorithm to given network as shown in Fig. P. 4.15.1 and find the least cost path between source node 1 to other nodes.




(G-1491) Fig. P. 4.15.1

**Soln. :**

**Part I : Dijkstra's algorithm : Refer Ex. 4.14.4.**

**Part II : Bellman Ford algorithm :**

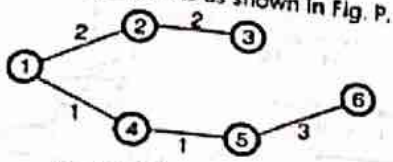
Let node 1 → A, 2 → B, 3 → C, 4 → D, 5 → E, 6 → F

- Node 1 is source node.
- Distance from node 1 to all other nodes is as shown in Table P. 4.15.1(a).

(G-2893)Table P. 4.15.1(a)

| Node | 1 arc distance | 2 arcs distance | 3 arcs distance |
|---|---|---|---|
| A | 0 | 0 | 0 |
| B | 2 | – | – |
| C | 4 | 4 (Due to B) | 7 (Due to D) |
| D | 1 | 4 (Due to B) | – |
| E | ∞ | 2 (Due to D) | 5 (Due to D) |
| F | ∞ | 9 (Due to C) | 5 (Due to E) |

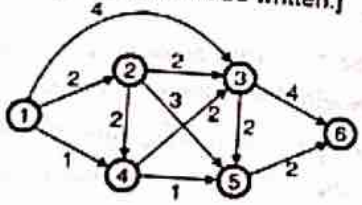
- Shortest path from node 1 to all other nodes using Bellman Ford algorithm is as shown in Fig. P. 4.15.1(a).




(G-1494) Fig. P. 4.15.1(a)

- The shortest path obtained from Dijkstra's and Bellman Ford algorithm is same.

**Ex. 4.15.2 :** Apply Djikstra's and Bellman-Ford routing algorithms to given network shown in Fig. P. 4.15.2 and find the least cost path from source node-1 to all other nodes. [Note : Steps of algorithms must be written.]




(G-1540) Fig. P. 4.15.2

**Soln. :**

**Part 1 : Using Dijkstra's algorithm :**

Let node 1 → A, 2 → B, 3 → C, 4 → D, 5 → E, 6 → E

**Step 1 :**

- Starting node is A. Enter it into group P as shown in Table P. 4.15.2(a).

(G-2894)Table P. 4.15.2(a)

| Permanent (P) | Temporary (T) |
|---|---|
| A | B (A, 2), D (A, 1), C (A, 4) |

- Add neighbours B and D to temporary group T.

**Step 2 :**

- Now pick up neighbour with smallest cost i.e. D and add it to group P.
- As D is added to P group we have to add neighbours of D to T group as shown in Table P. 4.15.2(b).

(G-1541) Table P. 4.15.2(b)

| Permanent (P) | Temporary (T) |
|---|---|
| A | B(A, 2), D(A, 1), C(A, 4) |
| A, D(A, 1) | B(A, 2), C(D, 3), E(D, 2), C(A, 4) |

**Step 3 :**

- Now move E (D, 2) from T to P and add neighbour F to T group as shown in Table P. 4.15.2(c).

(G-2895)Table P. 4.15.2(c)

| Permanent (P) | Temporary (T) |
|---|---|
| A | B (A, 2), D (A, 1), C (A, 4) |
| A, D (A, 1) | B (A, 2), C (D, 3), E (D, 2), C (A, 4) |
| A, D (A, 1), E (D, 2) | B (A, 2), C (D, 3), F (E, 4) |

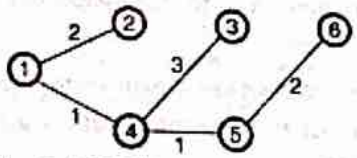
**Step 4 :**

- Now continue in the same manner to get final table as shown in Table P. 4.15.2(d) .

(G-2896)Table P. 4.15.2(d)

| Permanent (P) | Temporary (T) |
|---|---|
| A | B (A, 2), D (A, 1), C (A, 4) |
| A, D (A, 1) | B (A, 2), C (D, 3), E (D, 2), C (A, 4) |
| A, D (A, 1), E (D, 2) | B (A, 2), C (D, 3), F (E, 4) |
| A, D (A, 1), E (D, 2), B (A, 2) | C (B, 5), E (B, 5), D (B, 4), C (D, 4), F (E, 4) |
| A, D (A, 1), E (D, 2), B (A, 2), C (D, 4) | E (B, 5), D (B, 4), F (C, 8), E (C, 6), F (E, 4) |
| A, D (A, 1), E (D, 2), B (A, 2), C (D, 4), F (E, 4) | Null (stop) |

- Shortest path from node 1 to all other nodes is shown in Fig. P. 4.15.2(a).




(G-1542) Fig. P. 4.15.2(a) : Shortest path from node 1 to all other nodes

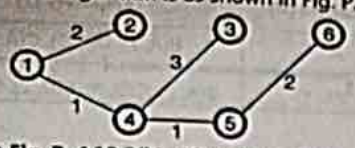
**Part 2 : Using Bellman Ford algorithm :**

- Let node 1 → A, 2 → B, 3 → C, 4 → D, 5 → E, 6 → F
- Node 1 is source node. Distance from node 1 to all other nodes is as shown in Table P. 4.15.2(e).

(G-2897)Table P. 4.15.2(e)

| Node | 1 arc distance | 2 arcs distance | 3 arcs distance |
|---|---|---|---|
| A | 0 | 0 | 0 |
| B | 2 | – | – |
| C | 4 | 4 (Due to D) | 7 (Due to D) |
| D | 1 | 4 (Due to B) | – |
| E | ∞ | 2 (Due to D) | 5 (Due to D) |
| F | ∞ | 8 (Due to C) | 4 (Due to E) |

- Shortest path from node 1 to all other nodes using Bellman Ford algorithm is as shown in Fig. P. 4.15.2(b).




(G-1543) Fig. P. 4.15.2(b) : Shortest path from node 1 to all other nodes

## 4.16 Path Vector Routing :

- It is different from both distance vector routing and link state routing. Table 4.16.1 shows the example of a path routing table.

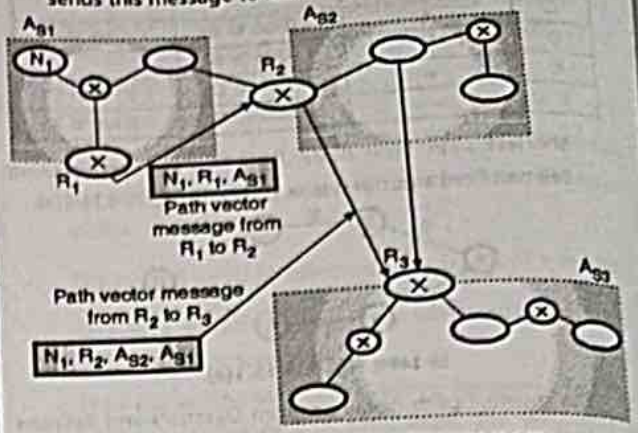
Table 4.16.1 : Path vector routing table

| Network | Next router | Path |
|---|---|---|
| N01 | R01 | AS 12, AS 21, AS 56 |
| N02 | R08 | AS 20, AS 57, AS 06 |
| . | . | . |
| . | . | . |
| . | . | . |

- Each entry in the routing table will have the information about the destination network, the next router and the path to reach the destination.

### 4.16.1 Path Vector Messages :

- The autonomous boundary routers participate in path vector routing.
- Their job is to advertise the reachability of networks present in their A.S. to the neighbour autonomous boundary router.
- Each router that receives a path vector message verifies whether or not the advertised path is according to its policy.
- Such a policy is made up of rules that are imposed by the router controlling administrator. If yes then the router will update its routing table and will modify the message before it is sent to the next neighbour.
- In the modified message it sends its own AS number and replaces the next router entry with its own identification. This process is demonstrated in Fig. 4.16.1.
- Fig. 4.16.1 shows an internet containing three autonomous systems $A_{S1}$ through $A_{S3}$. Router $R_1$ sends a path vector message to advertise that it is reachable to network $N_1$.
- Router $R_2$ on receiving this message will update its routing table. It then adds its own autonomous system ($A_{S2}$) to the path, inserts itself as the next router and sends this message to router $R_3$ as shown in Fig. 4.16.1.




(G-1788) Fig. 4.16.1 : Path vector messages

### 4.16.2 Loop Prevention :

- When a message is received, a router checks it to see if its autonomous system is in the path list to the destination.
- If it is present it indicates looping is involved which is undesirable and the message is ignored.
- In this way the looping problem and the associated instability which is present in distance vector routing is avoided in path vector routing.

### 4.16.3 Path Attributes :

- The path is specified in terms of attributes. Each attribute gives some information about the path.
- Hence the list of attributes helps the receiving router to make a better decision about when to apply its policy.
- Attributes are of two types :
  1. A well known attribute
  2. An optional attribute
- An attribute is called as a well known attribute if it is recognised by every BGP router. An optional attribute is the one that need not be recognised by every BGP router.
- The well known attributes are further classified into two categories :
  1. Well known mandatory attributes
  2. Well known discretionary attributes.

- The optional attributes also are classified into two types :
  1. An optional transitive attribute
  2. An optional nontransitive attribute.

## 4.17 Unicast Routing Protocols :

- We can define the unicast communication as the communication between one sender and one receiver.
- In short it is a one to one communication. We have already discussed about how the Internet has been divided in **administrative areas** called **Autonomous Systems** which helps in handling the exchange of routing information efficiently.
- In the following sections we are going to discuss some important routing protocols.

### 4.17.1 Routing :

- An Internet consists of many networks connected to each other by routers.
- A datagram passes through different routers when it travels from the source to destination.

### 4.17.2 Cost or Metric :

- As a router is connected to many networks it has to make a decision when it receives a packet from one of these networks, as to which network it should pass this packet to ?
- The router makes this decision on the basis of **Optimization.**
- That means it finds out which path is an optimum path to send the packet. But how does it define the term **optimum ?**
- One way is that a **cost** is assigned for passing through a network. This cost is also called as the **metric.** In connection with finding the optimum path, the network having a high cost is considered to be **bad** and to have a low cost is considered to be **good.**
- So in order to maximize the throughput the router should choose the networks (paths) having low costs. Similarly in order to minimize the delay, the router must choose the paths having low costs.

### 4.17.3 Routing Tables :

- The routing table for a host or a router consists of an entry for each destination, or a combination of destinations to route the IP packets.
- Routing tables can be of two types :
  1. Static routing tables
  2. Dynamic routing tables

**1. Static routing table :**

- The information in the static routing tables is entered manual.
- The route of a packet to each destination is entered into the table by the administrator.
- This routing table cannot update itself automatically. It has to be changed manually as and when required.
- Hence static routing table is useful only for small networks.

**2. Dynamic routing table :**

- The dynamic routing tables can get automatically updated by using a dynamic routing protocol such as RIP, OSPF or BGB.
- The structure of a dynamic routing table is shown in Table 4.17.1.

**Table 4.17.1 : Format of dynamic routing table**

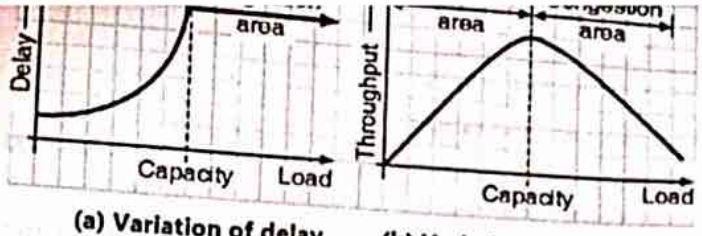
| Mask | Network Address | Next hop address | Interface | Flags | Reference count | Use |
|---|---|---|---|---|---|---|
| | | | | | | |
| | | | | | | |

## 4.18 Network Layer Congestion :

- The Internet model does not explicitly deal with the congestion at the network layer.
- However this study of network layer congestion is very important because it helps us to understand the congestion at the transport layer better.
- It also helps us to find remedies on congestion. The two important network performance issues that are related to the congestion at network layer are :
  1. Throughput and
  2. Delay
- The variation of these performance parameters with respect to load has been shown in Fig. 4.18.1(a) and (b) respectively.

**1. Variation of packet delay with load :**

- Consider Fig. 4.18.1(a) which shows that packet delay is very small when the load is much less than the capacity of the network.




**(a) Variation of delay with load** **(b) Variation of throughput with load**

**(G-2231) Fig. 4.18.1 : Packet delay and throughput as function of load**

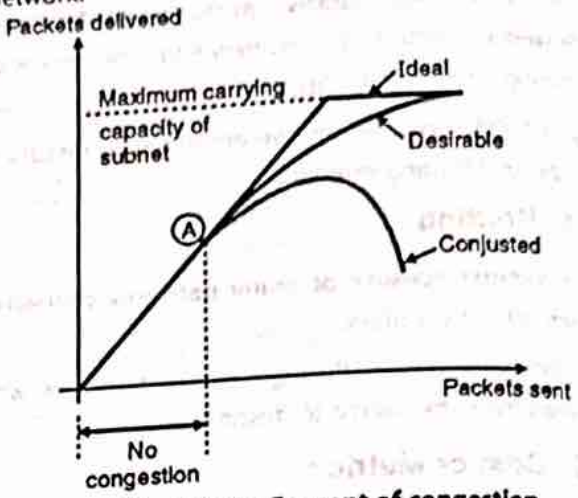
- This small delay is only due to the propagation delay and processing delay both of which have very small values.
- However as the load increases and reaches close to the network capacity the packet delay increases sharply due to the significant increase in the queuing delay.
- If the load is increased beyond the network capacity, the delay will become infinite, and congestion will result.

**2. Variation of throughput with load :**

- Now refer Fig. 4.18.1(b) which shows that the throughput increases with increase in load as long as the load is less than the network capacity.
- It is expected that for any load beyond the capacity, the throughput should remain constant. Instead it decreases sharply as the load exceeds the capacity of the network.
- This sharp reduction in throughput results due to discarding of packets by the routers.
- As the load is higher than the capacity, the queues at the routers overflow and some packets must be discarded.
- But packet discarding does not reduce the number of packets present in the network because every discarded packet is retransmitted by its source due to the time out mechanism.
- Therefore increasing the load beyond the capacity results in the congestion of network.

## 4.19 Congestion Control :

- An important issue in a packet switching network is congestion.
- If an extremely large number of packets are present in a part of a subnet, the performance degrades. This situation is called as **congestion**.
- Congestion... the network i.e. the number of packets sent to the network is greater than the capacity of the network (i.e. the number of packets a network can handle).
- Fig. 4.19.1 explains the concept of congestion graphically.
- Up to point A in Fig. 4.19.1, the number of packets sent into the subnet by the host is within the capacity of the network.




**(G-473) Fig. 4.19.1 : Concept of congestion**

- So all these packets are delivered. In short the number of packets delivered is proportional to number of packets sent and no congestion takes place.
- But after point A, the traffic increases too far. The routers cannot cope with the increased traffic and they begin to lose packets. The congestion begins here.
- As the traffic increases further, the performance degrades more and more packets are lost and congestion worsens.
- At very high traffic, the performance collapses completely and almost all packets are lost. This is the worst possible congestion.

### 4.19.1 Need of Congestion Control :

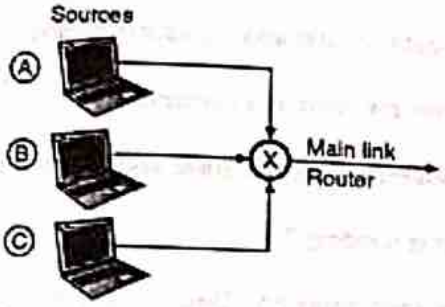
- It is not possible to completely avoid the congestion but it is necessary to avoid it otherwise control it.
- Congestion will result in long queues, which results in buffer overflow and loss of packets.
- So congestion control is necessary to ensure that the user gets the negotiated QoS (Quality of Service).

## 4.19.2 Causes of Congestion :

- Some of the causes of congestion are as follows :

1. **Sudden increase in flow of packets :**

- If suddenly a flow of packets start coming from three or four senders which all needing the same output line.
- Then a queue will become long. If the memory capacity is not sufficient to hold all these packets, some of them will be lost.
- This is shown in Fig. 4.19.2(a).

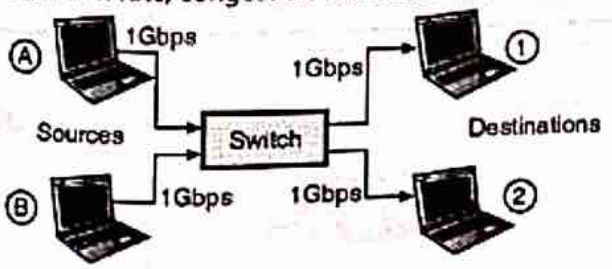



**(G-2232) Fig. 4.19.2(a) : Causes of congestion**

- This leads to congestion. Note that increasing the memory to infinity also does not solve the problem, in fact it worsens.

2. **Presence of slow and low bandwidth links :**

- Congestion is caused by slow and low bandwidth links. The problem will be solved when high speed links become available.
- It is not always the case, sometimes increases in link bandwidth can aggravate the congestion problem because higher speed links may make the network more unbalanced.
- For the configuration shown in Fig. 4.19.2(b), if both the sources begin to send to destination 1 at their maximum rate, congestion will occur at the switch.




**(G-2233) Fig. 4.19.2(b) : Network with high speed links**

- Higher speed links can make the congestion condition in the switch worse.

3. **Use of slow processors :**

- Congestion is caused by slow processors. The problem will be solved when processor speed is improved. Faster processors will transmit more data in unit time.
- If several nodes begin to transmit to one destination simultaneously at their maximum rate, the destination will be overwhelmed soon.

4. **Unwanted retransmission of packets :**

- Congestion can make itself worse. If a router does not have any free buffers it should ignore (discard) new packets arriving at it.
- But when a packet is discarded, the sender may retransmit it many times because it is not receiving the acknowledgement of the packet.
- This multiple transmission of packets will force the congestion to take place at the sending end.

## 4.19.3 Difference between Congestion Control and Flow Control :

- Congestion control makes it sure that the subnet is able to carry the offered traffic i.e. the subnet is able to carry all the packets sent by all the senders to their destinations.
- Congestion control is dependent on the behaviour of all the hosts, all the routers and other factors which reduce the carrying capacity of a subnet.
- On the contrary, the flow control is related to point to point traffic between a sender and its destination.
- Flow control ensures that a fast sender does not send data at a rate faster than the rate at which the receiver can receive it.
- Flow control involves some kind of feedback from the receiver, which can control the sending rate of the sender.
- It does this by comparing its bandwidth, buffer size, CPU speed etc. with the flow specifications.

Unit III

Chapter 5

# Network Layer Protocols

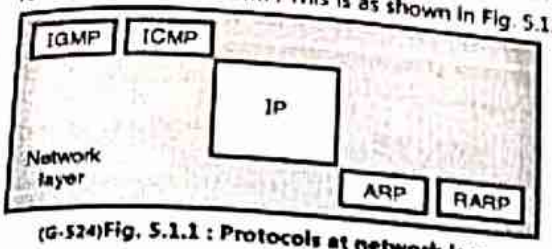
## Syllabus

**IP Protocol :** Classes of IP (Network addressing), IPv4, IPv6, Network address translation, Sub-netting, CIDR. **Network layer Protocols :** ARP, RARP, ICMP, IGMP. **Routing Protocols :** RIP, OSPF, BGP, MPLS. **Routing in MANET :** AODV, DSR, Mobile IP.

## Chapter Contents

## 5.1 Network Layer Protocols :

- The main protocols corresponding to the network layer in the TCP/IP suite as well as Internet layer are : ARP, RARP, IP, ICMP and IGMP. This is as shown in Fig. 5.1.1.




(G-524)**Fig. 5.1.1 : Protocols at network layer**

- Out of these protocols IP is the most important protocol. It is responsible for host to host delivery of datagrams from a source to destination.
- But IP needs to take services of other protocols. IP takes help from ARP in order to find the MAC (physical) address of the next hop.
- IP uses the services of ICMP during the delivery of the datagram packets to handle unusual situations such as presence of an error.
- IP is basically designed for unicast delivery. But some new Internet applications as well as multimedia need multicast delivery.
- So for multicasting, IP has to use the services of another protocol called IGMP.IPv4 is the current version of IP whereas IPv6 is the latest version of IP.

### 5.1.1 Why IP Address ?

- How does the Internet Protocol (IP) know about the source of a datagram and its destination ?
- For a common user the Internet should appear as a single network and all the incompatibilities of the physical networks that make the Internet should remain hidden from the common user.
- Also the people connected to these physical networks should be able use any technology of their choice. So we need to have a common interface which binds the end users of Internet and the people dealing with their own networks.
- To identify each computer connected to the Internet uniquely is a great challenge.
- Different networking technologies have different physical addressing mechanisms.
- A **physical address** is also known as the **hardware address** and there are three methods to assign the hardware address to a computer as follows :
  1. Static addresses
  2. Configurable addresses
  3. Dynamic addresses.

**1. Static addresses :**

- The static address is a physical address which is hard coded in the Network Interface Card (NIC) of the computer.
- This address is provided by the network hardware manufacturer and it does not change ever.

**2. Configurable addresses :**

- In this method, the physical address is configured inside a computer at the time of its first installation at its site.
- The configurable address allows the user to set up a physical address.

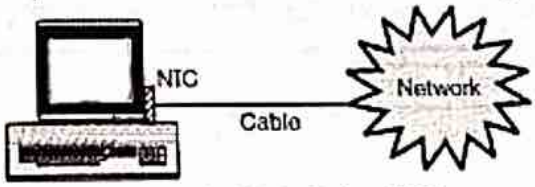
**3. Dynamic addresses :**

- In this method, a server computer dynamically assigns a physical address to a computer every time it boots.
- Thus the physical address of a computer changes everytime it is switch off and on.

**Note :** The method of static addresses is the simplest of all the three methods discussed so far.It is important to understand that every computer has a unique hardware or physical address and it is stored in the NIC of the computer.

**Role of NIC :**

- As discussed earlier, the NIC is an input/output interface on each computer.
- It allows the computer to communicate with all other computers on the network.This is as shown in Fig. 5.1.2.




(G-1439) **Fig. 5.1.2 : Role of NIC**

- The NIC acts as an interface between a computer and its network.

### 5.1.2 Logical Addresses (IP Addresses) :

- Logical addresses are required to facilitate universal communications in which different types of physical networks can be involved.

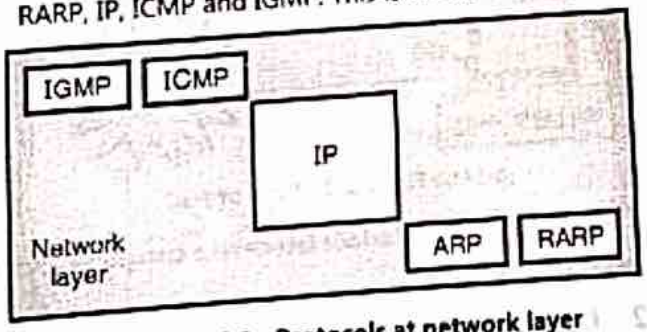
- The logical address is also called as the IP (internet protocol) address.
- The internet consists of many physical networks interconnected via devices like routers.
- Internet is a packed switched network that means the data from the source computer is sent in the form of small packets carrying the destination address upon them.
- A packet starts from the source host, passes through many physical networks and finally reaches the destination host.
- At the network level, the hosts and routers are recognised by their **IP addresses** or logical addresses.
- An IP address is an internetwork address. It is a universally unique address.
- Every protocol involved in internetworking requires IP addresses.
- The logical address used in internet is currently a 32-bit address. The same IP address can never be used by more than one computer on the Internet.

## 5.2 Internet Protocol Version 4 (IPv4) :

- We have already discussed the addressing mechanism, delivery and forwarding for the IP packets. Now we will discuss the format of IP packet in the next few sections.
- In the discussion we will see that an IP packet consists of a base header and options which are sometimes useful in controlling the packet delivery.

### 5.2.1 Position of IP :

- The main protocols corresponding to the network layer in the TCP/IP suite as well as Internet layer are : ARP, RARP, IP, ICMP and IGMP. This is as shown in Fig. 5.2.1.




(G-524)**Fig. 5.2.1 : Protocols at network layer**

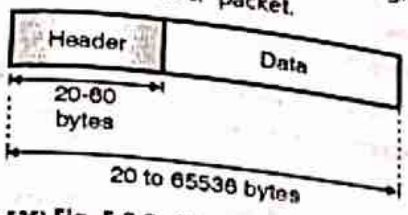
- Out of these protocols IP is the most important protocol. It is responsible for host to host delivery of datagrams from a source to destination.
- But IP needs to take services of other protocols. IP takes help from ARP in order to find the MAC (physical) address of the next hop.
- IP uses the services of ICMP during the delivery of the datagram packets to handle unusual situations such as presence of an error. IP is basically designed for unicast delivery.
- But some new Internet applications as well as multimedia need multicast delivery.
- So for multicasting, IP has to use the services of another protocol called IGMP. IPv4 is the current version of IP whereas IPv6 is the latest version of IP.

### 5.2.2 Internet Protocol (IP) :

- The Internet Protocol is the host to host delivery protocol which belongs to the network layer and is designed for the Internet.
- IP is used as the transmission mechanism by the TCP / IP protocols.
- That means the TCP or UDP packets are encapsulated in the IP packet and the IP carries it from source to destination.
- IP is a connectionless datagram protocol with no guarantee of reliability. It is an unreliable protocol because it does not provide any error control or flow control.
- IP can only detect the error and discards the packet if it is corrupted. If IP is to be made more reliable, then it must be paired with a reliable protocol such as TCP at the transport layer.
- Each IP datagram is handled independently and each one can follow a different route to the destination.
- So there is a possibility of receiving out of order packets at the destination. Some packets may even be lost or corrupted.
- IP relies on a higher level protocol to take care of all these problems. The version of IP that we are going to discuss is called as IPv4 i.e. IP version 4.
- IP is also called as a best effort delivery protocol. The meaning of the term best effort delivery is that the packet can get lost or corrupted or delayed.
- They may arrive out of order at the destination or may create congestion in the network.

**Datagrams :**

- Packets in IP layer are called datagrams. Fig. 5.2.2 shows the typical format of an IP packet.




**(G-525) Fig. 5.2.2 : IPv4 datagram format**

- A datagram has two parts namely the header and data as shown. The length of datagram is not fixed. It varies from 20 bytes to 65536 bytes. The length of the header is 20 to 60 bytes.
- The information necessary for the routing and delivery of the datagram has been stored in the header. The other part of the datagram is the data field which is of variable length.
- It is a custom in TCP/IP to show the header in 4-byte (32 bit) sections.

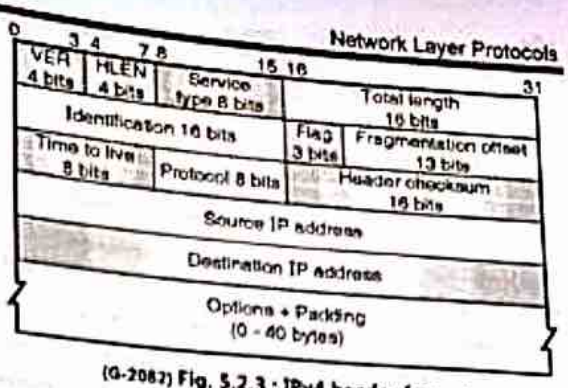
## 5.2.3 Various Network Layer Protocols :

- Various network layer protocols and their functions (significance) are as listed below.

| Sr. No. | Protocol | Function |
|---|---|---|
| 1. | IP | Transports datagram from sender to destination. It acts like the postal service it is responsible for host to host delivery. |
| 2. | ARP | It helps IP to find the MAC (physical address). It maps IP address to MAC address. |
| 3. | ICMP | It is used alongwith IP to report presence of error and sends control message on behalf of IP. It provides feedback on special conditions. |
| 4. | IGMP | It is a group management protocol used in multicasting environment alongwith IP. |
| 5. | RARP | Mapping MAC address to IP address. |

## 5.2.4 IPv4 Header Format :

- The IP frame header contains routing information and control information associated with datagram delivery. The IP header structure is as shown in Fig. 5.2.3.




**(G-2082) Fig. 5.2.3 : IPv4 header format**

- Various fields in the header format are as follows :

**1. VER (Version) :**

- This is a 4 bit field which is used to define the version of IP protocol. The current version of IP is 4 i.e. IPv4 but in future it may be completely replaced by the latest version of IP i.e. IPv6.
- This field will indicate the IP software running on the processing machine that this datagram belongs to IPv4 version. If the processing machine is using some other version of IP, then the datagram will be discarded.

**2. HLEN (Header length) :**

- This 4-bit long field is used for defining the length of the datagram header in 4-byte words.
- The value of this field is multiplied by 4 to get the length of the IPv4 header which varies between 20 and 60 bytes. When there are no options, the value of this field is 5 and the **header length** is 5 × 4 = 20 bytes.
- When the value of option field is maximum the value of HLEN field is 15 and the corresponding header length is maximum i.e. 15 × 4 = 60 bytes.

**3. Service type :**

- In the earlier designs of IP header, this field was called as **Type of Service (TOS)** field and its job was to define how the datagram should be handled.
- At that time, a part of this field used to define the precedence of datagram and the remaining part used to define the type of service out of different possible services such as low delay, high throughput etc.
- But now the interpretation of this field has been changed by IETF.
- This field is now supposed to define a set of **differential services.** Fig. 5.2.4 illustrates the new interpretation of the **service type** field.


(G-2083) **Fig. 5.2.4 : New interpretation of service type field**

- As seen in Fig. 5.2.4, in the new interpretation, the service type field is divided into two subfields namely, the 6 bit **codepoint** subfield and a 2 bit **unused** subfield.
- We can use the 6-bit **codepoint** subfield in two different ways, as follows :
  1. For the purpose of precedence interpretation.
  2. For the differential service interpretation.

**Precedence interpretation :**

- If the three right most bits are zeros, then the three leftmost bits are interpreted the same as the precedence bits in the service field (old interpretation).
- That means it is compatible with the old interpretation of this field.
- The precedence interpretation is used for defining the priority level of this datagram (from 0 to 7) in the situations like congestion.
- In the event of congestion, the datagrams with lowest precedence (0) will be discarded first.

**Differential service interpretation :**

- When the three rightmost bits are not all zeros, the 6 bit codepoint subfield is used for differential service interpretation.
- In that case these 6 bits can be used for defining a total of 56 (64 – 8) services, on the basis of the priorities assigned by the Internet or local authorities as per Table 5.2.1.

**Table 5.2.1 : Values of codepoints**

| Category | Codepoint | Assigning authority |
|---|---|---|
| 1. | xxxxx 0 | Internet |
| 2. | xxxx 1 1 | Local |
| 3. | xxxx 0 1 | Temporary or Experimental |

- The first, second and third categories contain 24, 16 and 16 service types respectively. The Internet authorities assign the first category.

one is temporary and can be used for experimental purposes.

4. **Total length :**

- This 16 bit field is used to define the total length of the IP datagram. The total length includes the length of header as well as the data field.
- The field length of this fields is 16 bits so the total length of the IP datagram is restricted to $(2^{16} - 1)$ = 65535 bytes out of which 20 to 60 bytes constitute the header and the remaining bytes are reserved to carry data from upper layers.
- This field allows the length of a datagram to be upto 65,535 bytes, although such long datagrams are impractical for most hosts and networks.
- All hosts must be prepared to accept datagram of upto 576 bytes, regardless of whether they arrive whole or in the form of fragments.
- The hosts are recommended to send datagram larger than 576 bytes only if the destination is prepared to accept larger datagram.
- We can find the length of data by subtracting the header length from the total length. As stated earlier the header length can be obtained by multiplying the contents of HLEN field by four.

  ∴ Length of data = Total length – header length
- The total length (maximum value) of 65,535 bytes might seem to be large but in future the size of IP datagram is likely to increase further because the improvement in technology will allow more bandwidth.

**Why do we need the total length field ?**

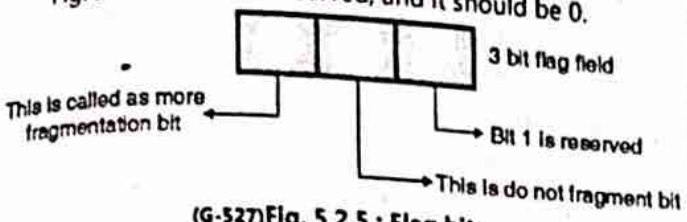
- We might feel that the **total length** field is not at all required because the host or router will drop the header and trailer when it receives a frame. Then why to include this field ?
- The answer to this question is that in many situations we do not need this field at all.
- But in some special situations, only the datagram is not encapsulated in the frame but there are some padding bits as well that are included.
- In such situations, the machine (host or router) that decapsulates the datagram, needs to check the **total length** field so as to understand how much is the data and how much is the padding ?

5. **Identification :**

- This field is used to identify the datagram originating from the source host. When a datagram is fragmented, the contents of the identification field get copied into all fragments.
- This identification number is used by the destination to reassemble the fragments of the datagram.

6. **Flags :**

- **Flags :** This is a three bit field. The 3 bits are as shown in Fig. 5.2.5. First bit is reserved, and it should be 0.

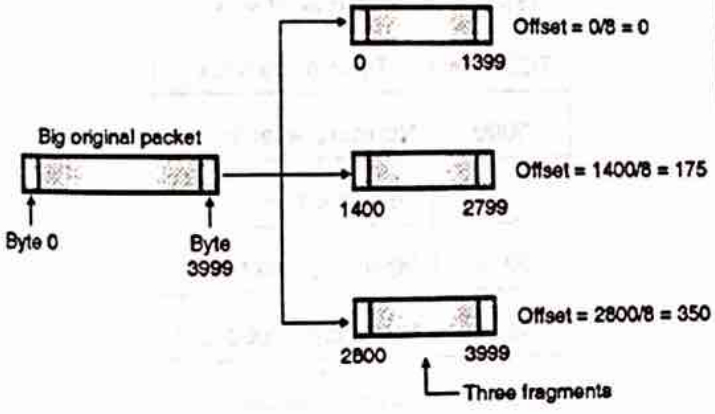



**(G-527)Fig. 5.2.5 : Flag bits**

- The second bit is known as the "Do Not Fragment" bit. If this bit is "1" then machine understands that the datagram is not to be fragmented. But if the value of this bit is 0 then the machine should fragment the datagram if and only if necessary.
- The third bit is known as "More Fragment Bit" (M). M = 1 indicates that the datagram is not the last fragment and M = 0 indicates that this is the last or the only fragment.

7. **Fragmentation offset :**

- This is a 13 bit field which is used to indicate the relative position of this fragment with respect to the complete datagram.
- It is the offset of the data in the original datagram measured in units of 8 bytes. To understand this refer Fig. 5.2.6.




**(G-528)Fig. 5.2.6 : Example of fragmentation**

- The original IP packet (datagram) contains 4000 bytes numbered from 0 to 3999. It is fragmented into three fragments.
- The first fragment contains 1400 bytes numbered from 0 to 1399. The offset for this fragment is 0/8 = 0. Similarly the offsets for the other two fragments are 1400/8 = 175 and 2800/8 = 350 respectively as shown in Fig. 5.2.6.
- The offset is measured in units of 8 bytes. Because the length of the offset field is 13 bits, so the fragments should be of size such that first byte number is divisible by 8.

8. **Time to Live (TTL) :**

- This is an 8-bit field which controls the maximum number of routers visited by the datagram during its lifetime. A datagram has a limited lifetime for travelling through an Internet.
- Originally the TTL field was designed to hold the **timestamp.** This timestamp value was decremented by one, everytime the datagram visits a router.
- As soon as the timestamp value reduces to zero the datagram is discarded.
- But for this scheme to become successful, all the machines must have synchronized clocks and they must know the time taken by a datagram to travel from one router to the other.
- Today the TTL field is used to **control** the maximum number of hops i.e. router by a datagram. At the time of sending a datagram, the source host will store a number in the TTL field.
- This number is approximately twice the maximum number of routers present between any two hosts.
- Everytime this datagram visits a router, this value is decremented by one.
- If after decrementing, the value of TTL field reduces to zero then that router discards the datagram.

**Need of TTL field :**

- Sometimes the routing tables in the Internet get corrupted, due to which a datagram may travel between two or more routers for a very long time but never ever gets delivered to the destination host.
- The TTL field is needed in such situations for **limiting the lifetime of a datagram.**
- The TTL field is also used to limit the journey of a packet intentionally.

- For example if a packet is to be confined to a local network only then a 1 is stored in the TTL field of this packet.
- As soon as it reaches the first router, then TTL field value is decremented from 1 to 0 and the packet will be discarded.

**9. Protocol :**

- This is an 8-bit field which is used for defining the higher level protocol which uses the services of IP layer.
- The data from different high level protocols can be encapsulated into an IP datagram. These protocols could be UDP, TCP, ICMP, IGMP etc.
- The protocol field contents would tell the name of the protocol at the final destination to which this IP datagram is to be delivered.
- At the destination, the value of this field helps in the process of demultiplexing. Table 5.2.2 shows some of the values of this field corresponding to different high level protocols.

**Table 5.2.2**

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

**10. Header checksum :**

- A checksum in IP packet covers on the header only. Since some header fields change, this field is recomputed and verified at each point that the Internet header is processed.

**11. Source address :**

- This field is used for defining the IP address of the source. It is a 32 bit field.

**12. Destination address :**

- This field is used for defining the IP address of the destination. It is also a 32 bit field.

**13. Options :**

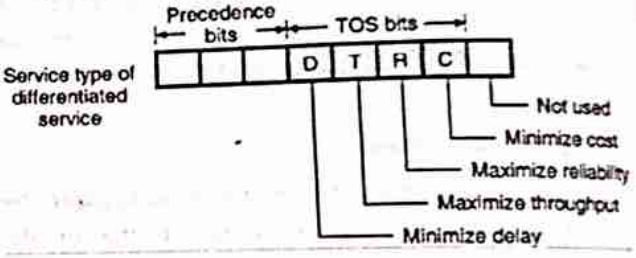
- Options are not required for every datagram. They are used for network testing and debugging.
- We have discussed all the options in detail, later in this chapter.

**Ex. 5.2.1 : Identify from IPv4 header :**

1. Which field gives number of hops count ?
2. What is the minimum and maximum length of HLEN ?
3. What are the differentiated services. Explain TOS bits ?
4. Which fields are related to fragmentation process ?
5. How to calculate total length ? Give the functional difference between IPv4 and IPv6.

**Soln. :**

1. The contents of Time To Live (TTL) field gives the number of hops. Every time the IP datagram is processed by a router the contents of TTL field will be decremented by 1.
2. The minimum contents of HLEN field is 5 i.e. 0101 corresponding to the minimum header length of 20 bytes and maximum contents will be 15 corresponding to the maximum header length of 60 bytes.
3. **TOS Bits :** The 8 bit **service type** field is also called as differentiated services. As shown in Fig. P. 5.2.1(a) the $4^{th}$ to $7^{th}$ bits are called as the TOS bits. Their meaning is included in Fig. P. 5.2.1(a).




**(G-1777) Fig. P. 5.2.1(a) : TOS bits and their meaning**

- Various combinations of the TOS bits will define the service type as shown in Table P. 5.2.1.

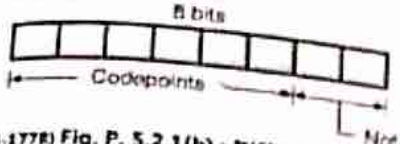
**Table P. 5.2.1 : Types of service**

| TOS bits | Type of service |
|---|---|
| 0000 | Normal (Default) |
| 0001 | Minimize cost |
| 0010 | Maximize reliability |
| 0100 | Maximize throughput |
| 1000 | Minimize delay |

**Differentiated service :**

- The interpretation and name of the 8 bit **service type field** has been changed by IETF as **Differentiated services.**

- As shown in Fig. P. 5.2.1(b) the first 6-bits correspond to the codepoint subfield while the last two bits are not used for any purpose.




(G-1778) **Fig. P. 5.2.1(b) : Differentiated services**

- The values of codepoints decide the assigning authority as shown in Table P. 5.2.1

**Table P. 5.2.1 : Values of codepoints**

| Category | Codepoint | Assigning authority |
|---|---|---|
| 1. | xxxxx 0 | Internet |
| 2. | xxxx 1 1 | Local |
| 3 | xxxx 0 1 | Temporary or Experimental |

**Note :** The above mentioned assignments have not been finalized yet

4. The fields related to the fragmentation process are : Identification (16 bits), Flags (3 bits) and Fragmentation offset (13 bits).
5. The total length of IP datagram (header + data) is calculated from the contents of the 16 bit field called "total length".

- Refer section 5.12 for the difference between IPv4 and IPv6.

**Ex. 5.2.2 :** Solve the following related of IP datagram :

(a) Which field shows number of hop count.

(b) If HLEN value is 5 and length of data is 24 bytes, calculate option.

(c) What are differentiate services ?

(d) Packet version of 010 is discarded. Justify.

**Soln. :**

(a) The time to live field controls the maximum number of routers visited by the datagram. So this field will control the number of hop count.

(c) Differentiated service (DS) is an 8 bit field. Its job is to define the class of datagram for QoS purpose.

(d) Packet version (VER) is a 4 bit long field which defines the version of IP. The current version is IPv4 and the latest version is IPv6. VER = 010 indicates IPv2 hence it is discarded.

## 5.3 Fragmentation :

- The network designers are not free to choose any size of the packet.

**Factors deciding the size of packets :**

- The maximum packet size varies network to network and the factors which decide the maximum packet size are as follows :
  1. Width of the TDM transmission slot.
  2. Protocols used.
  3. Type of operating system.
  4. International standards.
  5. Efforts to reduce retransmission.
  6. Desire to prevent one packet from occupying the channel too long.
- All these factors put a limit on the maximum packet size.
- The maximum payload size ranges from 48 bytes for an ATM cell to 65,515 bytes for an IP packet.
- When a large packet wants to travel over a network whose maximum packet size is very small, we face a problem.
- The solution to this problem is to avoid this situation in the first place by using a routing algorithm which will avoid sending packets through the networks that cannot handle them.
- But this solution cannot be exercised every time. The real solution to this problem is **Fragmentation.**

**Fragmentation :**

- The technique in which, the gateways break up large packets into smaller ones called as fragments.
- After fragmentation of a large packet, each fragment is sent as a separate internet packet.
- But the reverse process of putting the fragments together is substantially difficult.

**Recombination of fragments :**

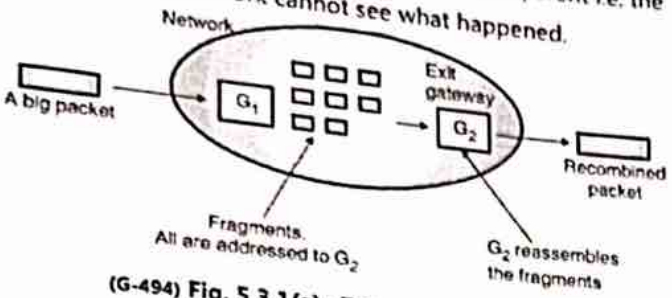
- The recombination of fragments can be done by using one of the following two strategies.
  1. Transparent strategy and
  2. Non-transparent strategy

### 5.3.1 Transparent Strategy :

- In this strategy, the fragmentation caused by a "small packet" network is made transparent to any subsequent network through which the packets will pass.

Fig. 5.3.1(a) it breaks the packet into fragments.

- Each fragment is then addressed to the same exit gateway. The exit gateway ($G_2$) recombines all these fragments.
- This strategy is illustrated in Fig. 5.3.1(a). In this way the small packet network has been made transparent i.e. the rest of the network cannot see what happened.




**(G-494) Fig. 5.3.1(a) : Transparent strategy**

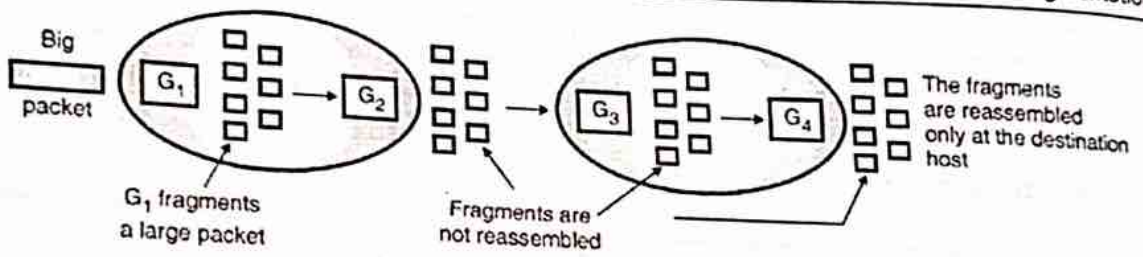
- The subsequent networks are not even aware that fragmentation has taken place. Fragmentation in ATM networks is called segmentation, but the concept is same.

**Disadvantages :**

- The disadvantages of transparent fragmentation are :
  1. The first problem with transparent fragmentation is that the exit gateway $G_2$ has to know that it has received all the pieces. For this a count field or an end of packet bit has to be included in each packet.
  2. Another important factor is that all the packets should exit via the same gateway.
  3. The last problem is the overhead required to repeatedly fragment and reassemble a large packet.

### 5.3.2 Non-transparent Strategy :

- In this strategy, the fragmented packets are not reassembled at any intermediate stage. That means the exit gateways will not reassemble the fragments.
- Instead each fragment is treated as a separate original packet. All these packets are passed through the exit gateway or gateways and their recombination is carried out at the destination host as shown in Fig. 5.3.1(b).
- This is called as a non-transparent fragmentation.




**(G-495) Fig. 5.3.1(b) : Non-transparent strategy**

**Advantage :**

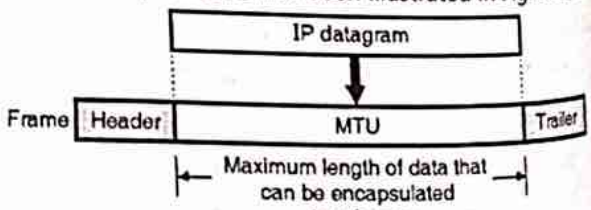
- The advantage of non-transparent strategy is that now we can use multiple exit gateways and improve the network performance.

**Disadvantages :**

- The disadvantages of non-transparent fragmentation are :
  1. Every host must be capable of reassembling the fragments.
  2. The total overhead increases due to fragmentation since each fragment has to have a header.
  3. When a packet is fragmented, the fragments will have to be numbered in such a way that the original data stream can be reconstructed at the destination.

### 5.3.3 Maximum Transfer Unit (MTU) :

- The frame format of each data link layer protocol is different in its own way.
- One of the important field in the frame format is the **maximum size of data field.**
- Therefore when we encapsulate an IP datagram in a frame, the datagram size should be less than the maximum data size specified by the maximum size field.
- The concept of MTU has been illustrated in Fig. 5.3.2.




**(G-2084) Fig. 5.3.2 : Concept of MTU**

- Now the problem is that the value of MTU changes from one protocol to the other used for the physical network.
- We have to make the IP protocol independent of the physical network. In order to do so the maximum length of IP datagram was decided to be equal to 65,535 bytes.
- If we use a physical network protocol which has MTU = 65,535 bytes, then the transmission will become more efficient.
- For the other protocols having MTU smaller than 65,535 bytes, the IP datagram is divided into small parts called **fragments** so that they can pass through the physical networks successfully.
- This processes of dividing the IP datagram in smaller parts is called as **fragmentation.**
- The fragmentation generally does not take place at the source because the transport layer there will adjust the segment size in such a way that they will fit in the IP datagrams and data link layer frames.
- After **fragmentation,** each fragment will have its own header. Most of the fields of the original header are copied into the fragment header but some fields are changed.

Such a fragmented datagram can be fragmented further if it comes across a network with even smaller MTU.

The fragmentation of a datagram can be carried by the source host or any router on the route of the datagram.

But the process of reassembly of all the fragments will be carried out only by the **destination host.**

All the fragments of a datagram are free to take any route and we do not have any control over them. In short each fragment acts as an independent datagram.

The reassembly of fragments is not done during the transmission because of the loss of efficiency associated with it.

At the time of fragmentation, all the required parts of the header are copied into the fragments. But the **options** field may or may not be copied as discussed later on.

The following three fields are altered when the host or router fragments a datagram :

1. Flags. 2. Fragmentation offset. 3. Total length.

- The remaining fields in the IP header are copied as it is.
- The value of checksum should be calculated again regardless of fragmentation. And the final point about fragmentation is that only data in a datagram is fragmented.

### 5.3.4 Fields Related to Fragmentation :

- The following three fields in an IP datagram header are related to the fragmentation and reassembly of an IP datagram :
  1. Identification.
  2. Flags and
  3. Fragmentation offset field.

## 5.4 IPv4 Addresses :

- Each computer connected to the Internet should be identified uniquely. The identifier used for this purpose is called as the Internet address or IP address.
- The hosts and routers on the Internet have unique IP addresses. The current version of IP (Internet Protocol) is IPv4 whereas the advanced version is IPv6.
- The IPv4 address is a 32-bit address and it is used for defining the connection of a host or router to the Internet. Thus an IP address is an address of the interface.

### 5.4.1 Uniqueness of IP Addresses :

- The IP address is **unique** and **universal.** That means each IP address defines only **one connection** to the Internet.
- At any given time, no two devices connected to the Internet can have the same IP address.
- But if a device is connected to the Internet via two connections through two different networks, then it can have two different IP addresses.
- All the IPv4 addresses are 32 bit long and they are used in the source address and destination address fields of the IP header.
- The IP addresses for hosts are assigned by the network administrator. For Internet it has to be obtained from the network information center.

### 5.4.2 Address Space :

- The IPv4 protocol has an address space. It is defined as the total number of addresses used by the protocol.

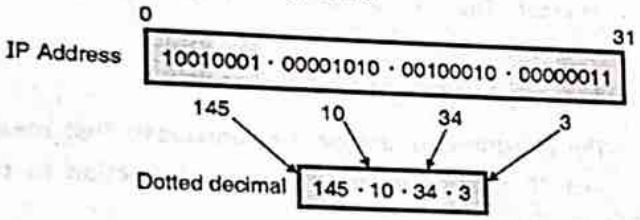
- ...space will be $2^N$ addresses.
- For IPv4, N is 32 bits. Hence its address space is $2^{32}$ or 4, 294, 967, 296 (more than 4 billion). So theoretically more than 4 billion devices could be connected to the Internet. Thus **the address space** of IPv4 is $2^{32}$.

### 5.4.3 Notation :

- The IPv4 addresses can be shown use three different notations as follows :
  1. Binary notations (base 2).
  2. Dotted decimal notation (base 256).
  3. Hexadecimal notation (base 16).
- Out of these the **dotted decimal** notation is most commonly used.

**Dotted decimal notation :**

- This notation has become popular because of the two advantages it offers. This notation makes the IPv4 address more compact and easy to read.
- The 32 bit IPv4 address is grouped into groups of 8-bits each separated by decimal points (dots). Each 8-bit group is then converted into an equivalent decimal number as shown in Fig. 5.4.1.



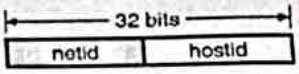

**(G-530) Fig. 5.4.1 : Dotted decimal notation**

- Each octet (byte) can take a value between 0 and 255. Therefore, the IPv4 address in the dotted decimal notation has a range from 0.0.0.0 to 255.255.255.255.
- For example the IPv4 address of 1001 0001.00001010 00100010 00000011 is denoted in the dotted decimal form as 145.10.34.3.

### 5.4.4 IPv4 Address Format :

**Net id and host id :**

- A 32 bit IPv4 address consists of two parts. The first part is called as **net id** i.e. network identification, which identifies a network on the Internet, and the second part is called as the **host id** that identifies a host on that network.
- Fig. 5.4.2 shows the IPv4 address format. Note that the **net id** and **host id** are of variable lengths depending on the class of address.




**(G-2002) Fig. 5.4.2 : IPv4 address format**

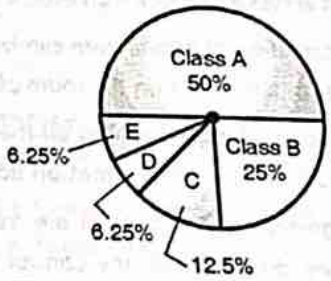
- Note that class D and E addresses are not divided into net id and host id for the reasons discussed later on.

## 5.5 Classful Addressing :

- The concept of IP addresses is few decades old. It uses the concept of **classes.** This architecture is called as the **classful addressing.**
- Later on in mid 1990s a new architecture of addressing was introduced which was known as **classless addressing.**
- This new architecture has superseded the original architecture. In this section we are going to discuss the classful addressing.

### 5.5.1 IPv4 Address Classes :

- In the classful addressing architecture, the IP address space has been divided into five classes : A, B, C, D and E. Fig. 5.5.1 shows the percentage of occupation of the address space by each class.




| Class | No. of addresses | |
|---|---|---|
| A | $2^{31}$ | 50% |
| B | $2^{30}$ | 25% |
| C | $2^{29}$ | 12.5% |
| D | $2^{28}$ | 6.25% |
| E | $2^{28}$ | 6.25% |

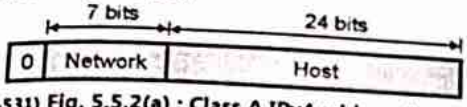
**(G-2003) Fig. 5.5.1 : Classful addressing occupation of address space**

- The number of class A addresses is the highest i.e. 50% and those of classes D and E is the lowest i.e. 6.25%.

## 5.5.2 Formats of Various Address Classes :

**Class A format :**

- The formats used for IPv4 address are as shown in Fig. 5.5.2. The IPv4 address for class A networks is shown in Fig. 5.5.2(a).

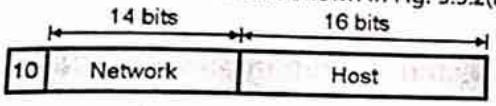



(G-531) **Fig. 5.5.2(a) : Class A IPv4 address formats**

- The network field is 7 bit long as shown in Fig. 5.5.2(a) and the host field is of 24 bit length. So the network field can have numbers between 1 to 126.
- But the host numbers will range from 0.0.0.0 to 127.255.255.255.Thus in class A, there can be 126 types of networks and 17 million hosts. The "0" in the first field identifies that it is a class A network address.

**Class B format :**

- The class B address format is shown in Fig. 5.5.2(b).

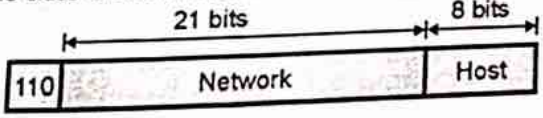



(G-532) **Fig. 5.5.2(b) : Class B format**

- The first two fields identify the network, and the number in the first field must be in the range 128 - 191.Class B networks are large.
- Host numbers 0.0 and 255.255 are reserved, so there can be upto 65,534 ($2^{16}$-2) hosts in a class B network. Most of the 16,382 class B addresses have been allocated.
- The first block covers address from 128.0.0.0 to 128.255.255.255 and the last block covers from 191.255.0.0 to 191.255.255.255.
- **Example :** 128.89.0.26, for host 0.26 on net 128.89.

**Class C format :**

- The class C address format is shown in Fig. 5.5.2(c).




(G-533) **Fig. 5.5.2(c) : Class C format**

- The first block in class C covers addresses from 192.0.0.0 to 192.0.0.255 and the last block covers addresses from 223.255.255.0 to 223.255.255.255.

**Class D format :**

- The class D address format is shown in Fig. 5.5.2(d).

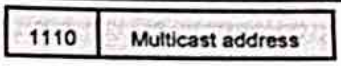



**Fig. 5.5.2(d) : Class D format**

- The class format allows for upto 2 million networks with upto 254 hosts each and class D format allows the multicast in which a datagram is directed to multiple hosts.

**Class E address format :**

- Fig. 5.5.2(e) shows the address format for a class E address.

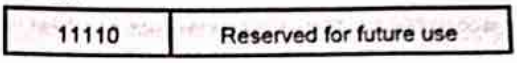
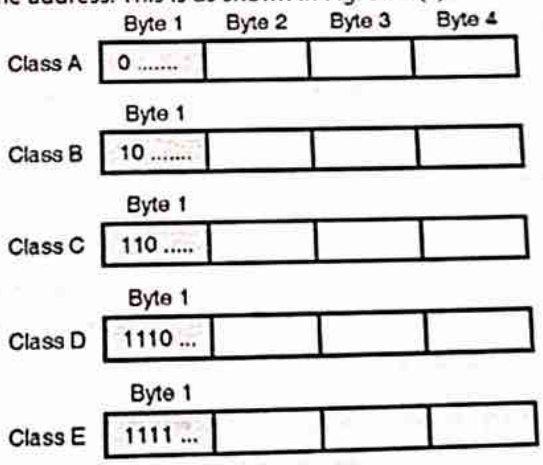



**Fig. 5.5.2(e) : IPv4 address for class E network**

- This address begins with 11110 which shows that it is reserved for the future use. The 32 bit (4 byte) network addresses are usually written in dotted decimal notation.
- In this notation each of the 4-bytes is written in decimal from 0 to 255.So the lowest IP address is 0.0.0.0 i.e. all the 32 bits are zero and the highest IPv4 address is 255.255.255.255.

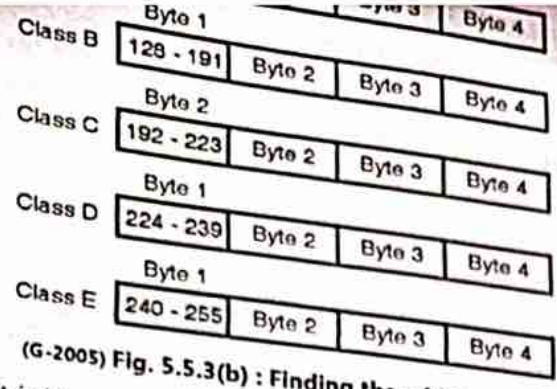
## 5.5.3 How to Recognize Address Classes ?

- When an IPv4 address is given to us either in the binary or dotted decimal notation, we can find the class of the address.
- If the given address is in the binary notation then we can identify its class by inspecting the first few bits of the address. This is as shown in Fig. 5.5.3(a).



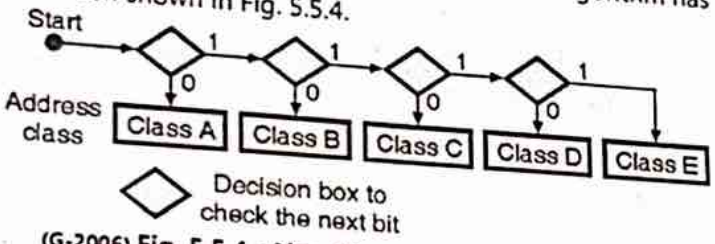

(G-2004) **Fig. 5.5.3(a) : Finding the address class**

- If the given address is in the dotted decimal notation then we can identify the address class by inspecting the first byte of the address. This is as shown in Fig. 5.5.3(b).




**(G-2005) Fig. 5.5.3(b) : Finding the address class**

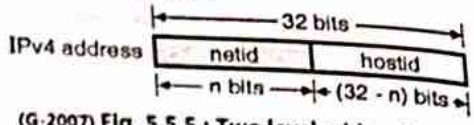
- It is important to note here that there are some special addresses which fall in class A or E.
- These special addresses are to be treated as the exceptions to the classful addressing. We have discussed them later in the chapter.
- In computers, the IPv4 addresses are generally stored in the binary notation format.
- Therefore it is possible to write an algorithm which can identify the address class by using the continuous checking process. The principle of such an algorithm has been shown in Fig. 5.5.4.




**(G-2006) Fig. 5.5.4 : Algorithm to identify address class**

## 5.5.4 Two Level Addressing :

- The IPv4 addressing is used for defining a destination for an Internet packet at the network layer. At the time when classful addresses were designed, the Internet was considered as the network of networks.
- In other words, the whole Internet was divided into a number of smaller networks with many hosts connected to each network.
- Normally an organization, which wants to connect to the Internet, creates a network and the Internet authorities allocate a block of address to the organization. These addresses can be in class A, B or C.
- All the addresses allotted to an organization belong to a single block.
- Therefore each IPv4 address in classful addressing system is made up of two parts namely **net id** and **host id** as shown in Fig. 5.5.5.




**(G-2007) Fig. 5.5.5 : Two level addressing in classful addressing**

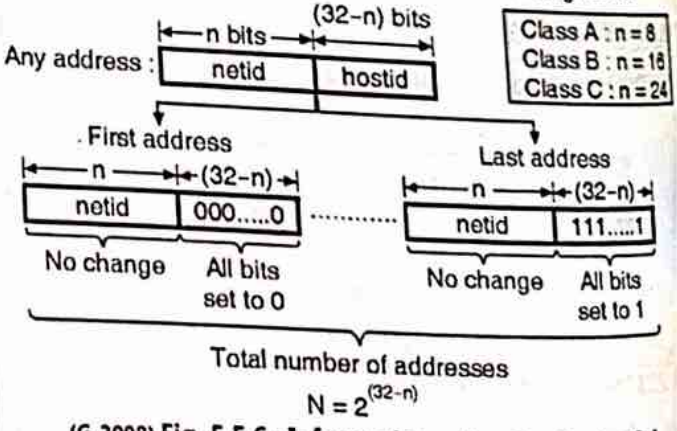
- The function of the **net id** is to define a network and that of the **host id** is to define a particular host in that network.
- As shown in Fig. 5.5.5 if n bits define **net id** then the remaining (32-n) bits define **host id.** The value of "n" is not same for all the classes. Infact it is depend on the class as shown in Table 5.5.1.

**Table 5.5.1**

| Class | Value of n |
|---|---|
| A | n = 8 |
| B | n = 16 |
| C | n = 24 |

## 5.5.5 Extracting Information in a Block :

- A block is nothing but a range of addresses. For any given block we would be interested to extract the following three pieces of information :
  1. The total number of addresses in the block.
  2. The first address of the block.
  3. The last address in the block.
- Before extracting all this information, we have to identify the class of the address as discussed earlier.
- Once we find the class of the block, we will have the values of "n" (the length of **net id** in bits) and (32 – n) i.e. the length of the **host id** in bits.
- It is now possible to obtain the three pieces of information mentioned above as shown in Fig. 5.5.6.




**(G-2008) Fig. 5.5.6 : Information extraction in classful addressing**

1. **Total number of addresses in the block :**

- The total number of IPv4 addresses in the given block will be equal to,

$$N = 2^{(32-n)} \quad \text{...(5.5.1)}$$

2. **First address in the block :**

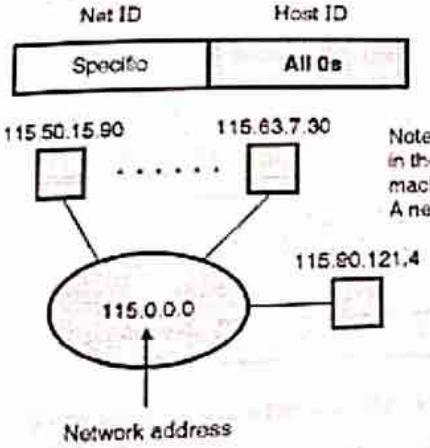
- The first address in the given block can be obtained by keeping the leftmost "n" bits in the address as it is and setting all the (32 – n ) rightmost bits to 0 as shown in Fig. 5.5.6.

3. **Last address in the block :**

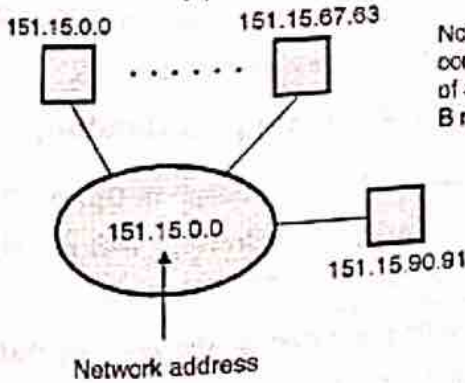
- The last address in the given block can be obtained by keeping the leftmost "n" bits in the address as it is and then setting all the (32 – n) rightmost bits to 1 as shown in Fig. 5.5.6.

## 5.5.6 Network Address :

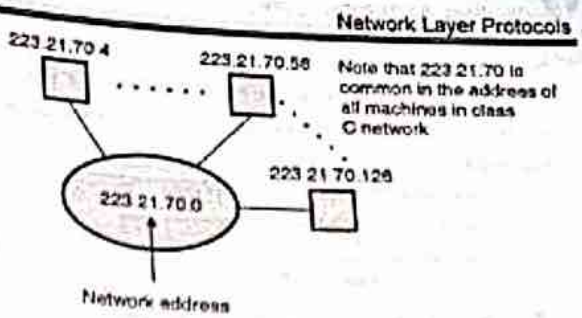
- The network address is an address that defines the network itself.
- It cannot be assigned to a host. Fig. 5.5.7 shows the examples of network addresses for different classes.




**(a) Class A network addres**




**(b) Class B network address**




**(c) Class C network address**

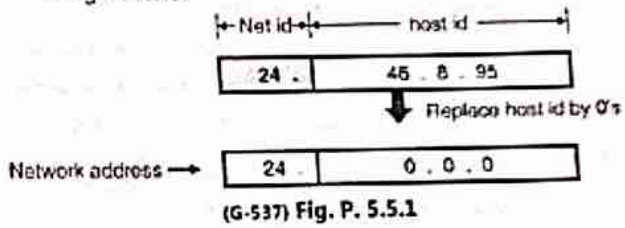
**(G-536) Fig. 5.5.7**

- The following examples will enable you to find the network address.

**Ex. 5.5.1 :** For the address 24.46.8.95 identify the type of network and find the network address.

**Soln. :**

- Examine the first byte. Its value is 24 i.e. it is between 0 and 127. So it is a class A network. So only the first byte defines the Net id. So we can find the network address by replacing the host id with 0s.
- The process of obtaining the network address is shown in Fig. P. 5.5.1.



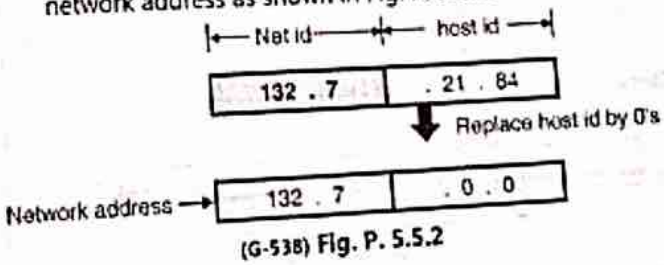

**(G-537) Fig. P. 5.5.1**

- So the network address is 24.0.0.0.

**Ex. 5.5.2 :** For the address 132.7.21.84 find the type of network and the network address.

**Soln. :**

- Examine the first byte. It is 132 i.e. between 128 and 192. So it is a class B network. So the first two bytes define the net id. Replace the host id with 0's to get the network address as shown in Fig. P. 5.5.2.



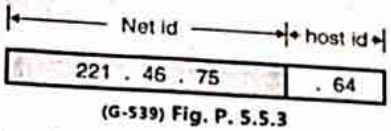

**(G-538) Fig. P. 5.5.2**

- So the network address is 132.7.0.0.

**Ex. 5.5.3 :** Find the class of the network if the address is 221.46.75.64.

**Soln. :**

- The first byte is 221 i.e. between 192 and 255. So this is a class C network. The net id and host id are as shown in Fig. P. 5.5.3.




**(G-539) Fig. P. 5.5.3**

**What is the difference between net id and network address ?**

- The network address is different from a net id. A network address has both net id and host id, with 0s for the host id.

**Where to use the network address ?**

- The network address is used to route the packets to the desired location.

### 5.5.7 Network Mask or Default Mask :

- Earlier we have discussed the methods for extracting different pieces of information. But all these methods are theoretical methods which are useful in explaining the concept.
- But practically these methods are not used. When a packet arrives at the input of the router in the Internet, it uses an algorithm to extract the **network address** from the destination address in the received packet.
- This can be achieved by using a **network mask.**

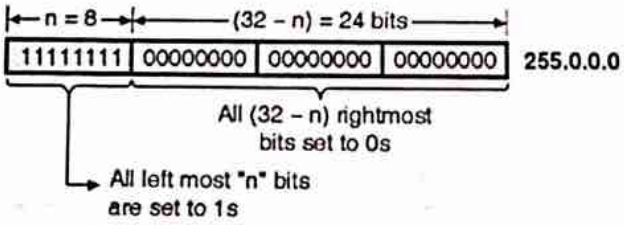
**Definition of default mask :**

- A **network mask** or **default mask** in classful addressing is defined as a 32-bit number obtained by setting all the "n" leftmost bits to 1s and all the (32 – n) rightmost bits to 0.

### 5.5.8 Default Masks for Different Classes :

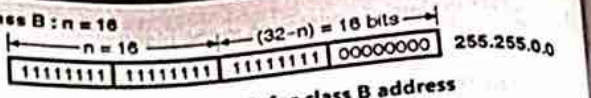
- We know that the value of n is different for different classes.
- Therefore their default masks also will be different. The default masks for class A, B and C addresses are as shown in Fig. 5.5.8.

**Class A : n = 8**



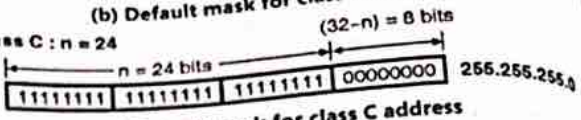

**(a) Default mask for class A address**

**Class B : n = 16**




**(b) Default mask for class B address**

**Class C : n = 24**




**(c) Default mask for class C address**

**(G-2009) Fig. 5.5.8**

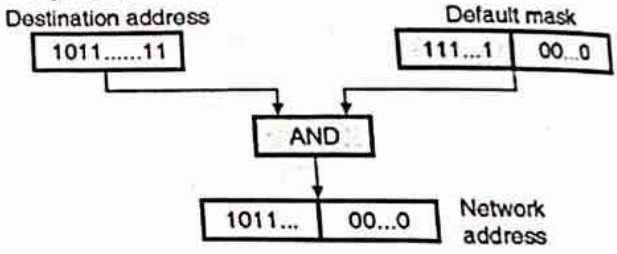
- Table 5.5.2 enlists the default masks of the three classes of IPv4 addresses.

**Table 5.5.2 : Default masks**

| Address class | Default mask |
|---|---|
| A | 255.0.0.0 |
| B | 255.255.0.0 |
| C | 255.255.255.0 |

### 5.5.9 Finding Network Address using Default Mask :

- The router uses the AND operation for extracting the network address from the destination address of the received packet.
- The router ANDs the destination address with the default mask to extract the network address as shown in Fig. 5.5.9.




**(G-2010) Fig. 5.5.9 : Finding a network address using the default mask**

- It is possible to use the default mask to find the number of addresses and the last address in the block.

### 5.5.10 Three Level Addressing : Subnetting :

- As discussed earlier, the originally designed IP addresses were with two level addressing with **net id** and **host id.**
- The two level addressing is based on the principle that in order to reach a host on the Internet, we have to reach the network first and then the host.

- But very soon it became evident that the two level addressing would not be sufficient for the following two reasons :

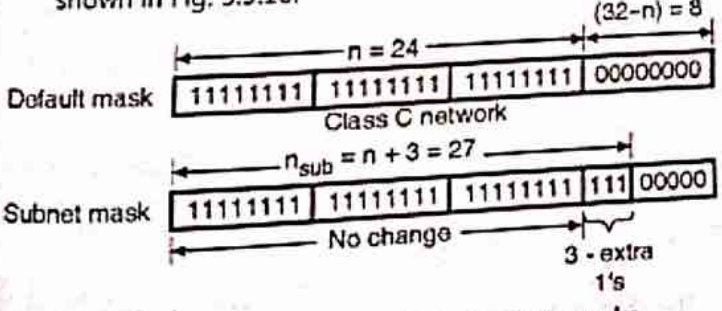
1. First it was needed to divide a large network of an organization (to which a block in class A or B is allotted) into many smaller **subnets** (subnetworks) for improved management and security.
2. Second reason is more important. The blocks in class A and B were almost depleted and the blocks in class C were smaller than the needs of most organization. Therefore the organizations had to divide their allotted class A or B block into smaller subnetworks and share them.

**Definition of subnetting :**

- We can define the subnetting as the principle of splitting a block of addresses into smaller blocks of addresses.
- In the process of subnetting we divide a big network into smaller subnetworks or **subnets.** Each such subnet has its own **subnet address.**

**Subnet mask :**

- The **network mask** or **default** mask that we discussed earlier is used when the given network is **not** to be divided into smaller subnetworks i.e. when **subnetting**is **not** to be done.
- But when the given network is to be divided into smaller subnets i.e. when subnetting is to be done, we need to create a **subnet mask** for each subnet. Fig. 5.5.10 shows the format of a subnet mask. Each subnet has its own **net id** and **host id.**
- If we want to divide a network into 8 subnets then the corresponding subnet mask will have three extra 1's because $2^3 = 8$, as compared to the default mask, as shown in Fig. 5.5.10.




**(G-2011) Fig. 5.5.10 : Default and subnet masks**

- In Fig. 5.5.10, we have shown the default mask and subnet mask when a class C network is to be divided into 8 subnets.

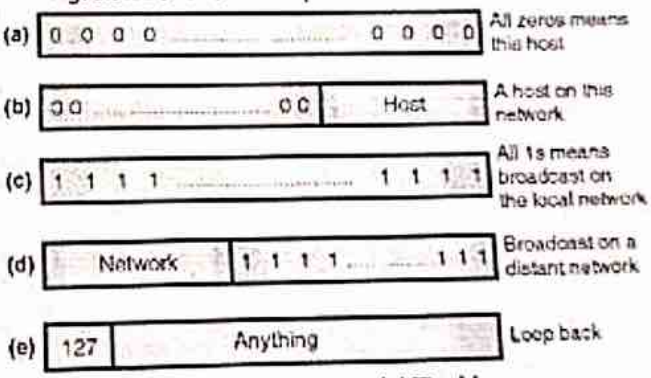
**Difference between subnet mask and default mask :**

**Table 5.5.3 : Difference between subnet mask and default mask**

| Sr. No. | Subnet mask | Default mask |
|---|---|---|
| 1. | To divide a given network address into two or more subnets, subnet mask is used. | The default mask signify a network without subnets. |
| 2. | In office network the subnet mask is used. | In home networks default mask is used. |
| 3 | A subnet mask can be changed as per the hosts / subnets requirement. | A default mask cannot be changed. It is fix for particular address class. |
| 4. | Subnet mask is used to distinguish network part and host part in a IP address | Default mask is subnet mask for a class of network. |
| 5. | Subnet mask represents the number of bits used by network and rest for host. | Default mask is inbuild network portion which is already defined in IPv4 classes. |

## 5.5.11 Special IP Addresses :

- Fig. 5.5.11 shows some special IP addresses.




**(G-540) Fig. 5.5.11 : Special IP addresses**

- All zeros means this host or this network and all 1s means broadcast address to all hosts on the indicated network.
- The IP address 0.0.0.0 is used by the hosts when they are being booted but not used afterward.
- The IP addresses with 0 as the network number refer to their own network without knowing its number as shown in Fig. 5.5.11(b).

- The address having all ones is used for broadcasting on the local network such as a LAN as shown in Fig. 5.5.11(c). Refer Fig. 5.5.11(d). This is an address with proper network number and all 1s in the host field.
- This address allow machines to send broadcast packets to distant LANs anywhere in the Internet. If the address is "127. Anything" as shown in Fig. 5.5.11(e) then it is a reserved address **loopback testing.** This feature is also used for debugging network software.

### 5.5.12 Limitations of IPv4 :

- The most obvious limitation of IPv4 is its address field. IP relies on network layer addresses to identify end-points on networks, and each networked device has a unique IP address. IPv4 uses a 32-bit addressing scheme, which gives it 4 billion possible addresses.
- With the proliferation of networked devices including PCs, cell phones, wireless devices, etc., unique IP addresses are becoming scarce, and the world could theoretically run out of IP addresses.
- If a network has slightly more number of hosts than a particular class, then it needs either two IP addresses of that class or the next class of IP address.
- For example, let us say a network has 300 hosts, this network needs either a single class B IP address or two class C IP addresses.
- If class B address is allocated to this network, as the number of hosts that can be defined in a class B network is $(2^{16} - 2)$, a large number of host IP addresses are wasted.
- If two class C IP addresses are allocated, as the number of networks that can be defined using a class C address is only $(2^{21})$, the number of available class C networks will quickly exhaust.
- Because of the above two reasons, a lot of IP addresses are wasted and also the available IP address space is rapidly reduced.
- Other identified limitations of the IPv4 protocol are: Complex host and router configuration, non-hierarchical addressing, difficulty in re-numbering addresses, large routing tables, non-trivial implementations in providing security, QoS (Quality of Service), mobility and multi-homing, multicasting etc.
- To overcome these problems the internet protocol version 6 (IPv6) which is also known as internet protocol, next generation (IPng) was proposed.
- In IPv6 the internet protocol was extensively modified for accommodating the unforeseen growth of the internet.
- The format and length of the IP addresses has been changed and the packet format also is changed.

### 5.5.13 Classless Addressing :

- Eventhough the number of actual devices connected to Internet is much less than 4 billion, the address depletion has taken place due to flaws in the classful addressing scheme.
- We have run out of class A and B addresses. To overcome these problems, the classless addressing is now being tried out.
- In the classless addressing, there are no classes but the address generation take place in blocks.

**Address blocks :**

- Address block is defined as the range of addresses in the classless addressing, when an entity wants to get connected to the internet, a block (range) of addresses is granted to it.
- The size of this block i.e. number of addresses depends on the size of the entity as well as its nature. That means for a small entity such as a household only one or two addresses will be given whereas for a larger entity like an organization, thousands of addresses can be allotted.

**Restrictions :**

- Some of the restriction on classless address blocks have been imposed by the internet authorities in order to simplify the process of address handling.
  1. The addresses in a block should be continuous i.e. serial in manner.
  2. The total number of addresses in a block has to be equal to some power of 2 i.e. $2^1$, $2^2$, $2^3$ ...etc.
  3. The first address should be evenly divisible by the number of addresses.

### 5.5.14 Supernetting :

- The class A and class B addresses are almost depleted.

- But class C addresses are still available. But the size of class C address with a maximum number of 256 addresses does not satisfy the needs of an organization. More addresses will be required.
- The solution to this problem is **supernetting.** In supernetting an organization combines several class C blocks to create a large range of addresses i.e. several networks are combined to create a supernetwork.
- By doing this the organization can apply for a set of class C blocks instead of just one.

**Example of supernetting :**

- If an organization needs 1000 addresses, they can be obtained by using four C blocks (one C block corresponds to 256 addresses).The organization can then use these addresses as one supernetwork as a whole.

**Note :** The classful addressing is almost obsolete now and it is being replaced with classless addressing.

## 5.5.15 Registered and Unregistered Addresses :

- Registered IP addresses are required for computers which are accessible from the Internet but not every computer that is connected to the Internet.
- For security reasons, networks use firewalls or some other technologies for protecting the computers. The firewalls will enable the workstations to access the Internet but do not allow the other systems on the Internet to access them.
- These workstations are given the unregistered private IP addresses.
- These addresses are assigned by the network administrator without obtaining them from an ISP (Internet Service Provider) or IANA. These are special network addresses in each class as shown in Table 5.5.4.

**Table 5.5.4 : IP addresses for private networks**

| Class | Network address |
|---|---|
| A | 10.0.0.0 through 10.255.255.255 |
| B | 172.16.0.0 through 172.31.255.255 |
| C | 192.168.0.0 through 192.168.255.255 |

- These addresses are to be used for private networks and are called **unregistered addresses.**
- We can choose any of these unregistered address while building our own private network.

## 5.5.16 Solved Examples :

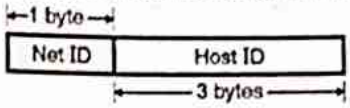
**Ex. 5.5.4 :** Show by calculations how many network each IP address class can have with one example ?

**Soln. :**

**Number of networks in different IP address :**

**Class A address :**

- The format of class A address is shown in Fig. P. 5.5.4(a).

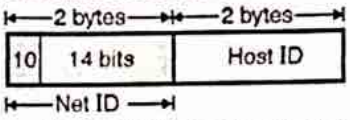



**(G-560) Fig. P. 5.5.4(a) : Class A address**

- Here one byte defines the network ID and three bytes define the host ID. The MSB in the network field is reserved. So actually there are only 7-bits in the network fields. So the number of networks in class A address will be 128.

**Class B address :**

- The format of class B address is shown in Fig. P. 5.5.4(b).

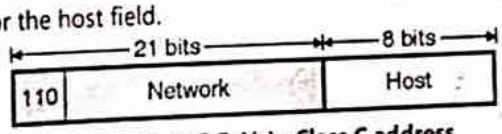



**(G-561) Fig. P. 5.5.4(b) : Class B address**

- Here 2-bytes are reserved for network field and remaining two bytes are for the host field. Out of 16-bits in the network field the first two bits (MSBs) are reserved.
- So actually 14 bits are available in the network field. So the number of networks in class B address is $2^{14}$ = 16, 368.

**Class C address :**

- The format of class C is shown in Fig. P. 5.5.4(c). Here 3-bytes are reserved for network field and only one byte for the host field.




**(G-562) Fig. P. 5.5.4(c) : Class C address**

- Out of 24-bits in the network field 3-bits are again reserved. So actually only 21-bits are available. So the number of networks in class C addresses is 2, 097, 152.

**Ex. 5.5.5 : A company is granted a site address 201.70.64.0. The company needs six subnets. Design the subnets.**

**Soln. :**

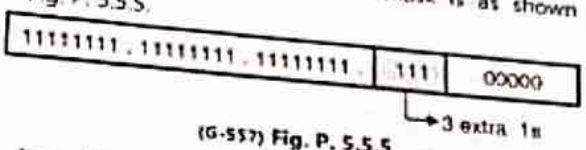
- This is a class C network. So the default mask is,

  255.255.255.0

- As we need 6 subnets, we need three extra 1s. So the subnet mask is,

  255.255.255.200

- In the binary form the subnet mask is as shown in Fig. P. 5.5.5.




(G-557) **Fig. P. 5.5.5**

- In order to have six subnets, we can have 6 different combinations of the 3-extra 1s as shown in Table P. 5.5.5(a).

**Table P. 5.5.5(a)**

| Combination | Subnet number |
|---|---|
| 0 0 0 | Subnet 1 |
| 0 0 1 | Subnet 2 |
| 0 1 0 | Subnet 3 |
| 0 1 1 | Subnet 4 |
| 1 0 0 | Subnet 5 |
| 1 0 1 | Subnet 6 |

- So the various addresses of 6 subnets are as shown in Table P. 5.5.5(b).

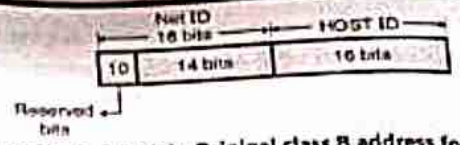
**Table P. 5.5.5(b)**

| Subnet number | Addresses |
|---|---|
| 1 | 201.70.64.0 to 201.70.64.31 |
| 2 | 201.70.64.32 to 201.70.64.63 |
| 3 | 201.70.64.64 to 201.70.64.95 |
| 4 | 201.70.64.96 to 201.70.64.127 |
| 5 | 201.70.64.128 to 201.70.64.159 |
| 6 | 201.70.64.160 to 201.70.64.191 |

**Ex. 5.5.6 : Suppose that instead of using 16-bits for the part of class B address originally, 20-bits had been used. How many class B network addresses would there have been? Give the range of IP addresses in decimal dotted form.**

**Soln. :**

- Fig. P. 5.5.6(a) shows the original class B address format :

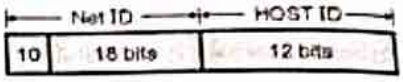



(G-567) **Fig. P. 5.5.6(a) : Original class B address format**

- The first two MSB bits of Net ID part are reserved. Hence, the number of bits actually available for network ID is 14.
- Hence the number of class B networks = $2^{14}$ = 16382.

**Modification :**

- Now with 20 bits instead of 16 being available for the Net ID part the actually available number of bits for Network part becomes 18. This is shown in Fig. P. 5.5.6(b).




(G-568) **Fig. P. 5.5.6(b) : Modified class B address format**

∴ Number of class B networks = $2^{18}$ = 2, 61, 888

- The range of IP addresses in the decimal dotted form would be 128.0.0.0 to 191.255.255.255.

**Ex. 5.5.7 : How many host per network in each IP address class can exist, show with example ?**

**Soln. :**

**Number of hosts in different IP addresses :**

**Class A :**

- There are 3-bytes (24-bits) in the host field.
- Hence the number of hosts in class A address will be $2^{24}$ = 16, 777,216.

**Class B :**

- There are 2-bytes (16-bits) in the host field. So the number of hosts in class B address will be 65536 i.e. $2^{16}$ per network.

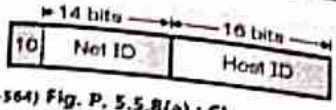
**Class C :**

- There is 1-byte (8-bits) in the host field. So number of hosts in class C address will be $2^{8}$ = 256 per network.

**Ex. 5.5.8 : A class B network on internet has a subnet mask of 255.255.240.0. What is the maximum number of hosts per subnet ?**

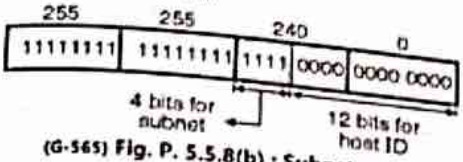
**Soln. :**

- The structure of class B address is as shown in Fig. P. 5.5.8(a).




(G-564) Fig. P. 5.5.8(a) : Class B address

- The given subnet mask is 255.255.2400. So it is as shown in Fig. P. 5.5.8(b).




(G-565) Fig. P. 5.5.8(b) : Subnet mask

- Thus there are 4 extra 1s as shown in Fig. P. 5.5.8(b). So there will be 16 subnets and each subnet can have $2^{12}$ = 4096 hosts.

---

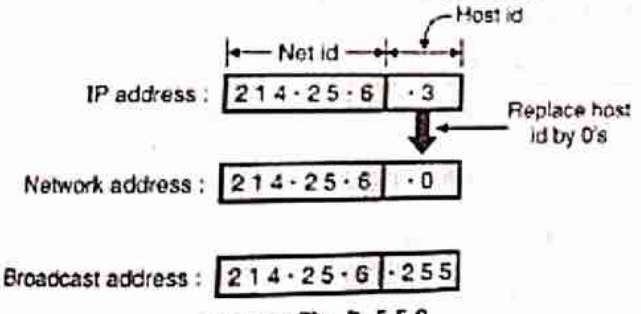
**Ex. 5.5.9 :** Identify class, subnet mask, network address and broadcast address of following IP addresses :

1. 214.25.6.3
2. 191.5.8.9
3. 5.6.45.4
4. 230.45.89.63

**Soln. :**

**1. 214.25.6.3**

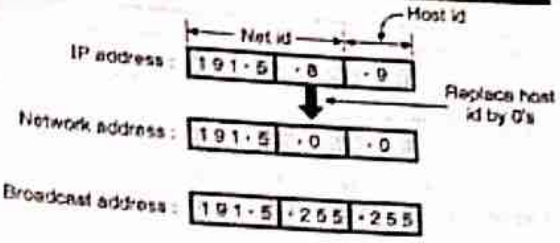
- Examine the first byte. Its value is 214 i.e. it is between 192-223.
- So it is class C network.
- Subnet mask for class C address is 255.255.255.0. The net id and host id are as shown in Fig. P. 5.5.9.




(G-1507) Fig. P. 5.5.9

**2. 191.5.8.9**

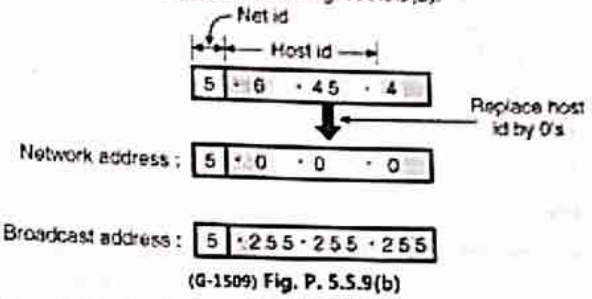
- Examine the first byte. Its value is 191 i.e. it is between 128-191.
- So it is class B network. Subnet mask for class B address is 255.255.0.0. The net id and host id are as shown in Fig. P. 5.5.9(a).




(G-1508) Fig. P. 5.5.9(a)

**3. 5.6.45.4**

- Examine the first byte. Its value is 5 i.e. it is between 0 to 127. So it is class A network.
- Subnet mask for class A network is 255.0.0.0.The net id and host id are as shown in Fig. P. 5.5.9(b).




(G-1509) Fig. P. 5.5.9(b)

**4. 230.45.89.63**

- Examine the first byte. Its value is 230 i.e. it is between 224-239. So it is class D network.The net id and host id are as shown in Fig. P. 5.5.9(c).

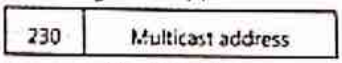



Fig. P. 5.5.9(c)

---

**Ex. 5.5.10 :** Consider a class-C network which needs to be subnetted into 3 subnets. Calculate the appropriate network mask. How many number of hosts can be supported by each subnet ?

**Soln. :**

**Given :** A class C network, 3 subnets.

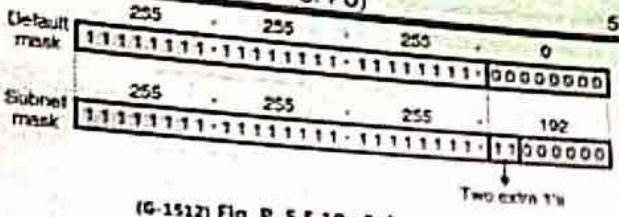
**To find :** 1. Network mask
2. Number of hosts per subnet

**Step 1 : Subnet mask :**

- The default mask for a class C network is

255 · 255 · 255 · 0

- In order to have three subnets, we must have 2 extra 1s. Hence the default mask and subnet mask are as shown in Fig. P. 5.5.10.




(G-1512) Fig. P. 5.5.10 : Subnet mask

**Step 2 : Number of hosts per subnet :**

- The two bits reserved for subnetting will have 4 combinations from 00 to 11, out of which any three combinations can be used for three subnets. We will use the combinations from 00 to 10 and will not use the combination 11.
- Thus each subnet will have six bits for host id. Therefore number of hosts per subnet will be $2^6 = 64$.

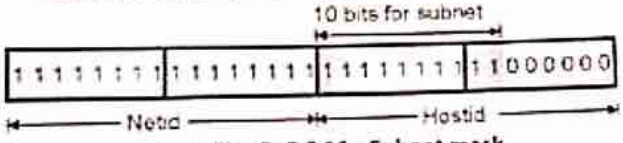
---

**Ex. 5.5.11 :** A company is granted the side address 181.56.0.0 (class B). The company needs 1000 subnets. Design the subnets.

**Soln. :**

**Given :** Class B network : 181.56.0.0

- The default subnet mask is 255.255.0.0. In order to have 1000 subnets we need to use 10 extra bits from the hostid field as shown in Fig. P. 5.5.11.




(G-2635) Fig. P. 5.5.11 : Subnet mask

∴ The new subnet mask is 255.255.255.192.

- The number of subnets is $2^{10} = 1024$.
- The number of address in each subnet is $2^6 = 64$.
- Table 1 shows the addresses of 1000 subnets.

**Table P. 5.5.11**

| Subnet number | Addresses |
|---|---|
| 1 | 181.56.0.0 to 181.56.0.63 |
| 2 | 181.56.0.64 to 181.56.0.127 |
| . | |
| . | |
| . | |
| . | |
| 1022 | 181.56.255.64 to 181.56.255.127 |
| 1023 | 181.56.255.128 to 181.56.255.191 |
| 1024 | 181.56.255.192 to 181.56.255.255 |

---

**Ex. 5.5.12 :** Differentiate between IPv4 and IPv6. Determine the class and network address for the following IP addresses (Assuming subnetting is not being used and use default mask)

1. 84.42.58.11
2. 195.38.14.13
3. 144.62.12.9

**Soln. :**

- For difference between IPv4 and IPv6 refer section 5.12

1. Examine the first byte. It is 84. i.e. between 0 and 127. Hence it is the class A address. The network is is given by 84.0.0.0
2. Examine the first byte. It is 195. i.e. between 192 and 255. So this is the class C address. The network id is given by 195.38.14.0.
3. Examine the first byte. It is 144. i.e. between 128 and 192. So it is a class B network. The network is is given by 144.62.0.0

---

**Ex. 5.5.13 :** Explain the different classes of IP addresses. Identify the class of the following IP addresses and give their default subnet masks :

1. 227.56.83.0
2. 114.22.43.21
3. 129.14.12.1

**Soln. :**

- Refer section 5.5.1 for different classes of IP addresses.

| Sr. No. | IP address | Class | Default subnet mask |
|---|---|---|---|
| 1. | 227.56.83.0 | Class C | 255.255.255.0 |
| 2. | 114.22.43.21 | Class A | 255.0.0.0 |
| 3. | 129.14.12.1 | Class B | 255.255.0.0 |

---

**Ex. 5.5.14 :** Identify the class of each addresses :

1. 14.23.120.8
2. 252.5.15.111
3. 200.58.20.165
4. 128.167.23.20
5. 205.16.37.32

**Soln. :**

1. Class A
2. Class C

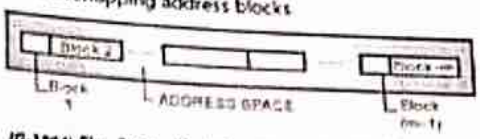
3. Class C
4. Class B
5. Class C

## 5.6 Classless Addressing in IPv4 :

- Eventhough, the number of actual devices connected to Internet is much less than 4 billion, the address depletion has taken place due to flaws in the classful addressing scheme.
- We have run out of class A and B addresses. To overcome these problems, the super netting and subnetting has been tried as discussed earlier.
- But subnetting and supernetting also could not solve the problem of address depletion in IPv4.
- Due to increased number of Internet users, it was evident that a larger address space would be required as a long term solution to this problem.
- For this the length of the IP address should be increased which means the IP packet itself must be changed. A long term solution is to switch to IPv6.
- But a short term solution which uses the same address space has been deviced for IPv4. It is known as **classless addressing.**
- In the classless addressing, there are no classes but the address generation take place in blocks.
- The classless addressing was announced by the Internet authorities in 1996 in which blocks of variable length which do not belong to any class are used.

### 5.6.1 Variable Length Blocks :

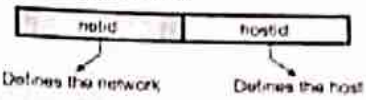
- Address block is defined as the range of addresses. In the classless addressing, when an entity wants to get connected to the Internet, a block (range) of addresses is granted to it.
- The size of this block i.e. number of addresses depends on the size of the entity as well as its nature.
- That means for a small entity such as a household only one or two addresses will be given whereas for a larger entity like an organization, thousands of addresses can be allotted.
- Fig. 5.6.1 shows how the address space is divided into non overlapping address blocks




**(G-1804) Fig. 5.6.1 : Variable length blocks in classless addressing**

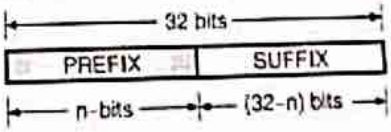
**Two level addressing :**

- We have discussed the two level addressing for classful addressing which divided an address into two parts namely : net id and host id.




**(G-1805) Fig. 5.6.2 : Two level addressing in classful addressing**

- The **net id** and **host id** define the network and host respectively.
- It is possible to use the same idea in the classless addressing as well. A block of addresses granted to an organization is divided into two parts called as the **prefix** and the **suffix.**
- The role of prefix is same as that of the net id whereas as the role of suffix is same as that of the host id.
- Thus in a block granted to an organization, all the addresses will have the **same prefix** but each address will have a different **suffix.**
- Thus the prefix defines the network (organization to which the address block has been granted) while the suffix defines individual hosts on the network.
- The concept of two level addressing in classless addressing using the prefix and suffix is as shown in Fig. 5.6.3.




**(G-1806) Fig. 5.6.3 : Two level addressing using prefix and suffix for classless addressing**

- The IPv4 address is 32 bit long out of which the prefix will be of length "n" which can take any value from 0 to 32 and the length of the suffix will be (32 – n) bits.

- Note that the value of "n" i.e. length of the prefix depends on the length of the address block allotted (granted) to an organization.

**Ex. 5.6.1 :** Find out the values of prefix and suffix lengths in classless addressing if all the available addresses in IPv4 is to be considered as one single block.

**Soln. :**

- The total addresses in IPv4 is $2^{32}$ = 4,294,967,296.
- We have to consider this as one block hence the prefix length n = 0.
- Whereas all the hosts will have their individual addresses. So all the 32 bits will be allotted to the suffix length.

**Ex. 5.6.2 :** For the same data of the previous example find out the values of prefix and suffix lengths if all the available IPv4 addresses are divided into 4,294,967,296 blocks with each block having only one host.

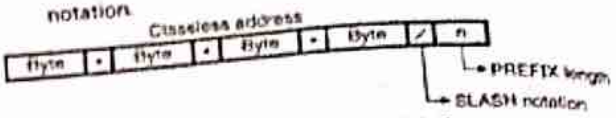
**Soln. :**

- Here the prefix length for each block is n = 32, and the suffix length would be (32-n) = 0.
- The address of the single host in each block will be same as its block address itself.

**Note :** The two previous examples show that the prefix number n and the number of addresses in a block are inversely proportional to each other. With increase in the value of n, the number of addresses in a block will decrease.

## 5.6.2 The Slash Notation (CIDR Notation) :

- If an address (classful or classless) is given to us and we want to extract information from it, then the net id in classful addressing or the prefix in classless addressing are extremely important and useful to us.
- However it is not easy to identify the prefix bits in a given classless address. It is easy to identify the net id from the given classful address.
- For a given classless address it is not possible to find the prefix length because the given address can belong to a block with any prefix length.
- Therefore, in classless addressing it is essential to include the prefix length to each address if the block of the given address is to be found.
- Hence the prefix length "n" is added to the classless address separated by a **slash** and the notation is known as the **slash notation.**
- Fig. 5.6.4 demonstrates a classless address with slash notation.




(G-1807) **Fig. 5.6.4 : Slash notation**

- The slash notation is also called as Classless Interdomain Routing or CIDR notation.

## 5.6.3 Network Mask :

- We have discussed the concept of network mask in the classful addressing. The same concept is also applicable in the classless addressing as well.
- A **network mask** in classless addressing is a 32 bit number.
- With its "n" left most bits (corresponding to the prefix) all set to 1s and the remaining (32-n) bits corresponding to the suffix all set to 0s.

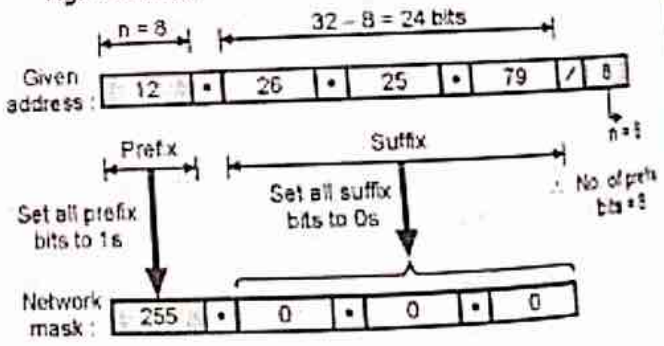
**Ex. 5.6.3 :** For the following addresses identify the number of prefix bits and write down the network mask :

1. 12.26.25.79 / 8
2. 130.12.230.156 / 16

**Soln. :**

**1. Classless CIDR address : 12.26.25.79 / 8**

- As per the slash notation we have n = 8 i.e. number of prefix bits is 8. Therefore the number of suffix bits = 32 – 8 = 24.
- In order to obtain the network mask the prefix bits all set to 1s and the suffix bits all set to zero as shown in Fig. P. 5.6.3(a).

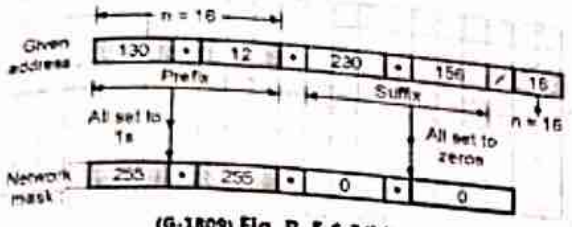



(G-1808) **Fig. P. 5.6.3(a)**

- Thus the network mask = 255.0.0.0

2. **Classless CIDR Address : 130.12.230.158 / 16**

- As per the slash notation, n = 16 i.e. number of prefix bits is 16. Number of suffix bits = 32 – 16 = 16
- In order to obtain the network mask, set all the prefix bits to 1s and set all the suffix bits to 0s as shown in Fig. P. 5.6.3(b).




(G-1809) **Fig. P. 5.6.3(b)**

- Thus the network mask = 255.255.0.0

## 5.6.4 Extracting the Block Information :

- We can extract all the required information from the given classless address in the CIDR notation.
- The information that we can obtain is as follows :
  1. The first address (network address)
  2. The number of addresses.
  3. The last address.
- We can obtain the number of addresses in a block as follows :

  Number of addresses in a block $N = 2^{(32-n)}$ ...(5.6.1)

  Where n = Number of prefix bits.
- The first address or network address in block can be obtained by ANDing the address with the network mask.

  First address = (Any address) AND (Network mask) ...(5.6.2)
- OR what we can do is keep the "n" leftmost bits of any address as it is and set the remaining (32-n) bits to 0s.
- This is equivalent to the ANDing operation mentioned above.
- In order to obtain the last address in the block we have to add the first address with the number of addresses in the block directly.

∴ Last address = First address + Number of addresses in the block ...(5.6.3)

- It is also possible to obtain the last address by ORing the address with complement of the network mask.

∴ Last address = (Any address) OR [NOT (Network Mask)] ...(5.6.4)

- One more way of obtaining the last address of the block is to keep all the "n" left most bits (prefix bits) as it is and set all the (32-n) bits (suffix bits) to 1s.

---

**Ex. 5.6.4 :** If an address in a block is given in CIDR classless notation as 64.32.16.8 / 27 then find the following :

1. Number of addresses in the block (N)
2. The first address and
3. The last address.

**Soln. :**

**Step 1 : Find n :**

Given address = 64.32.16.8 / 27

Hence n = 27 from the slash notation.

∴ n = 27 bits.

∴ Prefix bits = 27, suffix bits

= 32 – 27 = 5

**Step 2 : Number of addresses in the block (N) :**

$$N = 2^{(32-n)} = 2^5 = 32$$

**Step 3 : Find the first address :**

- Refer Fig. P. 5.6.4(a) to obtain the first address in the block. For this we have to AND the given address with the network mask.

  Network mask = [27 ones (n) | 5 zeros (32 – n)] (G-2708)

∴ Network mask = 255.255.255.224

- For ANDing write the given address and network mask in their binary notations as shown in Fig. P. 5.6.4(a).

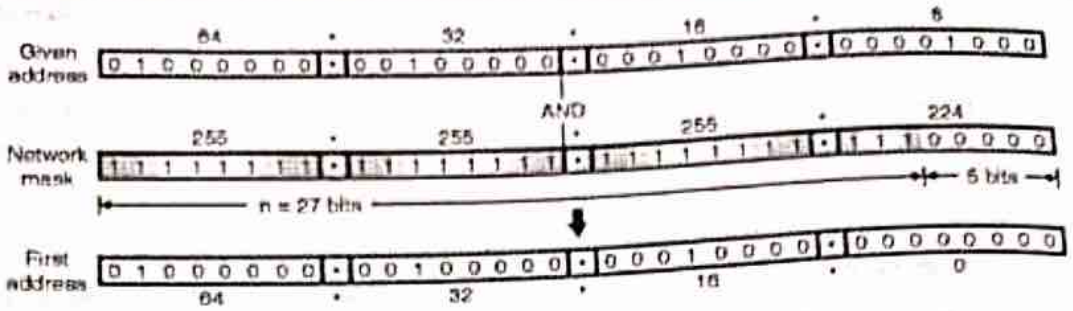
∴ From Fig. P. 5.6.4(a) we get the first address in the block as : ...Ans.

First address = 64.32.16.0 (G-2709)

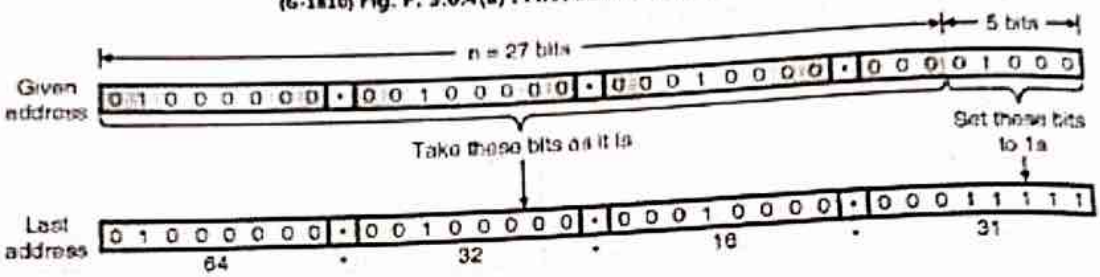
**Step 4 : Find the last address :**

- To obtain the last address in the block, we have to keep the left most 27 bits in the given address as it is and set the remaining 5 bits to 1s as shown in Fig. P. 5.6.4(b).
- From Fig. P. 5.6.4(b) we get the last address in the block as follows :

Last address = 64.32.16.31 (G-2709(a)) ...Ans.




(G-1810) Fig. P. 5.6.4(a) : First address in the block




(G-1811) Fig. P. 5.6.4(b) : Last address

**Ex. 5.6.5 :** For the classless address 129.65.33.01 / 24 find the following :

1. Number of addresses in the block (N)
2. The first address.
3. The last address.

**Soln. :**

**Step 1 : Find n :**

- Given address = 129.65.33.01 / 24 hence n = 24 from the slash notation.

$\therefore$ n = 24 bits

$\therefore$ Prefix bits = 24, suffix bits

= 32 – 24 = 8

**Step 2 : Number of addresses in the block (N) :**

$N = 2^{(32-n)} = 2^8 = 256$ ...Ans.

**Step 3 : Find the first address :**

- Refer Fig. P. 5.6.5(a) to obtain the first address in the block.
- For this we have to AND the given address with the network mask.

Network mask = | n : 24 ones | (32 – n) : 8 zeros |

(G-2710)

$\therefore$ Network mask = 255.255.255.0

- For ANDing write the given address and network mask in their dotted decimal notations as shown.

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Address : | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Network mask : | AND | 2 | 5 | 5 | . | 2 | 5 | 5 | . | 2 | 5 | 5 | . | 0 |
| First address (AND) : | | 1 | 2 | 9 | . | 6 | 5 | | . | | | | . | |

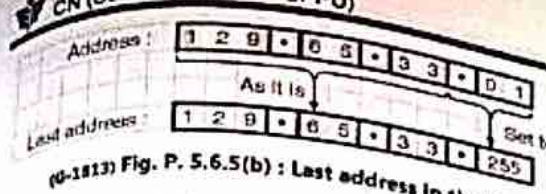
(G-1812) Fig. P. 5.6.5(a) : First address in the block

- From Fig. P. 5.6.5(a) we get the first address in the block as :

(G-2715) First address = 129.65.33.0 ...Ans.

**Step 4 : Find the last address :**

- To obtain the last address in the block, we have to keep the left most 24 bits in the given address as it is and set the remaining 8 bits to 1s as shown in Fig. P. 5.6.5(b).




(G-1813) **Fig. P. 5.6.5(b) : Last address in the block**

From Fig. P. 5.6.5(b) we get, the last address in the block is as follows :

(G-2711) Last address = 129.65.33.255 ...Ans.

**Ex. 5.6.6 :** Find the first addresses, last addresses and number of addresses of the following IP addresses :

1. 205.16.37.39/28
2. 123.56.77.29/27

**Soln. :**

**1. 205.16.37.39/28**

**Step 1 : Find n :**

Given address = 205.16.37.39/28

Hence n = 28 from the slash notation.

∴ n = 28 bits.

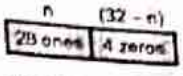
∴ Prefix bits = 28, suffix bits = 32 – 28 = 4

**Step 2 : Number of addresses in the block (N) :**

$N = 2^{(32-n)} = 2^4 = 16$

**Step 3 : Find the first address :**

- Refer Fig. P. 5.6.6(a) to obtain the first address in the block. For this we have to AND the given address with the network mask.

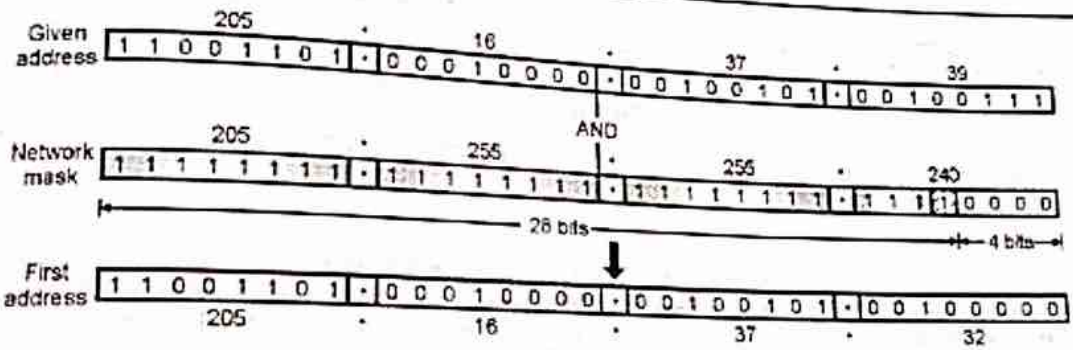



(G-2912)

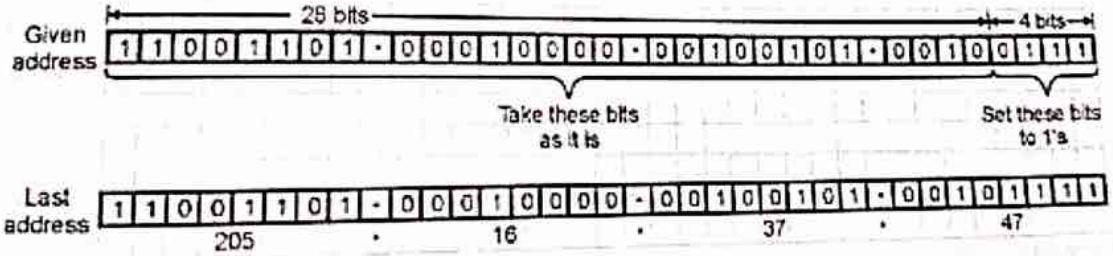
∴ Network mask = 255.255.255.240

- For ANDing write the given address and network mask in their binary notations as shown in Fig. P. 5.6.6(a).
- ∴ From Fig. P. 5.6.6(a) we get the first address in the block as :

(G-2913) First address = 205.16.37.32 ...Ans.




(G-2690) **Fig. P. 5.6.6(a) : First address in the block**




(G-2691) **Fig. P. 5.6.6(b) : Last address in the block**

**Step 4 : Find the last address :**

- To obtain the last address in the block, we have to keep the left most 27 bits in the given address as it is and set the remaining 5 bits to 1s as shown in Fig. P. 5.6.6(b).
- ∴ From Fig. P. 5.6.6(b) we get the last address in the block as follows :

(G-2909) Last address = 205.16.37.47 ...Ans.

**2. 123.56.77.29/27**

**Step 1 : Find n :**

Given address = 123.56.77.29/27

Hence n = 27 from the slash notation.

∴ n = 27 bits.

∴ Prefix bits = 27, suffix bits = 32 – 27 = 5

**Step 2 : Number of addresses in the block (N) :**

$$N = 2^{(32-n)} = 2^5 = 32$$

**Step 3 : Find the first address :**

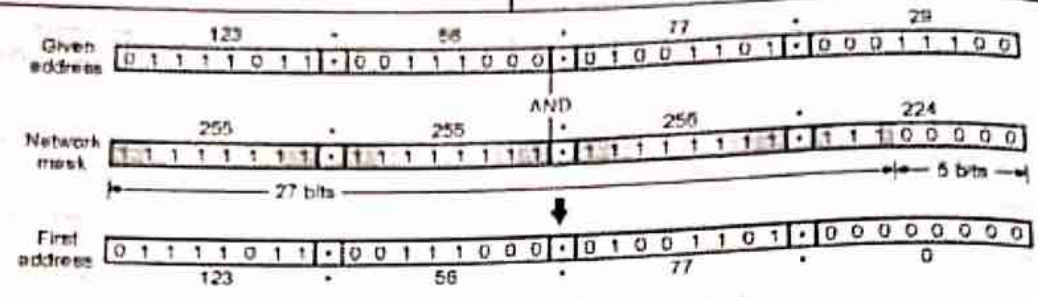
- Refer Fig. P. 5.6.6(c) to obtain the first address in the block.
- For this we have to AND the given address with the network mask.

Network mask = [27 ones | 5 zeros] (G-2909)

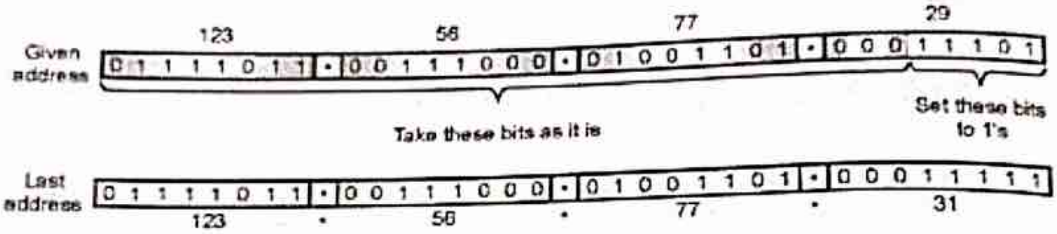
∴ Network mask = 255.255.255.224

- For ANDing write the given address and network mask in their binary notations as shown in Fig. P. 5.6.6(c).
- ∴ From Fig. P. 5.6.6(c) we get the first address in the block as :

(G-2909) First address = [123.56.77.0] ...Ans.




(G-2892) **Fig. P. 5.6.6(c) : First address in the block**




(G-2893) **Fig. P. 5.6.6(d) : Last address in the block**

**Step 4 : Find the last address :**

- To obtain the last address in the block, we have to keep the left most 27 bits in the given address as it is and set the remaining 5 bits to 1s as shown in Fig. P. 5.6.6(d).
- ∴ From Fig. P. 5.6.6(d) we get the last address in the block as follows :

(G-2911) Last address = [123.56.77.31] ...Ans.

## 5.6.5 Block Allocation :

- Now let us understand how to allocate the blocks in the classless addressing.
- The global authority for the block allocation is ICANA means Internet Corporation for Assigned Names and Addresses.
- But the individual addresses of the Internet users is not allotted by the ICANA.
- Instead ICANA will assign large blocks of addresses to various ISPs or large organizations.
- These ISPs or organization will assign addresses to the individual Internet users from their allotted blocks.

**Restrictions :**

- Some of the restriction on classless address blocks have been imposed by the internet authorities in order to simplify the process of address handling.
  1. The addresses in a block should be continuous i.e. serial in manner.
  2. The total number of addresses in a block has to be equal to some power of 2 i.e. $2^1, 2^2, 2^3$ ...etc.
  3. The first address should be evenly divisible by the number of addresses.

### 5.6.6 Relation to Classful Addressing :

- The classful addressing may be imagined as the special case of classless addressing such that the blocks of addresses in class A, B and C type addresses will have the prefix lengths $n_A = 8$, $n_B = 16$ and $n_C = 24$.
- Table 5.6.1 lists the prefix lengths for class A to F classful addresses and using this information we can change a block in classful addressing to a block in classless addressing.

**Table 5.6.1 : Prefix lengths for classful addressing**

| Class | Prefix length | Class | Prefix length |
|---|---|---|---|
| A | / 8 | D | / 4 |
| B | / 16 | E | / 4 |
| C | / 24 | | |

### 5.6.7 Subnetting :

- The concept of subnetting in classless addressing domain is similar to that discussed for the classful addressing.
- The subnetting is used for creating a three level hierarchy in the classless addressing domain. An organization or an ISP have a block of addresses granted to them.
- It can divide these addresses into several subgroups and each subgroup of addresses is assigned to a **subnetwork or subnet.**
- The subnetworks may be subdivided further if the organization want it that way.

### 5.6.8 Designing Subnets :

Let, N = Total number of addresses granted to an organization.

n = Prefix length

$N_{sub}$ = Assigned number of addresses to each subnetwork

$N_{sub}$ = Prefix length for each subnetwork

S = Total number of subnetworks.

- Now follow the steps given below to ensure that the subnetworks operate properly.

**Steps to follow :**

- The number of addresses in each subnetwork should always be equal to a power of 2. i.e. $2^0, 2^1, 2^2$....etc.
- We can use the following expression to find the prefix length of each sunetwork.

$$n_{subA} = n + \log_2\left[\frac{N}{N_{sub}}\right] \quad \text{...(5.6.5)}$$

- The starting address in each subnet should be divisible by the number of addresses in that subnework.
- To achieve this we need to first assign address to larger networks.

**Note :** These restrictions are similar to those applied when addresses to network were allocated.

### 5.6.9 Finding Information about Each Network :

- After designing the subnetworks, we can find the information about the subnets such as starting and last addresses, we can use the same procedure that was used to find the information about each network in the Internet.

**Ex. 5.6.7 :** A block of addresses granted to an ISP is given by 130.34.13.64 / 26. These addresses are to be divided into four subnetworks with equal number of hosts. Design the subnetworks and obtain all the information about each subnet.

**Soln. :**

**Step 1 : Find total number of addresses (N) :**

- From the given address we get n = 26 (prefix length).
- Hence the number of addresses in the whole network will be :

$$N = 2^{(32-n)} = 2^{(32-26)} = 2^6 = 64$$

- The first address in this block will be 130.34.13.64 / 26 whereas the last address will be 130.34.13.127 / 26.
- These values have been obtained using the procedure that we have discussed earlier.

**Subnet design :**

**Step 2 : Find number of hosts per subnetwork :**

- There are four subnetworks with equal number of guests.

  ∴ Number of hosts per subnetwork is given by,

$$N_1 = N_2 = N_3 = N_4 = \frac{N}{4} = \frac{64}{4} = 16 \quad \text{...Ans.}$$

- Note that the first requirement that 64 / 16 should be a power of 2 has been satisfied here.

**Step 3 : Find the prefix lengths of the subnets :**

- The prefix lengths of the four subnets are given by,

$$n_1 = n_2 = n_3 = n_4 = n + \log_2\left[\frac{N}{N_{sub}}\right]$$

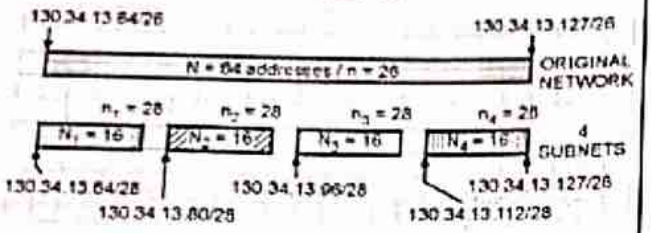
$$= 26 + \log_2\left[\frac{64}{16}\right]$$

$$= 26 + \log_2 4$$

$$\therefore n_1 = n_2 = n_3 = n_4 = 28 \quad \text{...Ans.}$$

**Step 4 : Starting and ending addresses of all the subnets :**

- Refer Fig. P. 5.6.7 which shows all the starting and ending addresses of the 4-subnets.
- It should be noted from Fig. P. 5.6.7 that all the starting addresses should be divisible by the number of addresses in the subnet i.e. by 16.




(G-1814) **Fig. P. 5.6.7**

## 5.6.10 Address Aggregation :

- Address aggregation is considered to be one of the advantages of CIDR architecture.
- As we know, ICANN assigns a large block of addresses to an ISP which is divided into smaller subnets and assigned to the customers by the ISPs.
- Thus many blocks of addresses are aggregated in one block and assigned to one ISP.

---

**Ex. 5.6.8 :** An organization is granted the block 130.34.12.64/26. The organization needs to have four subnets with equal number of addresses in each subnet. What are the subnet addresses and the range of addresses for each subnet ?

**Soln. :**

**Step 1 : Find total number of addresses (N) :**

- From the given address we get n = 26 (prefix length).
- Hence the number of addresses in the whole network will be :

$$N = 2^{(32-n)} = 2^{(32-26)} = 2^6 = 64$$

- The first address in this block will be 130.34.12.64 / 26 whereas the last address will be 130.34.12.127 / 26.

**Subnet design :**

**Step 2 : Find number of hosts per subnetwork :**

- There are four subnetworks with equal number of guests.

∴ Number of hosts per subnetwork is given by,

$$N_1 = N_2 = N_3 = N_4 = \frac{N}{4}$$

$$= \frac{64}{4} = 16 \quad \text{...Ans.}$$

- Note that the first requirement that 64 / 16 should be a power of 2 has been satisfied here.

**Step 3 : Find the prefix lengths of the subnets :**

- The prefix lengths of the four subnets are given by,

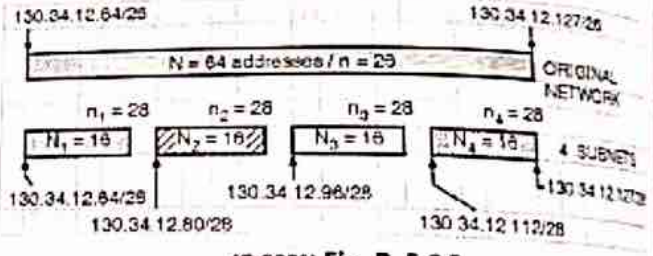
$$n_1 = n_2 = n_3 = n_4 = n + \log_2\left[\frac{N}{N_{sub}}\right]$$

$$= 26 + \log_2\left[\frac{64}{16}\right] = 26 + \log_2 4$$

$$\therefore n_1 = n_2 = n_3 = n_4 = 28 \quad \text{...Ans.}$$

**Step 4 : Starting and ending addresses of all the subnets :**

- Refer Fig. P. 5.6.8 which shows all the starting and ending addresses of the 4-subnets.
- It should be noted from Fig. P. 5.6.8 that all the starting addresses should be divisible by the number of addresses in the subnet i.e. by 16.




(G-2281) **Fig. P. 5.6.8**

---

**Ex. 5.6.9 :** A small organization is given a block with the beginning address and the prefix length 205.16.37.24/29 (in slash notation). What is the range of the block ?

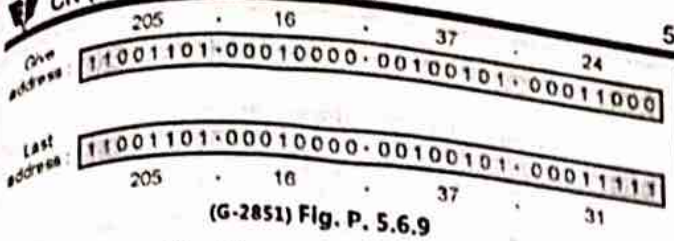
**Soln. :**

**Given :** 205.16.37.24/29 ...First address

**To find :** Range of block

**Step 1 : Find the last address :**

- To obtain the last address in the block, we have to keep the leftmost 29 bits in the given address as it is and set the remaining 3 bits to 1's as shown in Fig. P. 5.6.9.




(G-2851) Fig. P. 5.6.9

∴ Range of block is 205.16.37.24 to 205.16.37.31.

**Ex. 5.6.10 :** An ISP are granted a block of address starting with 127.60.4.0/20. The ISP wants with each organization receiving 8 address only. Design subblock and give the slash notation for each subblock.

**Soln. :**

**Given :**

- An ISP is granted a block of addresses starting with 127.60.4.0/20 among 100 organizations wherein each organization receives eight addresses.
- Let us consider that the address are divided into 128 sub-blocks each having 8-addresses.

Number of granted addresses to the ISP = $128 \times 8 = 1024$

⇒ Customer needs 8 addresses,

⇒ $\log 2^8$ bits are needed to define each host.

$\log 2^8 = \log 2^3$

$= 3 \log 2^2$

$= 3 \times 1$

$= 3$

- Prefix length = 32 – 3 = 29
- The address starts from 127.60.4.0/29 instead of 127.60.4.0/20.
- Since, there are 8 addresses distributed among 100 organization therefore, total number of allocated address = $100 \times 8 = 800$.

| Sub block | Starting address | Ending address |
|---|---|---|
| 1. | 127.60.4.0/29 | 127.60.4.7/29 |
| 2. | 127.60.4.8/29 | 127.60.4.15/29 |
| 3. | 127.60.4.16/29 | 127.60.4.23/29 |
| 4. | 127.60.4.24/29 | 127.60.4.31/29 |
| 5. | 127.60.4.32/29 | 127.60.4.39/29 |
| 6. | 127.60.4.40/29 | 127.60.4.47/29 |
| ⋮ | ⋮ | ⋮ |
| 10 | 127.60.4.72/29 | 127.60.4.79/29 |
| ⋮ | ⋮ | ⋮ |
| 32 | 127.60.4.248/29 | 127.60.4.255/29 |
| ⋮ | ⋮ | ⋮ |
| 64 | 127.60.5.248/29 | 127.60.4.255/29 |
| ⋮ | ⋮ | ⋮ |
| 98 | 127.60.7.8/29 | 127.60.7.15/29 |
| 99 | 127.60.7.16/29 | 127.60.7.23/29 |
| 100 | 127.60.7.24/29 | 127.60.7.31/29 |

Numbers of granted address = 1024

Number of allocated address = 800

Number of available address = Number of granted address– Number of allocated address.

= 1024 – 800 = 224

## 5.7 Special Addresses :

- In the classful addressing, some addresses were reserved for special purpose.
- Similarly in the classless addressing as well some addresses are reserved.

### 5.7.1 Special Blocks :

- Some address blocks have been reserved for special purpose.

### 5.7.2 All Zeros Address :

- The block 0.0.0.0 / 32 contains only one address. It is called as the all zero address and has a prefix length of n = 32.
- This address has been reserved for communication when a host has to send an IPv4 packet but it does not know its own address.
- In such situations, the host sends an IPv4 packet to a DHCP server using this all zero address as the source address and a limited broadcast address (all one address) as the destination address, so as to find its own address.

**All One Address-Limited Broadcast Address :**

- The block 255.255.255.255 / 32 contains only one address.
- It is called as an all one address and has a prefix length of n = 32.
- This all one address has been reserved for limited broadcast address i.e. if a host wants to send message to all the hosts simultaneously then the sending host can use all one address as a destination address inside the IPv4 packet.
- Such a broadcasting is confined to the network only because routers do not allow the all one packet to pass through them.
- The datagram sent with the all zero address as destination will be received and processed by all the hosts on the network.

### 5.7.4 Loopback Address :

- A loopback address is the address which is used to test the software on a machine.
- The block 127.0.0.0 / 8 with a prefix length of 8 is used for the loopback address.
- On using this address, a packet does not leave the machine at all but it returns to the protocol software.
- It can be used for testing the IPv4 software.

### 5.7.5 Private Addresses :

- The address blocks that are not recognized globally still assigned for private use are known as private addresses.
- These addresses are neither connected to nor isolated from the Network Address Translation (NAT) techniques.
- Table 5.7.1 depict such address blocks.

**Table 5.7.1 : Private addresses**

| Block | Number of addresses | Block | Number of addresses |
|---|---|---|---|
| 10.0.0.0 / 8 | 16,777,216 | 192.168.0.0 / 16 | 65,536 |
| 172.16.0.0 / 12 | 1,047,584 | 169.254.0.0 / 16 | 65,536 |

### 5.7.6 Multicast Addresses :

- The block 224.0.0.0 / 4 with a prefix length of n = 4 has been reserved for the multicast IP communication.

### 5.7.7 Special Addresses in Each Block :

- The usage of some address in each block for special addresses has been recommended.
- But it has not been made mandatory. These addresses are not assigned to any host.
- One important point to be remembered is that a very small block of addresses should not be used as special addresses.

### 5.7.8 Network Address :

- The network address is defined as the first address (with the suffix set all to 0s) in a block.
- It is used for defining the network itself. It does not define any host in the network.
- With the same principle, the first address in a subnetwork is called as the subnetwork address.

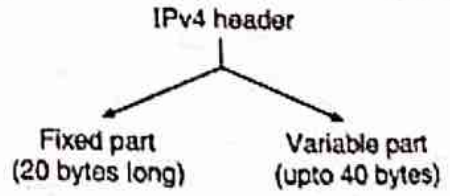
### 5.7.9 Direct Broadcast Address :

- We can use the last address in a block or subblock (with the suffix part set to all 1s), as a **direct broadcast address** for that block or subblock.
- A router generally uses this address for sending a packet to all the hosts connected to a specific network.
- This address is used as the destination address in the IPv4 packet and all the hosts will accept and process the datagram which has this destination address.

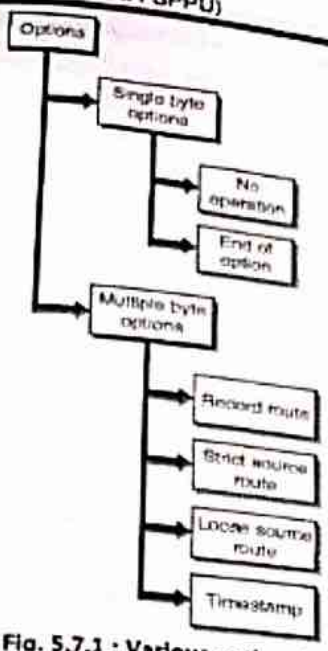
### 5.7.10 Options :

- The IPv4 datagram header consists of two parts as follows :

(G-1856)




- The options field is the part of the variable part the length of which is upto 40 bytes.
- Options is not needed for an IPv4 data. It is supposed to be used for testing and debugging purpose. Fig. 5.7.1 shows various options.




(G-1857) **Fig. 5.7.1 : Various options in IPv4**

- Let us discuss them one by one.

1. **No operation :** This is a one byte option which is used as a filter between options.
2. **End of option :** This is also a one byte option which is used for padding at the end of the option field. It can only be used as the last option.
3. **Record route :** This multiple byte option is used for recording the Internet routers which handle the datagram. It can prepare a list of upto nine addresses of such routers. This option may be used for debugging and management.
4. **Strict source route :** This multiple byte option is used by source for predetermining a route for the datagram. This is called dictation of the route by the source.
5. **Loose source route :** This multiple byte option is similar to the strict source route option but it is less rigid. It is necessary for the datagram to visit all the routers in the given list but the datagram is allowed to visit other routers as well (which are not on the list).
6. **Timestamp :** This multiple byte option is used for recording the time of datagram processing by the router. This time is measured from the midnight universal time (Green which time) and it is expressed in milliseconds.

## 5.8 IPv6 (Next Generation IP) :

- IPv6 is the next generation Internet Protocol designed as the next step of the IP version 4 IPv6 was designed to enable high-performance and larger address space
- This was achieved by overcoming many of the weaknesses of IPv4 protocol and by adding several new features

### 5.8.1 Advantages of IPv6 :

1. **Improved header format :**
- IPv6 uses an improved header format. In its header format the options are separated from the base header. These options are inserted when needed, between the base header and upper layer data
- The routing process is simplified due to this modification.
- The speed of the routing process increases and the routing time is reduced.

2. **Larger address space :**
- IPv6 has 128-bit address, which is 4 times wider in bits is compared to IPv4's 32-bit address space. So there is a large increase in the address space.

  Address space of IPv6 = $(2^{128})$

3. **New options :**
- IPv6 has increased functionality due to the addition of entirely new options that are absent in IPv4.

4. **More security :**
- IPv6 includes security in the basic specification.
- It includes encryption of packets (ESP : Encapsulated Security Payload) and authentication of the sender of packets (AH : Authentication Header) for enhancing the security.

5. **Possibility of extension :**
- The design of IPv6 is done in such a way that there is a possibility of extension of protocol if required.

6. **Support to resource allocation :**
- To implement better support for real time traffic (such as video conference), IPv6 includes flow label in the specification.
- With flow label mechanism, routers can recognize to which end-to-end flow the given packet belongs to.

7. **Plug and play :**
- IPv6 includes plug and play in the standard specification. It therefore must be easier for novice users to connect their machines to the network, it will be done automatically.

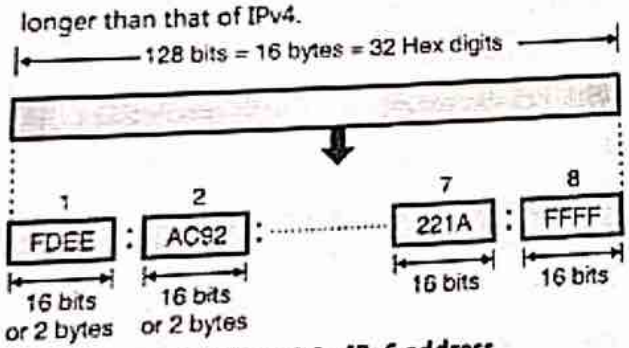
8. **Clearer specification and optimization :**
- IPv6 follows good practices of IPv4, and omits flaws/obsolete items of IPv4.

## 5.9 IPv6 Addressing :

- IPv6 is the next generation Internet Protocol designed as the next step of the IP version 4. IPv6 was designed to enable high-performance and larger address space.
- This was achieved by overcoming many of the weaknesses of IPv4 protocol and by adding several new features.
- The IPv6 was developed due to the address depletion of IPv4.
- The structure of IPv6 address is fundamentally different than that of IPv4.
- Therefore there is absolutely no possibility of address depletion taking place in future.

### 5.9.1 IPv6 Address :

- An IPv6 address is 128 bit long. It consists of 16 bytes as shown in Fig. 5.9.1. Thus the IPv6 address is 4 times longer than that of IPv4.




(G-545) **Fig. 5.9.1 : IPv6 address**

### 5.9.2 Notations :

- An address is stored in the computers in the binary form.
- But it is impossible for humans to handle a 128 bit binary address.
- Therefore many notations have been proposed to represent the IPv6 addresses, so that they become easier to handle for human beings.
- Some of the proposed notations are :
  1. Dotted decimal notation.
  2. Colon hexadecimal notation.
  3. Mixed representation.
  4. CIDR notation.

1. **Dotted decimal notation :**
- In order to maintain the compatibility with IPv4 addresses.
- We may feel tempted to use the dotted decimal notation.
- But practical observation is that this notation is convenient only for the 4 byte address of IPv4.
- It is not at all convenient for the 16 byte IPv6 addresses as it seems too long. Therefore this notation is very rarely used.

2. **Colon hexadecimal notation :**
- The 128 bit address can be made more readable and easy to handle. IPv6 has specified the colon hexadecimal notation.
- IPv6 uses a special notation called hexadecimal colon notation. In this, the total 128 bits are divided into 8 sections, each one is 16 bits or 2 bytes long.
- The 16 bits or 2 bytes in binary correspond to four hexadecimal digits of 4-bits each. Hence the 128 bits in hexadecimal form will have $8 \times 4 = 32$ hexadecimal digits.
- These are in groups of 4 digits as shown and every group is separated by a colon as shown in Fig. 5.9.2.

**AC 81 : 9840 : 0086 : 3210 : 000A : BBFF : 0000 : FFFF**

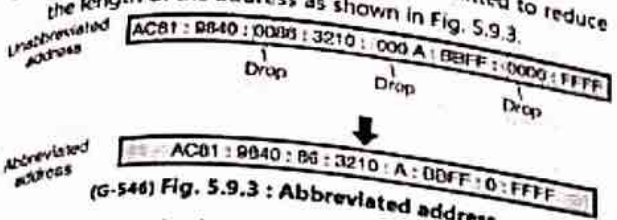
**Fig. 5.9.2 : Colon hexadecimal notation**

- IPv6 uses 128-bit addresses. Only about 15% of the address space is initially allocated, the remaining 85% being reserved for future use.
- These unused addresses may be used in the future by expanding the address spaces of existing address types or for totally new uses.

### 5.9.3 Abbreviation :

- The IPv6 address, in hexadecimal format contains 32 digits and it is very long. But in this address many hex digits are zero.

- We can take advantage of this to shorten the address by abbreviating it. A section corresponds to four digits between any two colons.
- The leading zeros in a section can be omitted to reduce the length of the address as shown in Fig. 5.9.3.

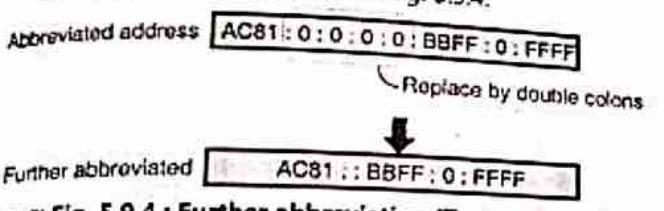



(G-546) **Fig. 5.9.3 : Abbreviated address**

- Note that only the leading zeros can be dropped but the trailing zeros cannot be dropped. This is illustrated in Fig. 5.9.3. Thus due to abbreviation the length of the address has reduced to 24 hex digits from 32.

**Further abbreviation :**

- We can make further abbreviation if there are consecutive sections consisting of only zeros. This is known as **zero compression.**
- We can remove the zeros completely and replace them with double colon as shown in Fig. 5.9.4.




(G-547) **Fig. 5.9.4 : Further abbreviation (Zero compression)**

- This further abbreviation has reduced the address length to just 13 hex digits. It is important to note that abbreviation can be done only once per address.
- Also note that if there are two sets of zero sections, then only one of them can be abbreviated.

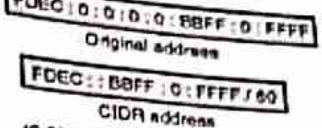
**3. Mixed representation :**

- Sometimes, the IPv6 address is represented using a mixed representation which combines the **colon hex** and **dotted decimal** notations.
- This notation is appropriate during the transition time during which an IPv4 address is being embedded in IPv6 address.
- In the mixed representation the rightmost 32 bits correspond to the IPv4 address. Hence they are represented by the dotted decimal notation.

Whereas the leftmost 96 bits (6 sections) are represented in colon hex notation.

**4. CIDR notation :**

- The type of addressing used in IPv6 is **hierarchical addressing.**
- Therefore IPv6 allows classless addressing and CIDR notation. Fig. 5.9.5 illustrates the CIDR address with a 60 bit prefix.

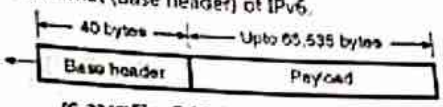



(G-2137) **Fig. 5.9.5 : CIDR address**

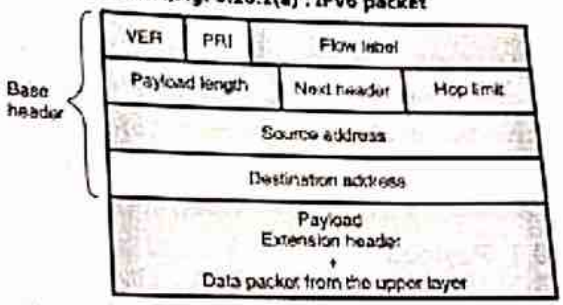
- It has been discussed later on in this chapter, how we can divide an IPv6 address into a prefix and a suffix.

## 5.10 IPv6 Packet Format :

- Fig. 5.10.1(a) shows IPv6 packet. Fig. 5.10.1(b) shows the packet format (Base header) of IPv6.




(G-2245) **Fig. 5.10.1(a) : IPv6 packet**




(G-550) **Fig. 5.10.1(b) : Format of an IPv6 datagram (Base header)**

- Each packet can be divided into two parts viz : base header and payload.
- Base header is the mandatory part and payload is an optional one. The payload follows the base header.
- The payload is made up of two parts :
  1. An optional extension headers and
  2. The upper layer data.
- The base header is 40 byte long whereas the payload consisting of the extension header and upper layer data can have information worth upto 65, 535 bytes.

**Base header :**

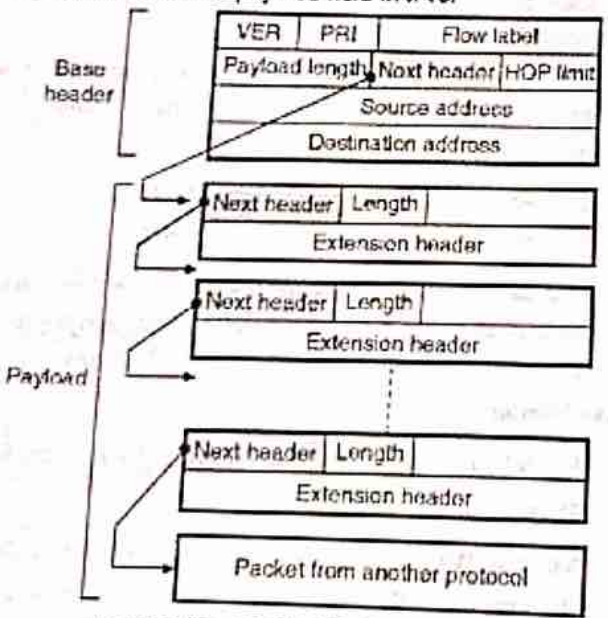
- Fig. 5.10.1(b) shows the base header. It has eight fields. These fields are as follows :

1. **Version (VER) :** The contents of this 4 bit field defines the version of IP such as IPv4 or IPv6. If VER = 6, then the version is IPv6.

2. **Priority :** This 4 bit field contents defines the priority of the packet which is important in connection with the traffic congestion.
3. **Flow label :** It is a 24 bit (3 byte) field which is supposed to provide a special handling for a particular flow of data.
4. **Payload length :** The contents of the 16 bit or 2 byte length field are used to indicate the total length of the IP datagram excluding the base header. That means it gives the length of only the payload part of the datagram.
5. **Next header :** It is an 8 bit field which defines the header which follows the base header in the datagram.
6. **Hop limit :** Contents of this 8 bit (1 byte) field have the same function as TTL (time to live) in IPv4.
7. **Source address :** It is a 16 byte (128 bit) Internet address which corresponds to the originator or source which has produced the datagram.
8. **Destination address :** This is a 16 byte (128 bit) Internet address which corresponds to the address of the final destination of datagram. But this field will contain the address of the next router and not the final destination if source routing is being used.

## 5.10.1 Payload :

- The meaning and format of payload field in IPv6 is different as compared to payload field in IPv4.
- *Fig. 5.10.2 shows payload field in IPv6.*




(G-2246) **Fig. 5.10.2 : IPv6 payload**

- In IPv6, the payload is combination of zero or more extension headers (options) which is followed by data from other protocols such as UDP, TCP etc.
- In IPv4, option is a part of the header, whereas in IPv6 it is designed as extension headers.
- Depends on the situation the payload can have as many extension headers as required.
- Extension header is made up of two mandatory fields : next header and the length which is followed by information which is related to the particular option.
- Value of next header field i.e. code defines which type of the next header is (e.g. source routing options, fragmentation option etc.)
- The last next header describes the protocol which carries the datagram. Some next header codes are listed in Table 5.10.1.

**Table 5.10.1 : Next header codes**

| Sr. No. | Code | Next header code |
|---|---|---|
| 1. | 00 | HOP by hop option |
| 2. | 02 | ICMPv6 |
| 3. | 06 | TCP |
| 4. | 17 | UDP |
| 5. | 43 | Source routing option |
| 6. | 44 | Fragmentation option |
| 7. | 50 | Encrypted security payload |
| 8. | 51 | Authentication header |
| 9. | 59 | Null (no next header) |
| 10. | 60 | Destination option |

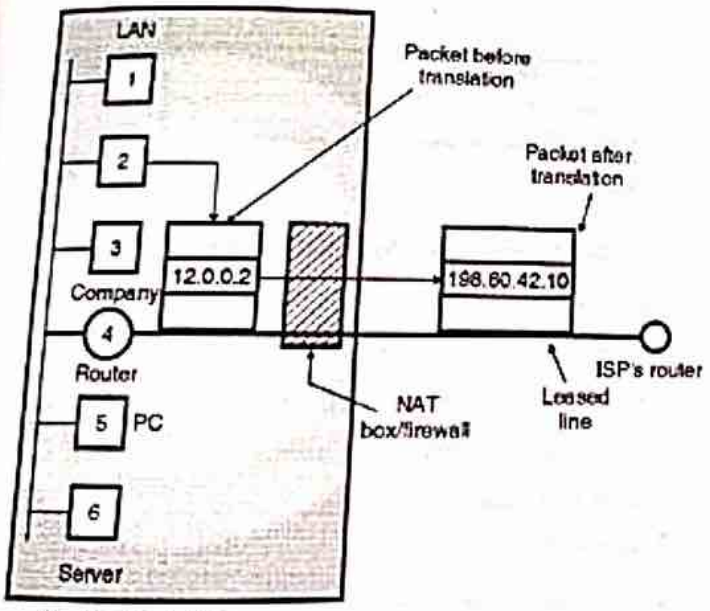
## 5.10.2 NAT – Network Address Translation :

- The problem that existing number of IP addresses is less than the actually required ones is practically important.
- A long term solution to this problem is that the whole Internet should be migrated from IPv4 to IPv6.
- This has begun, but will take year to get complete. (That means all the computers should have IPv6 addresses instead of IPv4 addresses).
- A quick solution to this problem is NAT i.e. Network Address Translation. It is described in RFC 3022.

- The basic idea in NAT is that each company is assigned a single IP address or at the most a small number of IP addresses so as to access the Internet.
- Within the company, every computer gets a unique IP address which is used for routing the internal traffic of the office.
- But when a packet goes out of the company, and goes to ISP, the translation of IP address takes place there.
- In order to make this scheme work, three ranges of IP addresses have been declared as private. Companies can use these addresses internally as per their requirement. However no packet containing these addresses is allowed to appear on the Internet. The three reserved ranges are as follows :

| | | |
|---|---|---|
| Range 1 | 10.0.0.0 to 10.255.255.255/8 | 16777216 Hosts |
| Range 2 | 172.16.0.0 to 173.31.255.255/12 | 1048 576 Hosts |
| Range 3 | 192.168.0.0 to 192.168.255.255/16 | 65 536 Hosts |

- Generally most companies choose the addresses from the first range. Refer Fig. 5.10.3 which explains the operation of NAT. It shows that within the company premises, every machine has a unique address of the form 12.a.b.c.

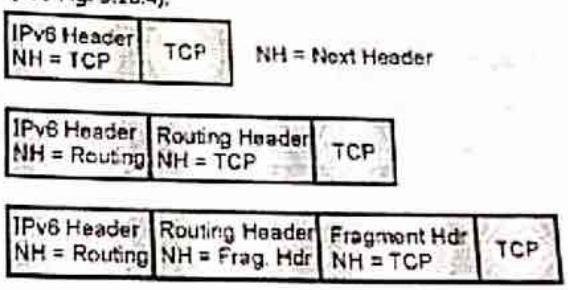



**(G-551) Fig. 5.10.3 : NAT**

- But when a packet leaves the company premises, it passes through the NAT box. This box converts the internal IP address 12.0.0.2 in Fig. 5.10.3 to the company's true IP address 198.60.42.10.
- The NAT box is generally combined with a firewall. It is also possible to integrate the NAT box into company's router.

### 5.10.3 Extension Headers :

- As stated earlier the length of the base header is 40 bytes and it always remains constant.
- But in IPv6, the fixed base header can be followed by upto six extension headers.
- In IPv4 these are optional headers. This gives more functionality to the IP datagram.
- The IPv4 header has space for some optional fields requiring a particular processing of packets.
- These optional fields are not used often, and they can deteriorate router performance because their presence must be checked for each packet. IPv6 replaces these optional fields by **extension headers.**
- In IPv6, optional Internet-layer information is encoded in separate headers that may be placed between the IPv6 header and the upper-layer header in a packet (see Fig. 5.10.4).




**(G-2712) Fig. 5.10.4 : Examples of headers chain**

- There are a small number of such extension headers, each identified by a distinct Next Header value.
- An IPv6 packet may carry zero, one, or more extension headers, each identified by the Next Header field of the preceding header.
- There are seven kinds of extension header :
- Extension headers are not examined or processed by any node along a packet's delivery path, until the packet reaches the node (or each of the set of nodes, in the case of multicast) identified in the Destination Address field of the IPv6 header, except for the Hop-by-Hop Options header and the Routing header.

- Therefore, extension headers must be processed strictly in the order of their appearance in the packet; a receiver must not, for example, scan through a packet looking for a particular kind of extension header and process that header before processing all the preceding ones.
- Each extension header has a length equal to a multiple of 64 bits (8 bytes).
- A full implementation of IPv6 must include support for the following extension headers :
- When more than one extension header is used in the same packet, it is recommended that those headers appear in the following order :

1. IPv6 header
2. Hop-by-Hop Options header
3. Destination Options header
4. Routing header
5. Fragment header
6. Authentication header
7. Encapsulating Security Payload header
8. Destination Options header
9. Upper-layer header

**1. Fragmentation :**

- The fragmentation in IPv6 is conceptually same as that discussed for IPv4, but the fragmentation in IPv6 takes place at a different place than that in IPv4.
- In IPv4 the fragmentation is done by the source or router, but in IPv6 the fragmentation may be carried out only by the original source.

**2. Authentication and Privacy :**

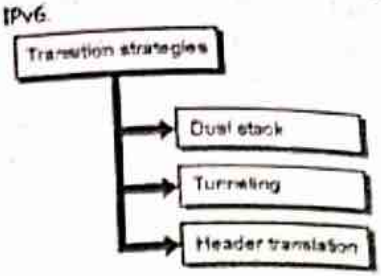
- IPv6 provides authentication and privacy using options in the extension header.

## 5.11 Transition from IPv4 to IPv6 :

- It is required to use a new version of the IP protocol.
- For that transition from IPv4 to IPv6 we have to define a transition day on that day each and every router or host stop using old version and should start using the new version.
- As there are huge number of systems in the Internet, transition from IPv4 to IPv6 is not practical suddenly.
- It will take some amount of time to move each and every system in the Internet from IPv4 to IPv6.
- The transition from IPv4 to IPv6 should be smooth to prevent any problems in the system.

### 5.11.1 Transition Strategies :

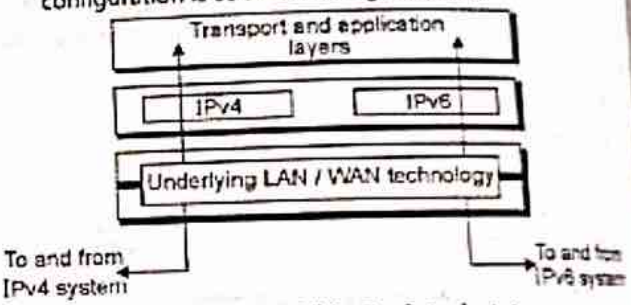
- Fig. 5.11.1(a) shows the strategies for transition from IPv4 to IPv6.




(G-2531) **Fig. 5.11.1(a) : Transition strategies**

**1. Dual stack :**

- Before completely migrating to version 6 it is recommended that all hosts should have a dual stack of protocols at the time of transition.
- Simultaneously station should run IPv4 and IPv6, until the Internet uses IPv6. The layout of dual stack configuration is as shown in Fig. 5.11.1(b).



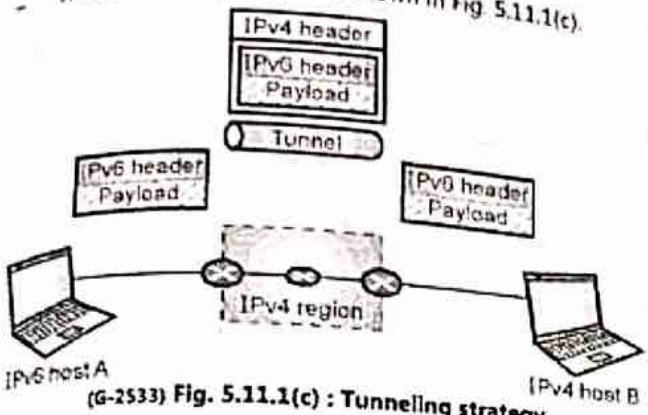

(G-2532) **Fig. 5.11.1(b) : Dual stack stategy**

- A source host send query to the DNS for deciding which version to use while sending a packet to a destination.
- A source host sends IPv4 packet if an IPv4 address is returned by the DNS, and sends IPv6 packet if DNS returns IPv6 address.

**2. Tunneling :**

- When two computers are using IPv6 want to communicate with each other and a region through which the packet must pass uses IPv4, in such case **tunneling** strategy is used.
- The packet should have IPv4 address while passing through this region.

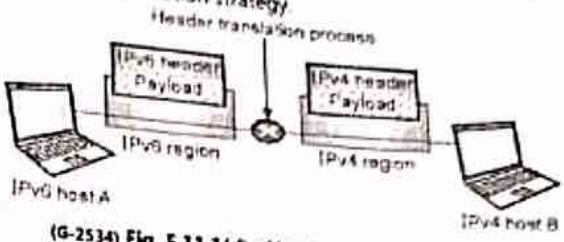
- When it enters in this region the IPv4 packet is encapsulated in IPv4 packet and when it exists the region it leaves its capsule.
- It looks like as if the IPv6 packet enters in a tunnel from one end and comes out from the other end. The protocol value is set to 41 for making it clear that IPv4 packet is holding an IPv6 packet as a data.
- The tunneling strategy is as shown in Fig. 5.11.1(c).




(G-2533) **Fig. 5.11.1(c) : Tunneling strategy**

3. **Header translation :**

- If some systems use IPv4 and the majority of the Internet has moved from IPv4 to IPv6, in that case header translation strategy is used where the receiver does not understand IPv6 but the sender wants to use IPv6 only.
- In this situation tunneling will not work because the packet should be in the IPv4 format which has to be understood by the receiver.
- In this strategy through header translation the format of header must be totally changed. The IPv6 packet header is converted into an IPv4 header. Fig. 5.11.1(d) shows header translation strategy.




(G-2534) **Fig. 5.11.1(d) : Header translation strategy**

### 5.11.2 Use of IP Addresses :

- A host may need to use both IPv4 and IPv6 addresses during the transition.
- IPv4 addresses must disappear after completion of transition.
- During the transition it is necessary that the DNS server is to be ready to map a host name to address type.
- After migrating all hosts in the world the IPv4 dictionary will disappear.

## 5.12 Comparison between IPv4 and IPv6 :

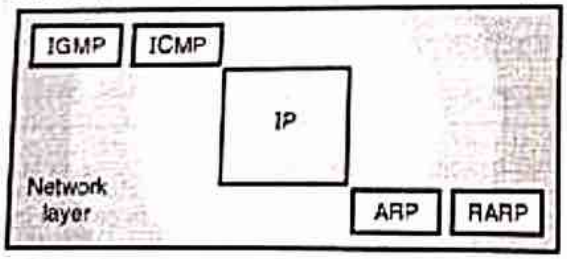
**Table 5.12.1 : Comparison between IPv4 and IPv6**

| Sr. No. | IPv4 | IPv6 |
|---|---|---|
| 1. | In IPv4 there are only $2^{32}$ possible ways to represent the address (about 4 billion possible addresses) | In IPv6 there are $2^{128}$ possible way (about $3.4 \times 10^{38}$ possible addresses) |
| 2. | The IPv4 address is written by dotted-decimal notation, e.g. 121.2.8.12 | IPv6 is written in hexadecimal and consists of 8 groups, containing 4 hexadecimal digits or 8 groups of 16 bits each, e.g. FABC: AC77: 7834:2222:FACB: AB98: 5432:4567. |
| 3. | The basic length of the IPv4 header comprises a minimum of 20 bytes (without option fields). The maximum total length of the IPv4 header is 60 bytes (with option fields), and it uses 13 fields to identify various control settings. | The IPv6 header is a fixed header of 40 bytes in length, and has only 8 fields. Option information is carried by the extension header, which is placed after the IPv6 header. |

| Sr. No. | IPv4 | IPv6 |
|---|---|---|
| 4. | IPv4 header has a checksum, which must be computed by each router | IPv6 has no header checksum because checksums are, for example, above the TCP/IP protocol suite, and above the Token Ring, Ethernet, etc. |
| 5. | IPv4 contains an 8-bit field called Service Type. The Service Type field is composed of a TOS (Type of Service) field and a procedure field | The IPv6 header contains an 8-bit field called the Traffic Class Field. This field allows the traffic source to identify the desired delivery priority of its packets |
| 6. | The IPv4 node has only Stateful auto-configuration. | The IPv6 node has both a stateful and a stateless address auto configuration mechanism. |
| 7. | Security in IPv4 networks is limited to tunneling between two networks | IPv6 has been designed to satisfy the growing and expanded need for network security. |
| 8. | Source and destination addresses are 32 bits (4 bytes) in length. | Source and destination addresses are 128 bits (16 bytes) in length. |
| 9. | IPsec support is optional. | IPsec support is required |
| 10. | No identification of packet flow for QoS handling by routers is present within the IPv4 header. | Packet flow identification for QoS handling by routers is included in the IPv6 header using the Flow Label field. |
| 11. | Address Resolution Protocol (ARP) uses broadcast ARP Request frames to resolve an IPv4 address to a link layer address. | ARP Request frames are replaced with multicast Neighbour Solicitation messages. |
| 12 | Must be configured either manually or through DHCP. | Does not require manual configuration or DHCP. |
| 13. | ICMP Router Discovery is used to determine the IPv4 address of the best default gateway and is optional | ICMP Router Discovery is replaced with ICMPv6 Router Solicitation and Router Advertisement messages and is required. |
| 14. | Header includes options | All optional data is moved to IPv6 extension headers. |

## 5.13 Internet Control Protocols :

- The main protocols corresponding to the network layer in the TCP/IP suite as well as Internet layer are : ARP, RARP, IP, ICMP and IGMP. This is as shown in Fig. 5.13.1.




(G-524)**Fig. 5.13.1 : Protocols at network layer**

- Out of these protocols IP is the most important protocol.
- It is responsible for host to host delivery of datagrams from a source to destination. But IP needs to take services of other protocols.
- IP takes help from ARP in order to find the MAC (physical) address of the next hop.
- IP uses the services of ICMP during the delivery of the datagram packets to handle unusual situations such as presence of an error.
- IP is basically designed for unicast delivery. But some new Internet applications as well as multimedia need multicast delivery.
- So for multicasting, IP has to use the services of another protocol called IGMP.

- IPv4 is the current version of IP whereas IPv6 is the latest version of IP.

## 5.14 ARP (Address Resolution Protocol) :

- ARP as defined in RFC 826 is Ethernet Address Resolution Protocol. ARP provides service to IP, which make us think that it is in the link layer TCP/IP model (or DLL of OSI model).
- But its messages are carried by DLL protocol and are not capsulated within IP datagrams. That is why it can be called as a network layer protocol as well.
- Thus ARP occupies an unusual place in TCP/IP suite. But the most important point is that ARP provides an essential service when TCP/IP is running on a LAN.
- An Internet consists of various types of networks and the connecting devices like routers. A packet starts from the source host, passes through many physical networks and finally reaches the destination host.
- At the network level, the hosts and routers are recognised by their IP addresses.

**IP address :**

- An IP address is an internetwork address. It is a universally unique address. Every protocol involved in internetworking requires IP addresses.

**MAC address :**

- The packets from source to destination hosts pass through physical networks.
- At the physical level the IP address is not useful but the hosts and routers are addressed by their MAC addresses.
- A MAC address is a local address. It is unique locally but it is not unique universally.
- The IP and MAC address are two different identifiers and both of them are needed, because a physical network can have two different protocols operating at the network layer at the same time.
- Similarly a packet may travel through different physical networks.
- So to deliver a packet to a host or a router, we require addressing to take place at two levels namely IP addressing and MAC addressing.
- Most importantly we should be able to map the IP address into a corresponding MAC address.

### 5.14.1 Mapping of IP Address into a MAC Address :

- We have seen the need of mapping an IP address into a MAC address.
- Such mapping can be of two types :
  1. Static mapping and 2. Dynamic mapping

**1. Static mapping :**

- In static mapping a table is created and stored in each machine. This table associates an IP address with a MAC address.
- If a machine knows the IP address of another machine then it can search for the corresponding MAC address in its table.
- The limitation of static mapping is that the MAC addresses can change. These changed MAC addresses must be updated periodically in the static mapping table.

**2. Dynamic mapping :**

- In dynamic mapping technique a protocol is used for finding the other address when one type of address is known. There are two protocols used for carrying out the dynamic mapping.
- They are :
  1. Address Resolution Protocol (ARP).
  2. Reverse Address Resolution Protocol (RARP)
- The ARP is used for mapping an IP address to a MAC address whereas the RARP is used for mapping a MAC address to an IP address.

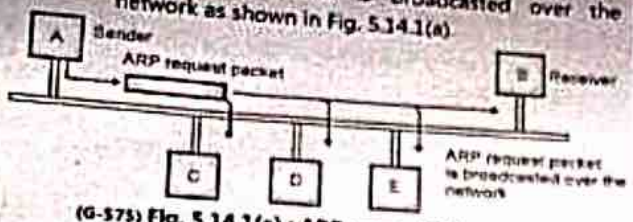
### 5.14.2 ARP Operation :

- ARP is used for mapping an IP address to its MAC address. For a LAN, each device has its own physical or station address as its identification.
- This address is stored on the NIC (Network Interface Card) of that machine.

**How to find the MAC address ?**

- When a router or a host (A) needs to find the MAC address of another host (B) the sequence of events taking place is as follows :
  1. The router or host A who wants to find the MAC address of some other router, sends an ARP request packet. This packet consists of IP and MAC addresses of the sender A and the IP address of the receiver (B).

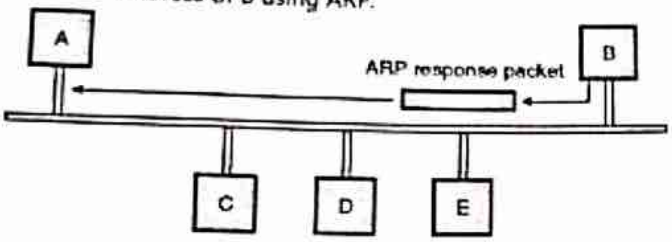
2. This request packet is broadcasted over the network as shown in Fig. 5.14.1(a).




(G-575) Fig. 5.14.1(a) : ARP request is broadcast

3. Every host and router on the network will receive the ARP request packet and process it. But only the intended receiver (B) will recognize its IP address in the request packet and will send an ARP response packet back to A.

4. The ARP response packet has the IP and physical addresses of the receiver (B) in it. This packet is delivered only to A (unicast) using A's physical address in the ARP request packet. This is shown in Fig. 5.14.1(b). Thus host A has obtained the MAC address of B using ARP.




(G-576) Fig. 5.14.1(b) : ARP response in unicast

### 5.14.3 Mapping Physical Address to Logical Address :

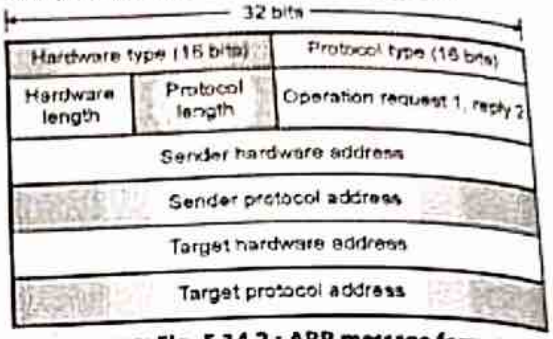
- Sometimes a host knows its physical address but needs to know its logical address. This can happen in the following two cases :
  1. If a diskless station has been just booted. This station can find its physical address by checking its interface but it does not know its logical address.
  2. An organization has less number of IP addresses. So it can not assign a separate IP address to each station. Hence it has to assign the IP addresses when a station demands for it.

### 5.14.4 ARP Cache Memory :

- The use of ARP would be inefficient if A needs to broadcast an ARP request for each IP packet that is to be sent to B, because instead of broadcasting the request it could have broadcast the IP packet itself.
- So ARP is efficient only if the ARP reply is stored in cache memory (cached) for a while.
- This is due to the fact that a system generally sends hundreds of packets to the same destination.
- Thus the system that receives an ARP reply stores the mapping in the cache memory and keeps it for 20 to 30 minutes.
- So if packets are again sent to the same destination then it could use this mapping instead of broadcasting an ARP request.
- Before sending an ARP request, the system checks its cache to see if the mapping could be found.

### 5.14.5 ARP Packet Format :

- ARP packet format is as shown in Fig. 5.14.2.




(G-2852) Fig. 5.14.2 : ARP message format

- The various fields in it are as follows :

1. **HTYPE (Hardware Type)** : This 16 bit field defines the type of network on which ARP is being run. ARP is capable of running on any physical network.
2. **PTYPE (Protocol Type)** : This 16 bit field is used to define the protocol using ARP. Note that we can use ARP with any higher-level protocol such as IPv4.
3. **HLEN (Hardware length)** : It is an 8 bit field which is used for defining the length of the physical address in bytes. For example, this value is 6 for Ethernet.
4. **PLEN (Protocol Length)** : This field is 8 bit long and it defines the length of the IP address in bytes. For IPv4 this value is 4.
5. **OPER (Operation)** : It is a 16 bit field which defines the type of packet. The two possible types of packets are ARP request (1) and ARP reply (2).
6. **SHA (Sender Hardware Address)** : This field is used for defining the physical address of the sender. The length of this field is variable.

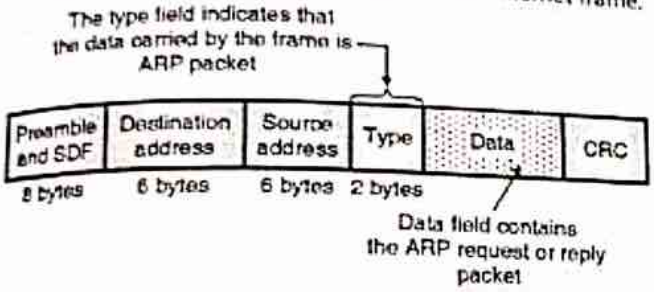
7. **SPA (Sender Protocol Address) :** This field defines the logical address of the sender. The length of this field is variable.

8. **THA (Target Hardware Address) :** It defines the physical address of the target. It is a variable length field. This field contains all zeros for the ARP request packet, because the receivers physical address is not known to the sender.

9. **TPA (Target Protocol Address) :** This field defines the logical address of the target. It is a variable length field.

### 5.14.6 Encapsulation :

- An ARP packet (request or reply) is inserted directly into the data link frame. Such an insertion is known as encapsulation.
- Fig. 5.14.3 shows an example of encapsulation in which an ARP packet being encapsulated in an Ethernet frame.




(G-578)**Fig. 5.14.3 : Encapsulation of ARP packet**

- The type field shows that the data carried by the frame is an ARP request or reply packet.

### 5.14.7 Operation of ARP on Internet :

**Working conditions :**

- The services of ARP can be used under the following working conditions when it is being operated on Internet :
  1. The sender is a host and wants to communicate with another host which is on the same network.
  2. The sender is a host and wants to communicate with a host on another network.
  3. The sender is a router. It has received a datagram with a destination address of a host on another network.
  4. The sender is a router. It has received a datagram which is meant for a host in the same network.

- Now let us see how ARP works on the internet.

**Operation :**

1. The sender (host or router) knows the IP address of the target.
2. IP orders ARP to create an ARP request message. The request packet consists of senders physical and IP addresses plus the IP address of the target but the physical address of the target is not known.
3. This ARP request packet is sent to the data link layer. Here the ARP request packet is inserted in a frame.
4. Every router or host receives this frame because it is broadcast. All the machines except the target drop this packet as discussed earlier.
5. The target machine sends back a reply packet which contains the target's physical address. This reply is unicast and addressed only to the sender.
6. The sender receives the reply packet. Hence the physical address of the target has been obtained.
7. The IP datagram carrying data for the target machine is inserted in a frame and the frame is unicast to the target machine.

## 5.15 The Reverse Address Resolution (RARP) Protocol :

- ARP is used for solving the problem of finding out which Ethernet address corresponds to a given IP address.
- That means ARP is used for the mapping of IP address to physical or MAC address.
- But sometimes we have to solve a reverse problem. That means we have to obtain the IP address corresponding to the given Ethernet (MAC) address.
- Such a problem can occur when booting a diskless workstation. The problem of obtaining the IP address when an Ethernet address is given, can be solved by using RARP (Reverse Address Resolution Protocol).
- The newly booted workstation is allowed to broadcast its Ethernet address.
- The RARP server after receiving this request, checks the Ethernet address in its files and finds the corresponding IP address.

- This IP address is then sent back. The disadvantage of RARP is that it uses a destination address of all 1s (limited broadcasting) to reach the RARP server.
- But such broadcasts are not forwarded by routers, so a RARP server is needed on each network. In order to get around this problem, another bootstrap protocol called BOOTP has been invented.
- Unlike RARP, it uses UDP messages which are forwarded over routers.
- It also provides a diskless workstation with additional information, including the IP address of the file server holding the memory image, the IP address of the default router and the subnet mask to use.

## 5.16 ICMPv4 :

- The long form of ICMPv4 is Internet Control Message Protocol version 4. The IP provides unreliable and connectionless datagram delivery, and makes an efficient use of network resources.
- IP is a best-effort delivery (which does not provide any guarantee) service that takes a datagram from its original source to its final destination. However, IP has two drawbacks :
  1. It does not have any error control mechanism.
  2. It does not have any assistance mechanism.
- The Internet Control Message Protocol (ICMP) is used to overcome these drawbacks.
- It is used along with IP. It reports presence of errors and sends the control messages on behalf of IP. ICMP does not attempt to make IP a reliable protocol.
- It simply attempts to report errors and provide feedback on specific conditions. ICMP messages are carried as IP packets and are therefore unreliable. ICMP is a network layer protocol.
- IP also lacks a mechanism for host and management queries. A host sometimes wants to know if a router or another host is operating or dead.
- And sometimes a network manager needs information from another computer on the network (such as host or router).
- There are two versions of ICMP protocol namely ICMPv4 and ICMPv6. In the following sections, we are going to discuss ICMPv4.

### 5.16.1 ICMP Encapsulation :

- ICMP operates in the network layer but its messages are not passed directly to the data link layer. Instead, the messages are first encapsulated inside IP datagrams and then sent to the lower layer.
- This is as shown in Fig. 5.16.1. The ping command uses ICMP as a probe to test whether a station is reachable.


(G-2103) Fig. 5.16.1 : ICMP encapsulation

- Ping packages an ICMP echo request message in a datagram and sends it to a selected destination.
- The user chooses the destination by specifying its IP address or name on the command line in a form such as :

ping 100.50.25.1

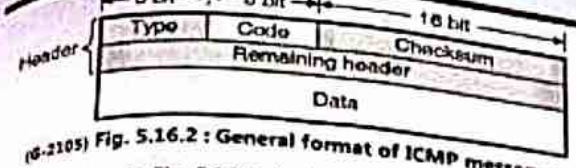
- When the destination receives the echo request message, it responds by sending an ICMP echo reply message.
- If a reply is not returned within a set time, ping resends the echo request several more times. If no reply arrives, ping indicates that the destination is unreachable.
- Another utility that uses ICMP is trace route, which provides a list of all the routers along the path to a specified destination.

### 5.16.2 ICMP Messages :

- ICMP messages are of two types :
  1. Error reporting messages
  2. Query messages.
- If a host or a router encounters a problem while processing an IP problem, then it uses the **error reporting messages** for reporting the problem.
- A host or a network manager can use the **query messages** to get some specific information from a router or another host.

### 5.16.3 Message Format :

- Fig. 5.16.2 shows the general format of ICMP messages.



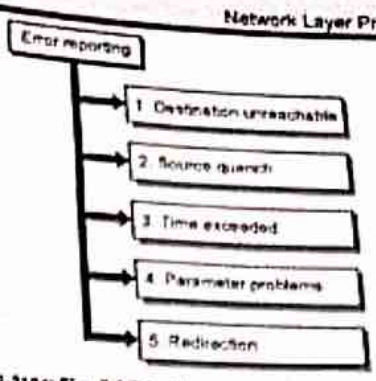

(G-2105) **Fig. 5.16.2 : General format of ICMP messages**

- As shown in Fig. 5.16.2, the header of an ICMP message is 8-byte long and the data section is of a variable size. The general header format for each ICMP message is different.
- But the first four bytes are common to all the message types.

1. **Type :**
   - This 8-bit field is used for defining the types of message.
2. **Code :**
   - This 8-bit field is used for specifying the reason for the particular message type.
   - The last common field is the **checksum** field which is 16 bit (2 byte) long. We will discuss it later in this chapter.
   - The information to find the original packet that had error is included in the **data section** of the error messages.
   - Whereas the **data section** in the query messages contains extra information depending on the type of query.

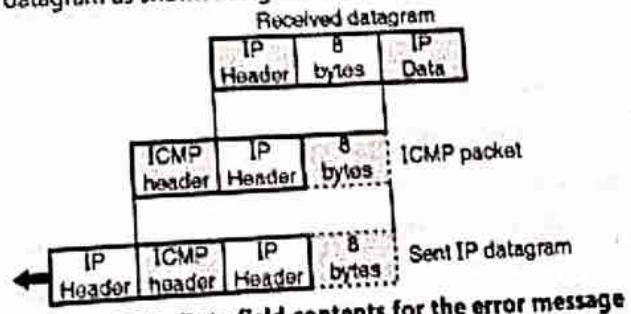
## 5.17 Error Reporting Messages in ICMPv4 :

- One of the important responsibilities of ICMP is to report the presence of an error. IP is an unreliable protocol.
- So error checking and control are not done by IP. So ICMP was designed to assist IP. But ICMP does not correct the errors. It simply reports them and leaves the error correction job to the higher level protocols.
- ICMP always sends the error reporting messages back to the original source. ICMP has five types of error reporting messages.
- Fig. 5.17.1 shows different types of error reporting messages. ICMP makes use of the source IP address for sending the error message back to original source of erroneous datagram.




(G-2104) **Fig. 5.17.1 : Error reporting messages**

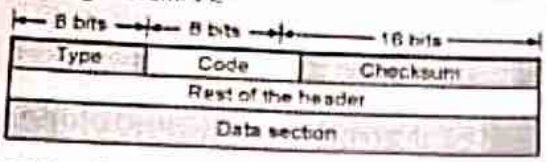
- Some of the important points about ICMP error messages are as follows :
  1. If a datagram containing and ICMP error message is received, then no ICMP error message will be generated in response to it.
  2. An ICMP error message will not be generated for a fragmented datagram that is not the first fragment.
  3. Any ICMP error message will not be generated for a datagram which has a multicast address.
  4. An ICMP error message will not be generated for a datagram which has a special address such as 127.0.0.0 or 0.0.0.0
- It is important to note that the data section of every error message, contains the IP address of the original datagram in addition to the 8 bytes of data in that datagram. The header of the original datagram is included in the error message, to ensure that the error message will reach the original source.
- The additional 8 byte data is included because in TCP and UDP, the first 8 bytes of information contains information about the port numbers for TCP and UDP and sequence number for TCP.
- The source can use this information and convey to TCP and UDP protocols that an error has occurred. Then the **error packet,** formed by ICMP, is encapsulated in an IP datagram as shown in Fig. 5.17.2.




(G-2106) **Fig. 5.17.2 : Data field contents for the error message**

### 5.17.1 General Format of Error Reporting Messages :

- Fig. 5.17.3 shows the general format of error reporting message in ICMPv4.




(G-2107(a)) **Fig. 5.17.3 : General format of error reporting message in ICMPv4**

- Depending on the values of the type and code fields, the type of error reporting message would change as shown in Table 5.17.1.

**Table 5.17.1**

| Sr. No. | Error reporting message | Type | Code |
|---|---|---|---|
| 1. | Destination unreachable | 03 | 0 to 15 |
| 2. | Source quench | 04 | 0 |
| 3. | Redirection | 05 | 0 to 3 |
| 4. | Time exceeded | 11 | 0 and 1 |
| 5. | Parameter prblem | 12 | 0 and 1 |

### 5.17.2 Destination Unreachable :

- When it is not possible for a router to route the datagram or when a host is unable to deliver a datagram, then the datagram is **discarded** and the **destination unreachable** error message is sent back by the respective host or router to the source host which originated the datagram.
- The general format of the destination unreachable error message is as shown in Fig. 5.17.3. The content of the type field for this error reporting message is 03.
- The code field for the destination unreachable error message has 16 different values (0 to 15) and each one specifies a reason for discarding a datagram.
- The destination host or routers can produce the destination unreachable message. Only the destination host can create code 2 and code 3 messages.
- The messages of other codes except codes 2 and 3 can be created only by the routers. The non-creation of destination unreachable message does not guarantee the delivery of datagram.
- It is not possible for the router to detect all the problems that prevent the packet delivery.

### 5.17.3 Source Quench Error Message :

- A host or router uses source quench messages in order to tell the original source that congestion has occurred and to request it to reduce its current rate of packet transmission.
- There is no flow control or congestion control mechanism in IP. So the source quench message in ICMP is designed to add some kind of flow control and congestion control to IP.
- This message serves two purposes :
  1. It tells the source that the packet has been discarded and,
  2. It gives a warning to the source that the source should slow down (quench) because congestion has taken place somewhere.
- Fig. 5.17.3 shows the format of the source quench error message. The content of the type and code fields for this error reporting message are 04 and 0 respectively.
- A source-quench message, one per discarded datagram due to congestion is sent back by a router or destination host, to the source host.
- But, the **congestion relieved** message cannot be sent to the source host as no such mechanism exists.
- As no such message could be sent back, the source host assumes that the congestion has continued to exist, and therefore it continues to reduce the rate of data transmission, until no more source-quench messages are received.
- The congestion can happen due to two types of communications :
  1. Due to one to one communication or
  2. Due to many to one communication.
- In the one to one communication, a single source host will be responsible for congestion because of its high data transmission rate.
- The source quench message will be useful under such operating conditions, for reducing the transmission rate of the source host and clear the congestion.
- But this message will not prove to be successful if congestion occurs in the many to one type communication.

- This is because the router or destination host does not know which source is fast and responsible for the congestion.
- As a result, it may discard the packets received from the slowest source instead of dropping them from a fast source which is actually responsible for congestion.

### 5.17.4 Time Exceeded Error Message :

- This message is generated in two cases :
  1. If a router receives a packet with a 0 in the TTL field then it discards that datagram and send a time exceeded message back to the source originating that packet.
  2. If all the fragments which are parts of a message do not arrive at the destination host within a certain time limit then time exceeded message is sent back.
- The format of the time exceeded message is as shown in Fig. 5.17.3. The content of the type and code fields for this error reporting message are 11 and (0 or 1) respectively.
- If **code = 0**, then the router will discard the datagram because the value of TTL (time to live) field is zero.
- If **code = 1**, then destination host discards the fragments of datagram because some fragments could not arrive at the destination host within the time limit.

### 5.17.5 Parameter Problem Error Message :

- There should not be any ambiguity in the header part of the packet.
- If a router or destination host comes across such ambiguity or missing value in any field of the datagram then it simply discards that datagram and sends the parameter problem message back to the source originating that message.
- This message can be created either by a router or the destination host. The content of the type and code fields for this error reporting message are 12 and (0 or 1) respectively.

1. **Code = 0 :** If code = 0, then the datagram is discarded because of an error or ambiguity present in one of the header fields. The erroneous byte is pointed at by the value of the pointer field. For example if pointer field = 0, then the first byte is an invalid field.
2. **Code = 1 :** If code = 1, then the datagram is discarded because required part of an option is missing.

- The format of the parameter problem is as shown in Fig. 5.17.3.

### 5.17.6 Redirection Error Message :

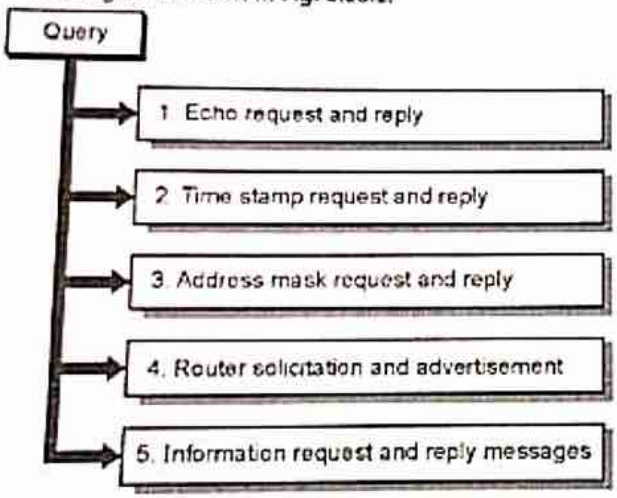
- If a router or host wants to send a packet to another network then it should know the IP address of the next router.
- The routers and hosts must have a routing table to find the address of the next router and the routing table has to be updated automatically on a continuous basis.
- The redirection message is used for such updating. The ICMP sends a redirection message back to its host to carry out an automatic periodic updating.
- In order to ensure higher efficiency, the hosts do not participate in the process of routing table update. This is because the number of hosts in the Internet is much higher than the number of routers.
- If the routing tables of hosts are updated dynamically then it creates an unwanted traffic. Generally the static routing is used by the hosts. That means the routing table of a host contains limited number of entries.
- Generally a host knows the IP address of only one router that is the default router. Due to this, a host can send a datagram which is destined for another network, to a wrong router.
- Here the datagram receiving router will route the datagram the correct router. However it sends a redirection message to the host to update the routing table of the host.
- The format of the redirection message is as shown in Fig. 5.17.3. The content of the type and code fields for this error reporting message are 05 and (0 or 3) respectively.
- The second row of the redirection message contains the IP address of the appropriate target router. It is important to understand that the redirection message is different from the other error message even though it is considered as an error reporting message.
- What is the difference ? In this case the router does not discard the erroneous datagram. Instead it is sent to the appropriate router.

- This process of redirection is narrowed down by the contents of the **code field** as follows :
  1. **Code = 0** : Redirection will be for a network specific route.
  2. **Code = 1** : Redirection is to be done for a host specific route.
  3. **Code = 2** : Redirection is to be done for a network specific route and based upon a specific type of service.
  4. **Code = 3** : Redirection is to be done for a host specific route on the basis of a specified type of service.

**Note :** A router sends the redirection message back to a host on the same local network.

## 5.18 Query Messages (ICMPv4) :

- The ICMP can diagnose some of the network problems.
- This is in addition with the error reporting feature. Such a diagnosis is done through the query messages.
- The query messages is a group of five different pairs of messages as shown in Fig. 5.18.1.

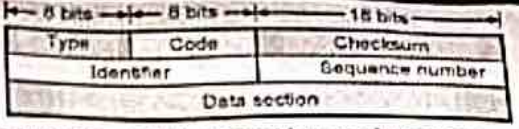



**(G-2112) Fig. 5.18.1 : Query messages**

- However out of these five pairs of messages, only two pairs are being used today.
- They are :
  1. Echo request and reply.
  2. Timestamp request and reply.

**General format of query messages :**

- Fig. 5.18.2 shows the general format of query message in ICMPv4.




**(G-2113(a)) Fig. 5.18.2 : General format of query message in ICMPv4**

- Depending on the values of the type and code fields, the type of query message would change as shown in Table 5.18.1.

**Table 5.18.1**

| Sr. No. | Error reporting message | Type | Code |
|---|---|---|---|
| 1. | Destination unreachable | 00 or 08 | 00 |
| 2. | Source quench | 13 | 14 |

### 5.18.1 Echo Request and Reply :

- This pair of query messages has been designed for the diagnostic purpose.
- This pair of messages is utilized by the network managers and users for identifying the network problems. The format of the echo request echo reply pair of messages is as shown in Fig. 5.18.2.
- The content of the type and code fields for Echo Request and Reply message are (00 and 08) and (0) respectively.
- This pair of query messages would determine whether the two given systems (either hosts or routers) can communicate with each other or not.
- The communication will take place as follows :
  1. A host or router sends the **echo-request** message to another host or router it wants to communicate to.
  2. The host or router which receives the echo request message will create an **echo-reply** message and sends it back to the original sender.
- We can also use the echo-request echo-reply pair to determine if the IP level communication is present or not.
- The network managers can use the echo request and echo reply pair of messages to check the operation of IP protocol.
- A host can also use this message pair to see if another host is reachable or not. At the users level, this is done by invoking the **packet Internet groper** command (ping).

- Now a days a version of ping command is provided by most systems which can create a string of echo-request and echo-reply messages for providing statistical information.
- It is also possible to check whether a node is functioning properly or not with the help of the echo-request echo reply pair of messages.
- In Fig. 5.18.2, the protocol does not formally define the identifier and sequence number fields. Therefore the sender can use them in an arbitrary manner.

### 5.18.2 Timestamp Request and Reply :

- This pair of messages can be used by the hosts and routers to find out the round trip time that an IP datagram needs to travel between them.
- It can also be used for synchronizing the clock signals used in the two machines (hosts or routers). Fig. 5.18.2 shows the format of these two messages.
- The content of the type and code fields for timestamp request and reply message are 13 and 14 respectively.
- As shown in Fig. 5.18.2, there are three timestamp fields and each field is 32-bit long.
- The number in each of these fields represents time in milliseconds from the midnight in Universal time.
- Even though, the 32 bit field can represent a number between 0 and 4,294,967,295 but a timestamp in this case can have the maximum value of 86,400,000 = $24 \times 60 \times 60 \times 1000$.
- The **timestamp request** message is created by the source. It fills the original timestamp field at departure time, and fills the other two timestamp fields will zeros.
- The **timestamp** reply message is created by the destination host.
- The original timestamp value from the timestamp request message is copied as it is into the original timestamp field in the timestamp reply message, by the destination.
- The destination then fills up the **receive timestamp** field by the time at which the request was received. At the end the destination fills up the **transmit timestamp** field with the **departure time of the reply** message.

**Computation of one way or round trip time (RTT) :**

- We can use the pair of timestamp messages to compute the one way or RTT i.e. the time required by the datagram to travel from source to destination and then come back to source again, as follows :

Sending time = receive timestamp – original timestamp.

Receiving time = returned time – transmit timestamp.

Round trip time = sending time + receiving time.

- If we want the calculations of the sending time and receiving time to be accurate, then the two clocks in the source and destination computers should be synchronized.
- But the calculation of RTT will be correct even if the clocks at the source and destination machines are not synchronized. We can calculate the one way time duration by dividing the RTT by two.

### 5.18.3 Deprecated Messages :

- IETF has declared the following three pairs of query messages as obsolete :
  1. Information request and reply messages.
  2. Address mask request and reply messages.
  3. Router solicitation and advertisement.

**1. The information request and reply messages :**

- These messages are not used now a days because the Address Resolution Protocol (ARP) is doing their duties.

**2. Address mask request and reply :**

- The IP address of a host contains a network address, subnet address and host identifier.
- A host may know its full IP address but may not know it is divided into three parts mentioned above.
- So it can send an address mask request message to the router. The router then sends back the address mask reply message.
- These messages are not being used today because their duties are done by the Dynamic Host Configuration Protocol (DHCP).

**3. Router solicitation and advertisement :**

- A host that wants to send data to a host on another network must know the address of routers connected to its own network.
- In such situations the router solicitation and advertisement messages can help.

- A host can broadcast or multicast a router solicitation message.
- The routers receiving this message can broadcast their routing information using the router advertisement message.
- These messages are not being used today because their duties are done by the DHCP.

### 5.18.4 Checksum :

- Earlier we have discussed the concept of checksum. In ICMP, the entire message (including the header and data) is considered for calculation of checksum.

**Checksum calculation :**

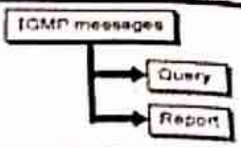
- The checksum calculation is done at the sending end by following the steps given below :
  1. Set the checksum field to zero.
  2. Calculate the sum of all the 16 bit words including header and data.
  3. Obtain the checksum by complementing the sum calculated in step 2.
  4. Store the checksum in the checksum field.

**Checksum testing :**

- The following steps are followed by the receiver using 1's complement arithmetic :
  1. Calculate the sum of all words (header and data).
  2. Complement the sum calculated in step 1.
  3. Accept the message if the result obtained in step 2 is 16 zeros. Otherwise the message is rejected.

## 5.19 Internet Group Management Protocol (IGMP) :

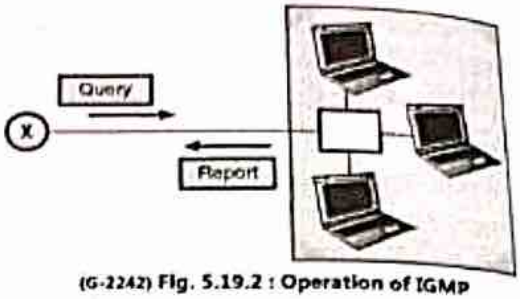
- Today, for collection of information about group membership the IGMP (Internet group management protocol) is used.
- IGMP is one of the auxiliary protocol defined at the network layer which is considered as a part of Internet protocol.
- IGMP messages are encapsulated in datagram similar to ICMP messages.
- Fig. 5.19.1 shows two types of messages in IGMP version 3.




(G-2241) Fig. 5.19.1

### 5.19.1 Operation of IGMP :

- Fig. 5.19.2 shows the operation of IGMP :




(G-2242) Fig. 5.19.2 : Operation of IGMP

**1. Query message :**

- A router sends query message periodically to all hosts which are attached to it to ask them for reporting their membership interest in group.
- In IGMPv3, a query message can be in any one of three forms :
  a. a general query message,
  b. a group specific query message,
  c. a source and group specific message.

**a. General query message :**

- In any group a general query message is sent about membership.
- With the destination address 224.0.0.1, a general query message is encapsulated in a datagram.
- All routers which are connected to the same network receive this general query message and inform them about the message which is already sent and avoid them from resending the message.

**b. Group specific query message :**

- To ask about a specific group membership, this message is sent from a router.
- If router do not receive any response about specific group and router want to make sure about a membership of that group in the network, then this group specific query message is sent. Multicast address (group identifier) is mentioned in the message.

- In a datagram, the message is encapsulated with destination address set to the corresponding multicast address.
- This message is received by all the hosts and those who are not interested they will drop this message.

c. **Source and group specific query message :**

- When the message comes from a specific source, router sends this message to ask about membership related to a specific group.
- If router is not able to hear a specific group related to a specific host, then again this message is sent.
- With the de
- stination address set to the corresponding multicast address the message is encapsulated in a datagram.
- This is message is received by all the host and those who are not interested they will drop this message.

2. **Report message :**

- To give a response to a query message host sends report message.
- Report message contains :
  1. List of records (in which each record gives the identifier of corresponding group i.e. multicast address).
  2. Addresses of all sources in which the host is interested in receiving messages.

3. **Addresses of sources from which the host do not want to receive a group message.**

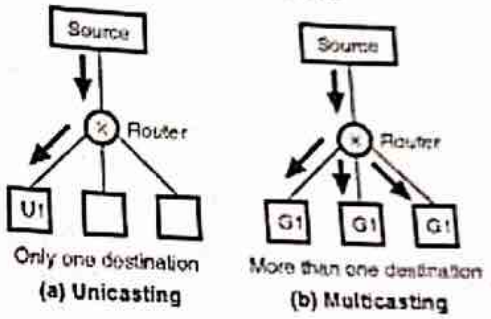
- In a datagram, the message is encapsulated with the multicast address 224.0.0.22 which is allocated to IGMPv3.
- In IGMPv3, if any host want to join a group, it will wait to receive a query message and then host sends a report message.
- If host want to leave the group then it will not respond to a query message.
- The group is eliminated from the router database if no hosts responds to corresponding message.

## 5.19.2 Multicast Forwarding :

- In multicasting, another issue is about the decision a router want to make for forwarding a multicast packet.
- In unicast and multicast communication, forwarding is different in two aspects which are discussed as follows :

1. **Destination in unicasting and multicasting :**

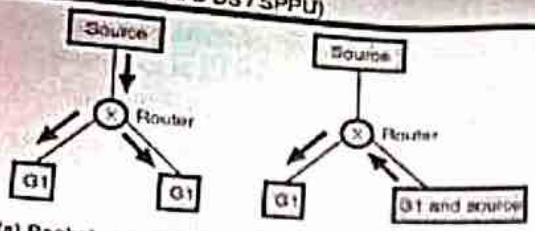
- In unicasting, only one single destination is defined by the destination address of the packet.
- It is necessary to send packet only out of one of the interface. And such interface is used which is the branch in shortest path tree which reaches the destination with minimum cost.
- In multicasting, one group is defined by the destination address of packet, but in the Internet that group can have more than one member.
- A router have to send the packet by using more than one interface to reach all of the destinations.
- This concept is as shown in Fig. 5.19.3. In unicast communication in the Internet U1 (destination network) cannot be in more than one part whereas G1(group) can have members in more than one part.




**(G-2243) Fig. 5.19.3 : Destination in unicasting and multicasting**

2. **Forwarding decision :**

- In unicasting, forwarding decision depends on the destination address of packet whereas in multicasting, it depends on both source and destination address of the packet.
- Forwarding in unicasting is based on where the packet should go, whereas forwarding in multicasting is based on where the packet has come from and where it should go. Forwarding concept is as shown in Fig. 5.19.4.
- Fig. 5.19.4(b) shows that the source is present in the part of the Internet where there is group member available.




(a) Packet sent out of two interfaces (b) Packet sent out of one interface

(G-2244) Fig. 5.19.4 : Forwarding

- To avoid sending second copy of packet from the interface it has arrived at the router must send the packet from only one interface. Fig. 5.19.4(b) shows that the member/members of the group G1 have already obtained a copy of packet.
- At the router when it arrives, if packet sent out in that direction does not help furthermore it generates more traffic.
- From this it is clear that in multicast communication forwarding depends on both source and destination address.

### 5.19.3 Multicasting Approaches :

- Similar to unicast routing, multicast routing need to generate routing tress to route optimally the packets from their source to destination.
- As discussed earlier the multicast routing decision at each router depends on the destination and source of the packet.
- As compared to the unicasst routing, the involvement of the source in the routing process makes multicast routing more difficult. For this reason following two approaches are designed in multicast routing :
  1. Routing using source based trees.
  2. Routing using group shared trees.

**1. Routing using source-based trees :**

- In this approach, each router is required to generate a separate tree for each combination of source and group.
- Suppose in the Internet there are m number of groups and n number of sources. Then a router need to generate (m × n) routing trees.
- In each tree, source is the root, the members of the group are leaves and router itself is present somewhere on the tree.
- In unicast routing router needs only one tree in which router itself acts as root and in the Internet all networks acts as the leaves.
- But it can appear, about all these trees router needs to store and create a large amount of information.
- In the Internet recently there are two protocols which use this approach which we will disuss later in this chapter.

**2. Routing using group shared tree :**

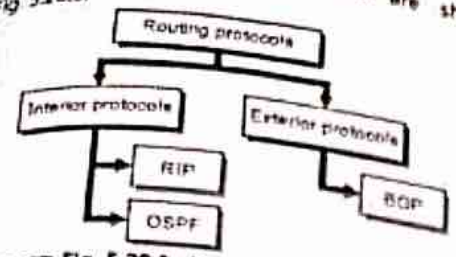
- In this approach, for each group we assign a router to act as the phony source.
- The allocated router which is known as **core router** which acts as the representative for the group.
- If source has a packet to send to a member of group then it send that packet to the core centre (i.e. unicast communication) and for multicasting core centre is responsible.
- One single routing tree is created by the core centre and itself is the root and in group any routers with active members acts as leaves.
- If there are m core routers and each has a routing tree for m trees.
- That means number of routing trees are m in this approach which are reduced from (m × n) in the source based tree.

## 5.20 Routing Protocols :

- Routing protocols are designed on the basis of the demand for dynamic routing tables.
- The router in an Internet are supposed to inform each other about changes.
- Routing protocols combine the rules and procedures which allow the routers to exchange information about these changes between themselves.
- We can divide the routing protocols into two categories **Interior protocols** and **exterior protocols.**
- We can define an **interior protocol** as the one that handles the **intradomain routing.**
- Similarly an exterior protocol is defined as the one which handles the **interdomain** routing.

### 5.20.1 Unicast Routing Protocols :

- Various unicast routing protocols are shown in Fig. 5.20.1.




(G-497) **Fig. 5.20.1 : Unicast routing protocols**

- The popular interior protocols are RIP (Routing Information Protocol) and OSPF (Open Shortest Path First).
- Whereas the exterior protocol used popularly is BGP (Border Gateway Protocol).
- RIP and OSPF are used to upgrade the routing tables inside an A.S. and BGP is used for upgrading the routing tables for the routers which join multiple A.S. together.

## 5.21 RIP (Routing Information Protocol) :

- RIP is used for updating the routing tables. The routing updates are exchanged between the neighbouring routers after every 30 seconds with the help of the RIP response message.
- These messages are also known as the RIP advertisements. These messages are sent by the routers or hosts. They contain a list of multiple destinations within an Autonomous System (AS).
- RIP is an interior routing protocol used inside an Autonomous System (AS). Its operation is based on distance vector routing.
- In the distance vector routing each router periodically shares its knowledge about the whole Internet with its neighbours.
- As stated earlier, RIP is a very simple intradomain or interior routing protocol which works inside an **Autonomous System (AS).**
- RIP implements the distance vector routing with the following considerations :

1. In an A.S. it has to deal with routers and networks (links) and not the nodes.
2. Now the destination in a routing table is a network. That is why, the network address in defined in the first column.
3. The **metric** used in RIP is called as the **hop count** and it is very simple. It is defined as the number of links a packet has to travel to reach its destination.
4. In RIP, the value of **infinity** is decided to be equal to 16. That is why the maximum **hop count** for any route inside an A.S. using RIP can be 15.
5. The next node column is used to define the address of the router to which the packet is to be dispatched.

**Routing table :**

- A typical routing table is shown in Table 5.21.1.

**Table 5.21.1 : Routing table**

| Destination | Hop count | Next router | Other information |
|---|---|---|---|
| | | | |

- Every router is supposed to keep such a table with it.
- Destination column consists of the destination network address.
- The hop count column consists of the shortest distance to reach the destination and the next router column consists of the address of the next router to which the packet is to be forwarded.
- The other information in Table 5.21.1 may include information such as subnet mask or the time this entry was last updated.

### 5.21.1 RIP Updating Algorithm :

- The routing table is updated when a RIP response message is received as stated earlier. The updating algorithm used by RIP is as follows :

**RIP updating algorithm :**

1. RIP response message is received.
2. Add one hop to the hop count for each advertised destination.
3. Repeat the following steps for each advertised destination :

- Add the advertised information to the table if the destination is not present in the routing table.

- Replace entry in the table with the advertised one if the next hop field is same.
- Replace entry in the routing table if advertising hop count is smaller than one in the table.

4. Return.

### 5.21.2 Initializing the Routing Table :

- When a new router is added to a network it initialises its routing table.
- Such a table consists of the information only about the directly attached networks and the corresponding hop counts.
- The next hop field which identifies the next router is empty.

### 5.21.3 Updating the Routing Table :

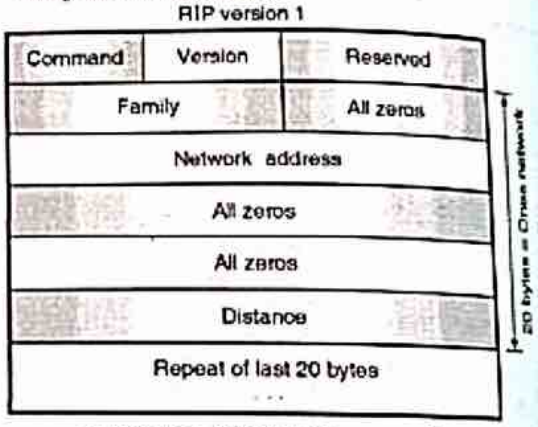
- When RIP messages are received, each routing table is updated using the RIP updating algorithm as discussed earlier.

### 5.21.4 RIP Operation :

- RIP work is a combination of a routing database that stores information on the fastest route from computer to computer, an update process that enables each router to tell other routers which route is the fastest from its point of view, and an update algorithm that enables each router to update its database with the fastest route communicated from neighboring routers.
- Each router on the Internet keeps a database that stores the following information for every computer in the same RIP network :
- **IP address** : The Internet Protocol address of the computer.
- **Gateway** : The best gateway to send a message addressed to that IP address.
- **Distance** : The number of routers between this router and the router that can send the message directly to that IP address.
- **Route change flag** : A flag that indicates that this information has changed used by other routers to update their own databases.
- **Timers** : Various timers.
- At regular intervals each router sends an update message which has full information about its routing database to all the other routers that are directly connected to it.
- Some routers will send this message as often as every 30 seconds, so that the network will always have up-to-date information.
- RIP uses the UDP network protocol because of its efficiency and there are no problems if a message gets lost due to any reason. This is because the next update will be coming in a short time.

### 5.21.5 RIP Message Format :

- RIP messages can be broadly classified into two types : messages that deliver routing information and messages that request routing information.
- Both use the same format which consists of a fixed header followed by an optional list of network and distance pairs.
- The summary of the RIP packet format fields illustrated in Fig. 5.21.1, is as follows :




(G-1998) **Fig. 5.21.1 : RIP message format**

1. **Command :**

- Indicates whether the type of the packet i.e. a request a response.
- The request asks that a router send all or part of its routing table.
- The response can be an unsolicited regular routing update or a reply to a request.
- Responses contain routing table entries. Multiple RIP packets are used to convey information from large routing tables.

2. **Version :**

- This field specifies the RIP version used. This field can signal different potentially incompatible versions.

3. **Zero :**

- This field is not actually used by RFC 1058 RIP. It was added just to provide backward compatibility with the older versions of RIP. Its name actually indicates its defaulted value: zero.

4. **Family :**

- This field is used to specify the address family used. RIP is designed to carry routing information for several different protocols.
- Each entry has an address-family identifier to indicate the type of address being specified. For example the value of AFI for IP is 2. Similarly different values indicate different protocols.

5. **Network address :**

- The network address field is used for defining the address of the destination network. In RIP this field is 14 bytes long, so that it can be used for any protocol.
- But the IPv4 address is only 4 byte long. Hence the remaining space in the address field is filled with zeros.

6. **Distance :**

- This field indicates the number of hops (routers) that have been traversed in the trip to the destination. This value is between 1 and 15 for a valid route, or 16 for an unreachable route.

## 5.21.6 Request Message :

- The request message is created in the following two situations :
  1. It is created by a router which has just come up.
  2. Or it is created by a router which has some time out entries.
- In a request message, information about some specific entries or all the entries is asked.
- Fig. 5.21.2(a) shows the format of the request message for one and Fig. 5.21.2(b) shows the format of request message for all.

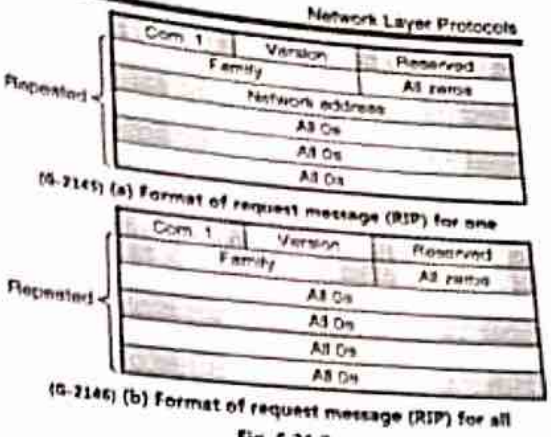



Fig. 5.21.2

## 5.21.7 Response Message :

- Response message in RIP can be one of the following two types :
  1. Solicited response or
  2. Unsolicited response.
- A **solicited response** is the one which is sent only as an answer to a request message.
- It carries with it the information about the destination specified in the request message.
- An **unsolicited response,** is not sent only once but it is sent periodically (every 30 seconds or so) when there is any change in the routing table. This response is also called as the update packet.

## 5.21.8 Timers in RIP :

- RIP uses three different timers as follows for supporting its options.
  1. The **periodic timer** to control the process of sending messages.
  2. The **expiration timer** is used for governing the validity of a route.
  3. The **garbage collection timer** is used for advertizing the failure of a route.

1. **Periodic timer :**

- The task of the periodic timer is to control the advertizing of the update messages regularly.
- As per protocol specifications, this timer should be set to 30 sec. but practically it is set randomly between 25 and 35 sec. Each router has one periodic timer.

- This timer counts down from the set value (25 to 35 sec.) and sends an update message when its count reaches a zero.
- Then the timer is set once again to a random value between 25 and 35 seconds.

2. **Expiration timer :**

- The responsibility of expiration timer is to govern the validity of a route. When a router gives out the update information about a route, the value of this timer is set at 180 sec or 3 minutes.
- This timer is reset, everytime a new update for that route is received, which under normal working conditions happen after every 30 sec.
- But due to some problem on the Internet, if a new update for that route is not received within 180 sec, then that route is considered expired and the hop count of that route is set to 16.
- This is an indication that the destination is not reachable. There is a separate expiration timer for each route.

3. **Garbage collection timer :**

- The router does not purge a particular route from its table even when the information about that route becomes invalid.
- Instead the router continues to advertise that route by increasing its metric value to 16 (destination is not reachable).
- At the same time, the router sets another timer called **garbage collection timer** to 120 sec. for this route.
- As soon as this count goes to zero, that route is purged from the router table,
- Due this timers the neighbours become aware that a particular root has become invalid, before its purging.

## 5.22 OSPF :

- The long form of OSPF is Open Shortest Path First protocol.
- This is another interior routing protocol. It is an intradomain protocol and it is based on the link state routing.
- For handling the routing efficiently and in a timely manner, the OSPF divides an A.S. into areas.

**Area :**

- Networks, hosts and routers are collectively called as an area. An autonomous system can be imagined to be made of various areas. All the networks inside an area should be connected.

**Area border routers :**

- These are special type of routers which are used at the borders of an area. These routers summarize the information about the area and sent it to the other areas.

**Backbone :**

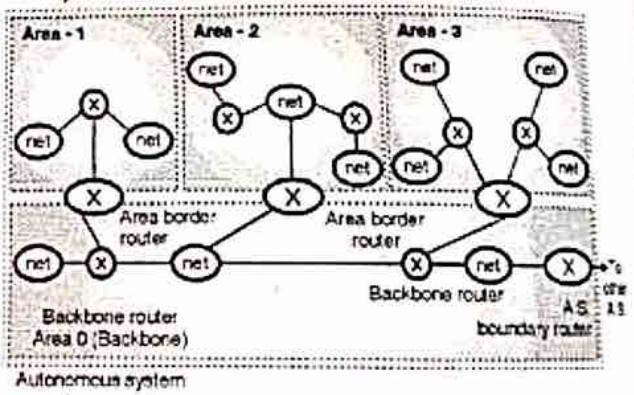
- A special area inside an autonomous system is called as backbone.
- All the areas inside an A.S. should be connected to the backbone. So backbone is the primary area and other areas are known as secondary areas.

**Backbone routers :**

- The routers inside the backbone are called as the backbone routers. But a backbone router can also work as an area border router.
- If the connectivity between a backbone and an area is broken, due to some problem, then the administration should create a **virtual link** between routers so that the backbone can continue to function as primary area.

**Area identification :**

- Each area has an area identification. The area identification of the backbone is zero. An autonomous system is as shown in Fig. 5.22.1(a).




(G-1786) **Fig. 5.22.1(a) : Autonomous system**

**Disadvantages of the RIP protocol :**

- The maximum distance between any two stations (the metric, measured in router hops) is 15 hops. A destination (network ID) whose hop count is 16 or more is considered to be unreachable.

- The cost to a destination network is measured in terms of number of hops.
- RIP determines a route based on a hop count that does not take into consideration any other criteria other than the number of routers between the source and destination networks.
- Due to this approach two-hop high-speed network will be ignored and a one-hop low-speed link would be used instead.
- We can make a router to take a better path by adjusting the hop-count metric on the router port, but this reduces the available diameter.
- RIP updates its entire table on a periodic basis using the broadcast address. (RIPv1; RIPv2 uses multicast or broadcast). But this would consume bandwidth.
- RIP sends its update with the help of a 576 byte datagram. If there are more entries than 512 bytes, then multiple datagrams must be sent.
- The biggest drawback of RIP is its slow convergence. In the worse case, a RIP update can take over 15 minutes end to end. This can lead to black holes, loops, etc. RIPv1 does not support VLSM.

**Remedies (What OSPF could do) :**

- The first shortest-path-first routing protocol was developed and used in the ARPAnet packet switching network all the way back in 1978.
- This research work was developed and used in many other routing protocol types and prototypes. One of those is OSPF.
- OSPF provides solutions to most of the drawbacks of RIP. Using OSPF we can scale up the routing architecture well beyond the maximum 16 hops supported by RIP.
- Rather than exchanging node (and network) reachability information, OSPF routers exchange link state information. Through the link state information, each router maintains its own copy of the network topology.
- From this link-state database, it is possible to find the shortest routing path.
- For those of you that are familiar with the OSI routing scheme, many of the features supported by OSPF are similar to the OSI IS-IS routing protocol.
- The original versions of OSPF are actually derived from some of the earlier versions of the IS-IS protocol.

### 5.22.1 Features of OSPF :

**1. Type of service routing :**

- It is possible to configure different routers to support different types of service requirements.
- For example, one router can be configured for high-throughput, while the other one is configured to support minimal delivery delay for some other application.

**2. Load balancing :**

- When multiple routes are available, traffic can be evenly distributed over the routes. This would obviously result in a higher network efficiency.

**3. Subdivision of autonomous systems :**

- It is possible to further divide the system into logical areas. This would improve the management of large autonomous systems.

**4. Security :**

- The data exchanges in OSPF are authenticated. Inadvertent or malicious transmissions from foreign routing nodes are disc arded.
- Only those hosts intended for the routing network are included.
- The network isn't vulnerable to the threat of having routing tables corrupted by faulty route information.

**5. Host :**

- OSPF supports specific, network and subnetwork routing.

**6. Special features are provided to support LAN environments :**

- Although the relationships between routers are maintained on a logical link basis, link state transmissions are minimized by the architecture. **Designated gateways** are responsible for transmitting the link state information for all information in their local area.

**7. OSPF is an open specification :**

- The OSPF has been published as an RFC and not defined as a defacto standard such as RIP.

...e to encourage many vendors to use it so that the users are not locked into a single vendor's equipment.

8. **OSPF area :**

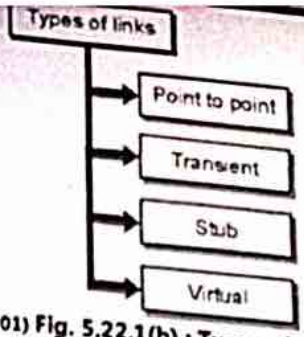
- OSPF divides the network into groups, called an **area**. The topology of an area is not known to the rest of the Autonomous System.
- This technique minimizes the routing traffic required for the protocol. When multiple areas are used, each area has its own copy of the topological database.
- Several concepts have been incorporated in the OSPF algorithm.
- The RIP treated an autonomous system as a monolithic collection of routes and subnets, but OSPF introduces the concept of areas.
- The concept of hiding the routing information within a OSPF routing domain (Internet autonomous system) has also been introduced.
- After dividing an autonomous system into a collection of logical areas, the OSPF can support different types of routing nodes (routers) such as internal routers, area border routers, backbone routers, and Autonomous System (AS) boundary routers. (See Fig. 5.22.1(a)).
- The protocols used to support OSPF routing include database broadcast packets and link state change broadcasts.
- A "Hello" protocol is used to detect changes in the availability of adjacent routers.

## 5.22.2 Metric :

- The cost assigned to each route by an OSPF administrator is called as metric of that route. In the OSPF protocol the metric can be based on a type of service.
- A router can have multiple routing tables which are based on different types of service.

## 5.22.3 Types of Links :

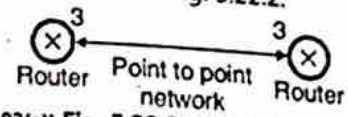
- In the OSPF protocol terminology, a connection is called as a link. OSPF defines four types of links called point to point, transient link, stub link and virtual links as shown in Fig. 5.22.1(b).




**(G-501) Fig. 5.22.1(b) : Types of links**

1. **Point to point link :**

- A point to point link is defined as the link (connection) that directly connects two router without any other host or router present in between.
- An example of such a link is two routers connected by a telephone line.
- Each router has only one neighbour at the other side of the link. This is shown in Fig. 5.22.2.

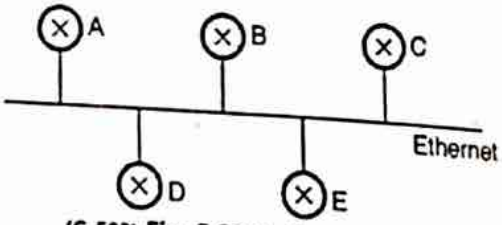



**(G-502(a)) Fig. 5.22.2 : Point to point link**

- It is not necessary to assign any network address to this link. The metric are shown at the two ends of the link and they are generally the same.

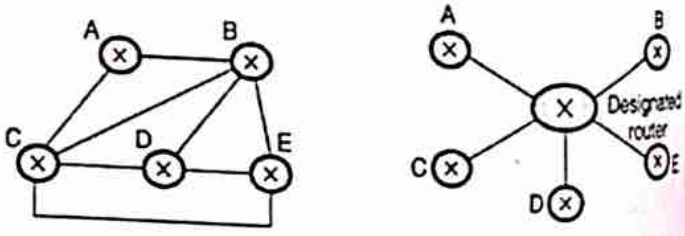
2. **Transient link :**

- It is a network having many routers attached to it as shown in Fig. 5.22.3.




**(G-503) Fig. 5.22.3 : Transient link**

- All LANs and some WANs are of this type.
- A, B, C ..... etc. are the routers. Each router has several neighbours.
- The relationship between the neighbouring routers is as shown in Fig. 5.22.4(a).




**(a) Unrealistic representation (b) Realistic representation**

**(G-1787) Fig. 5.22.4**

- Each router has been connected to every other neighbour.
- But this arrangement is extremely nonefficient and non-realistic.
- In order to make it more efficient and realistic, the configuration of Fig. 5.22.4(b) should be used. This is known as the transient network.
- The designated router is assigned to perform two tasks, one as a true router and the other as a designated router.
- Due to the realistic arrangement of Fig. 5.22.4(b) every router has only one neighbour i.e. the designated router (network), however the designated router has multiple (5 in this case) neighbours.
- The realistic arrangement reduces the number of announcement that each router has to make to a small number as compared to the unrealistic arrangement.
- Note that there is a metric from each node to designated router and there is no metric from the designated router to any other node.

3. **A stub link :**

- A stub link is a network that is connected to only one router as shown in Fig. 5.22.5.


**(a) Stub network** **(b) Representation**

**(G-504) Fig. 5.22.5**

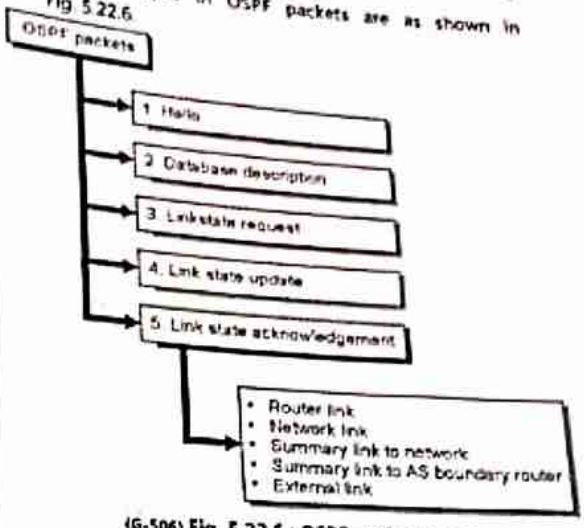
- The stub network of Fig. 5.22.5(a) is a special case of transient network. The data packets use the same link to enter and leave the network.
- This situation can be represented by using router A as a node and by replacing the network by a designated router as shown in Fig. 5.22.5(b).
- The link connecting router A and the designated router is unidirectional from router to network.
- When this link gets damaged the administration can create a virtual link between the two routers.

## 5.22.4 Virtual Link :

- The administration can create a **virtual link** between two routers, when a link between them gets broken due to some reason.
- Such a virtual link could be over a longer path which would go through many routers.

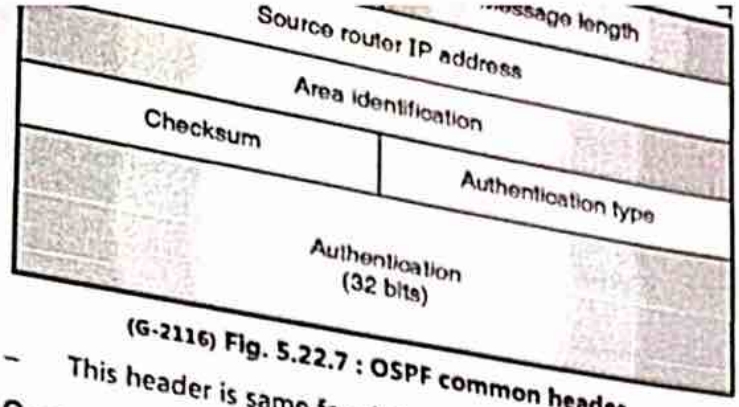
## 5.22.5 OSPF Packet Types :

- Different types of OSPF packets are as shown in Fig. 5.22.6.




**(G-506) Fig. 5.22.6 : OSPF packet types**

- The OSPF protocol runs directly over IP, and uses the assigned number 89.
- Each OSPF packet consists of an OSPF header followed by the body of a particular packet type.
- OSPF packets need to be sent to specific IP addresses in nonbroadcast multi-access networks.
- The OSPF operation consist of following stages :
- Neighbours are discovered by means of sending the Hello messages and designated routers are elected in multi-access networks. Adjacent routers are identified and link state databases are synchronized.
- Link State Advertisements (LSA) are exchanged among the adjacent routers so as to maintain the topological databases and also to advertise interarea and interAS routes.
- The routers use the information in the database to generate routing tables.
- All OSPF packets have the same common header which is as shown in Fig. 5.22.7.




(G-2116) Fig. 5.22.7 : OSPF common header

- This header is same for all the five packet types of OSPF.

**Common Header :**

- Various fields in the OSPF packet header are as follows :

**Version :**

- The contents of this 8-bit field tells us about the version of the OSPF protocol. It is currently version 2.

**Type :**

- This 8-bit field defines the type of the packet. There are five types of OSPF packets and they can defined by adjusting the contents of the type field from 1 to 5.

**Message length :**

- This 16-bit field defines the length of the total message which includes the header as well as the body.

**Source router IP address :**

- This 32-bit field defines the IP address of the router that sends the packet.

**Area identification :**

- This 32-bit field defines the area within which the routing takes place.

**Checksum :**

- This field is used for error detection on the entire packet excluding the authentication type and authentication data field.

**Authentication type :**

- This 16-bit field defines the authentication method used in this area.
- At this time, two types of authentication are defined : A 0 in this field shows that no authentication is being used and a 1 represents the use of password for authentication.

**Authentication :**

- This 64-bit field is the actual value of the authentication data.
- In the future, when more authentication types would be defined, this field will contain the result of the authentication calculation.
- For now, if the authentication type is 0, this field is filled with 0s. If the type is 1, this field carries an eight-character password.

## 5.22.6 Comparison between RIP and OSPF :

**Table 5.22.1 : Comparison between RIP and OSPF**

| Function/Feature | RIPv1 | OSPF |
|---|---|---|
| Standard number | RFC 1058 | RFC 2178 |
| Link-state protocol | No | Yes |
| Large range of metrics | Hop count (16=Infinity) | Yes, based on 1-65535 |
| Update policy | Route table every 30 seconds | Link-state changes, or every 30 [minutes] |
| Update address | Broadcast | Multicast |
| Dead interval | 300 seconds total | 300 seconds total, but usually much less |
| Supports authentication | No | Yes |
| Convergence time | Variable (based on number of routers X dead interval) | Media delay + dead interval |
| Variable-length subnets | No | Yes |
| Supports supernetting | No | Yes |
| Type of Service (TOS) | No | Yes |
| Multipath routing | No | Yes |
| Network diameter | 15 hops | 65535 possible |
| Easy to use | Yes | No |

# 5.23 Border Gateway Protocol (BGP) :

- BGP is an exterior routing protocol. It is a unica routing protocol.

It is used for the interautonomous system routing i.e. routing among different ASs.

It was introduced in 1989 and has four versions. BGP operation takes place on the basis of the routing method called **path vector routing.**

This principle is used because the distance vector routing and link state routing do not prove to be much suitable for interautonomous system routing.

### 5.23.1 Types of Autonomous Systems :

We have already discussed about autonomous systems. Now let us discuss about their types. The three categories of autonomous systems are as follows :

1. Stub AS 2. Multihomed AS 3. Transit AS.

**Stub AS :**

A stub AS is that type of AS which has only one connection to another AS. The hosts in the AS can send and receive data traffic to the hosts belonging to other AS.

But note that data traffic cannot pass through a stub AS. In other words the stub traffic can be either a source or sink.

**Multihomed AS :**

An AS which has more than one connection to other ASs is known as multihomed AS. But it is interesting to note that a multihomed AS is still only a source or sink for data traffic.

For a host in multihomed AS, it is possible to send and receive data traffic to from more than one AS. But it does not allow the **transient traffic.**

That means, the multihomed AS does not allow the data traffic coming from one AS to just pass through to the other AS.

**Transit AS :**

An AS which is a multihomed AS but also allows the transient data traffic is called as **transit AS.**

### 5.23.2 CIDR :

A Classless interdomain addressing is used in BGP. That means BGP makes use of the prefix (As discussed earlier) for defining a destination address.

### 5.23.3 Path Attributes :

The path for a destination address can be presented as a list of **attributes.** We get some information from each attribute about the path.

- The receiving router takes the help of this list of attributes for making a better decision when applying its policies.

### 5.23.4 Types of Attributes :

- There are two categories of attributes :
  1. A well known attribute. 2. An optional attribute.
- Every BGP router must recognize the **well known** attribute whereas the **optional attribute** is the one which need not be recognized by every router.
- The well known attributes are further classified into two types namely **mandatory** and **discretionary.** We define the well known mandatory attribute as the one which must appear in the description of router.
- On the other hand a well known discretionary attribute can be defined as the one which must be recognized by each router, but it need not be included in every update message.
- We can also subdivide the optional attributes into two categories as : **transitive** and **nontransitive** optional attributes.
- We may define the optional transitive attribute as the one which should be passed to next router that has not implemented this attribute.
- Similarly an optional nontransitive attribute is defined as the one which must be discarded if the receiving router has not implemented it.

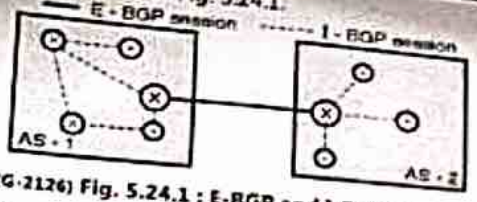
## 5.24 BGP Sessions :

- In a BGP **session,** the two routers using BGP exchange routing information between them. So we can define a session as **connection** which has been established between two BGP routers in order to exchange the routing information.
- In order to ensure a reliable session the BGP uses services of TCP. The speciality of such a connection that it lasts for a longer time until something unusual happens.
- Therefore the BGP sessions are called as the **semipermanent connections.**

### 5.24.1 External and Internal BGP :

- There are two types of BGP sessions as follows :
  1. External BGP (E-BGP) session.
  2. Internal BGP (I-BGP) session.
- We can use the **E-BGP** session for exchanging information between two nodes which are present in two different ASs.

- This is as shown in Fig. 5.24.1.

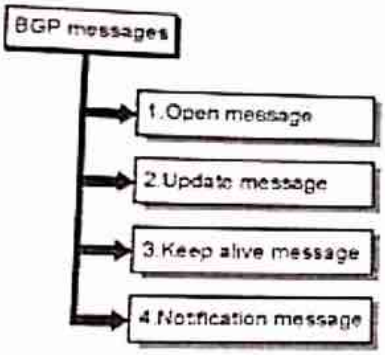



(G-2126) **Fig. 5.24.1 : E-BGP and I-BGP sessions**

- In Fig. 5.24.1, the session (connection) shown between AS-1 and AS-2 is E-BGP session. It is shown by a bold line. The two speaker routers $A_1$ and $B_1$ will exchange all the information which is known to them over the E-BGP session.
- But these routers collect information from the other routers belonging to their own A.S. using the I-BGP sessions shown by dotted lines in Fig. 5.24.1.

### 5.24.2 Types of Messages :

- BGP uses four different types of messages, as shown in Fig. 5.24.2.



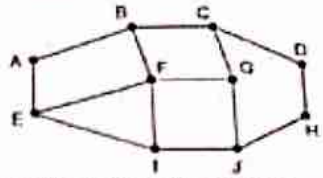

(G-508) **Fig. 5.24.2 : BGP message types**

### 5.24.3 Encapsulation :

- BGP messages are encapsulated in TCP segments by using the well known port 179. The error control and flow control are therefore not needed.
- After opening a TCP connection, the update, keep a live and notification messages are exchanged until a notification message is sent.

### 5.24.4 How does BGP Solve the Count to Infinity Problem ?

- The BGP is basically a distance vector protocol. But it is very much different from the most other protocols such as RIP.
- Instead of maintaining just the cost of each destination, each BGP router keeps track of the path used.
- Similarly instead of periodically giving each neighbour its estimated cost to each possible destination, each BGP router tells its neighbour the exact path that it is using.
- Fig. 5.24.3 shows a set of BGP routers and Table 5.24.1 shows the information that router F receives from its neighbours about "D".




(G-512) **Fig. 5.24.3 : A set of BGP router**

- BGP can solve the count to infinity problem easily. This can be explained as follows : Suppose that the router G in Fig. 5.24.3 crashes, or if the line FG becomes faulty, then router F receives routes from the remaining three neighbours i.e. B, I and E.
- As shown in Table 5.24.1, these routes are BCD, IFGCD and EFGCD.

**Table 5.24.1 : Information received by F from neighbours about D**

| Neighbour | Information |
|---|---|
| B | I use path BCD to reach D. |
| G | I use path GCD to reach D. |
| I | I use path IFGCD to reach D. |
| E | I use path EFGCD to reach D. |

- Looking at these routes, router F immediately understands that, the routes IFGCD and EFGCD are useless because they pass through F itself. So it decides to choose FBCD path as a new route. This avoids the count-to-infinity problem.

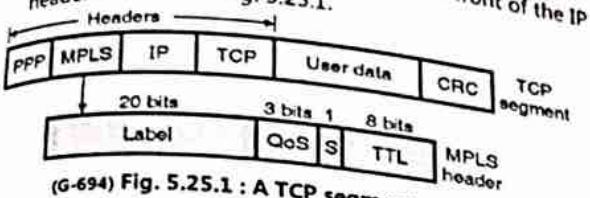
## 5.25 MPLS (Multi-Protocol Label Switching) :

- When IETF was developing integrated and differentiated services, several router vendors were developing a new and better forwarding method called as **label switching** or **tag switching**. IETF eventually standardized this idea under the name **MPLS (Multi Protocol Label Switching).**
- In this method, a label is added in front of each packet and the rooting is done on the basis of this label and not on the basis of the destination address.
- This label acts as an index into an internal table. Due to this, it becomes very easy to find the correct output line by referring to the table. This idea thus makes the routing process very fast.

- The concept of label switching is very close to that of the virtual circuits (used in X.25, Frame relay and ATM). In the virtual circuits also, they put the labels called Virtual Circuit Identifier (VCI), in each packet and rooting is carried out on the basis of VCI.
- MPLS is described in RFC 3031 and many other RFCs.

### 5.25.1 MPLS Header :

- The first step in MPLS will be to decide the place of the label in the IP packet. In the IP packet there is no place available for the label because it was not designed for virtual circuits.
- Therefore a new MPLS header is added in front of the IP header as shown in Fig. 5.25.1.




(G-694) **Fig. 5.25.1 : A TCP segment using IP, MPLS and PPP headers**

- The generic MPLS header has four fields :
  1. Label 2. QoS 3. S-field and 4. TTL.
- The **Label** field is a 20 bit field which holds the index as shown in Fig. 5.25.1. The 3-bit QoS (Quality of Service) field indicates the class of service.
- The 1-bit **S-field** relates to stacking of multiple labels in the hierarchical networks. If it hits a 0, then the packet is discarded. This feature avoids the infinite looping in the event of rooter instability.
- MPLS is to a large extent independent of both data link layer as well as network layer because the MPLS header is not a part of either the network layer packets or data link layer frames.
- It is therefore possible to build the MPLS switches that can forward both IP packets and ATM packets.
- That is why MPLS is called as a "Multi Protocol" switching technique.

### 5.25.2 How does MPLS Work ?

- When an MPLS packet or cell arrives at an MPLS router, the label is used as an index into the look up table to find out the correct outgoing line and also the new label to be used.
- This new label contains the address of the next MPLS router. The labels have to be remapped at every hop, similar to that in the virtual circuits.
- The routers normally group multiple flows that end at a particular router or LAN and use a single label for them. The flows grouped under a single label belong to the same **FEC (Forwarding Equivalence Class).**
- The FEC covers the following aspects :
  1. Where the packets are going.
  2. Their service class.
- All the packets under the same FEC are treated in the same way for forwarding purpose.
- With the virtual circuit switching it is not possible to group several distinct paths with different end points onto the same VCI. This is possible with MPLS because the packets contain the destination address as well as the label.

### 5.25.3 Forwarding Table :

- One main difference between MPLS and conventional VC (Virtual Circuit) is the way in which the forwarding table is constructed. In the VC networks, when a user wants to establish a new connection, it launches a set up packet into the network to create the path and make the entries into the forwarding table.
- The MPLS does not work this way as there is no set up phase for each connection.
- Instead there are two ways of creating the forwarding table entries. The two approaches are :
  1. Data driven approach
  2. Control driven approach.
- In the **data driven approach,** when a packet hits a router, that router will contact the next router where the packet will be going and asks it to generate a label for the flow.
- The protocols used in this approach use a technique called **coloured threads** to avoid loops.
- **Control driven approach,** is used on the networks that are not based on ATM. It has several variants. One of them is as follows :
- When a router is booted, it checks the roots depending on the final destination. It then creates one or more FECs for them, allocates a label for each one and passes the labels to its neighbours. The neighbouring routers will enter the labels in their forwarding tables and send new labels to their neighbours. This will continue till all the routers have received the information about the path.
- MPLS can operate at multiple levels simultaneously. The S-bit in Fig. 5.25.1 allows a router to remove a label to know if there are any additional labels left.
- The S-bit is set to 1 for the bottom label and 0 for all other labels.

...g is an activity which connects calls from source to destination in telecommunication networks which plays an important role in construction, design and operation of the network. Ad-hoc networks are wireless networks where multi-hop links are used for communication of nodes with each other.

- Routing is challenging task in MANET because there is constant change in topology of network due to mobility of nodes. To accomplish this task various protocols have been developed.

### 5.26.1 Problems with Routing in Mobile Ad-hoc Networks (MANET) :

**1. Asymmetric links :**

- In ad-hoc networks, nodes are mobile and keeps changing their position in the network. e.g. in MANET, where node A send a signal to node B but this does not tell about quality of connection in the reverse direction.

**2. Routing overhead :**

- Due to change of location of node in the network, unwanted routes may be generated which leads routing overhead.

**3. Interference :**

- Main problem with MANET is interference. As links enter and exit depending on characteristics of transmission, one transmission may interfere other and node may overhear other transmission which can disturb the total transmission.

**4. Dynamic topology :**

- Since the topology of network is not constant, the mobile node may change location or medium characteristics may change.

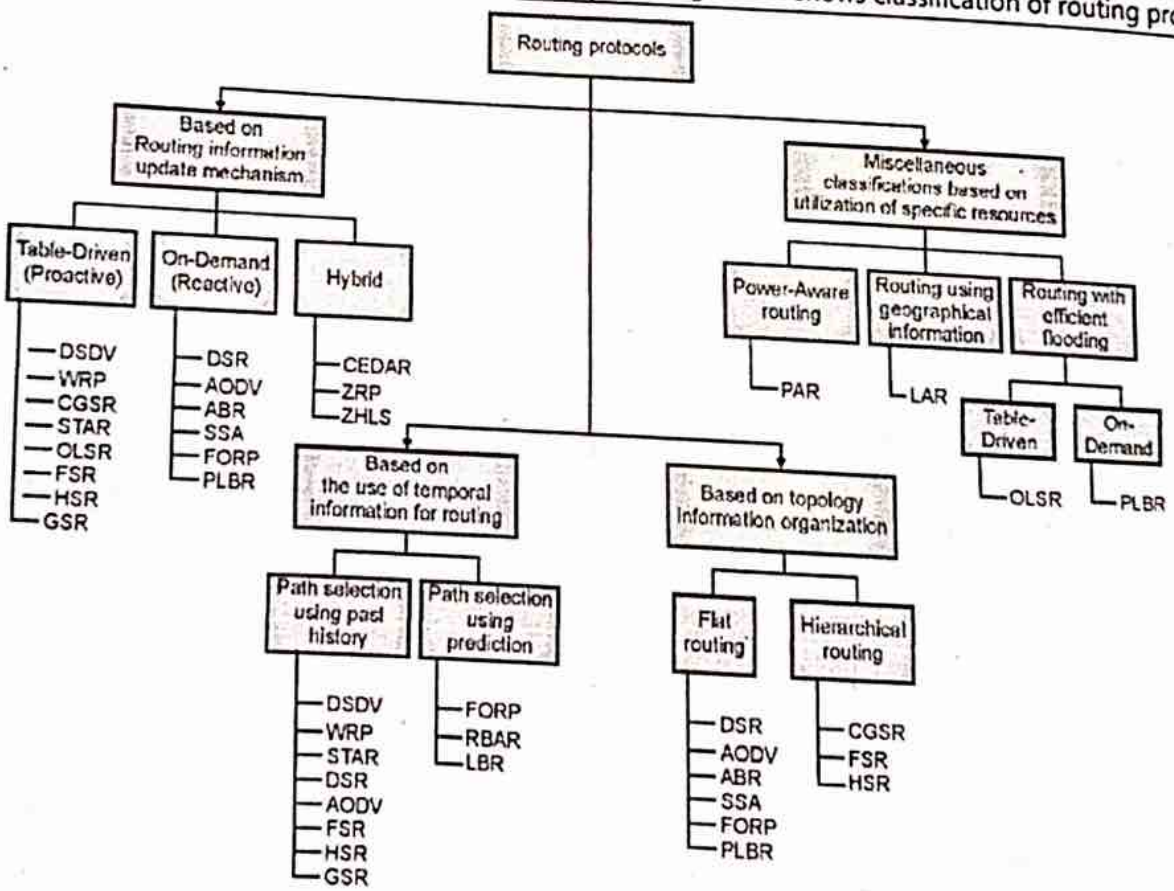
### 5.26.2 Characteristics of MANET Routing Protocol :

- To overcome the problems with routing in MANET routing protocols should have following characteristics :

1. It should be fully distributed.
2. It should be adjustable to frequent change i topology caused by the nodes mobility.
3. It must be localized.
4. It must be free from stagnant routes.
5. The convergence of routes must be quick.
6. Each node in the network should store informatio regarding local topology which is stable.
7. It should able to give good quality of service.

## 5.27 Classification of Routing Protocols i Adhoc Wireless Networks :

- Fig. 5.27.1 shows classification of routing protocols.




**(O-961) Fig. 5.27.1 : Classification of MANET routing protocols**

- Routing protocols for adhoc wireless network broadly classified into four categories based on :
  1. Routing information update mechanism.
  2. Use of temporal information for routing.
  3. Routing topology.
  4. Utilization of specific resources.

### 5.27.1 Based on the Routing Information Update Mechanism :

- Based on the routing information, adhoc wireless networks routing protocols can be classified into three groups.

**Pro-active or table-driven routing protocols :**

- In this protocol, every node maintains the network topology information. Information is maintained in the form of routing tables by constantly exchanging routing information.

**Reactive or on-demand routing protocols :**

- These protocols do not maintain topology information.
- By using a connection establishment process, these protocols obtain required path when required.

**Hybrid routing protocols :**

- These protocols combine the best features of above two categories.
- Table driven approach is used for routing within a particular geographical region. If nodes located beyond geographical zone, on demand approach is used.

### 5.27.2 Based on the use of Temporal Information for Routing :

- Use of temporal information is needed due to frequent path breaks and highly dynamic topology. These protocols are classified into two groups.

**Routing protocol using past temporal information :**

- Information of past status of the links is used in this protocol.

**Routing protocol that use future temporal information :**

- To make appropriate routing decisions, these protocols use information about expected future status of the wireless links.

### 5.27.3 Based on the Routing Topology :

- Adhoc wireless networks use either flat or hierarchical topology due to their relatively small number of nodes.

**Flat topology :**

- This topology uses presence of globally unique addressing method for nodes.

**Hierarchical topology :**

- This topology makes a use of logical hierarchy and an associated scheme of addressing. This topology is based on hop distance or geographical information.

### 5.27.4 Based on the Utilization of Specific Resources :

**Power-aware routing :**

- Aim of this routing protocol is to minimize important resources in adhoc network such as battery power consumption.

**Geographical information associated routing :**

- By utilizing the available geographical information, these protocols reduce the control overhead and improve the routing performance.

## 5.28 Table Driven (Proactive) Routing Protocols :

- These protocols are also known as proactive protocols because they maintain the routing information before it is required.
- They maintain topology information at each node in the form of table. To maintain consistent and accurate network state information these tables are frequently updated.
- Examples of table driven routing protocols are DSDV, WRP, HSR, GSR, FSR, FSLS etc. The proactive protocols are not suitable for large networks because they need to maintain table for each node.
- In the following subsection we will discuss proactive routing protocol : DSDV (Destination Sequenced Distance Vector Routing Protocol) and WRP (Wireless Routing Protocol).

### 5.28.1 Destination Sequenced Distance Vector Routing Protocol (DSDV) :

- DSDV is table routing protocol that is first protocol for adhoc wireless networks. DSDV is improved version of Bellman Ford algorithm where each node keeps a table, which contains shortest distance and the first node to every other node in the network on the shortest path.
- To prevent loops, to answer the count to infinity problem and faster convergence node includes table updates with increasing sequence number tags.

- As DSDV is table driven routing protocol at all times, routes to all destinations are readily available at every node.
- To maintain an upto date record of the network topology, the tables exchanged between neighbors at interval. If node finds major change in local topology, then tables are forwarded to neighboring nodes.
- There are two types of table updates :

1. Incremental updates. 2. Full dumps.

**1. Incremental updates :**

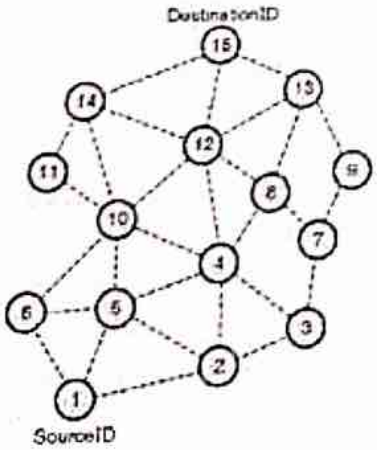
- An incremental update takes single Network Data Packet Unit (NDPU). When there is no significant change observed in topology, incremental updates are used.

**2. Full dumps :**

- Full dump takes multiple NDPU. A full dump is performed either when the local topology changes significantly or when an incremental update needs more than single NDPU.
- Destination nodes initiate the table update with new sequence number. Initiated sequence number is always greater than previous one.
- Once updated table is obtained, nodes update its tables with the help of either received data or hold it for some time to select the best metric. Metric is the smallest number of hops received from many kinds of the same update table from various neighboring nodes.
- A node may forward or decline the table based on the sequence number of the table update.

**Route establishment in DSDV :**

- Fig. 5.28.1 shows route establishment in DSDV.




| Dest | NextNode | Dist | SeqNo |
|---|---|---|---|
| 2 | 2 | 1 | 22 |
| 3 | 2 | 2 | 28 |
| 4 | 5 | 2 | 32 |
| 5 | 5 | 1 | 134 |
| 6 | 5 | 1 | 144 |
| 7 | 2 | 3 | 162 |
| 8 | 5 | 3 | 162 |
| 9 | 2 | 4 | 186 |
| 10 | 6 | 2 | 142 |
| 11 | 8 | 3 | 176 |
| 12 | 5 | 3 | 190 |
| 13 | 5 | 4 | 198 |
| 14 | 6 | 3 | 214 |
| 15 | 5 | 4 | 256 |

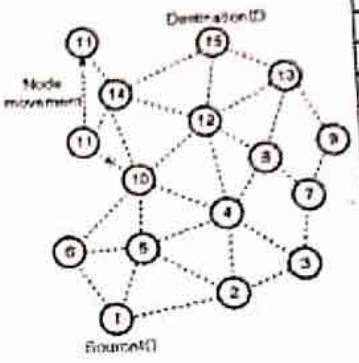
**(a) Topology graph of the network** **(b) Routing table for node 1**

**(G-1690) Fig. 5.28.1 : Route establishment in DSDV**

- In Fig. 5.28.1(a), node 1 is assumed as source node and node 15 is the destination node. The route already exists as shown in Fig. 5.28.1(b) because the nodes preserve universal topology data.
- The routing table of source node 1 shows that smallest route to the destination node 15 exists through node 5 and its minimum distance is 4 hops.

**Route maintenance in DSDV :**

- The reconfiguration of path used by on-going data transfer is handled by the protocol in the following way.




| Dest | NextNode | Dist | SeqNo |
|---|---|---|---|
| 2 | 2 | 1 | 22 |
| 3 | 2 | 2 | 28 |
| 4 | 5 | 2 | 32 |
| 5 | 5 | 1 | 134 |
| 6 | 5 | 1 | 144 |
| 7 | 2 | 3 | 162 |
| 8 | 5 | 3 | 162 |
| 9 | 2 | 4 | 186 |
| 10 | 6 | 2 | 142 |
| 11 | 5 | 4 | 186 |
| 12 | 5 | 3 | 190 |
| 13 | 5 | 4 | 198 |
| 14 | 6 | 3 | 214 |
| 15 | 5 | 4 | 256 |

**(a)** **(b)**

**(G-1691) Fig. 5.28.2 : Route maintenance in DSDV**

- The last node of broken link begin a table update message with the weight of broken link assigned to ∞ and with a sequence number larger than the registered sequence number for that destination node.
- Once, a node get an update table with weight as '∞', each node immediately circulate it to its adjacent nodes to broadcast the broken link data to the entire network.
- Hence, breaking of single link leads to the propagation of table update information to the entire network. A node allocates an odd sequence number to the link break record to distinguish it from the even sequence number generated by the destination node.
- Consider the case when node 11 moves from its current location, is as shown in Fig. 5.28.2(a).
- When an adjacent node observe the link break, it establishes all the paths passing through the broken link with distance as '∞'.

For example, when node 10 is aware of the link failure, it sets the path to node 11 as ∞ and transmits its routing table to its neighboring nodes.

- The neighboring nodes finding important changes in their routing tables retransmit it to their neighbors. In this way, broken link information spreads all over the network.
- Node 1 also establish the distance to node 11 as '∞'. When node 14 gets a table update message from node 11, it informs the neighbors about shortest distance to node 11. This information is circulated throughout the network.
- After receiving new update message with higher sequence number, all nodes saves the new distance to node 11 in their corresponding tables.
- Fig. 5.28.2(b) shows updated table at node 1, where the current distance from node 1 to node 11 is increased from 3 to 4 hops.

## 5.29 On-demand (Reactive) Routing Protocol :

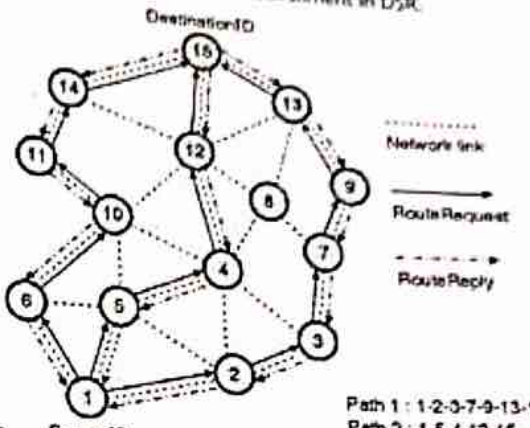
- On demand routing protocol is also called as reactive protocols.
- On demand routing protocol perform path finding procedure and exchange of routing information takes place only when path required by a node to make communication with a destination.
- If there is no communication between nodes they don't maintain routing information or activity hence these protocols are called as **reactive** protocols.
- If one node wants to send packet to other node this protocol finds the route in on-demand manner and it creates connection in order to transmit and receive the packet.
- In the following subsection we will discuss on-demand routing protocols : DSR and AODV.

### 5.29.1 Dynamic Source Routing Protocol (DSR) :

- Dynamic Source Routing Protocol (DSR) is an on demand routing protocol.
- It is to control the bandwidth consumed by control packets in adhoc networks by removing the periodic table update messages needed in the table driven method.
- As DSR protocol is beacon-less they do not need periodic packet transmissions. A node uses periodic beacon packet to inform its presence to neighboring node.

**Route establishment in DSR :**

- Fig. 5.29.1 shows route establishment in DSR.



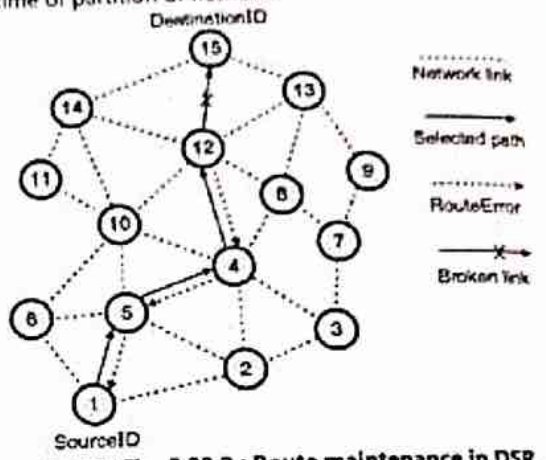

**(G-1692) Fig. 5.29.1 : Route establishment in DSR**

- During route construction phase, DSR establish a route by flooding RouteRequest packets over the network.
- After receiving RouteRequest packet, destination node responds by sending a RouteReply packet back to the source node carrying the route traversed by the RouteRequest packet received.
- Let a source node do not have a route to the destination.
- It initiates RouteRequest packet when source node have data packet to be sent to destination.
- This RouteRequest packet is flooded into the entire network.
- After receiving RouteRequest packet each node retransmits the RouteRequest packet to its adjacent node if it is not Every node sent already or if the node itself is not destination node provided Time to Live (TTL) counter of packet is not exceeded.

- Every RouteRequest packet contains sequence number generated by the source node and traversed path of it.
- Before forwarding RouteRequest packet node checks the sequence number of the packet.
- The packet is forwarded only if it is not duplicate RouteRequest packet.
- The sequence number on the packet is used to avoid loop formations and to prevent more transmission of the similar RouteRequest packets by intermediate node, which receives RouteRequest packet through many routes.
- Hence, during the route construction phase all nodes except destination node forward a RouteRequest packet.
- Upon receiving first RouteRequest packet, a destination node responds to the source node through the reverse path traversed by RouteRequest packet as shown in Fig. 5.29.1.
- As shown in Fig. 5.29.1, to obtain a path for destination node 15, source node 1 initiates RouteRequest packet.
- DSR protocol uses a route cache, which stores all possible data obtained from the source route obtained in data packet.
- Nodes can also study about the adjacent routes came across by data packets if nodes are operated in the promiscuous mode. (The mode in which node can receive the packet which are neither transmits nor addressed to itself).
- This route cache is also useful during the route construction phase.
- If RouteRequest packet is received by intermediate node, it has route to the destination node in its route cache then it respond to the source node by sending RouteReply with all route data from the source to the destination node.

**Optimizations :**

- In order to improve the performance of DSR protocol, many optimization techniques have been proposed. DSR protocol uses route cache at intermediate nodes.
- The route cache is settled with routes that can be removed from the information held in data packets that get forwarded.
- Intermediate nodes use this cache information to reply to the source node when they receive a RouteRequest packet.
- It also uses cache information if they find a route to the respective destination. An intermediate node discover about breaks in route when it operates in the promiscuous mode.
- Thus obtained information is useful to update the route cache so that the active routes kept in the route cache do not use such broken links.
- The affected node initiates RouteRequest packet at the time of partition of network.




**(G-1693) Fig. 5.29.2 : Route maintenance in DSR**

- An exponential back off algorithm is used to prevent Route Request flooding in the network when the destination node is in another disjoint set.
- Piggybacking of a data packet on the Route Request packet is used in DSR so that a data packet can be transmitted along with Route Request packet.
- In DSR, route construction phase becomes simple without optimization. If intermediate nodes are not redundant, they flood Route Request packet.

**Route maintenance in DSR :**

- As shown in Fig. 5.29.2, after getting the Route Request packet from node 1 all its adjacent nodes such as node 2, node 5 and node 6 forward Route Request packet. Node 4 gets Route Request packet from node 2 and node 5.
- Node 4 sends the first Route Request packet it gets from either node 2 or node 5 and rejects the other duplication or redundant Route Request packet.

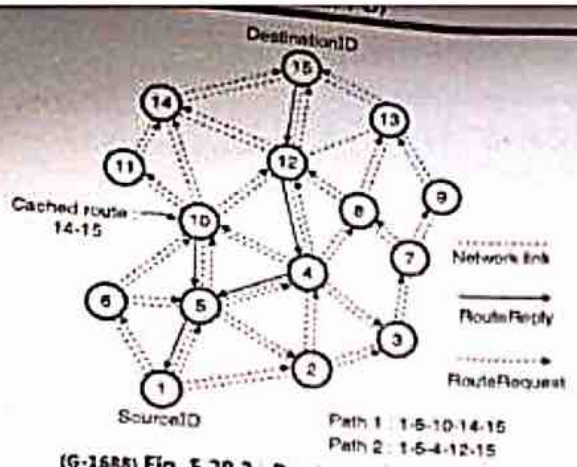
- the Route Request is circulated until it reaches at the destination node that is in the Route Reply.
- Source node may receive multiple replies if intermediate nodes are allowed to begin Route Reply packets.
- Suppose in Fig. 5.29.2 if the node 10 has a path to the destination node through node 14 it send the Route Reply to the source node.
- The source node chooses the recent and best route and selects that route for transmitting data packets. Each data packet carries the entire path to its destination node.
- When intermediate node is out of path causing link break (e.g. link between node 12 and 15). The neighboring node of failed link generates the Route Error message to inform the source node.
- The source node restarts the route establishment process. After receiving Route Error message, the cached entries at intermediate node and source node are removed.
- If a wireless link fails due to movement of node edges for example, node 1 and node 15, the source node again starts the route discovery process.

## 5.29.2 Adhoc on Demand Distance Vector Routing Protocol (AODV) :

- AODV routing protocol uses on demand method for discovering routes. i.e. route is created only when it is needed by source node for sending data packets.
- AODV protocol makes use of destination sequence number to identify the latest path.
- The main difference between AODV and Dynamic Source Routing (DSR) protocol is that DSR uses source routing in which a data packet contains the complete path to be traversed.
- In AODV protocol, the source nodes and intermediate nodes keep the next hop information related to each flow for data packet transmission.
- In on demand routing protocol the source node flood the RouteRequest packet in the network when there is no route available for the desired destination.
- From a single RouteRequest, it can obtain many routes of different destination.
- To obtain upto date path to the destination, AODV protocol uses destination sequence number (DestSeqNum).
- When the DestSeqNum of current packet is greater than the previous DestSeqNum stored at node, then a node updates its path information.
- A RouteRequest packet contains destination identifier (DestID), source identifier (SICID), source sequence number (SrcSeqNum) and destination sequence number (DestSeqNum), broadcast identifier (BcastID) and time to live (TTL) field. DestSeqNum shows that the route is accepted by source.
- When an intermediate node gets a RouteRequest packet, it either sends it or makes a RouteReply if it has valid routes to the destination.
- The validity of a route at the middle node is decided by comparing sequence number at middle node with the DestSeqNum in the RouteRequest packet.
- If a RouteRequest is received by many times, which is indicated by the broadcast identifier and source identifier, duplicate RouteRequest packets are rejected.
- All intermediate nodes having valid routes to the destination or destination node itself can allow to send RouteRequest packets to the source node.
- At the time of sending a RouteRequest packet, every intermediate node do entry of previous node address and its broadcast identifier (BcastID).
- A timer is used to delete this entry, if RouteReply is not received before time expires.
- As AODV does not employ source routing of data packet, this helps in storing an active path at the intermediate node.
- When a node receives a RouteReply packet, data about the previous node from which the packet was received is also stored in order to forward the data packet to next node.

**Route establishment in AODV :**

- Route establishment in AODV is as shown in Fig. 5.29.3.
- As shown in Fig. 5.29.3 source node 1 starts a path discovery process by initiating RouteRequest to the destination node 15 in the network.

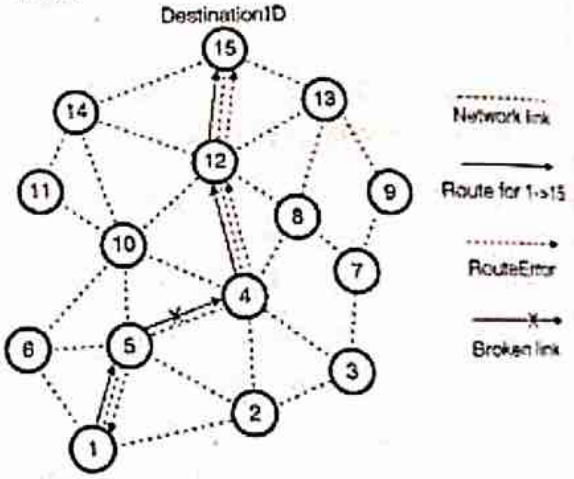



**(G-1688) Fig. 5.29.3 : Route establishment in AODV**

- As shown in Fig. 5.29.3 source node 1 starts a path discovery process by initiating RouteRequest to the destination node 15 in the network.
- It assumes the RouteRequest consists of the destination sequence number as 3 and source sequence number as 1.
- When nodes 2, 5 and 6 gets RouteRequest packet, these nodes verify their routes to the destination.
- In case the routes to node 15 do not exist, they send next RouteRequest packet to their neighbouring node.
- As shown in Fig. 5.29.3 node 3, node 4 and node 10 are the neighbours of node 2, node 5 and node 6 respectively.
- It is assumed that node 3 and node 10 have already existing routes to destination node 15. The route exists as 10-14-15 and 3-7-9-13-15 respectively.
- If the DestSeqNum at node 10 is 4 and at node 3 is 1, then only node 10 is permitted to reply along with the cached route to the source node 1.
- This is because intermediate node 3 has an oldest route to destination node 15 as compared to the route available at the source node 1.
- The DestSeqNum at node 3 is 1, whereas source node 1 has DestSeqNum 3.
- At the same time node 10 has recent route with DestSeqNum 4 to destination.
- If the RouteRequest reaches at destination node 15 through available path 4-12-15 or any other alternative route the destination node 15 sends a RouteReply to the source node 1.
- In such case multiple RouteReply packets reach the source node.
- All intermediate nodes getting RouteReply packet update their route tables with the recent DestSeqNum.
- They also update the routing information if it leads to smaller path between source and destination node.
- AODV do not repair a damaged path locally.

**Route maintenance in AODV :**

- When a wireless link breaks, it is determined by monitoring periodical beacons or through the notified link level acknowledgements at the end nodes (i.e source and destination nodes).
- Source node restarts the route to the destination node by the higher layers when it discovers the path break.
- If path break is found at intermediate node, the intermediate node update the source and destination nodes by sending unsolicited RouteReply with the value of hop count as '∞'.
- As shown in Fig. 5.29.4, when the path breaks between the nodes 4 and 5, both the nodes start RouteError messages to update their end nodes about the link break.




**(G-1689) Fig. 5.29.4 : Route maintenance in AODV**

- The end node removes the related entries from their tables.
- The source restarts the path finding process with new broadcast identifier and previous DestSeqNum.

**Advantages :**

1. In AODV protocol paths are established on demand and DestSeqNum are used to discover the recent route to destination.

2. The connection establishment delay is small.
3. As AODV is reactive in nature, it can handle highly dynamic behaviour of Ad-hoc networks.

**Disadvantages :**

1. Disadvantages of AODV protocol is that intermediate nodes can lead to conflicting routes if the source sequence number is not recent and intermediate nodes have greater but not the recent DestSeqNum, it results in having hard entries in table.
2. No reuse of routing information.
3. AODV does not discover a route until a flow is initiated.

## 5.30 Mobile IP :

- Mobile IP is the extension of IP protocol. It has been developed for the mobile and personal computers such as notebook.
- Mobile IP allows the mobile computers to get connected to the Internet at any location.

### 5.30.1 Addressing :

- Addressing is a very important problem in providing mobile communication using IP protocol. We will discuss its solution in this section.

#### 5.30.1.1 Addressing in Stationary Hosts :

- The original IP addressing was designed on the basis of two assumptions :
  1. The host is stationary.
  2. The host is connected to only one network.
- An IP datagram is routed by the routers on the basis of the IP address.
- As discussed earlier in this chapter, an IP address is made of two parts : a prefix and a suffix.
- A host gets associated with a network due to the prefix part of its IP address.
- That means a host cannot carry its IP address with itself from one place to the other.
- That means with change in place, the network changes and so does the IP address of the host.
- Routers use the fixed association between a host and its network for routing the packets to the network to which the host is attached.

#### 5.30.1.2 Mobile Hosts :

- The IP addressing structure needs to be changed when a host moves from one network to the other.
- To achieve this, various solutions have been suggested. Two of them are as follows :

1. **Changing the address :**
   - One of the solutions is to allow the mobile host to change its IP address as it changes the network.
   - This can be achieved by using DHCP. The mobile host can obtain a new IP address using DHCP and get associated with the new network.
   - But this technique has many drawbacks. Some of them are as follows :

**Drawbacks :**

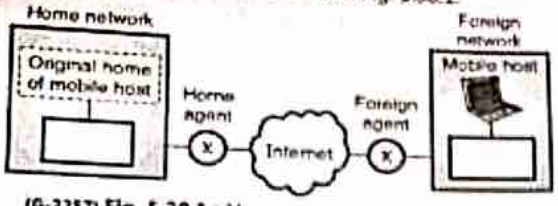
1. We need to change all the configuration files.
2. The mobile host would need rebooting, everytime it moves from one network to the other.
3. It would be necessary to revise the DNS table everytime so that all the other hosts on the Internet are aware of this address change.
4. If the mobile host moves from one network to the other when transmission is taking place, then the exchange of data will be interrupted because during the transmission, the client and server cannot change their port and IP addresses.

2. **Two addresses :**
   - Due to all the drawbacks of the first approach, the second approach of using **two IP addresses** for a mobile host is tried out and it is found to be a more feasible approach.
   - The two IP addresses assigned to a mobile host are :
     1. Home address and 2. Temporary address.
   - The **home address** is the original IP address of the mobile host, and the temporary address is called as the **care of address.**
   - The home address associated the host with its home network (i.e. the network which is permanent home of the host and it is its **permanent** IP address.
   - When the host moves to the other network, its temporary (care-of) address changes. This care-of address associates the host with the **foreign network.**

### 5.30.2 Agents :

- A **home agent** and a **foreign agent** are required for making the change of address transparent to the rest of Internet.

- The position of home agent with respect to the home network and that of the foreign agent with respect to the foreign network are shown in Fig. 5.30.1.




(G-2257) Fig. 5.30.1 : Home agent and foreign agent

- In Fig. 5.30.1 the home and foreign agents have been shown as routers. However actually they act as a router as well as a host.

**1. Home Agent :**

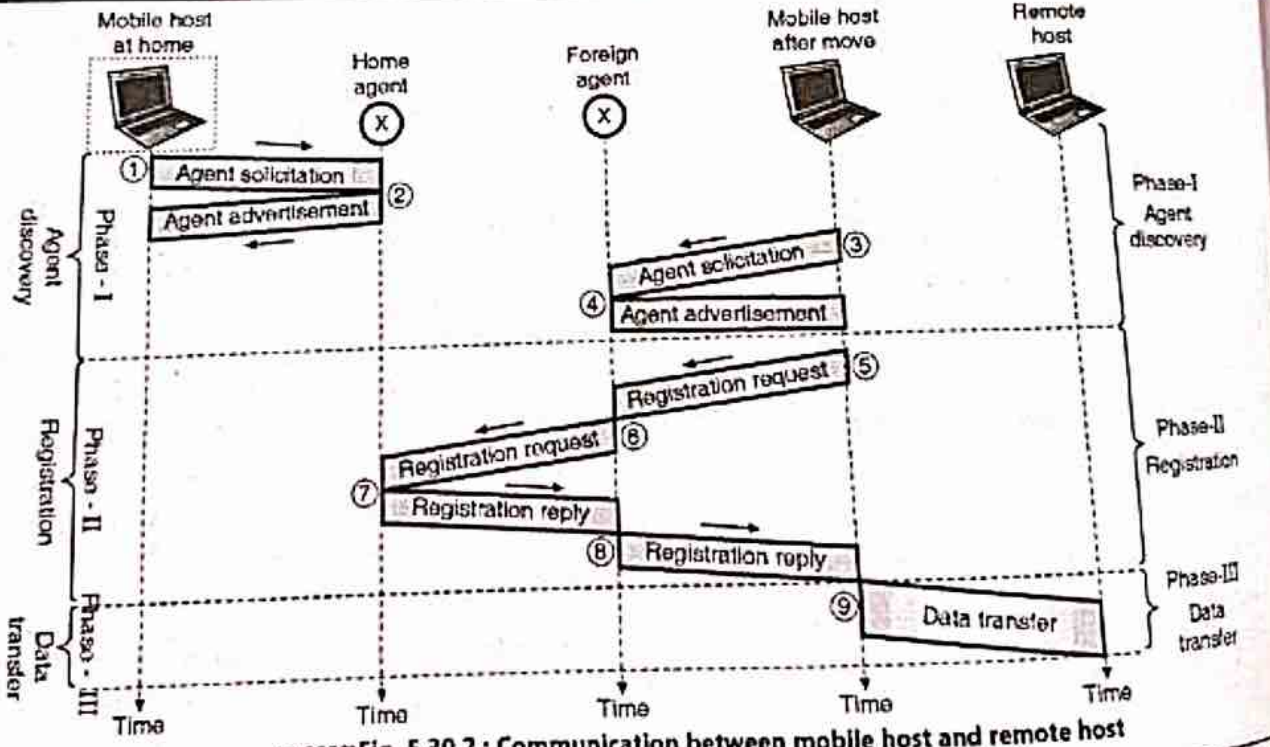
- A home router is basically a router attached to the home network of a mobile host.
- When a remote host sends a packet to the mobile host, the home agent acts on behalf of the mobile host, receives the packet and sends it to the foreign agent.

**2. Foreign Agent :**

- A foreign agent is a router connected to the foreign network. The packets sent by the home agent are received by the foreign agent and delivers them to the mobile host.
- Sometimes, a mobile host itself can act as foreign agent. Then there is no need of using a separate foreign agent.
- For this, the mobile host should have the ability to receive a care-of address on its own. This can be done using DHCP.
- In addition to this a special software needs to be installed at the mobile host to enable it to communicate with the home agent and to have the two addresses (home and temporary).
- It is necessary to keep the dual addressing transperent to the application programs.
- The care-of-address is called as **collocated care-of address** if a mobile host itself is acting as the foreign agent.
- The use of collocated care-of address has an advantage that the mobile host can move to any foreign network without even thinking about the availability of the foreign agent.
- However its disadvantage is that an extra software needs to be installed with the mobile host.

### 5.30.3 Three Phases :

- The communication of a mobile host with a remote host goes through the following three phases :
  1. Agent discovery
  2. Registration
  3. Data transfer.
- All these phases are shown in Fig. 5.30.2.




(G-2258) Fig. 5.30.2 : Communication between mobile host and remote host

1. **Phase-I : Agent Discovery (Steps 1 to 4) :**

- This is the first phase in mobile communication. It consists of the following two subphases :
  1. Agent solicitation and
  2. Agent advertisement.
- A mobile host must learn the address of (discover) its home agent before moving to any foreign network (Steps 1 and 2). The mobile host must also learn the address of (discover) the foreign agent once it moves to a foreign network (Steps 3 and 4).
- This process of address learning includes learning of both the care-of address and the foreign agents address.
- The agent discovery phase involves the discovery of home and foreign agents. This process requires the use of two messages namely :
  1. Advertisement message and
  2. Solicitation method.

2. **Phase-II : Registration (Steps 5 to 8) :**

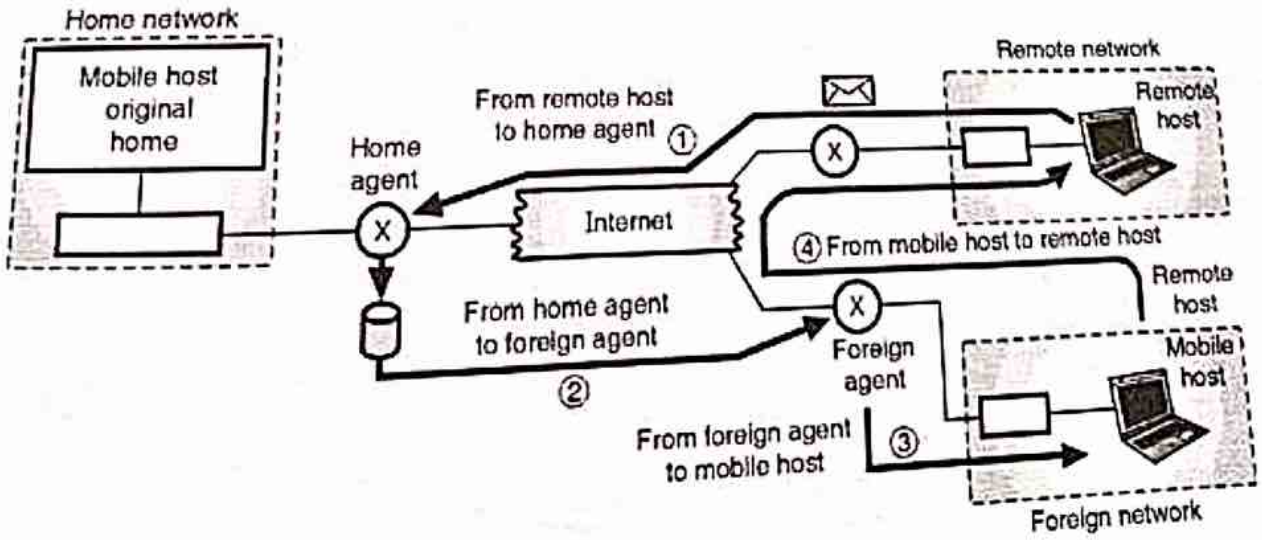
- This is the second phase of mobile communication. The mobile host first moves to the foreign network and discovers the foreign agent (Phase-I).
- After this it must undergo the registration phase, which corresponds to steps 5 to 8 in Fig. 5.30.2.
- The four aspects of registration are as follows :
  1. Registration of mobile host with the foreign agent (Step-5).
  2. Registration of mobile host with its home agent. This is normally done by the foreign agent on behalf of mobile host (Step 6).
  3. The mobile host must renew its registration if the registration has expired.
  4. The mobile host is supposed to cancel its registration when it returns back to its home network.
- The **registration request** and **registration reply** messages are used as shown in Fig. 5.30.2 for registration of mobile host with the home agent and foreign agent.

3. **Phase-III : Data Transfer :**

- This is the third phase in mobile communication after the agent discovery and registration. In this phase the mobile host can communicate with the remote host as shown in Fig. 5.30.2.

1. **From Remote Host to Home Agent :**

- If a packet is to be transferred from the remote host to mobile host, then the remote host uses its address as the **source address** and home address of mobile host as **destination address.**
- But practically the home agent is pretending as the mobile. So it will intercept the packet with the help of proxy ARP.
- Thus the communication from remote host to mobile host actually takes place between the remote host and home agent as shown in Fig. 5.30.2(a).




(G-2259) **Fig. 5.30.2(a) : Data transfer from remote host to home agent**

- The mobile communication between the Remote Host and Home agent has been marked by a thick path marked by "1" in Fig. 5.30.2(a).

2. **From Home to Foreign Agent :**

- As the packet is received by the home agent it sends the packet to the foreign agent using the concept of tunneling.
- The home agent encapsulates this received IP packet into a new IP packet by using its own address as the source address and foreign agents address as the destination address and sends this new IP packet to the foreign agent as shown by the thick path marked by "2" in Fig. 5.30.2(a).

3. **From Foreign Agent to Mobile Host :**

- From the IP packet received the foreign agent will recover the original packet by decapsulation process.
- However the recovered original packet has the home address of mobile host as its destination address.
- The foreign agent will refer to a registry table and finds the **care-of-address** of the mobile host. The original packet is then sent to the care-of-address as shown by the thick path marked by "3" in Fig. 5.30.2(a).

4. **From Mobile Host to Remote Host :**

- If a mobile host wants to send a packet to a remote host, it does it in a normal way.
- To do this, the mobile host creates a packet with its home address (and not the care-of-address) as source address and remote host's address as the destination address.
- It is very important to note that eventhough the packet originates from the foreign network, it has the home address of the mobile host.
- This communication has been shown by the thick path "4" in Fig. 5.30.2(a).

### 5.30.4 Transparency :

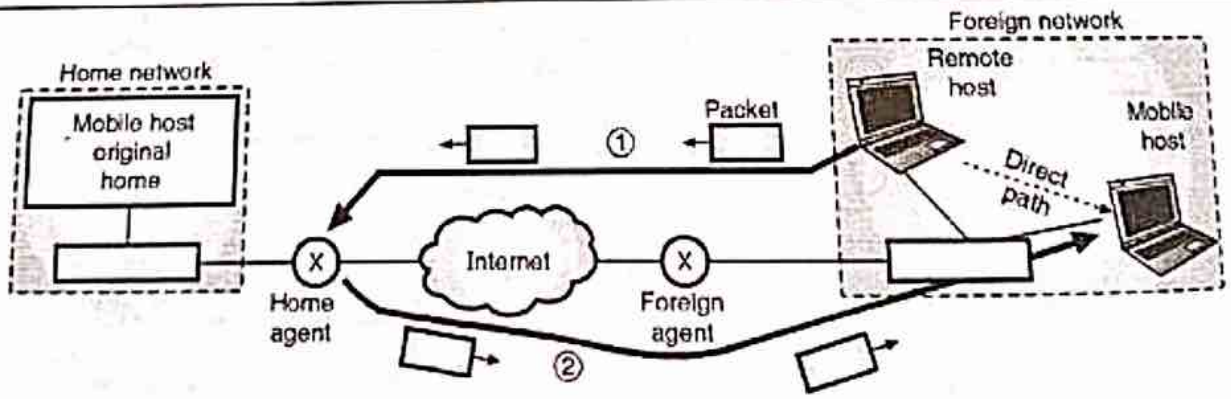
- In the entire data transfer process, the remote host absolutely does not know anything about the movement of the mobile host.
- Because, the remote host uses the home address as the destination address when sending a packet to the mobile host.
- Similarly the mobile host uses its home address as the source address while sending a packet to the remote host.
- In other words we say that movement of the mobile host is **totally transparent** because the rest of the Internet has absolutely no idea about the movement of the mobile host.

### 5.30.5 Inefficiency in Mobile IP :

- The communication done with the help of mobile IP can be moderately to severely inefficient.
- The case of moderate inefficiency is also called as the **triangle routing, or dog leg routing** whereas the case of severe inefficiency is also referred to as **Double Crossing** or **2X.**
- We will discuss both these cases in this section.

1. **Double Crossing or 2X :**

- Now consider a situation in which a remote host wants to communicate with a mobile host which has moved to the same network as that of the remote host as shown in Fig. 5.30.3.

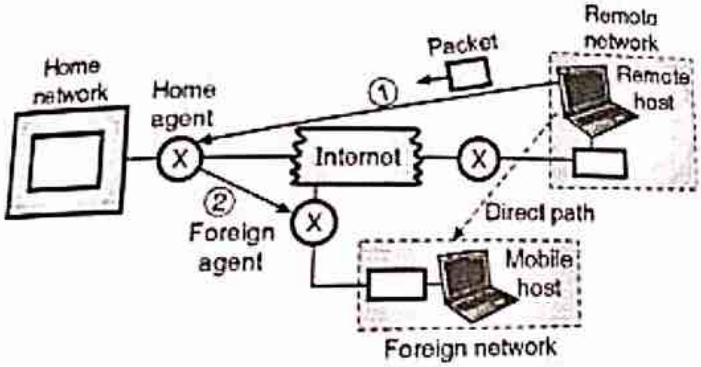



(G-2260)**Fig. 5.30.3 : Double crossing**

- This is called as a **double crossing or 2X** case i.e. the case of **severe inefficiency.**
- As discussed earlier, a mobile host can send a packet directly to the remote host. Therefore there is no loss of efficiency in this communication.
- However if the remote host wants to send a packet to the mobile host then it cannot do so directly (via the dotted direct path in Fig. 5.30.3).
- Instead the remote host has to send the packet first home agent (path-1 in Fig. 5.30.3) and the home agent will route the packet to the mobile host (path-2 in Fig 5.30.3).
- In this the packet has to cross the Internet twice. Thus the resources are used twice unnecessarily in this communication which reduces the efficiency severely.
- Hence the double cross case is called as the case of severe inefficiency.

**2. Triangle Routing or Dog Leg Routing :**

- A triangle routing or dog leg routing is a case of moderate inefficieny.
- It occurs when a remote host wants to send a packet to the mobile host which is not attached to its own (remote) network.
- This situation is illustrated in Fig. 5.30.4.




**(G-2261)Fig. 5.30.4 : Triangle routing**

- In this situation as well, if a mobile host wants to send a packet to a remote host it can do so directly without any loss of efficiency.
- But when a remote host wants to send a packet to a mobile host the packet has to first travel to the home agent and then to the mobile host as shown in Fig. 5.30.4.
- Thus the packet has to travel along two sides of a triangle instead of only one which is the direct path shown by a dotted line in Fig. 5.30.4.

### 5.30.6 Remedy :

- Binding the care-of address to the home address of mobile could be one of the solutions to the problem of inefficiency.
- That means when the home agent receives the first packet from the remote host and sends it to the foreign agent it should also send an **update binding packet** to the remote host.
- By doing this it is ensured that all the future packets to this mobile host can be sent to the **care-of-address** rather than **home** address.
- The remote host can save this information in a cache.
- However this remedy also has an inherent flaw. The **cache entry** would become outdated as the mobile host moves to a new network.
- To avoid this the **home agent** must send a **warning packet** to the remote host to inform that the mobile host has moved to a new network.

## Review Questions

Q. 1 Name different protocols in the network layer.

Q. 2 Write a note on IP.

Q. 3 Explain fragmentation in IP.

Q. 4 What is the name of a packet in IP ?

Q. 5 Explain the IP header.

Q. 6 What is MTU and how is fragmentation related to it ?

Q. 7 Compare IPv4 and IPv6.

Q. 8 State limitations of IPv4.

Q. 9 Write a note on mobile IP.

Q. 10 What is fragmentation ? Explain how is it supported in IPv4 and IPv6.

Unit IV

Chapter 6

# Transport Layer

## Syllabus

Process to process delivery, Services, Socket programming. **Elements of Transport Layer Protocols :** Addressing, Connection establishment, Connection release, Flow control and buffering, Multiplexing, Congestion control. **Transport Layer Protocols :** TCP and UDP, SCTP, RTP, Congestion control and Quality of Service (QoS), Differentiated services, TCP and UDP for Wireless networks.

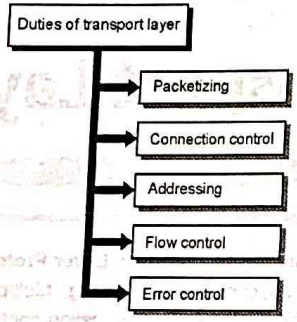
## Chapter Contents

## 6.1 Introduction :

- The transport layer is the core of the Internet model. The application layer programs interact with each other using the services of the transport layer.
- Transport layer provides services to the application layer and takes services from the network layer.
- The transport layer is fourth layer in the internet model.
- It connects the lower three layers to upper three layers of an OSI layer.

## 6.2 Transport Layer Duties :

- Transport layer is meant for the process to process delivery and it is achieved by performing a number of functions.
- Fig. 6.2.1 lists the functions of a transport layer.




(G-1407) **Fig. 6.2.1 : Duties of transport layer**

**1. Packetizing :**

- The transport layer creates packets with the help of encapsulation on the messages received from the application layer.
- Packetizing is a process of dividing a long message into smaller ones.
- These packets are then encapsulated into the data field of the transport layer packet.
- The headers containing source and destination address are then added.
- The length of the message which is to be divided can vary from several lines (e-mail) to several pages.
- But the size of the message can become a problem. The message size can be larger than the maximum size that can be handled by the lower layer protocols.
- Hence the messages must be divided into smaller sections. Each small section is then encapsulated into a separate packet. Then a header is added to each packet to allow the transport layer to perform its other functions.

**2. Connection control :**

- Transport layer protocols are divided into two categories :

  1. Connection oriented. 2. Connectionless.

**Connection oriented delivery :**

- A connection oriented transport layer protocol establishes a connection i.e. virtual path between sender and receiver. This is a virtual connection. The packet may travel out of order.
- The packets are numbered consecutively and communication is bi directional.

**Connectionless delivery :**

- A connectionless transport protocol will treat each packet independently. There is no connection between them. Each packet can take its own different route.

**3. Addressing :**

- The client needs the address of the remote computer it wants to communicate with. Such a remote computer has a unique address so that it can be distinguished from all the other computers.

**4. Flow and error control :**

- For high reliability the flow control and error control should be incorporated.
- **Flow control :** We know that data link layer can provide the flow control. Similarly transport layer also can provide flow control. But this flow control is performed end to end and not across a single link.
- **Error control :** The transport layer can provide error control as well.
- But error control at transport layer is performed end to end and not across a single link.
- Error correction is generally achieved by retransmission of the packets discarded due to errors.

**Congestion control and QoS :**

- The congestion can take place in the data link, network or transport layer.
- But the effect of congestion is generally evident in the transport layer.
- Quality of Service (QoS) can be implemented in other layers but its actual effect is felt in the transport layer.





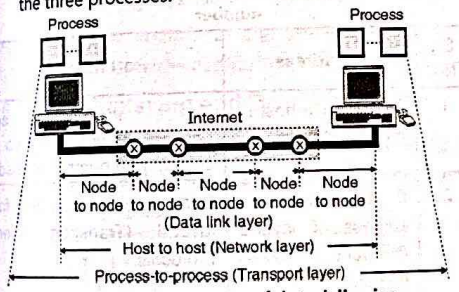
- The transport layer enhances the QoS provided by the network layer.

## 6.3 Transport Layer Services :

- In this section we are going to discuss the services provided by the transport layer.

### 6.3.1 Process-to-Process Communication :

- The data link layer performs a node to node delivery.
- The network layer carries out the datagram delivery between two hosts (host to host delivery).
- But the real communication takes place between two processes or application programs for which we need the **process-to-process delivery.**
- The transport layer takes care of the **process-to-process delivery.** In this a packet from one process is delivered to the other process.
- The relationship between the communicating processes is the client-server relationship. Fig. 6.3.1 demonstrates the three processes.




(G-594) **Fig. 6.3.1 : Types of data deliveries**

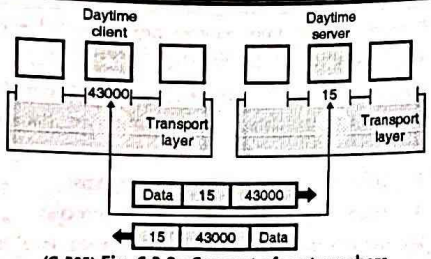
- There is a difference between host-to-host communication and process to process communication that we need to understand clearly. The host to host (computer to computer) communication is handled by the network layer.
- But this communication only ensures that the message is delivered to the destination computer. But this is not enough.
- It is necessary to handover this message to the correct process. The transport layer will take care of this.

### 6.3.2 Addressing : Port Number :

- There are several ways of achieving the process-to-process communication, but the most common method is using the client-server paradigm.
- **Client** is defined as the process on the local host.
- It needs services from another process called **server** which is on the other (remote) host.
- Both client and server have the same name. Some of the important terms related to the client-server paradigm are :

  1. Local host 2. Remote host
  3. Local process 4. Remote process

- We can use the IP addresses to define the local host and remote host.
- But this is not enough to define a process. In order to define a process, we have to use one more identifier called **Port Numbers.**
- In TCP/protocol suite, the port numbers are integers and they are numbered between 0 and 65,535. At the data link layer we need a MAC address, at the network layer we need to use an IP address.
- A datagram uses the destination IP address to deliver the datagram and uses the source IP address for the destination's reply.
- At the transport layer a transport layer address called a **port number** is required to be used to choose among multiple processes running on the destination host.
- The destination port number is required to make the packet delivery and the source port number is needed to return back the reply. In the Internet model, the port numbers are 16 bit integers.
- Hence the number of possible port numbers will be $2^{16}$ = 65,535 and the port numbers range from 0 to 65,535. The client program identifies itself with a port number which is chosen randomly.
- This number is called as **ephemeral port number.** Ephemeral means short lived. It is used because life of a client is generally short.
- The server process should also identify itself with a port number but this port number cannot be chosen randomly.
- The Internet uses universal port numbers for servers and these numbers are called as **well known port numbers.**
- Every client process knows the well known port numbers of the pre identified server process.
- For example, a Day time client process can use an ephemeral (temporary) port number 43000 for identifying itself, the Day time server process must use the well known (permanent) port number 15.
- This is illustrated in Fig. 6.3.2.




(G-595) **Fig. 6.3.2 : Concept of port numbers**

**Difference between IP addresses and port numbers :**

- The IP addresses and port numbers have altogether different roles in selecting the final destination of data.
- The destination IP address is used for defining a particular host among the millions of hosts in the world. After a particular host is selected, the port number is used for identifying one of the processes on this selected host.

**IANA Ranges :**

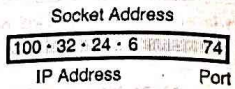
- The port numbers are divided into three ranges by IANA (International Assigned Number Authority).
- The ranges are as follows :
  1. Well known ports
  2. Registered ports
  3. Dynamic or private ports.

1. **Well known ports :** The ports from 0 to 1023 are known as well known ports. They are assigned as well as controlled by IANA.
2. **Registered ports :** The ports from 1024 to 49,151 are neither controlled nor assigned by IANA. We can only register them with IANA to avoid duplication.
3. **Dynamic or private ports :** The ports from 49,152 to 63,535 are known as dynamic ports and they are neither controlled nor registered.

- They can be used by any process. Dynamic ports are also known as private ports and dynamic port are called as ephemeral ports.

**Socket address :**

- Process to process delivery (transport layer communication) has to use two addresses, one is IP address and the other is port number at each end to make a connection.
- Hence a process to process delivery uses the combination of these two. The combination of IP address and port number is as shown in Fig. 6.3.3 and it is known as the socket address.




(G-1548) **Fig. 6.3.3 : Socket address**

- The client socket address defines the client process uniquely whereas the server socket address defines the server process uniquely.
- A transport layer protocol requires the client socket address as well as the server socket address. These two addresses contain four pieces. These four pieces go into the IP header and the transport layer protocol header.
- The IP header contains the IP addresses while the UDP and TCP headers contain the port numbers.
- If we want to use the transport layer services in the Internet, then we have to use a pair of socket addresses namely the clients socket address and the server's socket address.

**Difference between IP address and port number :**

**Table 6.3.1 : Difference between IP address and port number**

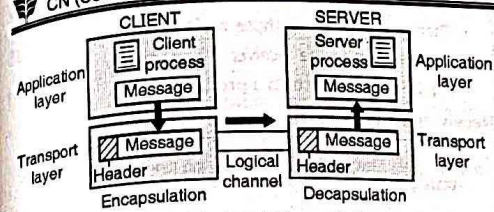
| Sr. No. | IP address | Port number |
|---|---|---|
| 1. | To identify a host IP address is used. | To identify an application/services on your system port number is used. |
| 2. | IP address is the address of layer-3 (Network layer) protocol. | Port number is used for layer-4 (Transport layer) protocols. |
| 3. | IP address is used by admin user of system. | Port number is provided by OS. This port number is called as port address. |
| 4. | IP address is used to send datagram traffic access network from source to destination machine. | After packet delivery to destination, with the help of the port numbers OS sends the data to correct application. |
| 5. | Eg. 191.168.0.1, 172.14.0.1 etc. | e.g. Port number 80, 68 etc. |

## 6.3.3 Encapsulation and Decapsulation :

- The transport layer carries out the **Encapsulation** of the message at the sending end and then **Decapsulation** at the receiving end when two computers communicate.
- This process has been illustrated in Fig. 6.3.4.








(G-2012) **Fig. 6.3.4 : Encapsulation and decapsulation**

**Encapsulation :**

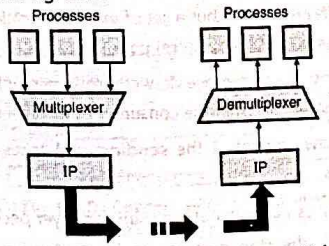
- At the sending end the process that has a message to send, will pass it to the transport layer alongwith a pair of socket addresses and some additional information.
- The transport layer adds its own header to this data. This packet at the transport layer in the Internet is known by different names such as **user datagram, segment or packet.**

**Decapsulation :**

- When the segment or datagram arrives at the receiving end, the header is isolated and destroyed, and the message is delivered to the process running at the application layer as shown in Fig. 6.3.4.
- The socket address of the sender process is then handed over to the destination process.

## 6.3.4 Multiplexing and Demultiplexing :

- The addressing mechanism allows multiplexing and demultiplexing taking place at the transport layer as shown in Fig. 6.3.5.




(G-597) **Fig. 6.3.5 : Multiplexing and demultiplexing**

**Multiplexing :**

- At the sending end, there are several processes that are interested in sending packets. But there is only one transport layer protocol (UDP or TCP).
- Thus it is a many processes-one transport layer protocol situation. Such a many-to-one relationship requires multiplexing.
- The protocol first accepts messages from different processes. These messages are separated from each other by their port numbers. Each process has a unique port number assigned to it.
- Then the transport layer adds header and passes the packet to the network layer as shown in Fig. 6.3.5.

**Demultiplexing :**

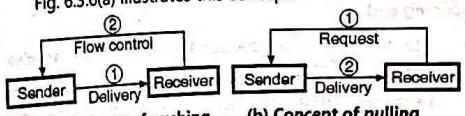
- At the receiving end, the relationship is one as to many. So we need a demultiplexer.
- First the transport layer receives datagrams from the network layer.
- The transport layer then checks for errors and drops the header to obtain the messages and delivers them to appropriate process based on the port number.

## 6.3.5 Flow Control :

- If the packets produced by the sender are at a rate X and the receiver is receiving them at a rate Y, then for X = Y, there will be a perfect balance observed in the system.
- But if X is higher than Y (source is producing packets at a rate which is higher than the rate at which the receiver is accepting them), then the receiver can be overwhelmed and has to **discard** some packets.
- And if X is less than Y (i.e. source is producing packets at slower rate than the rate of acceptance at the receiver) then system becomes **less efficient.**
- Flow control is related to the situation in which X > Y because it is very important to prevent data loss (due to discarding of packets) at the receiver site.
- There are two different ways of delivering the packets produced by the sender to the receiver. They are pushing or pulling.

1. **Pushing :**

- If the sender is sending the packets soon as they are produced, without receiving any prior request from the receiver then this type of deliver is called as **pushing.** Fig. 6.3.6(a) illustrates this concept.




**(a) Concept of pushing** **(b) Concept of pulling**

(G-2013) **Fig. 6.3.6**

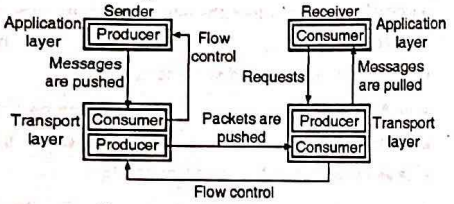
2. **Pulling :**

- If the sender sends the produced packets only when they are requested by the receiver then the delivery is called as **pulling.** Fig. 6.3.6(b) illustrates the principle of pulling.

- In case of **pushing** type delivery, if the packets are being sent at a higher rate than that of receiving, then the receiver will be **overwhelmed,** and some received packets will have to be discarded.
- In order to avoid discarding of packets, the **flow control** will have to be exercised.
- For this the receiver has to warn the sender to stop the delivery when it is overwhelmed and it has to inform the sender again to start delivery when it (receiver) is ready, to receive the packets.
- In case of **pulling type delivery,** the receiver is actually pulling the packets from the sender.
- It requests for the packets when it is ready. Therefore the flow control is not required in this case.

## 6.3.6 Flow Control at Transport Layer :

- The concept of flow control at transport layer has been illustrated in Fig. 6.3.7.




**(G-2014) Fig. 6.3.7 : Flow control at transport layer**

- It shows the communication taking place between a sender and a receiver.
- As shown in Fig. 6.3.7, there are four entities involved in this communication. They are as follows :
  1. Sender process.
  2. Sender transport layer.
  3. Receiver process.
  4. Receiver transport layer.
- We will discuss the flow control by considering the sending and receiving ends separately.

**Sending end :**

- The first entity on the sending end is the **sender process,** at the application layer. It works only as a **producer** which produces chunk of messages and pushes them to the transport layer on the sending end, as shown in Fig. 6.3.7.
- The second entity on the sending end is the **sender transport layer.** It has two different roles to play.
- First it acts as a **customer** and consumes all the messages produced and pushed by the producer.
- Then it encapsulates those messages into packets and pushes them to the receiver transport layer as shown in Fig. 6.3.7. Here it acts as a **producer.**

**Receiving end :**

- The first entity on the receiving end is the **receiver transport layer.** It also has two different roles to play.
- It acts as a **consumer** for the packets pushed by the senders transport layer and it also acts as the **producer.**
- It has decapsulate the messages and deliver them to the application layer as shown in Fig. 6.3.7.
- However the delivery of decapsulated messages to the application layer is a **pulling type delivery.**
- That means the transport layer waits till the application layer process requests for the decapsulated messages.

**Flow control :**

- As shown in Fig. 6.3.7, the flow control is needed for atleast two cases.
- First is from transport layer of sender to the application layer of sender.
- And secondly form the transport layer of receiver to the transport layer of sender.

**Buffers :**

- It is possible to implement the flow control in many different ways. One of the ways of implementation is to use two **buffers** one each at the sending and receiving transport layers.
- A **buffer** is nothing but a set of memory locations which can temporarily hold (store) packets.
- It is possible to exercise flow control communication by sending signals from the consumer to producer.
- The **flow control** at the **sending end** takes place as follows :
- As soon as the buffer at the transport layer becomes full it sends the stop message to its application layer in order to stop the chunk of messages that are being pushed into the buffer.
- The second flow control takes place at the receiver transport layer as follows :
- As soon as the buffer at receiver transport layer becomes full, it will inform the sender transport layer to stop pushing the packets.
- Whenever the buffer becomes partially empty, it again informs the sender transport layer to start sending the packets again.





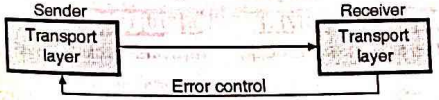
## 6.3.7 Error Control :

**Need of error control :**

- In the Internet, the network layer protocol IP has the responsibility to carry the packets from the transport layer at the sending end to the transport layer at the receiving end.
- But IP is unreliable. Therefore transport layer should be made reliable, in order to ensure reliability at the application layer.
- We can make the transport layer reliable by adding the **error control service** to the transport layer.

**Duties of error control mechanism :**

- Following are the important responsibilities of the error control mechanism introduced at the transport layer :
  1. To find and discard the corrupted packets.
  2. To keep the track of lost and discarded packets and to resend them.
  3. Identify the duplicate packets and discard them.
  4. To buffer out of order packets until the missing packets arrive.
- In the error control process, only the sending and receiving transport layers are involved.
- That means it is assumed that the chunk of messages exchanged between the application layers and transport layers are error free.
- The concept of error control at the transport layer level is demonstrated in Fig. 6.3.8.




**(G-2015) Fig. 6.3.8 : Concept of error control at the transport layer**

- The receiving transport layer manages the error control by communicating with the sending transport layer about the problem.

**Sequence numbers :**

- In order to exercise the error control at the transport layer following two requirements should be satisfied :
  1. The sending transport layer should know about the packet which is to be resent.
  2. The receiving transport layer should know about the packets which are duplicate or the ones that have arrived out of order.
- The requirements can be satisfied only if each packet has a unique **sequence number.**
- If a packet is either corrupted or lost the receiving transport layer will somehow inform the sending transport layer about the sequence number of those packets and request it to resend those packets.
- Due to the unique sequence number assigned to each packet it is possible for the receiving transport layer to identify the duplicate packets received.
- The out of order packets can also be recognized by observing gaps in the sequence numbers of the received packets.
- Packet numbers are given sequentially. But the length of the sequence number cannot be too long because the sequence number is to be included in the header of the packets.
- If the header of a packet allows "m" bits per sequence number, then the range of sequence number will be from 0 to $2^m$–1.
- For example if m = 3 then the range of sequence numbers will be from 0 to 7.
- Thus sequence numbers are modulo $2^m$.

**Acknowledgement :**

- The receiver side can send an acknowledgement (ACK) signal corresponding to each packet or each group of packets which arrived safe and sound.
- The question is what happens if a received packet is corrupted ? The answer is that the receiver simply discards the corrupted packet and does not send any ACK signal for it.
- The sender can detect a lost packet with the help of a timer. A timer is started at the sending end as soon as a packet is sent.
- If the ACK does not arrive before the expiry of the timer, then the sender treats the packet to be either lost or corrupted and resends it.
- The receiver silently discards the duplicate packets. It will either discard the out of order packets or stored until the missing packet is received.
- Note that every discarded packet is treated as a lost packet by the sender.

## 6.3.8 Combination of Flow and Error Control :

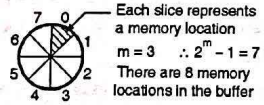
- Till now we have discussed the following important concepts :

1. We need to use buffers at the sending and receiving ends for exercising the flow control.
2. Also we have to use the sequence numbers and acknowledgements for exercising the error control.

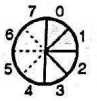
- We can combine these two concepts together by using two numbered buffers one at the sender and the other at the receiver, in order to exercise a combination of flow and error control.
- At the sending end, when a packet, is prepared to be sent, the number of the next free location (x) in the buffer is used as the sequence number of that packet.
- As soon as the packet is sent, its copy is stored at location (x) in the sending end buffer and the sender waits for the acknowledgement from the receiver.
- On reception of the acknowledgement of the sent packet, the copy of that packet is purged to make the memory location (x) free again.
- At the receiver, when a packet having a sequence number "y" arrives, it is stored at the memory location "y" in the receiver buffer until the receiver application layer is ready to receive it.
- The receiver will send the ACK message back to sender to inform it that packet "y" has arrived.

**Sliding window :**

- As the sequence numbers are modulo $2^m$, we can use a circle as shown in Fig. 6.3.9 to represent the sequence number from 0 to $2^m-1$.

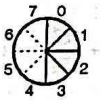



**(a) Sliding window in the circular format**

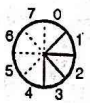


**(b) Two packets have been sent**

**(c) Three packets have been sent**



**(d) Four packets have been sent. The window is full**



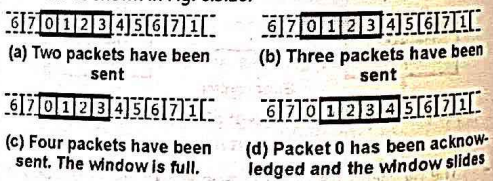
**(e) Packet 0 has been acknowledged and the window slides**

**(G-2017) Fig. 6.3.9**

- We can represent the buffer as a set of slices, called as the **sliding window** which will occupy a part of the circle at any time.
- In Fig. 6.3.9, we have assumed that m = 3. Therefore $2^m - 1 = 7$ and the sequence numbers are from 0 to 7.
- Hence the number of memory locations in a buffer will also be 8 i.e. 0 to 7.
- The sliding windows will correspond to the sender as well as receiver. On the sending side, when a packet is sent we will mark the corresponding slice.
- Therefore when marking of all the slices is done, it means the **sending buffer is full,** and it cannot accept any further messages from the application layer as shown in Fig. 6.3.9(d).
- When the acknowledgement for segment "0" arrives at the sending end, the corresponding segment (segment 0 ) is unmarked and window slides ahead by one slice as shown in Fig. 6.3.9(e). The size of the **sending window** is 4.
- Note that the sliding window is just an abstraction. In actual practice, computer variables are used to hold the sequence number of the next packet to be sent and the last packet sent.

**Sliding window in the linear format :**

- This is another way to diagrammatically represent a sliding window.
- It is as shown in Fig. 6.3.10.




(a) Two packets have been sent
(b) Three packets have been sent
(c) Four packets have been sent. The window is full.
(d) Packet 0 has been acknowledged and the window slides

**(G-2075) Fig. 6.3.10 : Sliding windows presented in the linear format**

- The principle of this type of sliding window is same as that of the circular representation.
- The linear format is the most preferred format. It needs less space on paper.
- Fig. 6.3.10(a), (b), (c) and (d) are the sliding windows presented in the linear format corresponding to Figs. 6.3.9(b), (c), (d) and (e) respectively in the circular presentation.



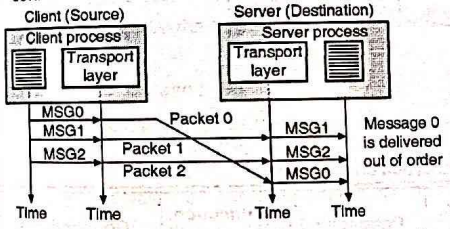


## 6.3.9 Connectionless and Connection Oriented Services (CLTS & COTS) :

- A transport layer protocol is capable of providing two types of services :
  1. Connectionless services.
  2. Connection oriented services.
- The meaning of the words connectionless and connection oriented is different at the transport layer than that at the network layer.
- A connectionless service at the network layer means different datagrams of the same message following different paths.
- However at the transport layer, the meaning of connectionless service is independency between different packets. On the other hand a connection oriented service means the packets are interdependent.

**Connectionless Transport Service (CLTS) :**

- Refer Fig. 6.3.11 to understand the concept of connectionless service.




**(G-2018) Fig. 6.3.11 : Concept of connectionless service**

- The source process at the application layer first divides its message in chunks of data the size of which is acceptable to the transport layer.
- These data chunks are then delivered to the transport layer one by one. These chunks are treated as independent units by the transport layer.
- Every data chunk arriving from the application layer is encapsulated in a packet by the transport layer and sent to the destination transport layer as shown in Fig. 6.3.11.

**Out of order delivery :**

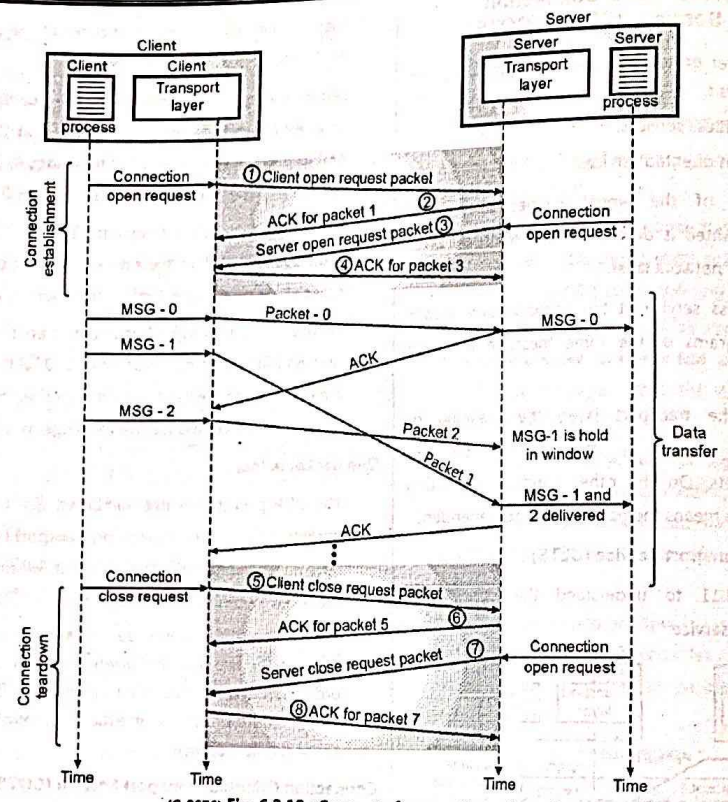
- In Fig. 6.3.11 we have considered three chunks of independent messages 0, 1 and 2.
- As the corresponding packets also are independent of each other and as they are free to follow their own path, these packets can arrive out of order at the destination as shown in Fig. 6.3.11.
- Naturally they are delivered to server process in an out of order manner. As seen in Fig. 6.3.11, at the sending end (client) the three chunks of messages 0, 1 and 2 are delivered to the transport layer in the order 0, 1, 2.
- But packet 0 travels a longer path and undergoes an extra delay. Therefore the packets are not delivered in order at the destination (server) transport layer.
- Therefore the message chunks delivered to the server process will also be out of order (1, 2, 0). If these chunks are of the same message then due to their out of order delivery the server will receive a strange message.

**One packet is lost :**

- The UDP packets are not numbered. So if one of the packets is lost, then the receiving transport layer will not have any idea about the lost packet. It will simply deliver the received chunks of messages to the server process.
- The above problems arise due to **lack of coordination** between the two transport layers. Due to this lack of co-ordination it is not possible to implement flow control, error control or congestion control in the connectionless service.

**Connection Oriented Transport Service (COTS) :**

- As we know, there are three stages involved in the connection oriented service.
- They are :
  1. Connection establishment.
  2. Exchange of data.
  3. Connection teardown.
- The connection oriented service is present at the network layer as well, but it is different from that at the transport layer.
- At the network layer, the meaning of connection oriented service involves the co-ordination between the hosts on either sides and all the routers between them.
- But at the transport layer, the meaning of connection oriented service is the end to end service that involves only the two hosts.
- Refer Fig. 6.3.12 to understand the concept of connection oriented service at the transport layer.




(G-2076) **Fig. 6.3.12 : Concept of connection oriented service**

- In Fig. 6.3.12, all the three stages namely connection establishment, data exchange and connection teardown have been shown.
- It is important to note that it is possible to implement the flow control, error control and congestion control in the connection oriented service.

**Comparison of CLTS & COTS :**

**Table 6.3.1 : Comparison of CLTS & COTS**

| Sr. No. | Parameter | Connection oriented | Connectionless |
|---|---|---|---|
| 1. | Reservation of resources | Necessary | Not necessary |
| 2. | Utilization of resources | Less | Good |
| 3. | State information | Lot of information required | Not much information is required to be stored |
| 4. | Guarantee of service | Guaranteed | No guarantee |
| 5. | Connection | Connection needs to be established | Connection need not be established |
| 6. | Delays | More | Less |
| 7. | Overheads | Less | More |
| 8. | Packets travel | Sequentially | Randomly |
| 9. | Congestion due to overloading | Not possible | Very much possible |

### 6.3.10 Quality of Service (QoS) :

- As mentioned earlier, the QoS parameters are as follows :





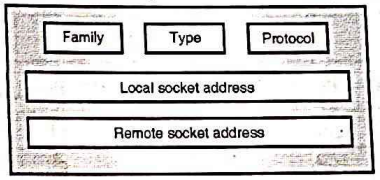
1. **Connection establishment delay :**
   - The time difference between the instant at which a request for transport connection is made and the instant at which it is confirmed is called as **connection establishment delay.**
   - This delay should be as short as possible to ensure better service.
2. **Connection establishment failure probability :**
   - Sometimes the connection may not get established even after the maximum connection establishment delay. This can be due to network congestion, lack of table space or some other problems.
3. **Throughput :**
   - It is defined as the number of bytes of user data transferred per second, measured over some time interval. Throughput is measured separately for each direction.
4. **Transit delay :**
   - It is the time duration between a message being sent by the transport user from the source machine and its being received by the transport user at the destination machine.
5. **Residual error ratio :**
   - It measures the number of lost or garbled messages as a percentage of the total messages sent.
   - Ideally the value of this ratio should be zero and practically it should be as small as possible.
6. **Protection :**
   - This parameter provides a way to protect the transmitted data against reading or modifying it by some unauthorised parties.
7. **Priority :**
   - Using this parameter the user can show that some of its connections are more important (have higher priority) than the other ones.
   - This is important when congestions take place. Because the higher priority connections should get service before the low priority connections.
8. **Resilience :**
   - Due to internal problem or congestion the transport layer spontaneously terminates a connection.
   - The resilience parameter gives the probability of such a termination.

## 6.4 Sockets :

- The socket interface was originally based on UNIX. It defines a set of system calls or procedure.
- The communication structure that we need in socket programming is called as a **socket.** A socket acts as an end point.
- Two processes can communicate if and only if both of them have a socket at their ends.

**Socket structure :**

- Fig. 6.4.1 shows a simplified socket structure.




(G-601) **Fig. 6.4.1 : Socket structure**

- Various fields in the socket structure are as follows :

1. **Family :** This field is used for defining the protocol group such as IPv4 or IPv6, UNIX domain protocol etc.
2. **Type :** This field is used for defining the type of socket such as stream socket, packet socket or raw socket.
3. **Protocol :** This field is usually set to zero for TCP and UDP.
4. **Local socket address :** It is used for defining the local socket address. This address is a combination of local IP address and the port address of the local application program.
5. **Remote socket address :** It is used for defining the remote socket address which is a combination of remote IP address and the port address of the remote application program.

### 6.4.1 Socket Types :

- There are three types of sockets :
  1. The stream socket
  2. The packet socket
  3. The raw socket
- All these sockets can be used in TCP/IP environment. Let us discuss them one by one.

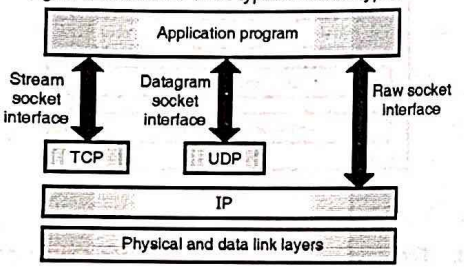
1. **Stream socket :**
   - This is designed for the connection oriented protocol such as TCP. The TCP uses a pair of stream sockets one each on either ends for connecting one application program to the other across the Internet.

**2. Datagram socket :**

- This type of socket is designed for the connectionless protocol such as UDP.
- UDP uses a pair of datagram sockets for sending a message from one application program to another across the Internet.

**3. Raw socket :**

- Raw sockets are designed for the protocols like ICMP or OSPF, because these protocols do not use either stream packets or datagram sockets.
- Fig. 6.4.2 shows the three types of socket types.




(G-602) **Fig. 6.4.2 : Type of sockets**

### 6.4.2 Berkeley Sockets :

- Table 6.4.1 lists various transport primitives used in Berkeley UNIX for TCP.

**Table 6.4.1 : Various transport primitives**

| Sr. No. | Primitive | Meaning |
|---|---|---|
| 1. | SOCKET | Create a new communication end point. |
| 2. | BIND | Provide a local address to a socket |
| 3. | LISTEN | Show willingness to accept connections |
| 4. | ACCEPT | Block the caller as long as a connection attempt does not arrive |
| 5. | CONNECT | Attempt to establish a connection |
| 6. | SEND | Send data |
| 7. | RECEIVE | Receive data |
| 8. | CLOSE | Release the connection |

- The first four primitives in the Table 6.4.1 are executed in the same order by the server. The SOCKET primitive creates a new end point and allocates table space for it within the transport entity.
- The newly created sockets do not have addresses. These are assigned using the **BIND** primitive.
- The **LISTEN** primitive allocates space to queue the incoming calls in case if several clients wish to connect at the same time.
- To block waiting for an incoming connection, the server executes an **ACCEPT** primitive.
- When a TPDU requesting for a connection arrives, the transport entity creates a new socket and returns a file descriptor for it.
- These were the primitives corresponding to server side. Now let us consider the client side.
- On the client side also a socket needs to be created first using the SOCKET primitive, however the BIND is not required.
- The **CONNECT** primitive blocks the caller and initiates the connection process.
- When it completes (which is indicated by an appropriate TPDU received from the server), the client process is unblocked and the connection is established.
- After this both the sides can use **SEND** and **RECEIVE** primitives to send and receive data.
- In order to release the connection, both sides have to execute a CLOSE primitive.

**Steps followed for socket programming :**

- The steps followed for the socket programming are as follows :

**Server side :**

1. Server creates a socket and checks for errors using SOCKET.
2. Assign address to the newly created socket using BIND.
3. Use the LISTEN to allocate space for the queue which is used for the incoming calls.
4. Execute an ACCEPT for blocking the waiting incoming connections.

**Client side :**

1. Create a socket using SOCKET.
2. Use CONNECT to initiate connection process.
3. Establish the connection.

## 6.5 Elements of Transport Protocols :

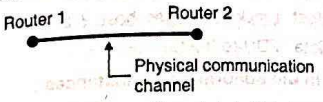
- In order to implement the transport layer services between the two transport entities, we have to use a **transport protocol.**
- The transport protocols have to deal with the following tasks :

1. Error control 2. Sequencing and 3. Flow control

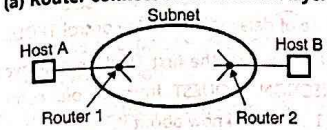




- The transport protocols are similar to the data link protocols in many ways but there are some dissimilarities as well.
- At the data link layer two router communicate directly via a physical channel as shown in Fig. 6.5.1(a), whereas at the transport layer the physical channel is replaced by the entire subnet as shown in Fig. 6.5.1(b).




**(a) Router connection at data link layer**




**(b) Router connection at transport layer**

(G-603) **Fig. 6.5.1**

- The difference between data link and transport layer communication is as given in Table 6.5.1.

**Table 6.5.1 : Difference between data link and transport layer**

| Sr. No. | Data link layer | Transport layer |
|---|---|---|
| 1. | Communication is through a physical channel. | Communication is through a subnet. |
| 2. | It is not necessary to specify the destination router. | Explicit addressing of destination is essential. |
| 3. | Establishing a connection is simple. | Initial connection establishment is more complicated. |
| 4. | No storage capacity. | There is some storage capacity in the subnet. |
| 5. | No additional delay. | Delay is introduced due to the storing capacity of subnets. |
| 6. | Different approaches are used for buffering and flow control by the two layers. | |

**Elements of transport protocols :**

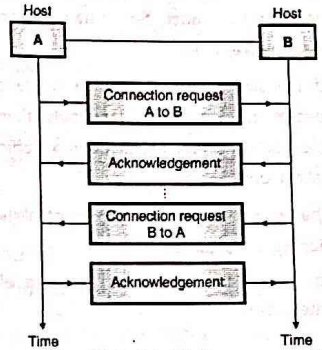
- Following are some of the important elements of transport protocols :
  1. Addressing
  2. Establishing a connection
  3. Releasing a connection
  4. Flow control and buffering
  5. Multiplexing
  6. Crash recovery

## 6.6 Connection Management :

- In a connection oriented service, a connection is established between source and destination. Then the data is transferred and at the end the connection is released.

### 6.6.1 Connection Establishment :

- Refer Fig. 6.6.1 to understand the connection establishment.




(G-604) **Fig. 6.6.1 : Establishing a connection**

- Following steps are taken to establish a connection :
  1. Host A sends a connection request packet to host B. This contains the initialisation information about data from A to B.
  2. Host B sends the packet of acknowledgement to confirm that it has received the request from A.
  3. Host B sends a connection request to A along with the initialisation information about traffic from B to A.
  4. Host A sends a packet of acknowledgement to confirm that it has received the request from B. It is possible to merge the steps 2 and 3.
- Note that each connection request must have a sequence number which is helpful in recovering from the loss or duplication of the packets. For the same reason, each acknowledgement also should have an acknowledgement number.
- The first sequence number in each direction should be random for each connection established. This is to ensure that a sender cannot create more than one connection which starts with the same sequence number (e.g. 2).
- This is important in recognizing the duplicate packets. Since a sequence number is required for each connection, the receiver has to keep the history of sequence numbers for each remote host for a specific amount of time but not indefinitely.

**Problems :**

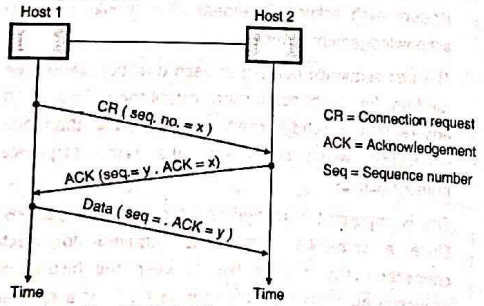
- Establishing a connection sounds easy. But actually it is a very tricky job.
- The problem occurs when the network can lose store and duplicate packets.
- The problems can be elaborated as follows :
  1. Due to congestion on a subnet, the acknowledgements do not get back in time from receiver to sender. So re-transmission of each packet takes place.
  2. If the subnet uses datagrams inside and every packet travels on a different route, then some of the packets might get stuck in a traffic jam and take a long time to arrive.
  3. The same connection getting re-established due to duplication of packet.
- So the crux of the problem is existence of delayed duplicate packets.

**Remedy :**

- The solution to this problem is to kill off the aged packets that are still wandering on the network.
- We should ensure that no packet lives longer than some predefined time.
- The packet lifetime can be restricted by using one of the following techniques :
  1. Restricted subnet design.
  2. Putting a hop counter in each packet.
  3. Time stamping each packet.

## 6.6.2 Three Way Handshake Technique :

- The delayed duplicate packet problem can be solved by using a technique called three way handshake.
- The principle of three way handshake is shown in Fig. 6.6.2(a).



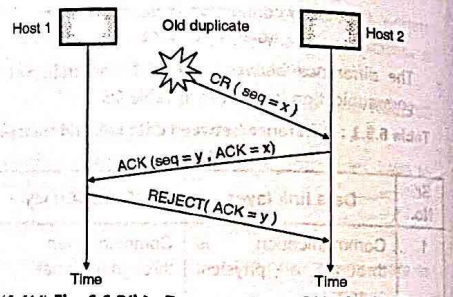

**(G-605) Fig. 6.6.2(a) : Three way handshake technique**

**Normal operation :**

1. Host 1 chooses a sequence number x and sends a TPDU containing the connection request (CR) TPDU to host 2.
2. Host 2 replies with a connection accepted TPDU to acknowledge x and to announce its own sequence number y.
3. Host 1 acknowledges host 2 and sends the first data TPDU to host 2.

**Operation in the abnormal circumstances :**

- Now let us see how the three way handshake works in presence of delayed duplicate control TPDUs.
- Refer Fig. 6.6.2(b). The first TPDU is a delayed duplicate CONNECTION REQUEST from an old connection. The HOST 1 does not know about it.

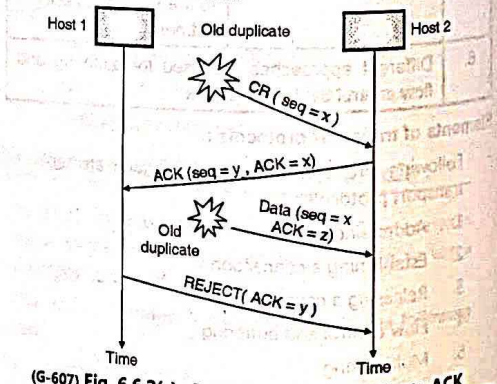



**(G-606) Fig. 6.6.2(b) : Response to an OLD duplicate**

- Host 2 receives this TPDU and sends to host 1, a connection accepted TPDU. But host 1 is not trying to establish any connection so it sends a REJECT along with ACK = y.
- So host 2 realizes that it was fooled by a delayed duplicate and abandons the connection.

**Duplicate CR and duplicate ACK :**

- This is another abnormal situation. Refer Fig. 6.6.2(c) to understand this situation.




**(G-607) Fig. 6.6.2(c) : Duplicate CR and duplicate ACK**





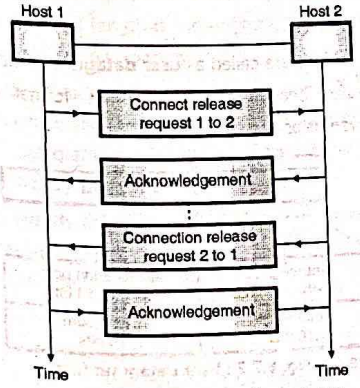
- This is the worst case in which delayed duplicates of both connection request (CR) and acknowledgement (ACK) are making rounds in the subnet.
- Host 2 gets a delayed duplicate CR and it replies to it by sending ACK. Note that host 2 has proposed a connection with a sequence number y.
- When the second delayed TPDU (duplicate) arrives at host 2 it understands that z has been acknowledged and not y. So it understands that this too is an OLD duplicate.

## 6.6.3 Connection Release :

- Any one of the two parties involved in data exchange can close the connection.
- But the problem is that, when connection is terminated from one end, the other party can continue to send data in the other direction.
- Hence, to ensure a proper connection release one has to follow the steps given below.

**Procedure to release a connection :**

- Refer Fig. 6.6.3 to understand this procedure.
  1. Host 1 sends a connection release request to host 2.
  2. Host 2 sends an acknowledgement to confirm the release request of host 1.
  3. After this the connection is closed in one direction (no data from host 1 to host 2) but host 2 can continue to send data to host 1.
  4. When host 2 finishes sending his data, it sends a connection release request to host 1.
  5. Host 1 acknowledges (confirms) the request made by host 2 and the connection is released from both ends.




**(G-608)Fig. 6.6.3 : Connection release**

- Releasing a connection is easier than establishing it. There are two styles of releasing a connection :

**Types of connecting release :**

1. Asymmetric release and
2. Symmetric release

**1. Asymmetric release :**

- In asymmetric release, when one party stops communicating the connection is broken. It is an abrupt release and it may lead to loss of data.

**2. Symmetric release :**

- Symmetric release treats the connection as two separate unidirectional connections and in order to release the connection each side must release the connection on its side.
- Symmetric release is as shown in Fig. 6.6.3 and there is no loss of data with the symmetrical release.

## 6.6.4 The Internet Transport Protocols :

- The Internet has two main protocols in the transport layer. One of them is connection oriented and the other one supports the connectionless service.
- TCP (Transmission Control Protocol) is a connection oriented protocol and UDP (User's Data Protocol) is the connectionless protocol. UDP is basically just IP with an additional short header.

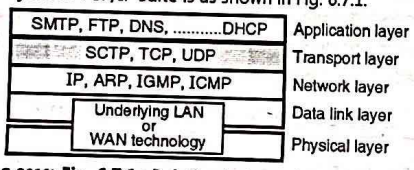
# 6.7 User Datagram Protocol (UDP) :

- The User Datagram Protocol is a very simple protocol. It adds little to the basic functionality of IP. Like IP, it is an unreliable, connectionless protocol.
- You do not need to establish a connection with a host before exchanging data with it using UDP, and there is no mechanism for ensuring that data sent is received.
- A unit of data sent using UDP is called a Datagram. UDP adds four 16-bit header fields (8 bytes) to whatever data is sent.
- These fields are : a length field, a checksum field, and source and destination port numbers. "Port number", in this context, represents a software port, not a hardware port.
- The concept of port numbers is common to both UDP and TCP. The port numbers identify which protocol module sent (or is to receive) the data.
- Most protocols have standard ports that are generally used for this.

- For example, the Telnet protocol generally uses port 23. The Simple Mail Transfer. Protocol (SMTP) uses port 25.
- The use of standard port numbers makes it possible for clients to communicate with a server without first having to establish which port to use.
- The port number and the protocol field in the IP header duplicate each other to some extent, though the protocol field is not available to the higher-level protocols. IP uses the protocol field to determine whether data should be passed to the UDP or TCP module.
- UDP or TCP use the port number to determine which application-layer protocol should receive the data.
- Although UDP isn't reliable, it is still a preferred choice for many applications.
- It is used in real-time applications like Net audio and video where, if data is lost, it's better to do without it than send it again out of sequence. It is also used by protocols like the Simple Network Management Protocol (SNMP).

**Relationship with other protocols :**

- The relationship of UDP with the other protocols and layers of TCP/IP suite is as shown in Fig. 6.7.1.




**(G-2019) Fig. 6.7.1 : Relation between UDP and other protocols**

- As shown, UDP is located between IP and application layer. It therefore works as an intermediary between application program and the network layer.

## 6.7.1 Responsibilities of UDP :

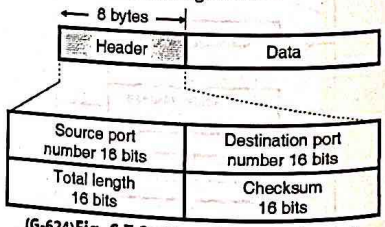
- Being a transport layer protocol, the UDP has the following responsibilities :
  1. To create a process to process communication, UDP uses port numbers to accomplish this.
  2. To provide control mechanisms at the transport layer, UDP does not provide flow control or acknowledgements. It provides error detection. The erroneous packet is discarded.
  3. UDP does not add anything to the services of IP except for providing process to process communication.

## 6.7.2 Advantages of UDP :

- UDP, despite all its simplicity and powerlessness is still used because it offers the following advantages :
  1. UDP has minimum overheads.
  2. UDP can be easily used if the sending process is not too bothered about reliability.
  3. UDP reduces interaction between sender and receiver.

## 6.7.3 User Datagram :

- User Datagram Protocol (UDP) provides a connectionless packet service that offers unreliable 'best effort' delivery.
- This means that the arrival of packets is not guaranteed, nor is the correct sequencing of delivered packets.
- Applications that do not require an acknowledgement of receipt of data, for example, audio or video broadcasting uses UDP.
- UDP is also used by applications that typically transmit small amounts of data at one time, for example, the Simple Network Management Protocol (SNMP).
- UDP provides a mechanism that application programs use to send data to other application programs.
- UDP provides protocol port numbers used to distinguish between multiple programs executing on a single device.
- That is, in addition to the data sent, each UDP message contains both a destination port number and a source port number.
- This makes it possible for the UDP software at the destination to deliver the message to the correct application program, and for the application program to send a reply.
- UDP packets are called as **user datagrams.** They have a fixed-size header of 8-bytes. The format of user datagram is as shown in Fig. 6.7.2.




**(G-624)Fig. 6.7.2 : User datagram format**





- The UDP header is divided into the following four 16-bit fields :
  1. Source port
  2. Destination port
  3. Total length
  4. Checksum.

**Source port number :**

- Source port is an optional field, when meaningful, it indicates the port of the sending process, and may be assumed to be the port to which a reply should be addressed in the absence of any other information.
- If not used, a value of zero is inserted. This is a 16 bit field. That means the port numbers can range from 0 to 65,535.
- If the source host is a client, means if a client is sending a request using UDP, then generally a **ephemeral (temporary)** port number is requested by the process and chosen by the UDP.
- If the source host is a server that means if a server is sending a response message, mostly the **well known port** number is used.

**Destination port number :**

- The destination port number also is a 16 bit number and this port number is used by the process running on the destination host.
- If the destination host is a server that means if a client is sending a request to it, then a **well known port** number is used in most cases. However if the destination host is a client than means if a server is sending its response to it, then the chosen port number is generally an **ephemeral port** number.

**Length :**

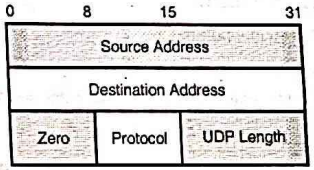
- It is also a 16 bit field which is used for defining the total length of the UDP datagram including header as well as data.
- Due to 16 bit length it can define a total length of the datagram up to 65,535 bytes.
- However practically the total length of a UDP datagram is much smaller than 65,535 bytes. This is because the UDP datagram is to be stored in an IP datagram which itself has a length of 65,535 bytes.
- The **length** field in the UDP datagram is actually not necessary, because this UDP datagram is actually encapsulated in an IP datagram and the IP datagram has its own length field.
- So without using the length field in UDP datagram, we can obtain the length of the UDP datagram as follows : UDP length = IP length – IP header length. Note that while delivering the UDP datagram to UDP layer, the IP software drops the IP header.

**UDP Checksum :**

- This is used to verify the integrity (i.e. to detect errors) of the UDP header.
- The checksum is performed on a "pseudo header" consisting of information obtained from the IP header (source and destination address) as well as the UDP header.

## 6.7.4 UDP Pseudo Header :

- The purpose of using a pseudo-header is to verify that the UDP packet has reached its correct destination.
- The correct destination consists of a specific machine and a specific protocol port number within that machine.




**(G-625) Fig. 6.7.3 : UDP pseudo header**

- The UDP header itself specifies only the protocol port number. Thus, to verify the destination, UDP on the sending machine computes a checksum that covers the destination IP address as well as the UDP packet.
- At the ultimate destination, UDP software verifies the checksum using the destination IP address obtained from the header of the IP packet that carried the UDP message.
- If the checksum agrees, then it must be true that the packet has reached the intended destination host as well as the correct protocol port within that host.

**User interface :**

- A user interface should allow the creation of new receive ports, receive operations on the receive ports that return the data octets and an indication of source port and source address, and an operation that allows a datagram to be sent, specifying the data, source and destination ports and addresses to be sent.

**IP interface :**

- The UDP module must be able to determine the source and destination Internet addresses and the protocol field from the Internet header.

- One possible UDP/IP interface would return the whole Internet datagram including the entire Internet header in response to a receive operation.
- Such an interface would also allow the UDP to pass a full Internet datagram complete with header to the IP to send. The IP would verify certain fields for consistency and compute the Internet header checksum.

**Protocol application :**

- The major uses of this protocol are the Internet Name Server, and the Trivial File Transfer.

**Protocol number :**

- This is protocol 17 (21 octal) when used in the Internet Protocol.

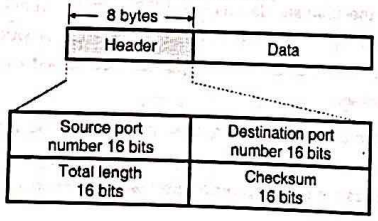
---

**Ex. 6.7.1 :** Explain UDP header. The following is a dump of a UDP header in hexadecimal format
06 32 00 0D 00 1C E2 17

1. What is source port number ?
2. What is destination port number ?
3. What is the total length of the user datagram ?
4. What is the length of the data ?
5. Is the packet directed from a client to a server or vice versa ?
6. What is the client process ?

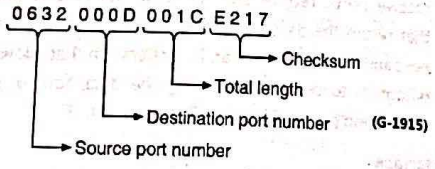
**Soln. :**

- Refer section 6.7.4 for UDP header.
- Fig. P. 6.7.1 shows the UDP header.




**(G-624) Fig. P. 6.7.1 : UDP header**

- The given UDP header is as follows :




1. Source port number = $(06\ 32)_H = 434$ ...Ans.
2. Destination port number = $(0D\ 00)_H = 832$ ...Ans.
3. Total length (header + data) = $(00\ 1E)_H = 30$ bytes ...Ans.
4. Length of data = 30 – 8 (header) = 22 bytes ...Ans.

## 6.8 UDP Services :

- In this section we are going to discuss the following important services provided by the UDP :
  1. Process to process communication.
  2. Connectionless services.
  3. Flow control.
  4. Error control.
  5. Checksum.
  6. Congestion control.
  7. Encapsulation and decapsulation.
  8. Queuing.
  9. Multiplexing and demultiplexing.

### 6.8.1 Process to Process Communication :

- We have already discussed the process to process communication in a general sense, earlier in this chapter. UDP also does it with the help of sockets which is a combination of IP address and port numbers.
- Table 6.8.1 shows different port numbers used by UDP. Some of these ports can be used by UDP as well as TCP.

**Table 6.8.1 : Well known ports used with UDP**

| Port | Protocol | Description |
|---|---|---|
| 7 | Echo | The received datagram is echoed back to sender. |
| 9 | Discard | Any received datagram is discarded. |
| 11 | Users | Active users. |
| 13 | Daytime | Return the day and the current time. |
| 17 | Quote | Return the quote of the day. |
| 19 | Chargen | To return a string of characters. |
| 53 | Nameserver | Domain Name Service (DNS). |
| 67 | BOOT PS | This is the server port to download the bootstrap information. |
| 68 | BOOT PC | This is the client port to download bootstrap information. |
| 69 | TFTP | Trivial File Transport Protocol. |
| 111 | RPC | Remote Procedure Call. |
| 123 | NTP | Network Time Protocol. |
| 161 | SNMP | Simple Network Management Protocol. |
| 162 | SNMP | Simple Network Management Protocol (Trap). |





### 6.8.2 Connectionless Services :

- As UDP is a connectionless, unreliable protocol, each user datagram sent using UDP is an independent datagram.
- Different user datagrams sent by the UDP have absolutely no relationship between them. This is true even for those datagrams which are originating from the same process and being sent to the same destination.
- The user datagrams do not have any number. Also the connection establishment and release are not at all required.
- So each datagram is free to travel any path. Only those processes which are sending very short messages can successfully use the UDP.

### 6.8.3 Flow and Error Control :

- Being a connectionless protocol, UDP is a simple, unreliable protocol. It does not provide any flow control, hence the receiver can overflow with incoming messages.
- UDP does not support any other error control mechanism, except for the checksum. There are no acknowledgements sent from destination to sender.
- Hence the sender does not know if the message has reached, lost or duplicated. If the receiver detects any error using the checksum, then that particular datagram is discarded.

### 6.8.4 Checksum :

- The calculation of checksum for UDP is different than that for IP. In UDP the checksum is calculated by considering the following three sections :
  1. A pseudoheader
  2. The UDP header.
  3. The data coming from the application layer.
- The checksum in UDP is **optional.** That means the sender can make a decision of not calculating the checksum.
- If so, then the checksum field is filled with all zeros before sending the UDP packet.
- In case if the calculated checksum is all zeros (when the sender decides to send checksum) then an all 1 checksum is sent.
- This solution works without any problem because, a checksum will never have an all 1 value.

### 6.8.5 Congestion Control :

- UDP does not provide any congestion control. It assumes that the UDP packets being small, will not create any congestion. But this assumption may not always be correct.

## 6.9 UDP Applications :

- Despite being connectionless, unreliable, no flow control, no error control, UDP is still preferred for some applications.
- This is because UDP has some advantages too. An application designer has to sometimes compromise between advantages and drawbacks to get the optimum.
- Some of the typical applications of UDP are as follows :
  1. UDP is suitable for the applications (processes) that have the following requirements :
     (a) A simple response to request is to be made.
     (b) Flow and error controls not essential.
     (c) Bulk data is not to be sent (like FTP).
  2. UDP is used for RIP (Routing Information Protocol).
  3. UDP is used for management processes such as SNMP.
  4. UDP is suitable for the processes having inbuilt flow and error control mechanisms, such as TFTP.
  5. UDP is suitable for the multicasting applications.
  6. UDP is also used in the real time applications which do not tolerate the uneven delays.

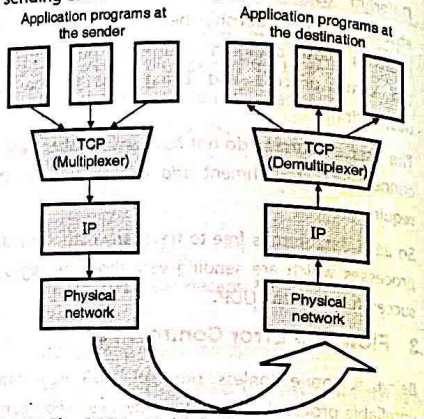
## 6.10 Transmission Control Protocol (TCP) :

- The TCP provides reliable transmission of data in an IP environment. TCP corresponds to the transport layer (Layer 4) of the OSI reference model.
- Among the services TCP provides are stream data transfer, reliability, efficient flow control, full-duplex operation, and multiplexing.
- TCP is the layer 4 protocol in the TCP/IP suite and it is a very important and complicated protocol. TCP has been revised multiple times in last few decades.
- With stream data transfer, TCP delivers an unstructured stream of bytes identified by sequence numbers. This service benefits applications because they do not have to chop data into blocks before handing it off to TCP.
- Instead, TCP groups bytes into segments and passes them to IP for delivery.

- TCP offers reliability by providing connection-oriented, end-to-end reliable packet delivery through an internetwork.
- It does this by sequencing bytes with a forwarding acknowledgment number that indicates to the destination the next byte the source expects to receive.
- Bytes not acknowledged within a specified time period are retransmitted.
- The reliability mechanism of TCP allows devices to deal with lost, delayed, duplicate, or misread packets.
- A time-out mechanism allows devices to detect lost packets and request retransmission.
- TCP offers efficient flow control, which means that, when sending acknowledgments back to the source, the receiving TCP process indicates the highest sequence number that it can receive without overflowing its internal buffers.
- TCP supports a full-duplex operation means that TCP processes can both send and receive at the same time.
- Finally, TCP's multiplexing means that numerous simultaneous upper-layer conversations can be multiplexed over a single connection.

### 6.10.1 Relationship Between TCP and IP :

- The relationship between TCP and IP is very interesting.
- Each TCP message gets encapsulated or inserted in an IP datagram and then this datagram is sent over the Internet to the destination.
- IP transports this datagram from sender to destination, without bothering about the contents of the TCP message.
- At the final destination the IP hands over the message to the TCP software running on the destination computer.
- IP acts like a postal service and transfers the datagrams from one computer to the other. Thus TCP deals with the actual data to be transferred and IP takes care of transfer of that data.
- Many applications such as FTP, Remote login TELNET etc. keep sending data to TCP software on the sending computer.
- The TCP software acts as a multiplexer at the sending computer.
- It receives data from various applications, multiplexes the data and hands it over to the IP software at the sending end as shown in Fig. 6.10.1.




(G-1440) **Fig. 6.10.1 : Multiplexing and demultiplexing using TCP**

- IP adds its own header to this TCP packet and creates an IP packet out of it.
- Then this packet is sent to its destination. At the destination exactly opposite process will take place.
- The IP software hands over the multiplexed data to the TCP software.
- The TCP software at the destination computer then demultiplexes the multiplexed data and gives it to the corresponding applications as shown in Fig. 6.10.1.

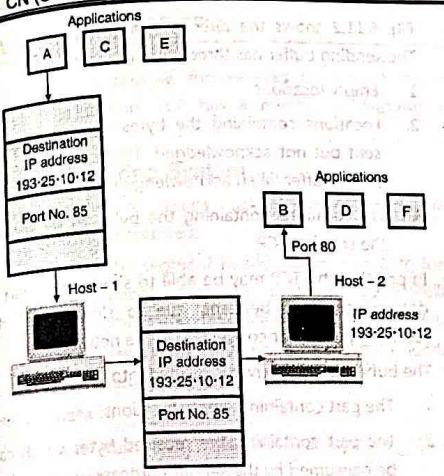
### 6.10.2 Ports and Sockets :

**1. Ports :**

- Applications running on different hosts communicate with TCP with the help of ports. Every application has been allotted a unique 16 bit number which is known as a **port.**
- When an application on one computer wants to communicate using a TCP connection to another application on some other computers these ports prove to be very helpful.
- Let an application A on host 1 wants to communicate with an application B on host 2. So the process takes place as shown in Fig. 6.10.2 and explained below.





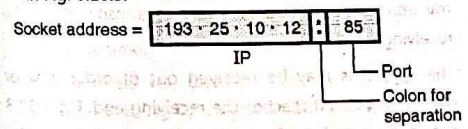



(G-1437) **Fig. 6.10.2 : Use of port numbers**

- Application A running on computer 1 provides the IP address of computer 2 and the port number corresponding to application B as shown in Fig. 6.10.2.
- Computer 1 communicates with computer 2 using the IP address and computer 2 uses the port number to direct the message to application B.

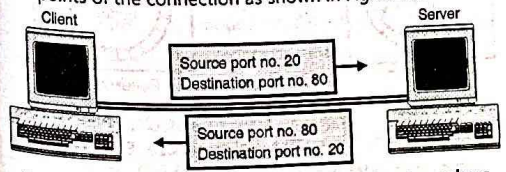
**2. Sockets :**

- A port is a 16 bit unique number used for identification of a single application. But socket address or simply socket would identify the combination of the IP address and the port number concatenated together as shown in Fig. 6.10.3.




(G-1438) **Fig. 6.10.3**

- For example if the IP address = 193.25.10.12 and the port number is 85. Then this port of this computer will have the following socket address.
- So a pair of sockets is required to identify a TCP connection between two applications on two different hosts. These two socket addresses specify the end points of the connection as shown in Fig. 6.10.4




(G-1436) **Fig. 6.10.4 : Source and destination port numbers**

- Generally the server port numbers are known as the **well known ports.**
- Some of the well known port numbers have already been mentioned for UDP and TCP earlier in this chapter.
- Multiple TCP connections between different applications or same applications on two hosts exist in practice.
- Here the IP addresses of the two hosts are same but the port numbers are different. The communication using port numbers is illustrated in Fig. 6.10.4

## 6.11 TCP Services :

- Following are some of the services offered by TCP to the processes at the application layer :
  1. Stream delivery service
  2. Sending and receiving buffers
  3. Bytes and segments
  4. Full duplex service
  5. Connection oriented service
  6. Reliable service.
  7. Process to process communication.

### 6.11.1 Process to Process Communication :

- The TCP uses port numbers a transport layer addresses.
- Table 6.11.1 shows some well known port numbers used by TCP.

**Table 6.11.1 : Well known ports used by TCP**

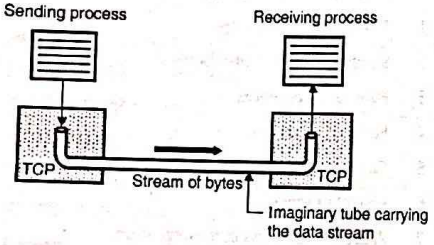
| Port | Protocol | Description |
|---|---|---|
| 7 | Echo | Sends received datagram back to sender |
| 9 | Discard | Discards any received packet |
| 11 | Users | Active users |
| 13 | Daytime | Sends the date and the time |
| 17 | Quote | Sends a quote of the day |
| 19 | Chargen | Sends a string character |
| 20 | FTP, Data | File Transfer protocol for data |
| 21 | FTP, Control | File Transfer protocol for control |
| 23 | TELNET | Terminal network |
| 25 | SMTP | Simple Mail Transfer Protocol |
| 53 | DNS | Domain Name server |
| 67 | BOOTP | Bootstrap Protocol |
| 79 | Finger | Finger |

| Port | Protocol | Description |
|---|---|---|
| 80 | HTTP | Hypertext Transfer Protocol |
| 111 | RPC | Remote Procedure Call |

- Note that if an application can use both UDP and TCP, the same port number is assigned to this application.

### 6.11.2 Stream Delivery Service :

- TCP is a stream oriented protocol. The sending process delivers data in the form of a stream of bytes and the receiving process receives it in the same manner.
- TCP creates a working environment in such a way that the sending and receiving processes seem to be connected by an imaginary "tube" as shown in Fig. 6.11.1.

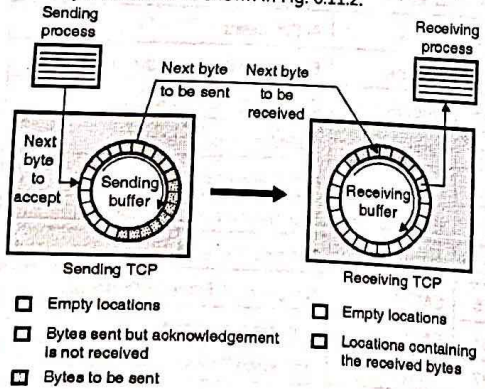



(G-621) **Fig. 6.11.1 : Stream delivery service**

- This is called as stream delivery service.

### 6.11.3 Sending and Receiving Buffers :

- The sending and receiving processes may not produce and receive data at the same speed. Hence TCP needs buffers for storage of data at both the ends.
- There are two types of buffers used in each direction :
  1. Sending buffer 2. Receiving buffer.
- A buffer can be implemented by using a circular array of 1 byte locations as shown in Fig. 6.11.2.

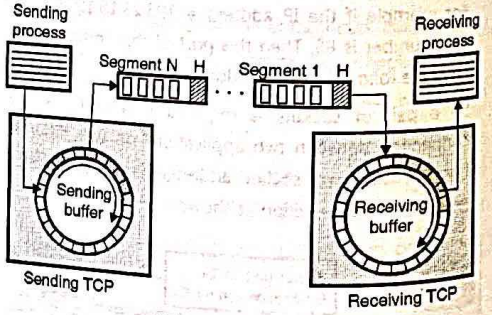



(G-622) **Fig. 6.11.2 : Sending and receiving buffers**

- Fig. 6.11.2 shows the direction of movement of data. The sending buffer has three types of locations :
  1. Empty locations
  2. Locations containing the bytes which have been sent but not acknowledged. These bytes are kept in the buffer till an acknowledgement is received.
  3. The locations containing the bytes to be sent by the sending TCP.
- In practice, the TCP may be able to send only a part of data which is to be sent, due to slowness of the receiving process or congestion in the network.
- The buffer at the receiver is divided into two parts :
  1. The part containing empty locations.
  2. The part containing the received bytes which can be consumed by the sending process.

### 6.11.4 Bytes and Segments :

- Buffering is used to handle the difference between the speed of data transmission and data consumption. But only buffering is not enough. We need one more step before sending the data.
- The IP layer, which provides service to TCP, has to send data in the form of packets instead of stream of bytes.
- At the transport layer, TCP groups a number of bytes to form a packet called a segment. A header is added to each segment for the purpose of exercising control.
- The segments are then inserted in an IP datagram and transmitted. The entire operation is transparent to the receiving process.
- The segments may be received out of order, lost or corrupted when it reaches the receiving end. Fig. 6.11.3 shows the creation of segments from the bytes in the buffers.




(G-623) **Fig. 6.11.3**

- The segments are not of the same size. Each segment can carry hundreds of bytes.





## 6.12 Features of TCP :

- In order to provide the services mentioned in the previous section, TCP has a number of features as follows :

### 6.12.1 Numbering System :

- The TCP software keeps track of the segments being transmitted or received.
- However in the segment header there is no field for a segment number value. But there are fields called sequence number and the acknowledgement number.
- Note that these fields correspond to the byte number and not the segment number.

**Byte numbers :**

- TCP give numbers to all the data bytes which are transmitted.
- The numbering is independent of the direction of data travel. The numbering does not always start from 0, but it can start with a randomly generated number between 0 and $2^{32} - 1$.

**Sequence number :**

- After numbering the bytes, the TCP assigns a sequence number to each segment that is being transmitted. The sequence number for each segment is same as the number assigned to the first byte present in that segment.

**Acknowledgement number :**

- The TCP communication is duplex. So both the communicating processes can send and receive data at the same time.
- Each process will give numbers to the bytes with a different starting byte number. Each party also uses an acknowledgement number to confirm the reception of bytes.
- The acknowledgement number is cumulative i.e. the receiver takes the number of the last byte received, adds 1 to it and uses this sum as the acknowledgement number.

### 6.12.2 Flow Control :

- TCP provides flow control (UDP does not). The receiver will control the amount of data to be sent by the sender. This will avoid data overflow at the receiver. The TCP uses byte oriented flow control.

### 6.12.3 Error Control :

- The error control mechanism is inbuilt for TCP. This allows TCP to provide a reliable service.
- The error control mechanism considers a segment as the unit of data for error correction however the byte oriented error control is provided.

### 6.12.4 Congestion Control :

- TCP takes the congestion in network into account. UDP does not do this. The amount of data sent by the sender depends on the following factors :
  1. The receivers decision (flow control).
  2. The network congestion.

**Summary of TCP features :**

1. TCP is a process-to-process protocol.
2. TCP uses port numbers.
3. It is a connection oriented protocol (creates a virtual connection).
4. It uses flow and error control mechanisms.
5. TCP is a reliable protocol.

## 6.13 The TCP Protocol :

- Let us take a general overview of the TCP protocol. Every byte on a TCP connection has its own 32-bit sequence number.
- These numbers are used for both acknowledgement and for window mechanism.

**Segments :**

- The sending and receiving TCP entities exchange data in the form of segments.
- A segment consists of a fixed 20 byte header (plus and optional part) followed by zero or more data bytes.

**Segment size :**

- The segment size is decided by the TCP software. Two limits restrict the segment size as follows :
  1. Each segment including the TCP header, must fit in the 65535 byte IP payload.
  2. Each segment must fit in the **MTU (Maximum Transfer Unit).** Each network has a maximum transfer unit. Practically an MTU which is a few thousand bytes defines the upper limit on the segment size.

**Fragmentation :**

- If a segment is too large, then it should be broken into small segments. Using fragmentation by a router.
- Each new segment gets a new IP header. So the fragmentation by router will increase the overhead.

**Timer :**

- The basic protocol used by TCP entities is the sliding window protocol.
- A sender starts a timer as soon as a sender transmits a segment.
- When the segment is received by the destination, it sends back acknowledgement along with data if any.
- The acknowledgement number is equal to the next sequence number it expects to receive.
- If the timer at the sender goes out before the acknowledgement reaches back, it will **retransmit** that segment again.

**Possible problems :**

- As the segments can be fragmented, a part of the transmitted segment only may reach the destination with the remaining part lost.
- Segments can arrive out of order.
- Segments can get delayed so much that timer is out and unnecessary retransmission will take place.
- If a retransmitted segment takes a different route than the original segment is fragmented then the fragments of original and retransmitted segments can reach the destination in a sporadic way.
- So a careful administration is required to achieve reliable byte stream. There is a possibility of congestion or broken network along the path.
- TCP should be able to solve these problems in an efficient manner.

## 6.13.1 TCP Segment :

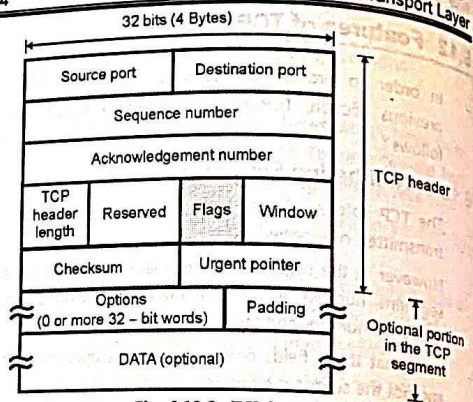
- The TCP segment as shown in Fig. 6.13.1.


**(G-1423)Fig. 6.13.1 : TCP segment**

- It consists of two parts :
  1. Header
  2. Data

## 6.13.2 The TCP Segment Header :

- Fig. 6.13.2 shows the layout of a TCP segment.




**(G-611)Fig. 6.13.2 : TCP header format**

- Every segment begins with a 20 byte fixed format header. The fixed header may be followed by header options.
- After the options, if any, upto 65535 – 20 – 20 = 65495 data bytes may follow.
- Note that the first 20 bytes correspond to the IP header and the next 20 correspond to the TCP header.
- The TCP segment without data is used for sending the acknowledgements and control messages.

**Source port :**

- A 16-bit number identifying the application the TCP segment originated from within the sending host.
- The port numbers are divided into three ranges, well-known ports (0 through 1023), registered ports (1024 through 49,151) and private ports (49,152 through 65,535).
- Port assignments are used by TCP as an interface to the application layer.

**Destination port :**

- A 16-bit number identifying the application the TCP segment is destined for on a receiving host.
- Destination ports use the same port number assignments as those set aside for source ports.

**Sequence number :**

- A 32-bit number identifying the current position of the first data byte in the segment within the entire byte stream for the TCP connection.
- After reaching $2^{32} - 1$, this number will wrap around to 0.





**Acknowledgement number :**

- A 32-bit number identifying the next data byte the sender expects from the receiver.
- Therefore, the number will be one greater than the most recently received data byte. This field is only used when the ACK control bit is turned on.

**Header length or offset :**

- A 4-bit field that specifies the total TCP header length in 32-bit words (or in ultiples of 4 bytes if you prefer).
- Without options, a TCP header is always 20 bytes in length. The largest a TCP header may be is 60 bytes.
- This field is required because the size of the options field(s) cannot be determined in advance. Note that this field is called "data offset" in the official TCP standard, but header length is more commonly used.

**Reserved :**

- A 6-bit field currently unused and reserved for future use.

**Control bits or flags :**

1. **Urgent pointer (URG) :** If this bit field is set, the receiving TCP should interpret the urgent pointer field.
2. **Acknowledgement (ACK) :** If this bit field is set, the acknowledgement field described earlier is valid.
3. **Push function (PSH) :** If this bit field is set, the receiver should deliver this segment to the receiving application as soon as possible.
   - An example of its use may be to send a Control-BREAK request to an application, which can jump ahead of queued data.
4. **Reset the connection (RST) :** If this bit is present, it signals the receiver that the sender is aborting the connection and all queued data and allocated buffers for the connection can be freely relinquished.
5. **Synchronize (SYN) :** When present, this bit field signifies that sender is attempting to "synchronize" sequence numbers.
   - This bit is used during the initial stages of connection establishment between a sender and receiver.
6. **No more data from sender (FIN) :** If set, this bit field tells the receiver that the sender has reached the end of its byte stream for the current TCP connection.

**Window :**

- A 16-bit integer used by TCP for flow control in the form of a data transmission window size. This number tells the sender how much data the receiver is willing to accept.
- The maximum value for this field would limit the window size to 65,535 bytes, however a "window scale" option can be used to make use of even larger windows.

**Checksum :**

- A TCP sender computes a value based on the contents of the TCP header and data fields. This 16-bit value will be compared with the value the receiver generates using the same computation.
- If the values match, the receiver can be very confident that the segment arrived intact.

**Urgent pointer :**

- In certain circumstances, it may be necessary for a TCP sender to notify the receiver of urgent data that should be processed by the receiving application as soon as possible.
- This 16-bit field tells the receiver when the last byte of urgent data in the segment ends.

**Options :**

- In order to provide additional functionality, several optional parameters may be used between a TCP sender and receiver.
- Depending on the option(s) used, the length of this field will vary in size, but it cannot be larger than 40 bytes due to the size of the header length field (4 bits).
- The most common option is the Maximum Segment Size (MSS) option.
- A TCP receiver tells the TCP sender the maximum segment size it is willing to accept through the use of this option.
- Other options are often used for various flow control and congestion control techniques.

**Padding :**

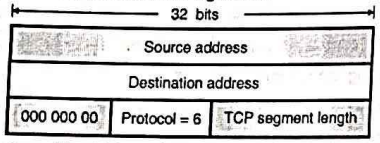
- Because options may vary in size, it may be necessary to "pad" the TCP header with zeros so that the segment ends on a 32-bit word boundary as defined by the standard.

**Data :**

- Although not used in some circumstances (e.g. acknowledgement segments with no data in the reverse direction), this variable length field carries the application data from TCP sender to receiver.
- This field coupled with the TCP header fields constitutes a TCP segment.

### 6.13.3 Checksum :

- A checksum is provided to ensure extreme reliability. It checksums the header, the data and the conceptual pseudo header shown in Fig. 6.13.3.




**(G-612)Fig. 6.13.3 : The pseudo header included in the TCP checksum**

- When the checksum is being computed, the TCP checksum field is set to zero, and the data field is padded out with an additional zero byte if its length is an odd number.
- Then all the 16 bit words are added in 1's complement and then 1's complement of the sum is taken to get the checksum.
- When a receiver performs the calculation on the entire segment including the checksum field, the result has to be zero.
- The pseudo header contains the 32 bit IP address of the source and destination machines, the protocol number for TCP i.e. 6 and the TCP segment length as shown in Fig. 6.13.3.

### 6.13.4 Encapsulation :

- The data coming from the application layer is encapsulated in a TCP segment.
- This TCP segment is then encapsulated in an IP datagram.
- The IP datagram is encapsulated in a frame at the data link layer.
- The process of encapsulation is shown in Fig. 6.13.4.


**(G-2072) Fig. 6.13.4 : Encapsulation**

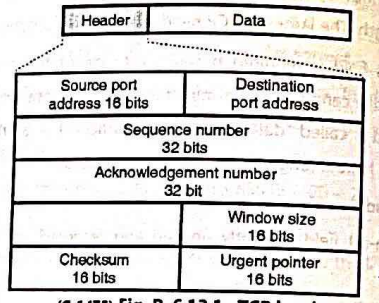
**Ex. 6.13.1 :** The following is a dump of TCP header in hexadecimal format

05320017 00000001 00000000 500207FF 00000000

1. What is the source port number ?
2. What is the sequence number ?
3. What is the acknowledgement number ?
4. What is the length of header ?
5. What is the window size ?

***Soln. :***

- The TCP segment format and its header format are as shown in Fig. P. 6.13.1.




**(G-1475) Fig. P. 6.13.1 : TCP header**

- The given TCP header consists of 5 words 32 bit each. We can identify them as follows : **(G-2364)**

1. 0532 0017 — Destination port address (16 bit)
2. 0000 0001 — Sequence number (32 bit)
3. 0000 0000 — Acknowledgement number (32 bit)
4. 5002 07FF — Window size (16 bit)

- Therefore :
  1. Sequence number = 0000 0001 = 1
  2. Destination port number = $(0\ 0\ 1\ 7)_H$ = 23
  3. Acknowledgement number = 0000 0000 = 0
  4. Window size = 07FF = 2047 bytes

## 6.14 A TCP Connection :

- TCP is a connection oriented protocol. Such a protocol would establish a virtual path between the sender and the receiver.
- Multiple segments corresponding to the message are then sent over this virtual connection.
- As TCP is using the same single path for the entire path, it can use the same path for acknowledgements and retransmission of damaged or lost packets.

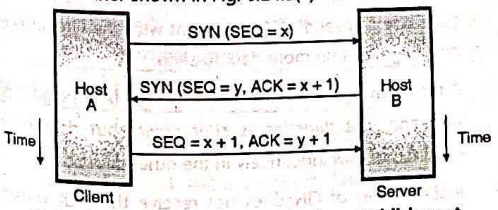




- While discussing the relation between TCP and IP we have seen how TCP uses the services of IP.
- TCP operates at a higher level than IP and the TCP connection is virtual and not physical.
- Though IP delivers the individual segments to the destination, the entire control on the connection is exercised by TCP.
- If a segment is lost or damaged, the TCP makes a decision of its retransmission, and IP does not know anything about it.
- The **three phases** in the connection oriented TCP transmission are as follows :
  1. Connection establishment
  2. Data transfer and
  3. Connection termination.

### 6.14.1 TCP Connection Establishment :

- To make the transport services reliable, TCP hosts must establish a connection-oriented session with one another. Connection establishment is performed by using a **three-way handshake** mechanism.
- A three-way handshake synchronizes both ends of a connection by allowing both sides to agree upon initial sequence numbers.
- This mechanism also guarantees that both sides are ready to transmit data and know that the other side is ready to transmit as well.
- This is necessary so that packets are not transmitted or re-transmitted during session establishment or after session termination.
- Each host randomly chooses a sequence number used to track bytes within the stream it is sending and receiving. Then, the three-way handshake proceeds in the manner shown in Fig. 6.14.1(a).

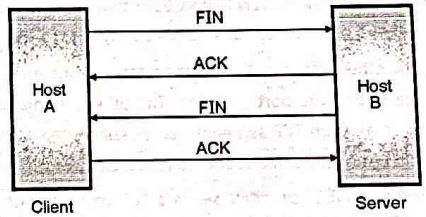



**(G-613) Fig. 6.14.1(a) : TCP connection establishment (Three-way handshake)**

- The requesting end (HOST A) sends a SYN segment specifying the port number of the server that the client wants to get connected to, and the client's initial sequence number (x).
- The server (HOST B) responds with its own SYN segment containing the server's initial sequence number (y).
- The server also acknowledges the client's SYN by acknowledging the client's SYN plus one (x + 1). A SYN consumes one sequence number.
- The client must acknowledge this SYN from the server by acknowledging the server's SYN plus one. (SEQ = x + 1, ACK = y + 1).
- This is how a TCP connection is established.

### 6.14.2 Connection Termination Protocol :

- While it takes three segments to establish a connection, it takes four to terminate a connection.
- Since a TCP connection is full-duplex (that is, data flows in each direction independently of the other direction), the connection should be terminated in both the directions independently. The termination procedure in each direction is shown in Fig. 6.14.1(b).



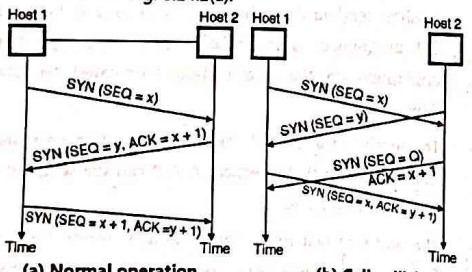

**(G-614) Fig. 6.14.1(b) : TCP termination**

- The rule is that either side can send a FIN when it has finished sending data (FIN indicates finished). When a TCP program on a host receives a FIN, it informs the application that the other end has terminated the data flow.
- The receipt of a FIN only means there will be no more data flowing in that direction. A TCP can still send data after receiving a FIN.
- The end that first issues the close (e.g., sends the first FIN) performs the active close and the other end (that receives this FIN) performs the passive close.
- Now refer Fig. 6.14.1(b). When the server receives the FIN it sends back an ACK of the received sequence number plus one.
- A FIN consumes a sequence number, just like a SYN. At this point the server's TCP also delivers an end-of-file to the application (the discard server).

- The server then closes its connection and its TCP sends a FIN to the client. The client's TCP informs the application and sends an ACK to server by incrementing the received sequence number by one.
- Connections are normally initiated by the client, with the first SYN going from the client to the server. A client or server can actively close the connection (i.e. send the first FIN).
- But in practice generally the client determines when the connection should be terminated, since client processes are often driven by an interactive user, who enters something like quit to terminate. This is how the TCP connection is released.

### 6.14.3 TCP Connection Management :

- Connections are established in TCP by following the three-way handshake technique. To establish a connection, one side, say the server, passively waits.
- It executes the LISTEN and ACCEPT primitives, to specify either a particular other side or nobody in particular.
- The other side (client) executes a connect primitive, with the IP and the port specified. The other information is the maximum TCP segment size, possible other options and optionally some user data (e.g. a password).
- The CONNECT primitive sends a TCP segment with the SYN bit on and the ACK bit off and waits for a response.
- The sequence of TCP segments sent in the normal case is shown in Fig. 6.14.2(a).




**(a) Normal operation** **(b) Call collision**

**(G-615) Fig. 6.14.2 : TCP connection management**

When the segment sent by host – 1 reaches the destination i.e. host – 2 the receiving server checks to see if there is a process that has done a LISTEN on the port given in the destination port field.

If not, it sends a reply with the RST bit on to reject the connection.

- Otherwise it gives the TCP segment to the listening process, which can accept or refuse (e.g. if it does not like the client) the connection.
- On acceptance a SYN is send, otherwise a RST. Note that a SYN segment occupies 1 byte of sequence space so it can be acknowledged unambiguously.

**Call collision :**

- If two hosts try to establish a connection simultaneously between the same two sockets then the events take place as shown in Fig. 6.14.2(b).
- Under such circumstances only one connection is established.
- Both the connections can not be established simultaneously because connections are identified by their end points.
- If the first set up results in a connection which is identified by (x, y) and second connection is also set up, then only one table entry will be made i.e. for (x, y).
- For the initial sequence number a clock based scheme is used, with a clock pulse coming after every 4 μsec.
- For ensuring an additional safety, when a host crashes, it may not reboot for 120 sec which is maximum packet lifetime.
- This is to make sure that no packets from previous connections are still alive and travelling around.

### 6.14.4 TCP Connection Release :

- A TCP connection is actually a full duplex connection but to understand the connection release we will assume that it is a pair of simplex connections.
- We can then think that each simplex connection is getting terminated independently.
- Releasing a TCP connection is identical on both ends. Each side can send a TCP segment with the FIN bit set, meaning it has no more data to send.
- After receiving a FIN, the Acknowledge (ACK) signal is sent and that direction is shut down, but data may continue to flow indefinitely in the other direction.
- If the sender of FIN does not receive the ACK within 2 maximum packet lifetimes, it releases the connection. The receiver will eventually notice that it receives no more data and time-out as well.
- Normally four TCP segments are required to release a connection i.e. one FIN and one ACK in each direction.





- However the first ACK and second FIN can be combined in the same segment.

**Connection reset :**

- The connection reset in TCP can take place when TCP at one end done any one of the following :
  1. It may deny a connection request.
  2. It may abort the existing connection.
  3. It may terminate an idle i.e. non operating connection.
- TCP does all the three with the help of the RST (reset flag).

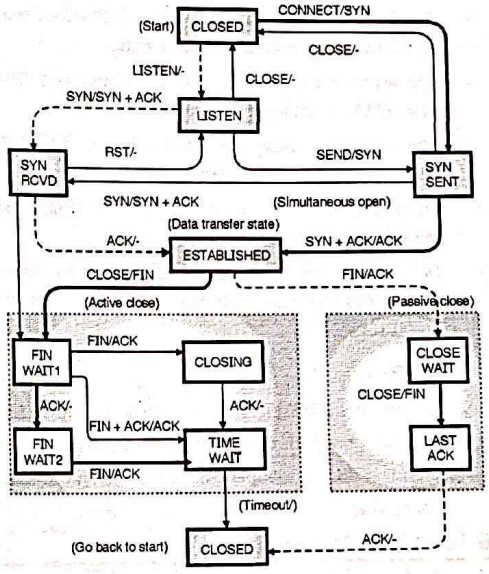
## 6.15 TCP State Transition Diagram :

- The steps to be followed in TCP connection establishment and release can be represented using a finite state machine.
- The total eleven states in such a state machine are given in Table 6.15.1.

**Table 6.15.1 : Different states in TCP finite state machine**

| State | Description |
|---|---|
| CLOSED | No connection is active or pending |
| LISTEN | The server is waiting for an incoming call |
| SYN RCVD | A connection request has arrived; wait for ACK |
| SYN SENT | The application has started to open a connection |
| ESTABLISHED | The normal data transfer state |
| FIN WAIT 1 | The application has said it is finished |
| FIN WAIT 2 | The other side has agreed to release |
| TIMED WAIT | Wait for all packets to die off |
| CLOSING | Both sides have tried to close simultaneously |
| CLOSE WAIT | The other side has initiated a release |
| LAST ACK | Wait for ack of FIN of last close |

- In each of the 11 states shown in Table 6.15.1, some specific events are considered to be legal events.
- Corresponding to every legal event some action may be taken, but if some event other than the legal one happens, then error is reported. The finite state machine is shown in Fig. 6.15.1.




**(G-696) Fig. 6.15.1 : TCP connection management final state machine**

- Each connection is always in the CLOSED state initially. It comes out of this state when it does either the passive open (LISTEN) or an active open (CONNECT).
- A connection is established, if the other side does the opposite and the state becomes ESTABLISHED.
- When both the sides initiate a connection release the connection is terminated and the state returns to CLOSED state.

**Various types of lines in the finite state machine drawing :**

- Various types of lines are used in the finite state machine drawing of Fig. 6.15.1. They have different meanings as stated below :
  1. **Heavy solid lines :** These lines show a client actively connecting to a passive server.
  2. **Heavy dotted lines :** These lines are used for the server.
  3. **The light faced lines :** These are for unusual event sequences.
- Over each line in Fig. 6.15.1 we have written the event / action pair.
- The event can either be a user-initiated system call (CONNECT, LISTEN, SEND or CLOSE), a segment arrival (SYN, FIN, ACK or RST), or a time-out.

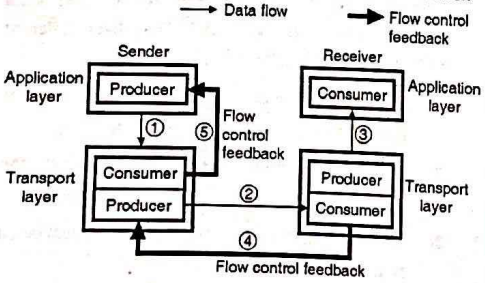
- For the TIMED WAIT state the event can only be a time-out of twice the maximum packet length.
- The action is the sending of a control segment (SYN, FIN or RST) or nothing.
- The time-outs to guard for lost packets (e.g. in the SYN SENT state) are not shown here.
- There are 11 states used in the TCP connection management finite state machine.
- Data can be send in the ESTABLISHED and the CLOSE WAIT states and received in the ESTABLISHED and FIN WAIT1 states.

**Explanation :**

- To understand the finite state machine of Fig. 6.15.1, first follow the path of a client i.e. the heavy solid line. After that follow the path of the server (the heavy dashed line).

## 6.16 Flow Control in TCP :

- The flow control is a technique used for controlling the data rate of the sender so that the receiver is not overwhelmed.
- In TCP the flow control has been kept separate from the error control.
- So when the flow control is being discussed, we will temporarily ignore the error control. i.e. we assume that the data transmission is taking place over an error free channel.
- Refer Fig. 6.16.1 which shows the data transfer taking place in only one direction from the sender to receiver.




(G-1802) **Fig. 6.16.1 : Data flow and flow control feedback in TCP**

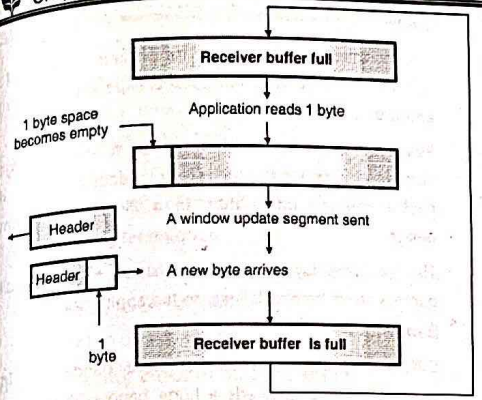
- We can apply the same principle to the bidirectional data transfer.
- Two different types of signals travel between the sending process and the receiving process in Fig. 6.16.1. They are data and flow control feedback signals.
- The data flow takes place from the sending process to the sending TCP (denoted by ①), then from sending TCP to receiving TCP (denoted by ②) and finally from receiving TCP to receiving process (denoted by ③).
- Thus flow of data takes place from sender to receiver. But the flow control feedback signals travel from the receiver to sender as shown.
- They flow from receiving TCP to sender TCP (denoted by ④) and from sending TCP to sending process (denoted by ⑤).
- Most TCP versions however, do not provide the flow control feedback facility. Instead the receiving process is allowed to pull data from receiving TCP whenever the receiving process becomes ready.
- Thus the receiving TCP controls the sending TCP (due to flow control feedback) and the sending TCP controls the sending process as far as the data rate of the sending process is concerned.
- Consider the flow control feedback path denoted by ⑤ in Fig. 6.16.1. This feedback is practically achieved by simply rejecting the data by sending TCP when its window is full.
- So now let us concentrate on the flow control feedback signal from receiving TCP to sending TCP, denoted by path ④ in Fig. 6.16.1. i.e. how does the receiving process control the sending TCP.

### 6.16.1 Silly Window Syndrome :

- This is another problem that can degrade the TCP performance.
- This problem occurs when the sender transmit data in large blocks, but an interactive application on the receiver side reads data 1 byte at a time.
- To understand this problem, refer Fig. 6.16.2.








(G-617) **Fig. 6.16.2 : Silly window syndrome**

1. Initially the receiver's buffer is full so it send a window size 0 to block the sender.
2. But the interactive application reads one byte from the buffer. So one byte space becomes empty.
3. The receiving TCP sends a window update to the sender informing that it can send 1 byte.
4. The sender send 1-new byte.
5. The buffer is full again and the window size is 0. This process can continue forever. This is known as the silly window syndrome.

### 6.16.2 Nagle's Algorithm :

- The Nagle's algorithm is very simple. It takes into account the speed of transmission of the sender and the speed of the network which is transporting the data.
- The algorithm is as follows :
  1. The first piece of data received from the sending application program is send by the sending TCP even if it is only 1 byte.
  2. Once the first segment is sent, the sending TCP will wait and accumulate data in the output buffer until either the acknowledgement is received from the receiving TCP or sufficient data is accumulated to fill the maximum size segment.
  3. Step 2 is repeated for the remaining transmission.
- If the sending application program data rate is higher than the speed of data transporting network then the segments are larger (maximum size segments).
- On the other hand if the sending application program is slower than the data transport network, the segments will be smaller than the maximum segment size.

**Clark's solution to silly window syndrome :**

- Clark suggested a solution to silly window syndrome as follows.
- He suggested that the receiver should not send a window update for 1 byte.
- Instead the receiver must wait until it has a considerable amount of buffer space available and then send the window update.
- To be specific, the receiver should wait until it can handle the maximum window size it has advertised at the time of establishing a connection or its buffer is half empty, whichever is smaller.
- The sender can also help to improve the situation.
- It should not send tiny segments. Instead it must wait and send a full segment or at least one containing half of the receivers buffer size.

## 6.17 Timers in TCP :

1. **Persistence timer :**
   - The second timer in TCP is called **persistence timer.** It is designed to solve the following problem :
     1. The receiver sends an ACK with window size = 0. So the sender will wait for the receivers buffer to have some free space.
     2. After the receiver buffer becomes partially empty it sends a window update to the sender asking it to send.
     3. But the packet containing this window update is lost on its way to sender.
     4. So both sender and receiver will be waiting for ever.
   - To solve this problem, the persistence timer is used. If it goes off, then sender transmits a probe to the receiver. The receiver sends the window size in response to this probe.
   - If the window size is still zero then the persistence timer is set again and the cycle repeats. But if the window size is nonzero then sender can send data.
2. **Keepalive timer :**
   - This is the third timer in TCP. It is used when a connection is idle for a long time. When a connection is idle for a very long time, the Keepalive timer may go off. This will cause one side to check if the other side is still there.

- If the other side does not respond, then the connection is terminated.

**3. Timer for TIMED WAIT state :**

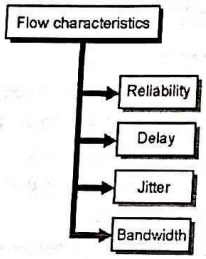
This timer is used in the TIMED WAIT state while closing. This timer is set to a time equal to twice the maximum packet lifetime to ensure that after closing a connection all the packets created by it die off.

## 6.18 Quality of Service (QoS) :

- The long form of QoS is quality of service and it is an internetworking issue. We can informally define quality of service as something flow seeks to attain.

**Flow characteristics/ QoS parameters :**

- There are four important characteristics of data flow : reliability, delay, jitter and bandwidth. These characteristics are shown in Fig. 6.18.1.




(G-480) **Fig. 6.18.1 : Flow characteristics / QoS parameters**

**1. Reliability :**

- A data flow must have some level of reliability. Lack of reliability means a packet or acknowledgment, will be lost and retransmission will be required.
- However, each application programs has a different demand for reliability.
- For example, it is more important that electronic mail, file transfer, and Internet access have reliable transmissions than telephony or audio conferencing.

**2. Delay :**

- Source-to-destination delay is another important flow characteristic.
- Again delay tolerance of different applications will be different.
- In this case, telephony, audio conferencing, video conferencing, and remote log-in need minimum delay, while file transfer or email are delay tolerant applications.

**3. Jitter :**

- Jitter is the variation in delay for packets belonging to the same flow. i.e. different packets experience different amounts of delays.
- Real-time audio and video cannot tolerate a large amount of jitter. On the other hand, it does not matter if packet carrying information in a file have different delays.
- The transport layer at the destination waits until all packets arrive before delivery to the application layer.

**4. Bandwidth :**

- Different applications need different bandwidths. In video conferencing needs a huge bandwidth whereas an email may not need a large bandwidth.

### 6.18.1 Techniques for Achieving Good QoS :

- Some of the techniques useful in achieving good QoS are as follows :
  1. Buffering
  2. Traffic shaping
  3. Leaky bucket algorithm
  4. Token bucket algorithm
  5. Resource reservation
  6. Admission control
  7. Proportional routing
  8. Packet scheduling.

### 6.18.2 Traffic Shaping :

- One of the important reason behind congestion is the bursty nature of the traffic. If the traffic has a uniform data rate then congestion problem will not be very common.
- Traffic shaping is an open loop control of congestion control. It manages the congestion by making the packet transmission rate to be more predictable.
- This will make the data rate more uniform and bursty traffic is reduced. Thus traffic shaping will regulate the average rate or the burstiness of data transmission.
- The process of monitoring a traffic flow is called as **traffic policing.** Here the principle followed is to check if a packet stream (connection) obeys the rules and if it violates the rules then, give penalty !
- For this the network would like to monitor the traffic flow during the connection period. The process of monitoring and enforcing the rules to regulate traffic flow is called **traffic policing.**





- Penalty for breaking the rules will be :
  1. Drop packets that violate the rules.
  2. Give low priority to them.

**Definition of traffic shaping :**

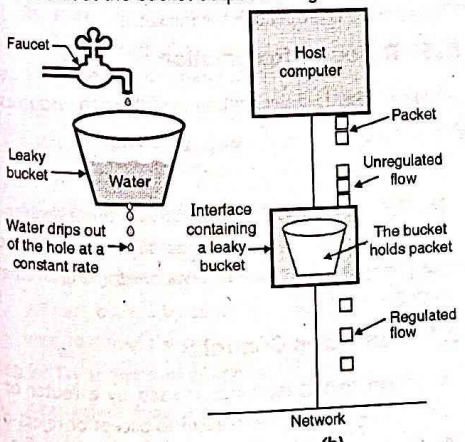
- Traffic shaping is defined as a mechanism to control the amount and rate of the traffic sent to the network.

**Traffic shaping techniques :**

- The two popularly used traffic shaping techniques are :
  1. Leaky bucket algorithm
  2. Token bucket algorithm

### 6.18.3 Leaky Bucket Algorithm :

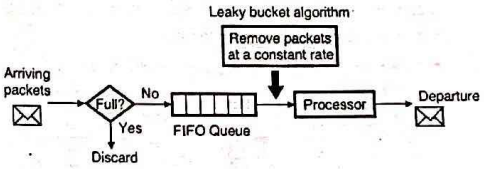
- Leaky bucket algorithm is used to control congestion in network traffic. As the name suggests it's working is similar to a leaky bucket in real life.
- The principle of leaky bucket algorithm is as follows :
- Leaky bucket is a bucket with a hole at bottom. Flow of the water from bucket is at a constant rate (data rate is constant) which is independent of water entering the bucket (incoming data).
- If bucket is full, any additional water entering in the bucket is thrown out (Packets are discarded). Same technique is applied to control congestion in network traffic.
- Every host in the network is having a buffer (equivalent to a bucket) with finite queue length. Packets which are put in the buffer when buffer is full are thrown away.
- The buffer may send some number of packets per unit time onto the subnet (helpful if packets vary greatly in size) as shown in Fig. 6.18.2.
- The data flow at the input of the bucket is unregulated but that at the bucket output is a regulated one.




(G-481) **Fig. 6.18.2 : Leaky bucket algorithm**

**Leaky bucket implementation :**

- Fig. 6.18.3 shows the implementation of leaky bucket principle.




(G-482) **Fig. 6.18.3 : Implementation of leaky bucket**

- A FIFO (First In First Out) queue is used for holding the packets which is equivalent to the leaky bucket.
- The implementation of Fig. 6.18.3 can be discussed under two different operating conditions, namely :
  1. For packets of fixed size.
  2. For packets of variable size.

**1. Fixed size packets :**

- If the arriving packets are of fixed size (e.g. cells in ATM networks), then the process of Fig. 6.18.3 will allow the removal of a fixed number of packets from the queue corresponding to every tick of the clock.

**2. Packets of variable size :**

- If the packets at the input of the process are of different size, then the fixed output rate will not correspond to the number of packets leaving the process but it will correspond to the number of bits leaving the process.

**Algorithm :**

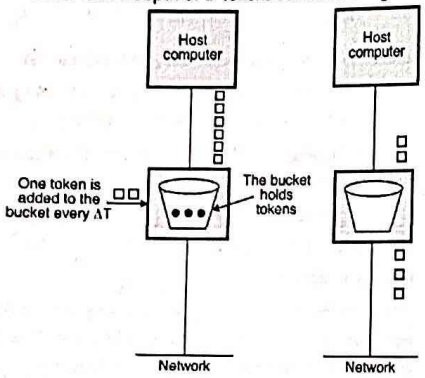
- The algorithm for variable length packets is as follows :
  1. Initialize a counter to a number "n" at the tick of the clock.
  2. If "n" is greater than the packet size, then send the packet and decrement the counter by the packet size.
  3. Repeat step 2 until "n" becomes smaller than the packet size.
  4. Reset the counter and go back to step 1.

**Note :** Thus a leaky bucket algorithm shapes the bursty traffic to convert it into a fixed rate traffic. It does so by averaging the data rate. It drops the packets if the bucket (buffer) is full.

### 6.18.4 Token Bucket Algorithm :

- This algorithm is similar to the leaky bucket but it is possible to vary output rates. This is useful when larger burst of traffic is received.

- It enforces a long-term average transmission rate while permitting bounded bursts.
- In this approach, a token bucket is used to which manages the queue regulator that ultimately controls the rate of packet flow into the network.
- A token generator continuously produces tokens at a rate of R tokens per second and puts them into a token bucket with a depth of D tokens as shown in Fig. 6.18.4.

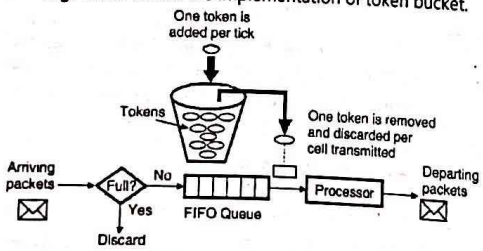



(G-483) Fig. 6.18.4 : Token bucket algorithm

- If the token bucket gets full then the extra tokens are discarded.
- Token bucket algorithm is a variant of leaky bucket algorithm discussed earlier. Here the bucket is filled with tokens. A packet which grabs and destroys a token is allowed to leave the bucket.
- Due to this mechanism, the packets never get lost but they just have to wait to grab a token.
- At the same time, an unregulated stream of packets arrive and are placed into a packet queue that has a maximum length of L.
- If the flow delivers more packets than the queue can store, the excess packets are discarded.

**Implementation of token bucket :**

- Fig. 6.18.5 shows the implementation of token bucket.




(G-484) Fig. 6.18.5 : Implementation of token bucket

- The token bucket can be easily implemented with a counter. The token is initialized to zero.
- Every time a token is added, the counter is incremented by 1 and every time a packet is dispatched, the counter is decremented by 1. If the counter contents go to zero, the host cannot send any data.

**Note :** The token bucket allows the bursty traffic at maximum possible rate.

**Token bucket performance :**

Let, s = Burst length (seconds),
c = Bucket capacity (bytes),
ρ = Token arrival rate (bytes/second),
and m = Maximum source rate (bytes/second)

**What is the duration of a maximum-rate burst through a token bucket ?**

1. Maximum bytes sent from the token bucket during a burst is, $c + \rho \cdot s$
2. Maximum bytes the source can send during a burst is, $m \cdot s$
3. Setting the two equal and solving for s,

$$s = \frac{c}{m - \rho}$$

### 6.18.5 Combination of Token Bucket and Leaky Bucket :

- The token bucket and leaky bucket techniques can be combined to obtain the following advantages :
  1. To credit an idle host 2. To regulate the traffic
- The token bucket is used first followed by the leaky bucket technique.
- The rate of leaky bucket needs to be higher than the rate of tokens dropped in the bucket.

### 6.18.6 Resource Reservation :

- The data flow is dependent on the following resources :
  1. Buffer 2. Bandwidth
  3. CPU time
- The QoS can be improved by reserving these resources. The QoS model called integrated services operates on the principle of resource reservation, for improvement in QoS.

### 6.18.7 Admission Control :

- Admission control technique is used by a router or a switch. They use this mechanism to accept or reject the data flow based on predefined parameters called flow specifications.





- Before accepting a flow for processing, a router checks the flow specifications and finds out if it is possible to take up and handle this new data flow.
- It does this by comparing its bandwidth, buffer size, CPU speed etc. with the flow specifications.

## 6.19 TCP Congestion Control :

- We have already discussed the reasons of congestion in networks and the Internet is no exception. So there are congestions occurring on Internet too.
- The network layer detects the congestion by looking at the growing queues at the routers and tries to manage it by dropping packets.
- The network layer has to give feedback to the transport layer about the possible congestion because only then the transport layer can reduce the sender's data rate.
- In the Internet, TCP plays a major role in controlling congestion. A control law called AIMD (Additive Increase Multiplicative Decrease) can be used in response to binary congestion signals received from the network.
- According to this law, in response to congestion signals the transport protocol should converge to a fair and efficient bandwidth allocation.
- TCP congestion control is based on this approach using a **window** and with a loss of packet used as the binary signal to indicate congestion.

**Principle of congestion control :**

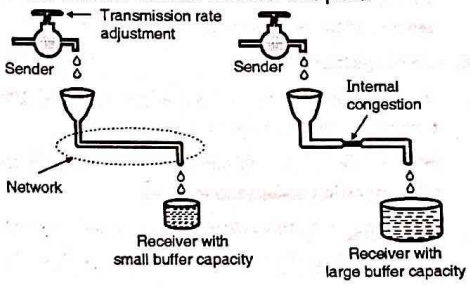
- The basic principle is do not inject a new packet into the network until an old one is delivered. TCP tries to do this by dynamically adjusting the window size.
- The steps followed in achieving the congestion control in TCP are as follows :

**Step 1 : Detect the congestion :**

- This is the first step in congestion control. Now-a-days packet loss due to transmission errors is very rare because the optical fiber links are being used.
- So most transmission time-outs (loss of packets) are due to congestions.
- So all the Internet TCP algorithms assume that time-outs are caused by congestion and so time outs can be used to detect the congestion.

**Step 2 : Try to prevent congestion :**

- After establishing a connection, a suitable window size is to be chosen. The receiver window size is based on its buffer capacity.
- If the sender adjusts its transmission rate according to this capacity as shown in Fig. 6.19.1(a), the congestion due to buffer overflow will never take place.




(G-618) Fig. 6.19.1 : Congestion

- Now consider Fig. 6.19.1(b). The sender is slow, the receiver has a large buffer capacity but the problem is low internal carrying capacity of the network.
- If the sender is too fast, the water will back up and some will be lost (loss of packets) and congestion will take place.

**Conclusion :**

- To prevent congestion TCP has to deal with two problems separately – receiver capacity and network capacity.

**Solution :**

- To deal with the two problems mentioned earlier each sender maintains two windows : the window the receiver has granted (which indicates the receiver capacity) and the **congestion window** (which indicates the network capacity).
- The first window that indicates the receiver capacity is called as the **flow control window.**
- The size of the congestion window is equal to the number of bytes the sender may have in the network at any time.
- Hence the corresponding **sending rate** is equal to the ratio of **congestion window size** and the **RTT** of the connection.
- TCP adjusts the size of window as per the AIMD rule.
- The **congestion window** is maintained in addition to the **flow control window** (Which specifies the number of bytes that the receiver can buffer).
- Both these windows are considered simultaneously.
- Both the windows indicate the number of bytes the sender may transmit and the number can be different.

- Therefore the number of bytes that may be sent by the sender is the minimum of the two windows.
- So the effective window is the minimum of what the sender and the receiver both think is all right.

**Modern congestion control :**

- Modern congestion control was added to TCP in 1988 through the efforts of Van Jacobson.
- In 1986 due to growing number of Internet users the first **congestion collapse** took place.
- As a response to this collapse Jacobson approximated an AMID congestion window and added it to the existing TCP.
- While doing so he made following two important considerations :
  1. The rate at which the acknowledgements return to the sender is approximately equal to the rate at which packets can be sent over the slowest link in the path. This is the rate a sender wants to use to avoid congestion. This timing is known as **ACK clock** and it is an essential part of TCP. Using ACK clock TCP smoothes out traffic and avoids congestion.
  2. The second consideration was that AMID rule will take a very long time to reach the desired operating point on fast networks if the congestion window is started from a small value. The start up time can be reduced by using a large initial window. But a too large starting window would cause congestion in slow or short links.
- Hence Jacobson mixed both linear and multiplicative increase in the window size in his solution to resolve congestion. This modified algorithm is known as the **slow start** algorithm.

### 6.19.1 Slow Start Algorithm :

- After establishing a connection, the sender initialises the congestion window to the size which is equal to the maximum segment in use on the connection. It then sends one maximum segment.
- If this segment is acknowledged by the receiver indicating no congestion, it adds bytes corresponding to one full segment to the congestion window. So now the congestion window size is equal to two maximum size segments. The sender then sends two segments.
- As each of these segments is acknowledged indicating that there is no congestion, the size of congestion window is increased by one maximum segment size. This is shown in Fig. 6.19.2. This is the exponential growth of the congestion window size.
- When the congestion window is of n segments, if all n segments are acknowledged before time-out takes place, the congestion window is increased by the byte count corresponding to n segments.
- But there is a limit on the exponentially growing congestion window. The congestion window stops growing as soon as either the time-out occurs or the receiver's window size is reached.
- If the congestion window can grow to 1024 (1 kbyte) byte, 2048 byte, but a burst of 4096 bytes gives a time-out then we have to set the congestion window at 2048 in order to avoid congestion.
- Once this is done, no data bursts longer than 2048 bytes will be sent by the sender even if receiver grants a wider window.
- The name of this algorithm is slow algorithm and it is required to be supported by all the TCP implementations.

### 6.19.2 Internet Congestion Control Algorithm :

- Till now only two parameters have been used namely receiver window and congestion window. But in the algorithm we are going to discuss, a third parameter called **threshold** is used.
- Initially the threshold is set to 64 kbyte. When the time-out occurs, the threshold is set to half of the current congestion window i.e. 32 k bytes and the congestion window is reset to one maximum segment.
- The slow start algorithm is then used to find what the network can handle. But most importantly the exponential growth of the congestion window is stopped as soon as it reaches the **threshold.**
- After this point (threshold point), the congestion window grows linearly (and not exponentially) by one maximum segment for each burst instead of one per segment. This is illustrated in Fig. 6.19.2.
- Table 6.19.1 is used to plot the graph of Fig. 6.19.2. See how the threshold point acts as the boundary of the exponential growth and linear growth of the congestion window.



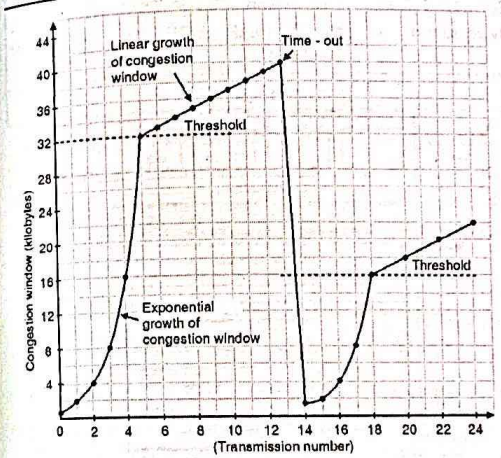


(G-619(a)) **Table 6.19.1**

Threshold — Time - out

| Transmission number | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Congestion window kilobytes | 2 | 4 | 8 | 16 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 1 |

Exponential growth — Linear growth




(G-619) **Fig. 6.19.2 : Internet congestion control algorithm**

- The maximum segment size here is 1024 i.e. 1 kbyte. Initial value of congestion window was 64 k, but time-out occurs.
- So threshold is set to 32 k and congestion window to 1 k at 0. (Original point in Fig. 6.19.2.)
- Then the congestion window grows exponentially till the congestion window size reaches the threshold of 32 k.
- The threshold occurs at 32 k and the congestion window grows linearly after this point.
- The time-out occurs as the 13th transmission. Therefore the new threshold is set to half the current window (i.e. at 16 k) and slow start is initiated again. The process will repeat thereafter.
- If no more time-outs occur, the size of congestion window continues to grow upto the size of the receiver window.

## 6.20 Comparison of UDP and TCP :

**Table 6.20.1 : Comparison of UDP and TCP**

| Characteristic / Description | UDP | TCP |
|---|---|---|
| General Description | Simple, high-speed, low-functionality "wrapper" that interfaces applications to the network layer and does little else. | Full-featured protocol that allows applications to send data reliably without worrying about network layer issues. |
| Protocol Connection Setup | Connectionless; data is sent without setup. | Connection-oriented; connection must be established prior to transmission. |
| Data Interface To Application | Message-based; data is sent in discrete packages by the application. | Stream-based; data is sent by the application with no particular structure. |
| Reliability and Acknowledgments | Unreliable, best-effort delivery without acknowledgments | Reliable delivery of messages; all data is acknowledged. |
| Retransmissions | Not performed. Application must detect lost data and retransmit if needed. | Delivery of all data is managed, and lost data is retransmitted automatically. |
| Features Provided to Manage Flow of Data | None | Flow control using sliding windows; window size adjustment heuristics; congestion avoidance algorithms. |
| Overhead | Very low | Low, but higher than UDP |
| Transmission Speed | Very high | High, but not as high as UDP |

| Characteristic / Description | UDP | TCP |
|---|---|---|
| Data Quantity Suitability | Small to moderate amounts of data (up to a few hundred bytes) | Small to very large amounts of data (up to gigabytes) |
| Types of Applications That Use The Protocol | Applications where data delivery speed matters more than completeness, where small amounts of data are sent; or where multicast/broadcast are used. | Most protocols and applications sending data that must be received reliably, including most file and message transfer protocols. |
| Well-Known Applications and Protocols | Multimedia applications, DNS, BOOTP, DHCP, TFTP, SNMP, RIP, NFS (early versions). | FTP, Telnet, SMTP, DNS, HTTP, POP, NNTP, IMAP, BGP, IRC, NFS (later versions). |
| Error control | Only checksum. | Provided. |

## 6.21 Protocols for Real Time Interactive Applications :

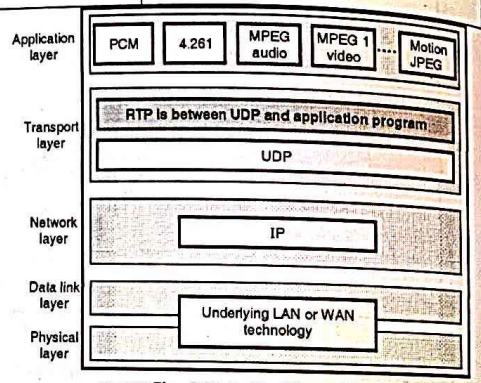
- Real time interactive applications such as Internet phone and video conferencing have become extremely popular, now a days.
- So the standard bodies such as IETF and ITU are busy in laying out standards for this class of applications.
- The protocols used for real time interactive applications are as follows :
  1. RTP (Real time protocol)
  2. SIP
  3. H.323.

### 6.21.1 RTP [Real Time Protocol] :

- RTP is the protocol designed to handle real time traffic on the Internet.
- It does not have a delivery mechanism like multicasting, port numbers and so on. Therefore we must use it with UDP.
- The position of RTP is between UDP and the application program as shown in Fig. 6.21.1.
- The main uses of RTP are as follows :
  1. Time stamping
  2. Sequencing
  3. Mixing facilities.

**Position of RTP in protocol suite :**

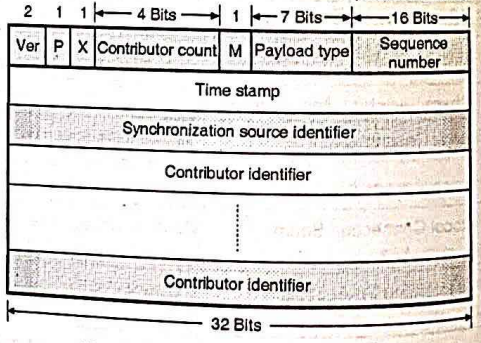
- Fig. 6.21.1 shows the position of RTP in the Internet protocol suite.
- As shown, RTP is placed in the transport layer between UDP and the application layer.




(G-742) **Fig. 6.21.1 : Position of RTP**

**RTP packet format :**

- Fig. 6.21.2 shows the format of the RTP packet header.




(G-743) **Fig. 6.21.2 : RTP packet format**

- The format is very simple and general in nature so as to cover all real time applications.
- The description of various fields is as given below :
  1. **Ver :** This 2-bit field is used to define the version number of the RTP. The current version is 2.
  2. **P :** This is a 1-bit field,

     If P = 1, then it, indicates that padding is present at the end of the packet.





If P = 0, it indicates that, there is no padding.

3. **X :** This is a 1-bit field,

   If X = 1 then it indicates the presence of an extra extension header between the basic header and the data.

   If X = 0 then, no extra extension header is present.
4. **Contributor count :** This 4-bit field indicates the number of contributors. We can have at the most 15 contributors because a 4-bit field can form combinations only between 0 and 15.
5. **M :** This 1-bit field is used as marker. The application used it to indicate, the end of its data.
6. **Payload type :** This is a 7-bit field which indicates the type of the payload. Different types of payloads have been defined so far. Some common applications are listed below :

**Payload types :**

| Type | Application | Type | Application |
|---|---|---|---|
| 0 | PCMM audio | 7 | LPC audio |
| 1 | 1016 | 8 | PCMA audio |
| 2 | G721 audio | 9 | G722 audio |
| 3 | GSM audio | 14 | MPEG audio |

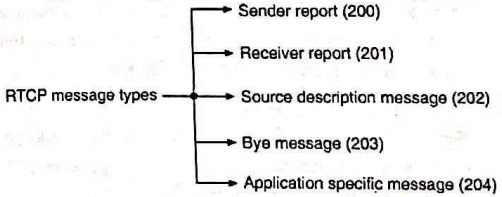
7. **Sequence number :** This is a 16-bit field. It is used to assign number to the RTP packets. The sequence number of the first packet is randomly selected and it is incremented by 1 for each subsequent packet. The sequence number is used by the receiver to detect lost or out of order packet.
8. **Time stamp :** This is a 32-bit field that indicates the time relationship between packets. The value of time stamp for the first packet is a random number. For each succeeding packets, the value of time stamp is equal to the sum of the preceding time stamp and the time the first byte is produced (sampled).
9. **Synchronization source identifier :** If there is only one source, this 32-bit field defines the source. However if there are multiple sources, the mixer is treated as the synchronization source and the other sources will be considered as contributors. The protocol provides a strategy if there is any conflict.
10. **Contributor identifier :** This is a 32-bit field. Each identifier (a maximum of 15) defines a source. When there is more than one source in a session, the mixer is treated as the synchronization source and the remaining sources are considered as the contributors.

**UDP port :**

- RTP is a transport layer protocol, but the RTP packet is not directly encapsulated in the IP datagram.
- RTP is treated like an application program and is encapsulated in a UDP user datagram. However no-well-known port is assigned to RTP. This is different as compared to other application programs.
- The port can be selected whenever required with only one restriction. The **port number must be an even number.** The next number (an odd number) is used by the companion of RTP which is the Real Time Transport Protocol (RTCP).

### 6.21.2 RTCP [ RTP Control Protocol ] :

- RTP supports only one type of message, i.e. the one that is used to send data from source to the destination. But only one message type won't always be enough.
- Sometimes additional message types are required to be used, for controlling the flow and quality of data and to permit the destination to send feedback to the source or sources.
- The protocol designed for this purpose is Real time Transport Control Protocol (RTCP). This protocol supports five types of messages as shown in Fig. 6.21.3.




(G-744) **Fig. 6.21.3 : RTCP message types**

- The number next to each message type indicates the type of the message (200, 201.....,204 in Fig. 6.21.3).

1. **Sender report :**

- All the active senders send the sender report periodically in a conference which tells about transmission and reception statistics for all RTP packets sent during the interval. The sender report includes an absolute time stamp, which helps the receiver to synchronize different RTP message.
- This is very important when both audio and video are being transmitted.

- Note that the relative time stamps used for audio and video transmission are different from each other.

**2. Receiver report :**

- The receiver report informs the sender and other receivers about the quality of service. This is for passive participants, i.e. the participants which do not send, RTP packets.

**3. Source description message :**

- Some additional information about itself like name, e-mail address, telephone number and address of the owner or controller of the source could be given by the source by periodically sending a source description message.

**4. Bye message :**

- A stream can be shut down if a source sends a bye message. By sending the bye message the source will announce that it is leaving the conference. The other sources can as it is detect the absence of a source. But this message is considered as a direct announcement. It is also very useful to a mixer.

**5. Application specific message :**

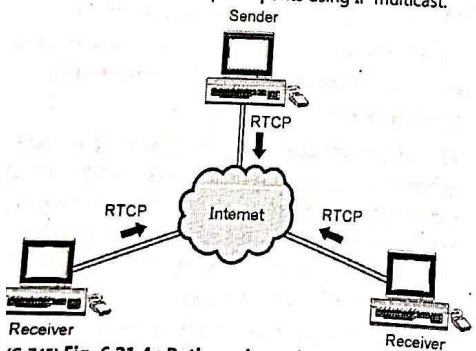
- The application specific message is a packet for the application that wants to use new applications. This message is used for defining a new message type.

**6. UDP port :**

- RTCP uses a temporary port and not a well known UDP port like RTP. The UDP port selected should have a number that immediately follows the UDP port selected for RTP i.e. it will be an odd numbered port.

### 6.21.3 RTCP Packets :

- Fig. 6.21.4 shows that in the multicast scenario the RTCP packets are transmitted by each participant in RTP session to all other participants using IP multicast.




(G-745) **Fig. 6.21.4 : Both sender and receiver send RTCP messages**

- We can distinguish between the RTP and RTCP packets by using the distinct port numbers. The RTCP port number is chosen to be equal to the RTP port number plus 1.
- The RTCP packets do not contain audio and/or video information but they contain the sender and/or receiver reports.
- They also contain the information about number of packets sent, number of packets lost and interactive jitter.

**RTCP packet types :**

- For each RTP stream received by a receiver, the receiver generates a reception report. The receiver puts all its reception reports into a single RTCP packet. We have already discussed different types of RTCP packets.
- This packet is then sent to all the participants.
- The reception report is made of many fields. Some of the important ones are as follows.

**Reception report fields :**

1. The SSRC of the RTP stream for which the reception report is being generated.
2. The fraction of packets lost within the RTP stream.
3. The last sequence number received in the stream of RTP packets.
4. The inter arrival jitter.

- For each RTP stream transmitted by the sender it creates and transmits RTCP sender report packets.
- These packets include following information about RTP stream :

1. The SSRC of the RTP stream.
2. The time stamp and wall clock time of the most recently generated RTP packet in the stream.
3. The number of packets sent in the stream.
4. The number of bytes sent in the stream.

**RTCP Bandwidth Scaling :**

- RTCP has a potential scaling problem. Consider an RTP session which consists of one sender and a large number of receivers.
- If all the receivers periodically generate RTCP packets, then the total transmission rate of RTCP packets can be much higher than the rate of RTP packets sent by the sender.
- Note that the amount of RTP traffic does not change with the change in number receivers.





- But the amount of RTCP will increase linearly with increases in the number of services. This is known as the scaling problem.
- In order to solve this scaling problem, the RTCP modifies the rate at which a participant sends RTCP packets into the multicast tree as a function of the number of participants in the session.
- Also because each participant sends control packets to everyone else, each participant can estimate the total number of participants in the session.
- RTCP attempts to limit its traffic to 5% of the session bandwidth. That means if there is only one sender, sending at a rate of say 2 Mbps, then RTCP tries to limit its traffic to 100 kbps (5% of 2 Mbps) as follows.
- The protocol RTCP gives 75% of this rate i.e. 75 kbps to the receivers and it gives the remaining 25% of the rate i.e. 25 kbps to the senders.
- The 75 kbps is devoted to the receivers is equally shared among the receivers. So if there are X number of receivers, then each one of them gets 75/X kbps.
- A participant (sender or receiver) determines the RTCP packet transmission period by dynamically calculating the average RTCP packet size and dividing the average RTCP packet size by its allocated rate.
- The period for transmitting RTCP packets for a sender is given by,

$$T = \frac{\text{Number of senders}}{0.25 \times 0.05 \times \text{Session bandwidth}} \times \text{(average RTCP packet size)}$$

- The period for transmitting RTCP packets for a receiver is,

$$T = \frac{\text{Number of receivers}}{0.75 \times 0.05 \times \text{Session bandwidth}} \times \text{(average RTCP packet size)}$$

## 6.22 Stream Control Transmission Protocol (SCTP) :

- SCTP is a new transport layer protocol. The multimedia and steam traffic is increasing day by day on the Internet. SCTP is a general purpose transport layer protocol which is designed to handle the multimedia and stream traffic.
- SCTP is a new reliable, message oriented transport layer protocol.
- The position of SCTP is between the application layer and network layer.
- It provides the interface between the application programs and the network operations.
- SCTP is designed mostly for the newly designed and recently introduced Internet applications.
- These new applications are as follows :

1. H. 323 (IP telephony)
2. SIP (IP telephony)
3. H. 248 (Media gateway control)
4. IUA (ISDN over IP)

- It is not possible to use TCP for these applications because they need more sophisticated service than that provided by TCP. SCTP is capable of providing the required services with better performance and reliability.

### 6.22.1 UDP Performance for Internet Applications :

- The following features of UDP are desirable when it is to be used for the Internet application like IP telephony.

**Desirable features :**

1. It is a message oriented protocol.
2. It conserves the boundaries of the message.
3. All UDP messages are independent of each other.

- However UDP is not suitable for the applications such as IP telephony or real time data transmission due its following undesirable features.

**Undesirable features :**

1. UDP is unreliable. So the sender does not know anything if the datagram is lost or duplicated or discarded or received out of order.
2. No congestion control.
3. No flow control.

### 6.22.2 TCP Performance for Internet Applications :

- Some of important desirable features of TCP are as follows :

**Desirable features :**

1. TCP is a byte oriented protocol.
2. TCP is a reliable protocol.
3. TCP can detect the duplicate segments.
4. In TCP the lost segments are resent.
5. Bytes are delivered in order.
6. TCP has flow control.
7. TCP has congestion control.

**Undesirable features :**

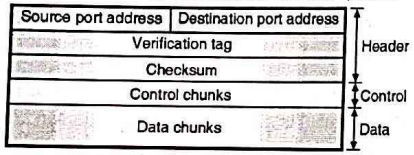
1. TCP does not preserve the message boundaries.

### 6.22.3 Features of SCTP :

- The important features of SCTP are as follows :
  1. SCTP combines all the desirable features of UDP and TCP.
  2. It is a message oriented protocol (like UDP).
  3. It is a reliable protocol (like TCP).
  4. It preserves the message boundaries (like UDP).
  5. It can detect lost, duplicate or out of order data (like TCP).
  6. It provides flow control and congestion control (like TCP).
  7. SCTP has new features that are not available in UDP and TCP.

### 6.22.4 SCTP Packets :

- The SCTP packet design is completely different than that of TCP. In SCTP, the data is carried in the form of data chunks while control information is carried as control chunks. Fig. 6.22.1 shows the SCTP packet.

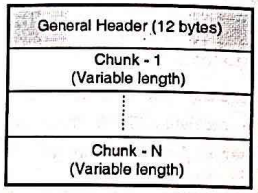



(G-2048) **Fig. 6.22.1 : An SCTP packet**

- The role of an SCTP packet is same as that of a TCP packet. In SCTP the control information is not a part of the header, but it is included in the control chunks. The control chunks are of different types.
- In SCTP the data is not treated as one entity. Instead it is in the form of several data chunks, and each chunk can belong to a different stream.
- There is no option section is SCTP like TCP. We have to define new chunk types to handle options in SCTP.
- The length of general header in SCTP is 12 bytes as compared to 20 bytes in TCP. The checksum length in SCTP is 32 bit as compared to 16 bits in TCP.
- The verification tag field in SCTP packet is used as an association identifier. Each association is defined by a unique verification tag. We can have multihoming in SCTP by using different IP addresses.
- In an SCTP packet several different data chunks will be present and each one is defined by TSN, IS and SSN. In SCTP, control information and data information are carried in separate chunks. In SCTP the TSN, IS and SSN numbers (identifiers) are used only to identify the data chunks.
- The control chunks never use these three identifier. In SCTP the data is contained in data chunks, streams and packets.
- The relationship between these three is as follows :
  1. An association may send many **packets.**
  2. Each **packet** may contain many **chunks.**
  3. These **chunks** may belong to different **streams.**

### 6.22.5 SCTP Packet Format :

- We are now going to discuss the packet format in SCTP and different types of chunks. Fig. 6.22.2 shows the SCTP packet format, which shows that an SCTP packet consists of two parts namely :
  1. A mandatory header and
  2. A set of blocks called chunks.




(G-2049) **Fig. 6.22.2 : SCTP packet format**

- In SCTP, there are two types of chunks :
  1. Data chunks and
  2. Control chunks
- An association in SCTP is controlled and maintained by the control chunks whereas the user data is carried by the data chunks.

### 6.22.6 General Header :

- The general header or packet header in SCTP is used for defining the end points of each association to which that packet belongs to.
- It also guarantees that the packet belongs to a particular association.

## 6.23 Socket Programming :

- Many network applications consist of two programs namely a client program and a server program.
- When these programs are executed a client and a server process are created which communicate with each other by reading from and writing through the sockets.
- When creating a network application, a developer has to write the code for both client and sever programs.
- There are two different types of network applications.





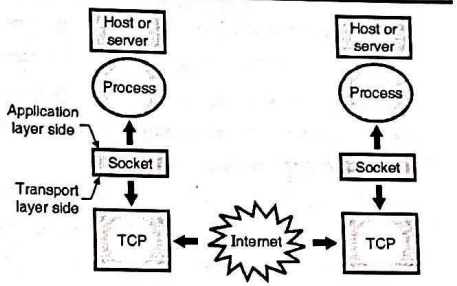
- The first type of network application is an implementation of a protocol standard defined in, for example RFC.
- For such an implementation, the client and server programs must be written as per the rules of RFC.
- It is possible for two independent developers to write the client and server programs that can operate with each other properly.
- The other type of network application is a proprietary application. In this case the application layer protocol used by the client and server programs may not conform to any existing RFC.
- A single developer or developing team writes the client and server programs.
- As the code does not implement a public domain protocol, the other independent developers can not develop code that interoperates with the application.
- So when developing a proprietary application, the developer should not use one of the well known port numbers defined in the RFCs.

**Key issues in developing proprietary application :**

- When developing a proprietary type application, the developer needs to first decide whether the application is to run over TCP or UDP.
- TCP is connection oriented and provides a reliable byte-stream channel for the data to flow between the end systems.
- The UDP is connectionless and sends data in packets between the end systems.
- But it is an unreliable protocol. These TCP and UDP applications are written in Java. It is possible to write the code in C or C++ but Java has many advantages.

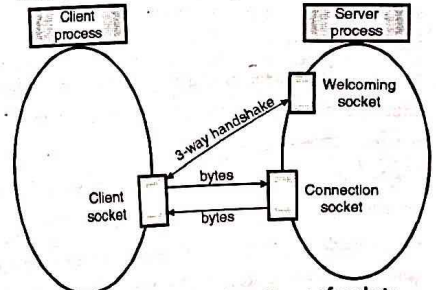
### 6.23.1 Socket Programming with TCP :

- The processes running on different machines communicate with each other by sending messages into sockets. This is demonstrated in Fig. 6.23.1.
- Socket acts as a door between the application process and TCP as shown in Fig. 6.23.1. The application developer controls everything on the application layer side of the socket but does not have any control over the transport layer side of the socket.




(G-630) **Fig. 6.23.1 : Communicate between processes through TCP sockets**

- The interaction of the client and server takes place as follows. The client has to initiate contact with the server and when such a contact is being initiated, the server should be ready.
- That means the server must be a running process (not dormant) when a client initiates contact and the server process must have a socket to welcome the initial contact from the client.
- With the server process running, the client process can initiate a TCP connection to the server.
- This is done in the client program by creating a socket.
- When the client socket is created, the client specifies the address of the server process i.e. the IP address of the server process i.e. the IP address of the server host and the port number of the server process.
- Then the TCP on the client side initiates a three way handshake and establishes a connection with the server.
- The three way handshake and the TCP connection establishment is shown in Fig. 6.23.2.




(G-1247) **Fig. 6.23.2 : Different types of sockets**

- During the three way handshake the client process knocks on the welcoming socket of the server process.
- The server process responds to this knocking by creating a new socket called **connection socket** which is dedicated to that particular client.
- In the last phase of the three way handshake a TCP connection is established between the client socket and the connection socket as shown in Fig. 6.23.2.
- The TCP connection is equivalent to a direct virtual pipe between the clients socket and server's connection socket to allow a reliable byte-stream service between the client process and server process.

### 6.23.2 Socket Programming with UDP :

- As discussed in the previous section, when two processes communicate over a TCP connection, it is equivalent to communicating over a virtual pipe between the two processes.
- This pipe will remain in place until one of the processes terminates the TCP connection. The sending process does not have to insert the destination address to the bytes to be sent because the virtual connection is existing.
- Also the pipe provides a reliable byte transfer without altering the sequence in which the bytes are received.
- Like TCP, the UDP also allows two or more processes running on different hosts to communicate. But there is a major difference.
- The first difference is that UDP provides a connectionless service so there is no handshaking process in order to establish the virtual pipe like TCP.
- As there is no virtual pipe existing, when a process wants to send a batch of bytes to the other process, the sending process has to attach the address of the destination process.

The destination address is a tuple consisting of the IP address of the destination host and the port number of the destination process.

The IP address and port number together are called as **"packet".**

UDP provides an unreliable message oriented service in which there is no guarantee that the bytes sent by the sending process will reach the destination process.

- After creating a "packet", the sending process will push the packet into the network through a socket.
- This packet is then driven in the direction of destination process.
- The code for UDP socket programming is different than that for TCP in the following ways :
  1. No need for a welcoming socket as no handshaking is needed.
  2. No streams are attached to the socket.
  3. The sending host has to create packets.
  4. The receiving process has to obtain information from each received packet.

## 6.24 Integrated Services and Differentiated Services :

- To provide the required QoS, two architectures have been proposed.
- They are as follows :
  1. The integrated services (Intserv)
  2. The differentiated services (Diffserv).
- **Intserv** is defined as a framework which is developed within the IETF for providing QoS guarantees to individual application sessions.
- The goal of **Diffserv** is to provide the ability to handle different classes of traffic in different ways within the Internet.

### 6.24.1 Intserv :

- The intserv architecture has two key features
  1. Reserved resources
  2. Call set up.

**1. Reserved resources :**

- A router must know about how much of its resources such as buffers, link bandwidth have already reserved for the current or ongoing session.

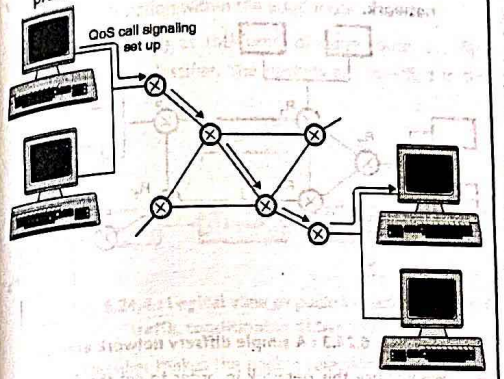
**2. Call set up :**

- A session requires certain QoS guarantee. Therefore to fulfil the requirements sufficient resources at each router should be first reserved.
- The call set up is also called as call admission. Each router has to determine the local resources required by the session and also should fined the amounts of its resources that are already occupied etc.





- Refer Fig. 6.24.1 to understand the call set up procedure.




(G-761) **Fig. 6.24.1 : The call set up procedure**

- The steps involved in call set up process are as follows :
  1. Traffic characterization and specification of desired QoS.
  2. Signaling for call set up.
  3. Pre-element call admission.

**1. Traffic characterization and specification of desired QoS :**

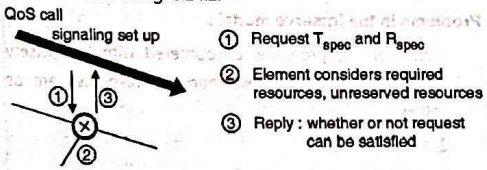
- The session has to declare its QoS requirement. Then the router can determine whether its resources are sufficient to meet these requirements or not.
- The session must also characterize the traffic that it will be sending into the network.
- In the Intserv architecture the $R_{spec}$ (R for reservation) defines the specific QoS being requested by a connection and the $T_{spec}$ (T for traffic) characterizes the traffic sent by the sender.
- Depending on the type of service requested the specific form of $R_{spec}$ and $T_{spec}$ will vary.
- The $R_{spec}$ and $T_{spec}$ are defined in part in RFC 2210 and RFC 2215.

**2. Signaling for call set up :**

- In order to reserve the resources for the session, a session's $T_{spec}$ and $R_{spec}$ should be conveyed to the router.
- The signaling protocol in the Internet is the RSVP protocol. RFC 2210 describes how to use the RSVP resource reservation protocol with the Intserv architecture.

**3. Pre-element call admission :**

- The router receives the $T_{spec}$ and $R_{spec}$ for a session. Then it can determine whether it is possible to go ahead with the call or not.
- The decision of call admission is dependent on factors like traffic specification, requested type of service, and existing resource allotment made by the router to the ongoing session.
- The pre-element call behaviour of a router has been illustrated in Fig. 6.24.2.




(G-762) **Fig. 6.24.2 : Pre-element call behaviour**

### 6.24.2 Classes of Service :

- The Intserv architecture defines two classes of services as follows :
  1. Guaranteed service. 2. Controlled-load service.

**1. Guaranteed quality of service :**

- This type of service specifications are defined in RFC 2212. It defines limits on the queuing delays experienced by a packet when it is routed through a router.
- The traffic characterization of a source are given by a leaky bucket with parameters (r, b) and the characteristics of the requested service are given by the transmission rate R at which the packets are transmitted.
- Thus a session requesting guaranteed service is expecting that the bits in its packet be transmitted at the forwarding rate of R bits / sec.
- As the traffic is specified by leaky bucket characterization and transmission rate is R, it is possible to limit the maximum queuing delay at the router.

**2. Controlled load network service :**

- The RFC 2211 states that a session receiving controlled load service, will receive a QoS which is very close the QoS received from an unloaded network element.
- That means it is assumed that a very high percentage of the packets will be successfully passed through the router without getting dropped.

- The delay in the router experienced by these packets is assumed to be equal to zero.
- However note that the controlled load service does not make any quantitative guarantees about performance.
- The controlled load service is used for real time multimedia applications on the Internet.
- That is why the performance of these applications is load dependent. Their performance is good well when the load on the network is small, but the performance degrades with increase in load.

**Problems in the Intserve model :**

- Some of the problems encountered with the Intserv model and per flow reservation of resources are as follows :
  1. Scalability
  2. Flexible service model.

**1. Scalability :**

- For per flow resource reservation, the router needs to process resource reservations and maintain the per flow state for each flow passing through the router. Per flow reservation can increase the overhead in large networks to a great extent.

**2. Flexible service models :**

The Intserv framework provides for a small number of pre specified service classes. Class A service is preferred over class B service and so on.

**Remedy :**

- The solution to these difficulties is to use the differentiated service (Diffserv) which provides a scalable and flexible service differentiation.

## 6.24.3 Differentiated Services (Diffserv) :

- Diffserv has the ability to handle different classes of traffic in different ways within the Internet.
- The scalability is needed because hundreds of thousands of source-destination traffic flows may be present at the router of the Internet.

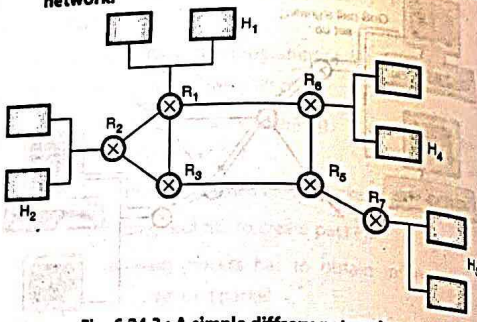
The flexibility is required because new service classes may arise and old service classes may become obsolete.

The Diffserv architecture is flexible. It does not define specific service classes.

Instead Diffserv provides the functional components. Functional components are pieces of the network architecture with which such services can be built.

**A simple Diffserv network example :**

- Refer Fig. 6.24.3 which shows a simple **diffserv network.**




(G-763) **Fig. 6.24.3 : A simple diffserv network example**

- We will use this network in order to set the framework for defining the architectural components of the Diffserv model.
- The Diffserv architecture consists of two sets of functional elements namely :

  1. Edge function 2. Core function.

**1. Edge functions : Packet classification and traffic conditioning :**

- At the input of the network the incoming packets are marked. The differentiated service (DS) field of the packet header is set to some value.
- For example in Fig. 6.24.3 packets which are sent from $H_1$ to $H_3$ travel through the routers $R_1$, $R_3$, $R_5$ and $R_7$ and they may be marked at $R_1$.
- Whereas the packets being routed from $H_2$ to $H_4$ travel through $R_2$, $R_1$ and $R_6$ and they may be marked at $R_2$.
- Such type of mark received by a packet will identify the class of traffic to which it belongs. Different classes of service will then receive different service within the core network.

**2. Core functions : Forwarding :**

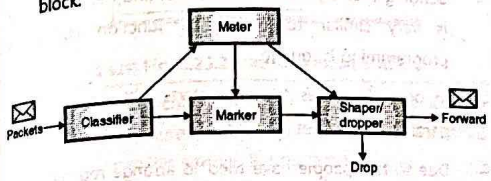
- The DS marked packet arrives at the Diffserv capable router. The router understands the class of received packet.
- The per hop behaviour of the router is associated with the packet class. The router will forward the packet onto their hop according to the per hop behaviour.
- The per hop behaviour is important because it decides how a router's buffers and link bandwidth are shared among various service classes.





**Diffserv traffic classification and conditioning :**

- Refer Fig. 6.24.4 which provides a classification and marking function within the edge router.
- Packets arriving at the input of edge router are first applied to a classifier. The packets are classified in this block.




(G-764) **Fig. 6.24.4 : Logical view of packet classification and traffic conditioning at the end**

- The classifier makes the packet selection on the basis of the values of one or more packet header fields and forwards the packet to the proper marking function.
- As long as the user is sending packets into the network without violating the negotiated traffic rules, the packets get their priority marking and they are forwarded towards the destination without any penalty.
- But if the traffic rules are violated, then the out of profile packets might be marked in a different way and as a penalty they might be shaped or might be dropped.
- The metering block in Fig. 6.24.4 has a function of comparing the incoming packet flow with the negotiated traffic profile and to find out whether the packet is following the negotiated traffic profile or not.
- The actual decision making such as remark, forward, delay or drop a packet is not the Diffserv architecture's job. Instead it is done by the network administrator.

**Per-hop behaviours :**

- The second key component of the Diffserv architecture involves the per hop behaviour (PHB) which is performed by Diffserv capable routers.
- PHB is defined as a description of the externally observable forwarding behaviour of a diffserv node applied to a particular diffserv behaviour aggregate.

## 6.25 Wireless TCP and UDP :

- Theoretically the transport protocols should be independent of the technology of the underlying network layer.
- However in practice most TCP implementations have been carefully optimized on the basis of some assumptions that are true for wired networks but not true for wireless networks.
- Therefore, they work correctly for wireless network, but the performance is not optimum.

**Problem :**

- The main problem in application of TCP on wireless networks is the congestion control algorithm. As stated earlier, almost all the TCP implementations assume that time-outs take place due to congestion and not due to lost packets.
- So when a timer times-out, the TCP assumes that congestion has taken place and slows down the transmission rate of the sender. This is to reduce the load on the network and avoid congestion.
- But the wireless links are not very reliable and they always keep losing packets. So if a packet is lost, then the sender should re-transmit it as early as possible and should not slow down.

**Conclusion :**

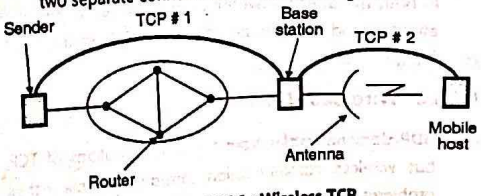
- The conclusion is when a packet is lost on the wired network, the sender should slow down. But when a packet is lost on a wireless network the sender should retransmit.
- This problem arises when the sender does not know the type of network (wired or wireless).

**Practical difficulty :**

- In practice, the path from sender to receiver is not homogeneous.
- A part of it can be over a wired network and the remaining may be wireless. Under such circumstances the decision about time-out becomes more difficult.

### 6.25.1 Solution (Indirect TCP) :

- The solution suggested for this problem is to use indirect TCP in which the TCP connection is split into two separate connections as shown in Fig. 6.25.1.




(G-627) **Fig. 6.25.1 : Wireless TCP**

- The first connection is between sender and base station and the second one is between base station and receiver which is a mobile host. The base station only copies the packets between the connections in both the directions.
- The first connection is on the wires whereas the second one is wireless. The problem of non-homogeneous connections has been solved. The first connection can respond to the time-outs by slowing down the sender where the second one will speed up the sender if time-outs take place.

### 6.25.2 Alternative Solution :

- In an alternative solution, the TCP is not split as discussed earlier but modifications are made to network layer code in the base station.
- One of the important change is that a **snooping agent is added.** It observes and caches TCP segments going out to the mobile host and the ACKs being received from it.
- The snooping agent has a timer, which is set every time a segment goes out to the mobile host. If no ACK is received before the expiry of the timer, then the snooping agent re-transmits that segment without informing the source.
- A disadvantage of this scheme is that if the wireless link is losing too many packets, then the source may time-out waiting for ACK and will go into the congestion control mode.
- This problem will not be there with indirect TCP because there the congestion control will be exercised if and only if there is congestion in the wired part of the network.
- What if segments originating from the mobile host are lost ? The solution is when the base station observes that there is a gap in the inbound sequence numbers then, it sends a request for selective repeat.
- Thus the alternative solution proposed by Balakrishnan will ensure that the wireless link becomes more reliable in both the direction without the source even knowing about it and there is no change in the semantics of TCP.

### 6.25.3 Wireless UDP :

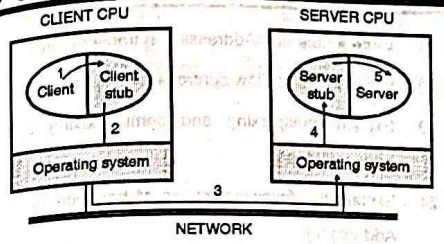
- UDP does not suffer from the same problems of TCP but wireless communication introduces some other problems in UDP.
- The main problem with UDP is that its reliability decreases to a great extent as compared to that over the wired networks.

### 6.25.4 RPC (Remote Procedure Call) :

- Sending message to a remote host and getting a reply is very similar to making a function call in a programming language.
- In both the cases the user starts with one or more parameters and get back a result.
- Due to this people have tried to arrange request reply interactions on networks in the form of procedure calls. This makes network applications very easy to program and deal with.
- For example, consider a procedure called get-IP-address (host-name) in which a UDP packet is sent to a DNS server and the reply is awaited. If timed out, it tries again. In this way all the details of networking can be hidden from the programmer.
- An extremely important work in this area was done by Birrell and Nelson.
- They suggested to allow programs to call procedures located on remote hosts.
- When a process on machine-1 calls a procedure on machine-2, then the calling process on 1 is suspended and execution of the called procedure takes place on machine-2.
- No message passing is visible to the programmer. This technique is called as **RPC (Remote Procedure Call).** Traditionally the calling procedure is known as client and the called procedure is known as the server. These names are used for RPC too.
- The principle behind RPC is to make a remote procedure call look like a local call.
- To call a remote procedure, the client program should be bound with a small library procedure called as **client stub** which represents the server procedure in the client's address space.
- Similarly a server is bound with a procedure called as the **server stub.**
- Fig. 6.25.2 shows the actual steps in making an RPC.








(G-628) **Fig. 6.25.2 : Steps in making RPC**

**Step 1 :** Client calls the client stub. This is a local procedure call and the parameters are pushed on to the stack in the normal way.

**Step 2 :** Client stub encapsulates the parameters into a message and makes a system call and sends the message. Packing the parameters into a message is called as marshaling.

**Step 3 :** The message is sent from client machine to server machine.

**Step 4 :** The received packet by the server is passed to the server stub.

**Step 5 :** Server stub calls the server procedure with the unmarshaled parameters.

- The reply from server to client follows the same path in the opposite direction.
- It is important to note that, the client procedure written by the user makes a normal (local) procedure call to the client stub (which has the same name as the server procedure).
- The client procedure and client stub are in the same address space.
- Therefore the parameters are passed in the normal manner. On the server side, for the server procedure nothing is new.
- Thus instead of I/O being done on socket, the network communication takes place by **faking** a normal procedure call.

**Problems with RPC :**

- RPC is conceptually good but has some problems. The biggest one is the use of pointer parameters. Generally passing a pointer to a procedure is not a problem.
- The called procedure can use the pointer in the same way the caller procedure uses it, because both the procedures have the same virtual address space.
- But with RPC it is not possible to pass pointers because client and server are in different address space.
- It is however possible to use tricks to pass pointers. But even this cannot pass all the pointers. (pointers pointing to a graph or other complex data structure). Hence some restrictions should be placed on parameters in case of RPC.
- The second problem is that in the languages such as C, we can write a procedure to compute the inner product of two vectors (arrays) even though the size of vectors is not specified.
- Each vector could be terminated by a special value which is known only to the calling and called procedures.
- Under these circumstances, it becomes impossible for the client stub to pack those parameters, the size of which is not known.
- The third problem is that it is not always possible to find out the types of the parameters, not even from a formal specification or code itself.
- For example **printf**, it can have any number of parameters of different types.
- Trying to call printf as a remote procedure would not be practically possible.
- The fourth disadvantage is related to the use of global variables.
- Generally the calling and called procedures can communicate by using global variables in addition to using parameters.
- But if the called procedure is present on a remote machine, then the code will fail because the global variables are not being shared anymore.
- All these problems put some limitations on the use of RPC.
- It is not compulsory that we use UDP packets for RPC but UDP is very well suited for RPC.
- But sometimes a TCP connection is required to be set up rather than using UDP.

## Review Questions

Q. 1 What do you mean by congestion control and QoS ?

Q. 2 What are the parameters of QoS ?

Q. 3 Define the term : Socket.

Q. 4 List the types of socket.

Q. 5 What are the steps used for socket programming ?

Q. 6 What are the elements of transport layer ?

Q. 7 What is difference between IP addresses and port number ?

Q. 8 What are the functions of client and server ?

Q. 9 What problems will occur in establishing a connection ?

Q. 10 What is TCP and UDP ?

Q. 11 Define threshold condition in congestion.

Q. 12 Explain the significance of listen call. Does it apply to all sockets ?

Q. 13 What parameters are specified by its various arguments.

Q. 14 Explain in detail how TCP provides flow control.

Q. 15 Define a term silly window syndrome and possible solution to overcome its effect.

Q. 16 What are the techniques used to improve QoS ?

Q. 17 What are the duties of transport layer ? Explain in brief.

Q. 18 Draw and explain the relation between network layer, transport layer and application layer.

Q. 19 What are the transport service primitives ?

Q. 20 Draw and explain the various fields of socket structure.

Q. 21 Write a note on : Addressing in transport layer.

Q. 22 Write note on : Flow control and buffering.

Q. 23 Explain multiplexing and demultiplexing used in transport layer.

Q. 24 Explain the following issues of transport protocol : Addressing.

Q. 25 Explain how you will choose between TCP and UDP ? Compare them.

Q. 26 How does TCP tackle congestion problem using the internet congestion control algorithm.

Q. 27 Explain how TCP connections are established using the three way handshake. What happens when two hosts simultaneously try to establish a connection.

Q. 28 Explain a congestion control algorithm.

Q. 29 Explain the TCP transmission policy, congestion control.

Q. 30 Explain the following issues of transport protocol :

1. Establishing a connection.
2. Releasing a connection.

Q. 31 Give the structure of UDP header.

Q. 32 Explain the TCP header and working of the TCP protocol.

Q. 33 Explain the various fields of TCP header with the help of neat diagram.

Q. 34 Explain the various steps that are followed in releasing a TCP connection.

Q. 35 Explain Token bucket algorithm.

Q. 36 Explain Leaky bucket algorithm.

Q. 37 Compare Token and Leaky bucket algorithms.

□□□

Unit V

Chapter 7

# Application Layer

## Syllabus

Client server paradigm, Peer to peer paradigm, Communication using TCP and UDP services, Domain Name System (DNS), Hypertext Transfer Protocol (HTTP), **Email :** SMTP, MIME, POP3, Webmail, FTP, TELNET, Dynamic Host Control Protocol (DHCP), Simple Network Management Protocol (SNMP).

## Chapter Contents

## 7.1 Introduction :

- Application layer is the topmost layer in the TCP/IP protocol suite.
- The hardware and software of the Internet was designed and developed for providing various types of services at the application layer.
- All the other layers (4 of them) make these services possible.
- We will discuss various services provided at the application layer first (in this chapter) and later on study the supporting role of the other layers, providing these services.
- Many application programs have been created and used during the lifetime of the Internet. Some of them could never become standards.
- Some others have become obsolete. Some have been modified, other have been replaced by new ones. But some applications have survived the test of time and have become standard applications.
- Everyday new application protocols are being added to Internet.

The Internet can provide services via two types of applications :

1. The traditional applications.
2. The new applications.

The traditional applications make use of the **client server** paradigm whereas the new applications are based on the **peer-to-peer** paradigm.

The application layer provides communication with the help of a **logical connection** which is an **imaginary connection** between the application layers of the two communicating computers. This is not the physical connection.

The actual communication however involves all the lower layer and different types of devices such as routers, switches etc.

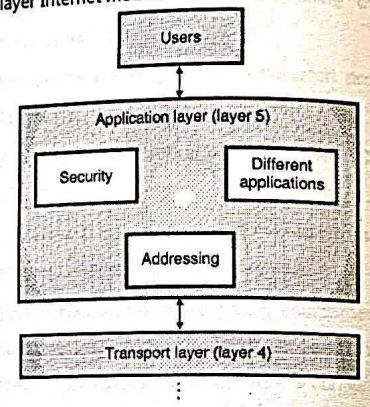
### 7.1 Position of Application Layer :

The application layer is the topmost (fifth layer) of the Internet model. This is layer where all the interesting applications are found.

People can use the Internet due to the presence of application layer.

The layers below the application layer provide reliable transport but they do not do any real work for the users.

- In other words, the other four layers are created so that people can use the various application programs. Fig. 7.1.1 shows the position of application layer in the 5-layer Internet model.




(G-629)**Fig. 7.1.1 : Position of application layer**

- The application layer provides services to the users. The users can be humans or software. It enables the user to access the network.
- The application layer receives services from the transport layer.

**Support protocols :**

- For the real applications in the application layer to function, there is a need of support protocols.
- The three areas or protocols required for such support are :
  1. Network security.
  2. Domain Name Service (DNS).
  3. Network management.
- Security is not a single protocol but it contains a large number of concepts and protocols used for providing privacy.
- DNS is used to handle naming or addressing within the Internet. The third support protocol is network management.
- In this chapter we are going to discuss some common client - server applications that are used in the Internet. Some of the important applications discussed in this chapter are : DNS, FTP, HTTP, SMTP and MIME.

## 7.2 Providing Services :

- All the communication networks which were designed to be used in the era prior to the Internet era were designed to provide a specific type of service.





- An example of such a service is the telephone service. The network for telephony was originally designed to provide only the voice service. Later on the same network was used to provide some other services such as the FAX.
- In a similar manner, the Internet also was designed for providing service to the users all over the world. But the Internet is more flexible than the other services such as postal service or telephone service, due to the layered architecture of TCP/IP suite.
- Application layer being the topmost layer in the TCP/IP suite, is slightly different from the other layers.
- The application layer protocols only take services from the other layer protocols but they do not provide any service to the protocols belonging to the other layers in TCP/IP suite.
- Therefore it is easily possible to add or remove protocols to/from the application layer. This layer is the only layer which can provide services to the Internet users.
- Due to the flexibility of the application layer, it is possible for us to add new application protocols to the Internet.

### 7.2.1 Standard and Non-standard Protocols :

- The protocols belonging to the first four layers of the TCP/IP suite have to be standardized and documented in order to ensure proper operation of the Internet.
- These protocols are generally included in the package along with an operating system such as windows or UNIX.
- However the application programs can be either standard or nonstandard, for ensuring flexibility.

**1. Standard protocols (Application layer) :**

- In our day to day life, we use several application layer programs for our interaction with the Internet. These programs are standardized and well documented by the Internet authorities.
- Each standard protocol is in the form of a pair of computer programs.
- These programs have been designed to interact with the user and the transport layer so as to provide a specific service to the user.

**2. Nonstandard protocols (Application layer) :**

- By writing two programs which can interact with a user and the transport layer to provide a specific service to the user, any programmer can create a nonstandard application layer program.
- The creation of a nonstandard protocol does not need any approval of the Internet authorities if it is used privately.
- The Internet has become so popular because of these nonstandard application layer protocols.

## 7.3 Application Layer Paradigms :

- During the life time of the Internet, three different paradigms have been developed.
- They are as follows :
  1. Client-server paradigm.
  2. Peer to peer paradigm.
  3. Mixed Paradigm.

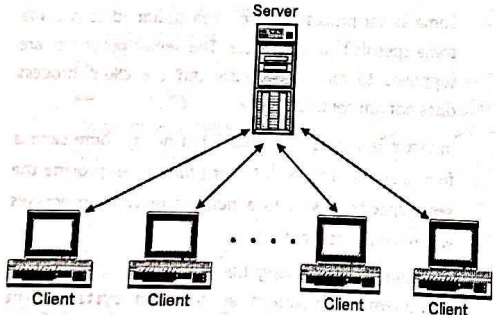
### 7.3.1 Traditional Paradigm : Client Server :

- The **client-server paradigm** is a traditional application layer paradigm which was the most popular paradigm until a few years ago.

**Server and client processes :**

- An application program called as the server process is basically the service provider in this paradigm.
- The server process runs continuously and waits for another application program called as the **client process** to make a connection through the Internet to ask for a service.
- Some server processes have been designed to provide some specific type of services. The server processes are supposed to run continuously but the client process does not run continuously.
- In fact it is started when the client needs some service from a server process. A server process can provide the same specific service to a number of client processes which request for that service.
- In computer networking the computers connected to the Internet are known as the **end systems.** The examples of end systems are as follows :
  1. Desktop computers
  2. PCs
  3. Workstations
  4. Household applications
  5. Web TVs and set top boxes
  6. Digital cameras etc.

- The end systems are also known as **hosts** because they run application programs such as Web browser program, or a Web server program etc.
- Hosts can be of two different categories as follows :
  1. Client 2. Server
- In client-server network relationships, some computers act as server and other act as clients.
- A **server** is a computer, which makes the network resources available to other computers when they request it.
- It also provides some services to them. A **client** is the computer running a program that requests the service from a server.
- Local Area Networking (LAN) uses the client-server network relationship for its operation.
- You can construct a client server network by using one or more powerful computers as a servers and the remaining computers as clients.
- Client-server network typically uses a directory service to store information about the network and its users.
- All available network resources such as files, directories, applications and shared devices, are centrally managed and hosted by the server and then are accessed by client in a client-server network.
- Fig. 7.3.1 shows client-server network relationship. The server provides security and administration of the network.




**(G-41) Fig. 7.3.1 : Client server network relationship**

- In client-server networks the processing tasks are divided between clients and servers. .
- Clients request services such as file storage and printing and servers deliver them.

**Applications :**

- The following traditional services are still using the client server paradigm for their operation :
  1. WWW : World wide web
  2. HTTP : Hyper Text Transfer Protocol
  3. FTP : File Transfer Protocol
  4. email
- Some of these protocols have been discussed in this chapter.

### 7.3.2 New Paradigm : Peer-to-Peer (P2P) :

- In response to the needs of some new applications on the Internet, a new paradigm called as peer to peer paradigm has emerged in recent days.
- It is also known as the **P2P** paradigm. Here the continuously running server process is not needed. Instead the responsibility of the server process is shared by the **peers.**
- Most of the Internet applications available today operate on the client-server paradigm. But gradually the **peer-to-peer (P2P) paradigm** also has gained some importance.
- The principle of P2P paradigm is that two peers (laptops, desktops or mainframes) can exchange services by communicating directly with each other.
- If the file requested by a client to server is a large file such as a music or video file, then it puts a lot of load on the server machine.
- In such situations the **P2P** paradigm becomes attractive. The P2P paradigm is also attractive in a situation in which two peers want to exchange files without involving the server.
- However it should be noted that the **P2P** paradigm does not ignore the client-server paradigm completely. Instead the P2P allows some users to share the duty of the server.
- Instead of sharing of a big file using client-server connection, the P2P paradigm will let the server download a part of that file and then share it among themselves.
- Thus in P2P paradigm the same computer has to sometimes behave like a client and at some other time like a server.
- In other words, the same computer will be a client for some applications for certain amount of time and server at other times.
- However such applications are not a part of the Internet, but they are controlled commercially.
- In P2P paradigm any computer connected to the Internet can provide service as well as request for a service.





- That means it can work as a **server** at one time and as a **client** at some other time.
- One of the best examples of Internet application in which the P2P paradigm is used is **Internet.**

**Telephony :**

- Another situation in which the P2P paradigm can be more useful is when one Internet users wants to share something (a file for example) with another Internet user.

**Advantages :**

- The main advantages of P2P paradigm are as follows :
  1. It is easily scalable.
  2. It is cost effective because an expensive server need not be used.

**Disadvantages :**

- Along with the above stated advantages, there are some drawbacks of P2P paradigm :
  1. Providing a secured communication is difficult.
  2. This paradigm cannot be used by all the Internet applications.

**Applications :**

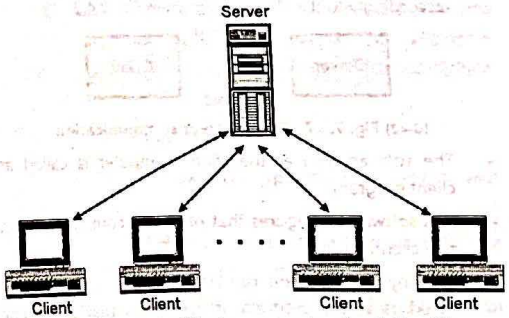
- The following Internet applications use the P2P paradigm :
  1. Skype
  2. Internet telephony
  3. IPTV.

### 7.3.3 Mixed Paradigm :

- In order to get the benefits of both the paradigms, some applications may try to use the mixture of the two paradigms.

## 7.4 Client Server Paradigm :

- In computer networking the computers connected to the Internet are known as the **end systems.**
- The examples of end systems are as follows :
  1. Desktop computers
  2. PCs
  3. Workstations
  4. Household applications
  5. Web TVs and set top boxes
  6. Digital cameras etc.
- The end systems are also known as **hosts** because they run application programs such as Web browser program, or a Web server program etc.
- Hosts can be of two different categories as follows :
  1. Client 2. Server
- In client-server network relationships, some computers act as server and other act as clients. A **server** is a computer, that makes the network resources available to other computers when they request it. It also provides some services to them. A **client** is the computer running a program that requests the service from a server.
- Local Area Networking (LAN) uses the client-server network relationship for its operation. You can construct a client server network by using one or more powerful computers as a servers and the remaining computers as clients. Client-server network typically uses a directory service to store information about the network and its users.
- All available network resources such as files, directories, applications and shared devices, are centrally managed and hosted by the server and then are accessed by client in a client-server network.
- Fig. 7.4.1 shows client-server network relationship. The server provides security and administration of the network.




**(G-41) Fig. 7.4.1 : Client server network relationship**

- In client-server networks the processing tasks are divided between clients and servers. Clients request services such as file storage and printing and servers deliver them.

**Client :**

- The individual workstations in the network are called as the clients. A client can also be a mobile PC, PDA and so on.

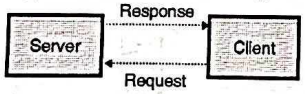
- In short we can define a **client** as a program running on the local machine which requests some services from the server.
- It is said that the client program is a **finite** program. We have discussed it later on in this chapter.

**Server :**

- The central computer which is more powerful than the clients and which allows the clients to access its softwares and database is called as the server.
- Server computers typically are more powerful than client computers or are optimised to function as servers.
- No user can access the resources of the servers until he has been authenticated (permitted) by the server to do so.
- Generally we can define as **server** as a program which is running on the remote server computer to provide service to all the clients.
- It only initiates a service when requested by that client computer.
- The server program is called as an **infinite** program. We have discussed the reason for it later on in this chapter.

**Communication in client-server configuration :**

- Fig. 7.4.2 explains the principle of communication in the client server configuration.
- The client places a request on the server machine when he wants an access to the centralised resources.
- The server responds to this request and sends the signal accordingly to the client as shown in Fig. 7.4.2.




(G-42) **Fig. 7.4.2 : Client/server communication**

- The software run at the client computer is called as client program.
- This software configures that particular computer to act as a client.
- Similarly the software run on the server computer is called as server program. It configures that particular computer to act as a server.
- A server program when started, will run infinitely unless it faces some problem. Therefore it is called as an **infinite** program.
- A server program waits for incoming requests from clients. On receiving a request, it will respond to the request in one of the following way :
  1. Iteratively 2. Concurrently
- A client program will be started by the user and gets automatically terminated when the service is complete. Therefore it is called as the **finite** program.
- Generally the communication with the server is initiated by the client by using the IP address of the remote machine and the well known port address of the specific server program which running on that machine.
- The request respond process in client - server communication can get repeated multiple times. But still this process will eventually come to an end.
- Therefore it is called as the finite process.

### 7.4.1 Concurrency :

- We will discuss the concurrency in the client as well as server because both can operate in the concurrent mode.

#### 7.4.1.1 Concurrency in Clients :

- It is possible to run the clients on a machine either in the iterative mode or in the concurrent mode.
- The operation of clients in the **iterative** mode involves running them (clients) one by one. That means one client must first complete the cycle of start, run and terminate after which the machine can start running some other client. But this method is too time consuming.
- Therefore the computers now a days make use of the **concurrent** clients where the machine can run two or more clients simultaneously. This would save time to a great extent.

#### 7.4.1.2 Concurrency in Servers :

- The severs can be of two types : namely **iterative** servers or **concurrent** servers. The iterative server is capable of processing only one request at a time.
- That means the server should first complete the cycle of receiving a request, processing it and sending the response to the requesting client, after which it can handle another request.
- However the **concurrent** server can handle many requests simultaneously on the time sharing basis.

**Protocols used :**

- The protocols that are commonly used by the servers are UDP, TCP or SCTP. Therefore the operation depends upon the following two factors :
  1. The transport layer protocol.
  2. The service method.



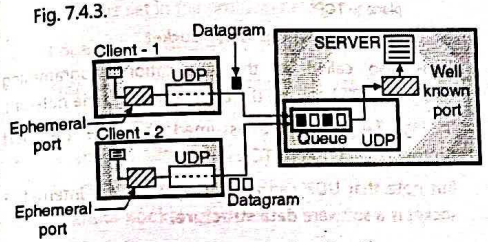


### 7.4.2 Types of Servers :

- We can classify the servers into following four types :
  1. Connectionless iterative (uses DDP).
  2. Connectionless concurrent.
  3. Connection - oriented iterative.
  4. Connection - oriented concurrent (Uses TCP/SCTP).
- Let us discuss them one by one.

#### 7.4.2.1 Connectionless Iterative Server :

- This type of server processes one request at a time, and it uses UDP for communication.
- The principle of operation of this server is as shown in Fig. 7.4.3.

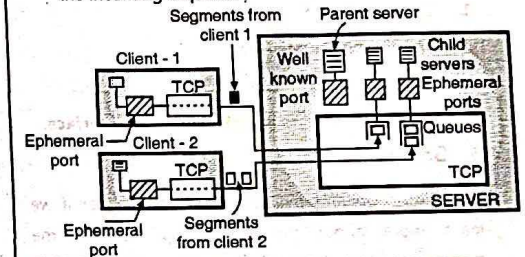



(G-1987) **Fig. 7.4.3 : Connectionless iterative server**

- As shown, a server receives a request through UDP and it sends its response back through UDP.
- The server pays attention to only one datagram at a time and ignores all others. These datagrams are stored in a queue as shown in Fig. 7.4.3 and wait for their turn to come. These datagrams could be from different clients or from the same client. But irrespective of that, the datagrams will be processed one by one.
- The server uses only one well known port for this purpose.

#### 7.4.2.2 Connection - Oriented Concurrent Server :

- The concurrent connection - oriented servers generally use TCP or SCTP. This type of server can serve many clients simultaneously.
- The communication between the server and clients is connection oriented type. Therefore the first step will be to establish a connection between the server and each client.
- This connection will remain open as long as all the data stream is not processed. After processing the entire stream, the connection is terminated.
- In this type of communication, each connection between a client and server needs a separate port and many such connections are open simultaneously. Hence this type of server cannot use only one port.
- So many ports are required to be used. But a server can use only one well-known port.
- In order to solve this problem, they use **only one** well known port and multiple **ephemeral** ports as shown in Fig. 7.4.4. The server uses a will known port to accept the incoming requests.




(G-1988) **Fig. 7.4.4 : Connection oriented concurrent server**

- Once the connection is established with the requesting client, the server assigns a temporary port to this connection and makes the well known port free.
- Now the transfer of data can take place between the temporary port at a client and the temporary port at the server.
- Another client can now make a request at the well known port which has been freed.
- As shown in Fig. 7.4.4, the server creates **child processes** which are copies of the original **(parent)** process, in order to serve many clients simultaneously.
- Also observe in Fig. 7.4.4, that a separate queue has been assigned to each connection. The segments received from the client are stored in the appropriate queues.

### 7.4.3 Socket Interface :

- The question is how a client process communicate with a server process ?
- A computer does this by executing a program which is a set of sequentially written instructions.
- A computer program consists of different sets of instructions for mathematical operations, string manipulations, input / output access etc.
- If it is necessary for a program on one machine to communicate with another program running on some other machine, **then a set of new instructions should be written** in order to instruct the transport layer to establish a connection, complete the bidirectional data transfer and terminate the connection.

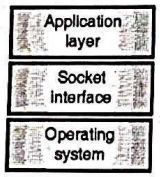
- Such a set of new instructions designed for interaction between two entities is called as **Interface.**

### 7.4.4 Types of Interface :

- Three different interfaces that have been designed for communication are :
  1. Socket interface.
  2. Transport layer interface.
  3. STREAM.
- Out of these we will discuss only the **socket interface.**

### 7.4.5 Socket Interface :

- We can understand the socket interface better if we learn more about the relationship between the operating system (Unix, Windows etc.) and the TCP/IP protocol suite.
- Refer Fig. 7.4.5 to understand the conceptual relationship between an operating system and TCP/IP suite.



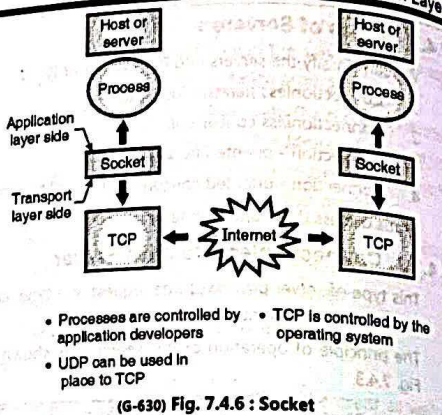

**(G-1989) Fig. 7.4.5 : Relation between TCP/IP suite and operating system**

**Definition :**

- We may define the **socket interface** as the set of instructions which helps an application access the services provided by the TCP/IP protocol suite.
- It is located between the application program and operating system.

### 7.4.6 Socket :

- In most applications there exists a pair of communicating processes.
- They send messages to each other. These messages must travel the underlying network.
- The sending process sends messages into the network through its **socket** and receiving process receives messages from the network through its socket as shown in Fig. 7.4.6.
- Thus **socket** is defined as an interface between the application layer and the transport layer within a host.

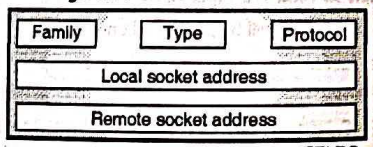



**(G-630) Fig. 7.4.6 : Socket**

- It is also called as the Application Programming Interface (API) between the application and the network.
- In Fig. 7.4.6 we have assumed that the transport protocol being used is TCP.
- But note that UDP can also be used. In the Internet, a socket is a software **data structure.**

**Data structure :**

- A socket is defined with the help of the format of data structure. This format of data structure is dependent on the language used by the processes.
- In C language the socket is defined as a five field structure. It is also called as **struct** or **record** and is as shown in Fig. 7.4.7.




**(G-1990) Fig. 7.4.7 : Socket data structure in C**

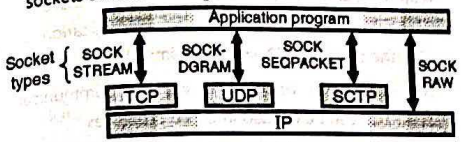
- It is important to understand that a programmer is not supposed to modify this structure, which is already defined as shown in Fig. 7.4.7.
- The programmer is supposed to only use the header file which contains the definition of the structure.
- The five fields defined in the data structure are as follows :

1. **Family :**
   - This field in the data structure will define the protocol group e.g. IPv4, IPv6, UNIX domain protocols etc.
   - The family type used for TCP/IP can be defined by the constant IF_INET for IPv4 and IF_INET for IPv6 protocols.





2. **Type :**
   - This field in the data structure will define four types of sockets as shown in Fig. 7.4.8.




**(G-1991) Fig. 7.4.8 : Socket types**

3. **Protocol :**
   - This field in the data structure is used to define the protocol which uses the interface. For TCP/IP protocol the value set in the protocol field is 0.
4. **Local socket address :**
   - This field in the data structure is used for defining the local socket address.
   - A socket address is obtained by combining an IP address and a port number.
5. **Remote socket address :**
   - This field is used for defining the remote socket address.

**Functions :**

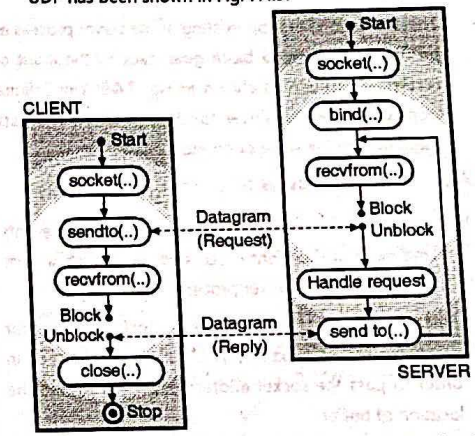
- A list of predefined functions is used to facilitate the interaction between a process and the operating system.
- These functions are combined to create processes.
- Various important functions are as follows :
  1. The socket function.
  2. The bind function.
  3. The connect function
  4. The listen function.
  5. The accept function.
  6. The fork function.
  7. The send and receive function.
  8. The sendto and recvfrom function.
  9. The close function.
  10. Byte ordering functions.
  11. Memory Management functions.
  12. Address conversion functions.

**Header files :**

- We need to use the header files in order to use the functions stated earlier.
- This header file is defined in a separate file named as **headerfiles.h.**
- This file will then be included in the program so as to avoid inclusion of long lists of header files.
- All these header files may not be needed in all the programs but still they are recommended to be included.

### 7.4.7 Communication using UDP :

- A simplified flow diagram for the communication using UDP has been shown in Fig. 7.4.9.




**(G-1992) Fig. 7.4.9 : Connectionless iterative communication using UDP**

- It diagrammatically illustrates the client - server communication with the help of UDP.
- The connectionless iterative communication using UDP can be divided into two processes :
  1. Server process 2. Client process.

### 7.4.8 Server Process :

- Refer Fig. 7.4.9. The server process will start first. The first step is that the server process calls the **socket** function for creating a socket.
- Then the server process calls the **bind** function which binds the socket to its well known port, and also to the IP address of computer which is running the server process.
- Next the **recvfrom** function is called by the server process which blocks the process until it receives the **request** datagram from the client.

- When the request datagram is received by the server process, the **recvfrom** function will unblock the server process, extracts the client socket address and address length from the request datagram, and retunes this information to the server process.
- The server process saves this information and calls a procedure or function in order to handle the request received from the client process.
- After readying the results it will call the function **sendto** and sends results to the requesting client process by making the use of the saved information.
- There is an infinite loop existing at the server process as the output of **sendto** block goes back to the input of **recvfrom** function as shown in Fig. 7.4.9. This infinite loop is used by the server to respond to the requests originated by the same or different clients.

### 7.4.9 Client Process :

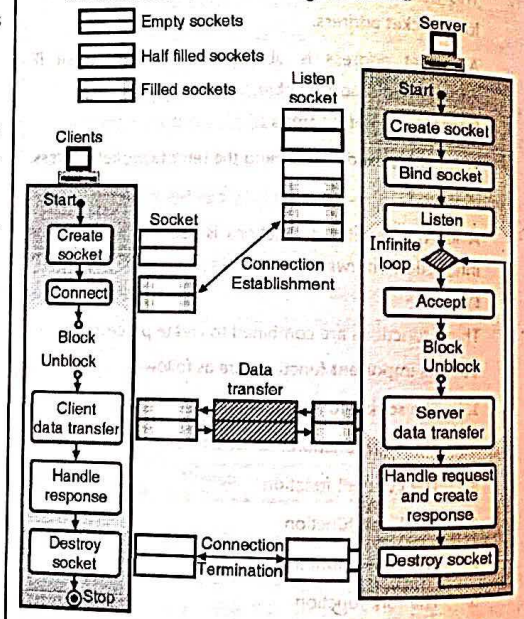
- Refer Fig. 7.4.9 to understand the sequence of events taking place at the client process which is much simpler as compared to the server process.
- First of all the client process calls the **socket** function for creating a socket. Next step is to call **sendto** function in order to pass the socket address of the server and the location of buffer.
- The UDP is supposed to take the data from this buffer and make the datagrams.
- As the next step, the client process calls the **recvfrom** function which blocks the client process until it receives the **response** message from the server process.
- As soon as the **response** from the server is received, the UDP delivers the received data to the client process.
- Due to this the **recv** function unblocks and delivers the received data to the client process.
- For all this discussion it was assumed that the client message is very small which can fit into one single datagram.
- But if the client message is long then we have to repeat the two functions **sendto** and **recvfrom.**
- But the server process is not aware of the multiple datagrams sent by the same client for the same communication. So it handles each request as an independent one.

## 7.5 Communication using TCP :

- After discussing the connectionless iterative communication using UDP, now let us discuss the connection oriented concurrent communication using TCP (the case of SCTP would be similar).
- The connection oriented concurrent communication using TCP can be divided into two processes :
  1. Server process 2. Client process.
- The flow diagram for TCP based communication is as shown in Fig. 7.5.1.

### 7.5.1 Server Process :

- Refer Fig. 7.5.1, in which it is the server process that starts first. The first step is that the server process calls the **socket** function for creating the socket.




**(G-2203) Fig. 7.5.1 : Flow diagram of TCP based connection oriented concurrent communication**

- This socket is called as **listen socket** because it is going to be used only during the establishment of connection.
- As the next step, the server process calls the **bind** function, which will bind this connection to the socket address of the server computer.
- The server function then calls the **accept** function which is basically a **blocking** function. It will block the server process until the TCP receives a connection request (SYN segment) from a client.





- On receiving the **request** from the client process, the **accept** function is unblocked and a new socket is created which is called as the **connect** socket, which carries the socket address of the requesting client.
- As soon as the **accept** function is unblocked the server understands that the requesting client needs some service.
- The server process or parent process now calls the **fork** function, in order to provide the concurrency.
- The **fork** function will create a **child process** which is a new process and it is an exact replica of the **parent process.**
- Thus after we call the **fork** function, two processes are running simultaneously (parent and child) i.e. concurrently, but each one is capable of handling different things.
- Each process now has two sockets namely **listen** and **connect** sockets. Now the parent process will handover the task of serving the existingly served client to the **child process** and calls the **accept** function again to wait for the request from another client, because the newly created child process is now ready to serve the client who is already being served and make the parent process.
- The child process will first close the **listen socket** and then call the **recv** function so as to receive data from the client.
- The **recv** function is very similar to the **recvfrom** function is basically a blocking function. It is blocked upto the instant when a segment is received.
- As shown in Fig. 7.5.1, the child process uses a loop and keeps calling the **recv** function repeatedly as long as it does receive all the segments sent by the client.
- All this data is then given to a function called as **handle Request** by the child process for handling the request and send the results to the requesting client.
- The **send** function is then called for sending the results to the client.
- All the discussion till now was based upon certain assumption. They were as follows :
  1. We have used the simplest possible flow diagrams.
  2. The size of data sent to client is very small and we can send it in just one call of the **send** function.
  3. Third assumption was that the TCP is able to send the entire client data in one segment.
- Actually if the data sent to the client is not small, the server may have to call the **send** function repeatedly.
- In reality TCP may require several segments to send the client data.
- Therefore the client may not receive all the data it requested for in one segment.

### 7.5.2 The Client Process :

- Refer Fig. 7.5.1 to understand the client process. As per the flow graph given there, the first step the client takes is it calls the **socket** function for creating a socket.
- Next it (client process) calls the **connect** function for making a request to connect to the server.
- As we know, the **connect** function is a blocking function. It will remain blocked until the connection is established between client TCP and server TCP.
- On the return of **connect** function, the client calls the **send** function for sending data to the server.
- The server may need to call the **send** function only once or several times depending on the size of data. For calling the **send** function repeatedly, we have to use a loop in the flow diagram.
- Next the client calls the **recv** function. This function will remain blocked as long as the first segment of data does not arrive.
- The server may be able to send all the data by calling the **send** function only once but the TCP may not be able to send it using only one segment.
- Therefore the **recv** function may have to be called repeatedly by the client process to receive all the data.

### 7.5.3 Peer to Peer Paradigm :

- Most of the Internet applications available today operate on the client-server paradigm. But gradually the **peer-to-peer (P2P) paradigm** also has gained some importance.
- The principle of P2P paradigm is that two peers (laptops, desktops or mainframes) can exchange services by communicating directly with each other.
- If the file requested by a client to server is a large file such as a music or video file, then it puts a lot of load on the server machine.
- In such situations the **P2P** paradigm becomes attractive. The P2P paradigm is also attractive in a situation in which two peers want to exchange files without involving the server.

- However it should be noted that the **P2P** paradigm does not ignore the client-server paradigm completely. Instead the P2P allows some users to share the duty of the server.
- Instead of sharing of a big file using client-server connection, the P2P paradigm will let the server download a part of that file and then share it among themselves.
- Thus in P2P paradigm the same computer has to sometimes behave like a client and at some other time like a server.
- In other words, the same computer will be a client for some applications for certain amount of time and server at other times. However such applications are not a part of the Internet, but they are controlled commercially.

### 7.5.4 P2P File Sharing :

- P2P means process to process file sharing.
- The P2P file sharing is the most important Internet application because the highest amount of Internet traffic, corresponds to the P2P file sharing.
- Modern P2P file sharing system shares MP3 (3 to 8 M bytes), videos (10 to 1,000 M bytes), images, software documents etc.
- In this section, we will discuss the protocols and networking issues in P2P file sharing.
- Before going into details of P2P file sharing system, let us take an example. Suppose **Rahul** uses the P2P file sharing application for MP3 downloading. He runs the P2P file sharing software on his home PC (peer). He uses an ADSL connection to access the Internet. He shuts down his PC every night and does not have a hostname. So everytime he connects to the Internet the ISP will assign a new IP address to his PC.
- Suppose that Rahul is connected to the Internet and searching for the MP3 for a particular song of a particular artist.
- As soon as he goes into search, the P2P application displays a list of those peers who are currently connected to the Internet and have a copy of that song, for sharing.
- Each one of them is an ordinary PC owned by an ordinary Internet user like Rahul.
- Rahul then requests the required MP3 file from one of the peers say Preeti's PC. Then a direct TCP connection gets established between Rahul and Preeti's PC and the MP3 file is sent from Preeti's PC to Rahul's PC.
- If Preeti disconnects her PC from the Internet in the middle of this download, then Rahul's P2P file sharing software may attempt the remaining part of the MP3 file from the other peer.
- Also when the download from Preeti to Rahul is going on, some other user can download some other song from Rahul's PC.
- Thus the P2P file sharing allows direct sharing of information without any independent server getting involved. However P2P file sharing operates on the client server principle. The requesting person acts as a client and the requested user acts as the server. The file is sent using the File Transfer Protocol (FTP).
- In P2P file sharing system, typically a large number of users are connected to Internet and each user has objects such as MP3, videos, software and images for sharing.

## 7.6 Domain Name System (DNS) :

**Definition of DNS :**

- Domain name system (DNS) is a system that translates domain names into IP addresses. The DNS servers are Internet's equivalent of a phone book. They maintain the directory of domain names and translate them into IP addresses

**Addressing :**

- For communication to take place successfully, the sender and receiver both should have addresses and they should be known to each other.
- The addressing in application program is different from that in the other layers.
- Each program will have its own address format. For example an e-mail address is like abc@vsnl.net whereas the address to access a web page is like http :// www.google.com/
- It is important to note that there is an alias name for the address of remote host. The application program uses an alias name instead of an IP address.
- This type of address is very convenient for the human beings to remember and use. But it is not suitable for the IP protocol.
- So the alias address has to be mapped to the IP address. For this an application program needs service of another entity.
- This entity is an application program called DNS. Note that DNS is not used directly by the user. It is used by another application programs for carrying out the mapping.





### 7.6.1 How does DNS Work ?

- To map a name onto an IP address, an application program calls a library procedure called the **resolver.** The name is passed on to the resolver as a parameter.
- The resolver sends a UDP packet to a local DNS server which looks up the name and returns the corresponding IP address to the resolver.
- The resolver then sends this address to the caller. Then the program can establish a TCP connection with the destination or sends in the UDP packets.

### 7.6.2 Name Space :

- The names assigned to machines should be selected carefully from the name space.
- There should be a complete control over the relation between the names and the IP addresses.
- The names and corresponding addresses are uniquely defined. A name space maps each address to a unique name.
- It can be arranged in two different ways :
  1. Flat name space.
  2. Hierarchical name space.

### 7.6.3 Flat Name Space :

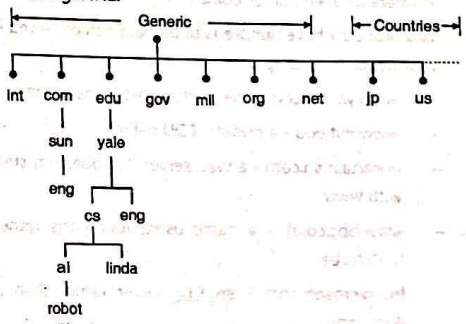
- In a flat name space, a name is assigned to every address. This type of name is simply the sequence of characters.
- That means it does not have any structure. The flat name space is not suitable for large systems like Internet, because there can be ambiguity and / or duplication.

### 7.6.4 Hierarchical Name Space :

- In the hierarchical name space, each name is made of many parts.
- The first part may correspond to the name of an institution, the second part may define the department and so on.
- The part that defines the nature of institution and name of institution is assigned by a central authority.
- The responsibility of deciding the rest of the name can be given to that institute itself.
- That institute can add suffix or prefix to the name for defining its host or resources.

## 7.7 Domain Name Space :

- Conceptually the Internet has been divided into hundreds of top level domains. Each domain covers many hosts.
- Each domain is divided into several sub domains and they are further partitioned and so on.
- These domains can be represented by a tree as shown in Fig. 7.7.1.

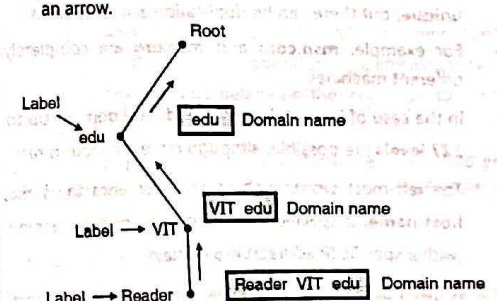



(G-631)**Fig. 7.7.1 : A portion of Internet domain name space**

- The top level domains are of two types namely generic and countries.

**Generic domains :**

- The generic domains are com (commercial), edu (educational institutions), gov (government), int (some international organizations), mil (military), net (network providers) and org (nonprofit organizations).
- The country domains include one entry for every country.
- Each domain is named by following an upward path. The components are separated by dots e.g. **eng.sun.com.** This is called hierarchical naming.
- Another example of hierarchical naming is shown in Fig. 7.7.2. The upward followed path has been shown by an arrow.




(G-632) **Fig. 7.7.2 : Domain names, labels and hierarchical naming**

**Label :**

- Each node in the tree has a label (or component) and it can be specified using up to 63 characters.
- If we had to remember the IP addresses of all of the Web sites we visit every day, we would all go nuts.
- Human beings just are not that good at remembering strings of numbers.
- We are good at remembering words, however, and that is where domain names come in.
- You probably have hundreds of domain names stored in your head. For example :
  - www.yahoo.com - the world's best-known name
  - www.mit.edu - a popular EDU name
  - encarta.msn.com - a Web server that does not start with www
  - www.bbc.co.uk - a name using four parts rather than three
  - ftp.microsoft.com - an **FTP** server rather than a Web server
- The COM, EDU and UK portions of these domain names are called the **top-level domain** or **first-level domain.**
- There are several hundred top-level domain names, including COM, EDU, GOV, MIL, NET, ORG and INT, as well as unique **two-letter combinations for every country.**
- Within every top-level domain there is a huge list of **second-level domains.** For example, in the COM first-level domain, you have got :
  - yahoo
  - msn
  - microsoft
  - plus millions of others.
- Every name in the COM top-level domain must be **unique,** but there can be duplication across domains.
- For example, **msn.com** and **msn.org** are completely different machines.
- In the case of bbc.co.uk, it is a third-level domain. Up to **127 levels** are possible, although more than four is rare.
- The left-most word, such as **www** or **encarta**, is the **host name.** It specifies the name of a specific machine (with a specific IP address) in a domain.
- A given domain can potentially contain millions of host names as long as they are all unique within that domain.

**Absolute and relative domain names :**

- Domain names can be of two types : absolute or relative.
- An absolute domain name always ends with a dot (or period as it was called). For example eng. sun. com. But the relative domain does not end with a dot.

**Are domain names case sensitive ?**

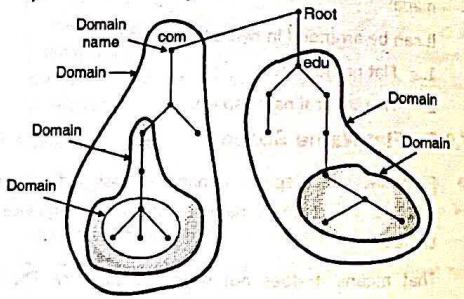
- No they are not case sensitive. So com and COM means the same thing.

**How many characters ?**

- Component names can have upto 63 characters and the full path name can at the most have 255 characters. Each domain controls how it allocates the domain under it.
- To create a new domain we have to take a permission of the domain in which it is to be included.

**Domain :**

- A domain can be defined as a subtree of the DNS name space as shown in Fig. 7.7.3.




**(G-633) Fig. 7.7.3 : Domains**

- The name of the domain is the domain name of the node at the top of the subtree as shown in Fig. 7.7.3. e.g. com or edu.
- A domain can be divided into subdomains as shown in Fig. 7.7.3. Note that the naming follows organizational boundaries, not physical networks.
- That means even if two different departments are located in the same building, they can have distinct domains.
- But the computers belonging to the same department kept in two different buildings will not have different domains.

## 7.8 Distribution of Name Space :

- The information contained in the domain name should be stored.

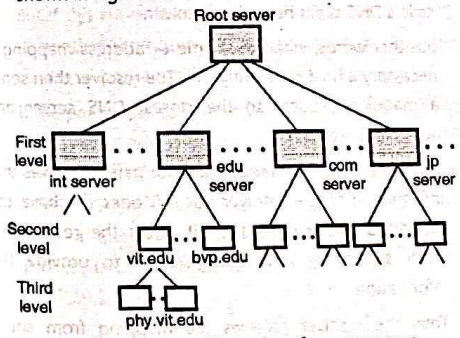




- But this is a huge information and if we store it on one computer then the system would be highly inefficient and unreliable.
- It will be an inefficient system because the system will be heavily loaded by the requests coming from all over the world.
- It will be unreliable because failure of one computer will make the data inaccessible. If we make a distributed name space then all these problems can be overcome.

### 7.8.1 Hierarchy of Name Servers :

- Name server contains the DNS database i.e. the various names and their corresponding IP addresses. Theoretically a single name server could contain the entire DNS database.
- But practically to store such a huge information at one place is inefficient and unreliable. Such a server will be soon overloaded and be useless and worst thing is if it ever goes down the entire Internet will go down.
- The solution to this problem is to distribute the information among many computers called **DNS servers.**
- Then we have to use a hierarchy of the Name servers as shown in Fig. 7.8.1.

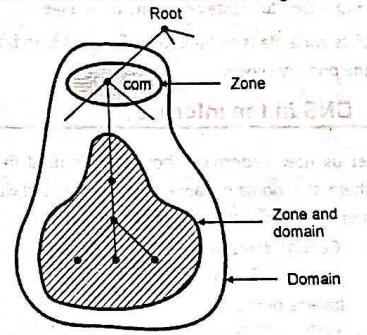



**(G-634)Fig. 7.8.1 : Hierarchy of name servers**

- First the whole space is divided into many first level domains.
- The root server stands alone and can create as many first level domains as required.
- The first level domains are further divided into smaller sub domains called second level domains. They can be further divided as shown in Fig. 7.8.1.
- Each server can be responsible (authoritative) to either a large or small domain. Note that the hierarchy of servers is similar to the hierarchy of names.
- The whole DNS name space is divided up into non overlapping **zones.** The concept of zones is as explained below.

**Zones :**

- With a number of DNS servers being used instead of a single one, we have to define the area over which each server has an authority. What a server is responsible for or has authority over is called as a zone.
- If a server is appointed for a domain and the domain is not further divided into sub-domains then the domain and zone will be the same as shown in Fig. 7.8.2.




**(G-635)Fig. 7.8.2 : Domains and zones**

- The server makes a database called a zone file. It keeps all information about every node under that zone.
- But if a server divides its domains into sub domains and delegates a part of its authority to other servers then domain and zone will be different from each other. This is shown in Fig. 7.8.2.
- The information about the nodes that belong to the sub domains is stored in the servers at the lower levels.
- The higher level and original server keeps some sort of reference of these lower level servers.

**Root server :**

- A root server is defined as a server whose zone consists of the whole DNS tree. It does not store any information about domains but delegates the authority to other servers.
- It only keeps the reference of these servers. There are more than 13 root servers and they are distributed all around the world.

**Primary and secondary servers :**

- DNS defines two types of servers namely the primary servers and the secondary servers.

**Primary server :**

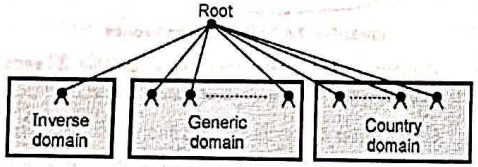
- It is a server which stores a file about its zone. It is authorized to create, maintain and update the zone file. It stores the zone file on a local disk.

**Secondary server :**

- This server transfers complete information about a zone from another server which may be primary or secondary server.
- The transferred information is saved on the disc storage of the secondary server. The secondary server is not authorized to create or update a zone file.
- If its zone file is to be updated, then it is to be done by the primary server.

## 7.9 DNS in the Internet :

- Let us now understand how DNS is used in Internet where the domain name space (tree) is divided into three different sections as shown in Fig. 7.9.1.
  1. Generic domain
  2. Country domain
  3. Inverse domain.




(G-636)**Fig. 7.9.1 : Use of DNS in Internet**

### 7.9.1 Generic Domains :

- The registered hosts are defined in the generic domains according to their generic behaviour e.g. com for commercial organizations.
- The first level in the generic domains section allows 14 possible lables. Some of them are given in Table 7.9.1.

**Table 7.9.1 : Generic domain lables**

| Lable | Description |
|---|---|
| aero | Airline or aerospace related companies. |
| com | Commercial organizations. |
| coop | Cooperative business organizations. |
| edu | Educational institutions. |
| gov | Government institutions. |
| int | International organizations. |
| mil | Military organization. |
| net | Network support centers. |
| org | Non-profit organizations. |

### 7.9.2 Country Domain :

- This domain section uses two character country abbreviations eg. US for united states. Second lable in this domain can specify organization or national designations.

### 7.9.3 Inverse Domain :

- The inverse domain is used for mapping an address to a name. This is exactly the opposite process discussed so far in which a name is mapped onto the address.

## 7.10 Name Address Resolution :

- The process of mapping a name to an address or vice versa is called as name address resolution.

**Resolver :**

- DNS application is based on the client server model. If a host wants to map a name to address or vice versa it calls a DNS client named as resolver.
- In other words, when the name ↔ address mapping is necessary a host calls a resolver. The resolver then sends a mapping request to the closest DNS server and accesses its storage.
- If this server has the requested information, it gives that information to the resolver but if it does not have the requested information, then it refers the resolver to other servers or asks other servers to provide the information.
- Thus the resolver receives the mapping from some source.
- It then checks for errors and if found error free delivers the mapping to the requesting process.

**Mapping names to addresses :**

- Generally the resolver gives a domain name to the server and requests for the corresponding IP address. The server checks the generic or country domains to get the corresponding address.
- If the domain name is from the generic domain section then the resolver receives a domain name such as,

xxx.yyy.zzz.edu





- The query is sent to the local DNS server for resolution by the resolver. If the local server does not get the answer then, it will refer the resolver to other servers or asks them directly.
- The same procedure is followed for a name from country domain.

**Mapping addresses to names :**

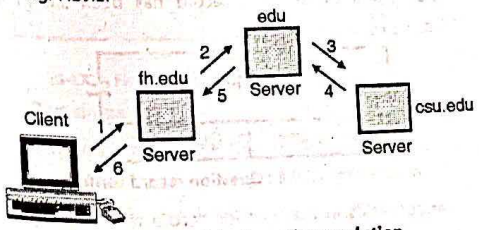
- Here, a client sends an IP address to a server and requests for its name. This type of query is called as PTR query.
- To answer the PTR query, the DNS uses the inverse domain.
- If the IP address is 142.36.48.118 then the resolver first inverts the address and adds two labels "in_addr" and "arpa" to it. So the domain name sent is :

  118.48.36.142.in_addr.arpa.

- This is received by the local DNS and resolved.

### 7.10.1 Recursive Resolution :

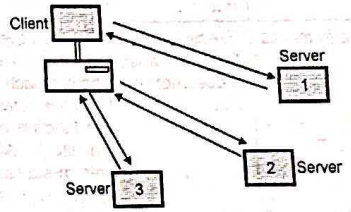
- Sometimes a client (resolver) requests for recursive or final answer from a name server. If this server is authorized for the domain name, it checks its database and sends a reply.
- But if this server is not authorized it diverts this request to another server (usually the parent server) and waits for the response.
- If the parent has the authority, then it sends the answer, otherwise it diverts the query to another server. When the query is solved, the response is returned back to the requesting client.
- Such a query is called as recursive query and the process is called recursive resolution. It is illustrated in Fig. 7.10.1.




(G-637)**Fig. 7.10.1 : Recursive resolution**

### 7.10.2 Iterative Resolution :

- This type of mapping can be done if the client does not ask for recursive answer. In iterative resolution, if the server has authority for the name it will send the answer.
- But if it does not have the authority then it returns to the client the IP address of the server that holds the answer to the query.
- The client has to repeat the query to this new server. If this server also cannot answer the query then it sends the IP address of another server to the client.
- Now the client should send the query to this third server.
- This process is called as iterative resolution because client sends the same query to different servers.
- Fig. 7.10.2 illustrates the iterative resolution.




(G-638)**Fig. 7.10.2 : Iterative resolution**

**DNS examples :**

- The DNS system is a **database**, and no other database on the planet gets this many requests.
- No other database on the planet has millions of people changing it every day, either. That is what makes the DNS system so unique!

**For example :**

- www.yahoo.com - the world's best-known name
- www.mit.edu - a popular EDU name
- encarta.msn.com - a Web server that does not start with www
- www.bbc.co.uk - a name using four parts rather than three
- ftp.microsoft.com - an FTP server rather than a Web server
- www.spce.ac.in - Server in India 'in' domain.
- The COM, EDU and UK portions of these domain names are called the **top-level domain** or **first-level domain.**
- There are several hundred top-level domain names, including COM, EDU, GOV, MIL, NET, ORG and INT, as well as unique two-letter combinations for every country.

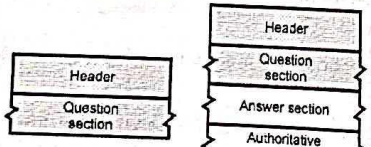
**Comparison of Iterative Resolution and Recursive Resolution :**

**Table 7.10.1 : Comparison of Iterative Resolution and Recursive Resolution**

| Sr. No. | Parameter | Iterative resolution | Recursive resolution |
|---|---|---|---|
| 1. | Definition | Iteration refer to a situation where some statements are executed again and again using loops until some condition is true | Recursion referes to a situation where a function calls itself again and again until some base condition is not reached. |
| 2. | Performance | Execution is faster because it does not use stack. | Comparatively slower because before each function call the current state of function is stored in stack. After the return statement the previous function state is again restored from stack. |
| 3. | Memory | As it does not use stack memory usage is less. | Memory usage is more as stack is used to store the current information |
| 4. | Code size | Bigger | Smaller |

## 7.10.3 The DNS Message Format :

- DNS has two types of messages as follows and both of them have the same format.
  1. Query 2. Responses or reply
- The formats of the two DNS messages are as shown in Fig. 7.10.3.







**(a) Query** **(b) Response or reply**

**(G-639) Fig. 7.10.3**

- Both query and reply messages have the same header format with some fields set to zero for query messages.
- The header is 12 byte long. The header format for both the types of messages is shown by shaded portions in Fig. 7.10.3.

## 7.10.4 Caching :

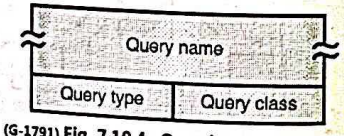
- Every time a query is asked, the server has to spend time in searching the corresponding IP address.
- If this searching time is reduced then efficiency would go up. The searching time can be reduced by using a technique called caching.
- When a server asks for a mapping from another server and receives the response, it stores this information in its cache memory before sending it to the client.
- If the same or other client request for the same mapping, it can check its cache memory and resolve the problem at its own level. This will certainly save a lot of time.
- But the problem with caching is that, if a server caches (stores) a mapping for a long time then the mapping may get outdated and the client will not get the latest mapping.
- This problem can be solved by adding the time to live information (TTL) to the mapping and each server is asked to keep a TTL counter for each mapping in its cache.

## 7.10.5 DNS Records :

- There are two types of records used in DNS as follows :
  1. Question records and
  2. Resource records
- The question records are used in the query and response messages whereas the resource records are used in the response messages.

**Question records :**

- The client uses the question record to get the required information from the server.
- The format of question record has been shown in Fig. 7.10.4.




**(G-1791) Fig. 7.10.4 : Question record format**

- Various fields in the question record format are query types and query class.

**Query name :**

- This field has a variable length and it contains a domain name. The count field tells us how many characters are present in each section.





**Query type :**

- This field is 16-bit long and it defines the type of query.
- Some of the commonly used query types are A, NS, CNAME, SOA, ANY etc.

**Query class :**

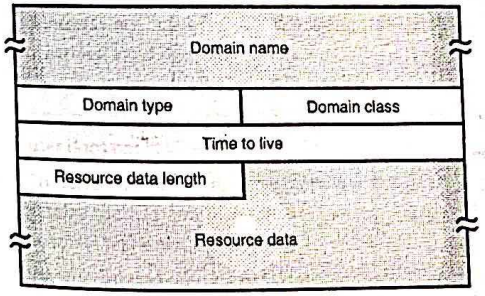
- This field also is 16-bit long. It defines the specific protocol using DNS.
- Table 7.10.1 has listed some of possible classes. However the most important class would be IN i.e. internet (class 1).

**Table 7.10.1 : Query classes**

| Class | Mnemonic | Explanation |
|---|---|---|
| 1. | IN | Internet |
| 2. | CSNET | CSNET network (Not used now) |
| 3. | CS | COAS network |
| 4. | HS | The Hesiod server (MIT) |

**Resource record :**

- Each domain name i.e each node on the tree in DNS is associated with the resource record which is a part of the server database.
- Resource records are returned by the server to the client. The format of RR has been shown in Fig. 7.10.5.




**(G-1792) Fig. 7.10.5 : Format of resource record**

1. **Domain name :**
   - This field contains the domain name and its length is not fixed.
   - It has a variable length. The domain name in the question record is duplicated here.
2. **Domain type :**
   - This field and the query type field in the question record are the same except the last two types i.e. AXFR and ANY are not allowed.
3. **Domain class :**
   - This field is same as the query type field in the question record.
4. **Time to - live :**
   - This field is 32 bit long and it defines the time for which the answer is valid (in seconds). If the contents of this field is zero, then it indicates that the resource record is used only in a single transaction.
5. **Resource data length :**
   - This field is 16 bit long. It is used for defining the length of the resource data.
6. **Resource data :**
   - As shown in Fig. 7.10.5 the resource data field is a variable length field. It contains the answer to the query or domain name or the additional information.
   - The format and contents of this field depend on the value of the type field. It can be one of the following :
     1. A number
     2. A domain name
     3. An offset pointer
     4. A character string.

**Encapsulation :**

- DNS can use either TCP or UDP. It may choose any one of these protocol but in either case the server uses port 53.
- The UPD is preferred if the length of response message is upto 512 bytes whereas TCP is used if the message length is larger than 512 bytes.

**Registrars :**

- New domains are added to DNS through a registrar, which is a commercial entity.
- Whenever an organization applies for DNS domain name, a registrar first checks that the requested domain name is unique.

## 7.11 World Wide Web (WWW) :

- People have become aware of the power of Internet through WWW. HTTP is a file transfer protocol which is specifically designed to facilitate access to the WWW.
- The World Wide Web is an architectural framework for accessing documents which are spread out over a number of machines over Internet.
- It has a colourful graphical interface which is easy for the beginners to use. It provides information on almost every subject. The web (also known as WWW) began in 1989 at CERN the European center for nuclear research.
- The web was designed basically to connect scientists stationed all over the world. The web is basically a client-server system.

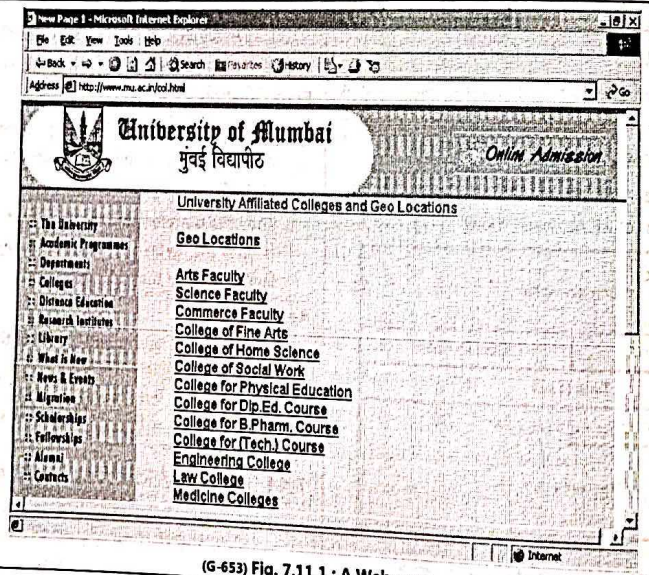
- The web pages are written in the languages HTML and Java. The growth of the World-Wide Web (WWW or simply Web) today is simply phenomenal.
- Each day, thousands of more people join the Internet (above 100 million users at recent estimates).
- Easy retrieval of electronic information along with the multimedia capabilities of Web browsers (like Mosaic or Netscape) are the factors responsible for this explosion.
- This topic provides some basic information behind some of this technology used in accessing the World-Wide Web.

**Difference between Web and Internet :**

- The Web and the Internet are not the same thing. The Web is a collection of standard protocols or instructions, sent back and forth over the Internet to gain access to information.
- The Internet, on the other hand, is a "network of networks" -- a more physical entity.

### 7.11.1 Web from the Users Side :

- The user (client) looks at the web as a collection of vast worldwide collection of documents called **pages** in short.
- **Links or pointers :** Each page may contain links or pointers to it, related pages, anywhere in the world. A user can follow a link by clicking on it.
- This will take him to the pages pointed by the links. This process can be repeated indefinitely.
- **Hypertext :** Pages which point to the other pages are said to use hypertext.
- **Browser :** The program used for viewing pages is called as a browser.
- The job of a browser is to fetch the page requested by the user, interprets the text and formatting commands which it contains, display the page with proper format on the screen. An example of a web page is shown in Fig. 7.11.1.




(G-653) Fig. 7.11.1 : A Web page

- A web page starts with a title and contains the following :
  1. Some information
  2. Strings of text, linked to other pages
  3. E-mail address of the page's maintainer.

**Hyperlinks :**

- Strings of text that are links to other pages are called hyperlinks.
- They are high lightened by underlining, using special colour or both.
- In order to follow a link, the user has to place the cursor on the high lightened area using the mouse or arrow keys and select it by clicking the mouse or pressing the ENTER key.
- The browsers can be of two types, namely the graphical browsers and non-graphical browsers.





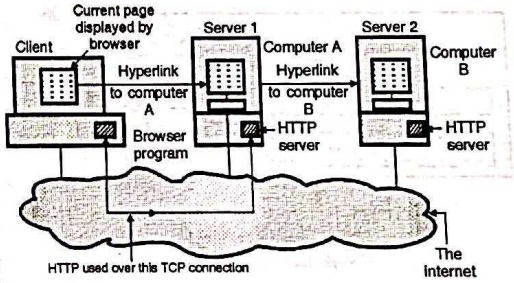
- But the graphical browsers are more popular. Voice based browsers are also being developed.
- Most browsers have a large number of buttons and features which make the navigation on web easier. There can be a button to back to the previous page or a button for going forward to the next page.
- Some browsers can provide a facility of having a button or menu item to set a bookmark on a given page and another one to display the list of bookmarks.
- This makes it possible to revisit any of them with a single click on mouse.
- It is also possible to save pages or print them. Lot of options are available to control the screen layout and setting various preferences of the users.
- The web pages can also contain line drawings, icons, maps, photographs etc and they can be linked (if required) to another page.

**Hypermedia :**

- All pages may not be viewable in the conventional way because some pages may contain audio tracks, video clips or both.
- If the hypertext pages are mixed with other media, the result of such a mixing is called as hypermedia. Some browsers are capable of displaying all kinds of hypermedia but others cannot do so.
- Many web pages contain large images that take a long time to load. When the images are being loaded, the user does not have anything to see.
- To solve this problem, some browsers first fetch and display the text and then get the images. The user can read the text when images are getting loaded.
- Another strategy can be to provide an option to disable the automatic fetching and displaying of images.
- One more alternative opted by some page writers is to display the full image in a coarse resolution and then to fill up the details gradually.
- Some web pages display forms requesting the user to fill up information.
- This is meant for searching a database for a user supplied item or ordering a product etc.
- Some web pages contain maps which allow the users to click on them to get the zooming facility or get information about the clicked geographical area.
- For hosting a web browser a machine should be directly connected to the Internet or at least have a SLIP or PPP connection to a router or other machine which is directly connected to Internet.
- This is because of the manner in which the browser fetches a page. To fetch a page it has to establish a TCP connection to the machine from where the page is to be fetched.

### 7.11.2 Web from the Servers Side :

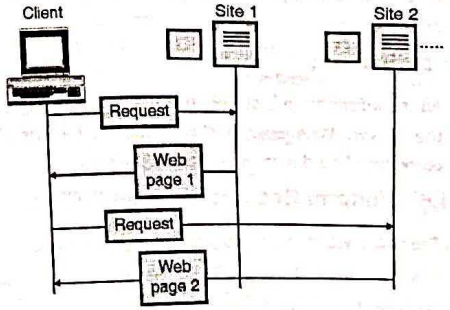
- Every website has a server process. It is listening to TCP port 80 on which incoming clients (browsers) are connected.
- Once a connection is established, the client sends a request and the server sends a reply for that. Then the connection is released.
- The protocol used for defining the legal request and replies is called HTTP. Fig. 7.11.2 shows various parts of the web model.




(G-654) Fig. 7.11.2 : Web model

### 7.11.3 WWW Architecture :

- The WWW is a distributed client/server service. A client (user) uses a browser to access a service using a server.
- But the service provided is distributed over a number of separate locations called as sites. Fig. 7.11.3 shows the architecture of WWW.

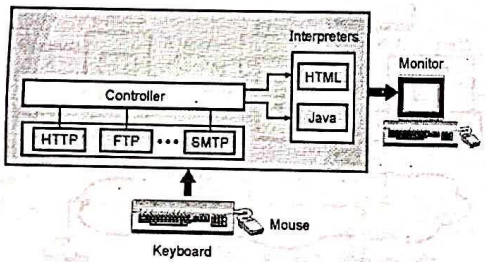



(G-655) Fig. 7.11.3 : WWW architecture

- As shown in Fig. 7.11.3, there are number of sites and each site holds a number of web pages. These pages can be retrieved and viewed by using browsers.
- The client sends a request through its browser to get a web document from a particular site. This request contains the site address and web page address (called URL) along with some other information.
- The server at the requested website finds the document and sends it to the client.

### 7.11.4 Browser (Web Client) :

- Even though a number of browsers are available around, the browser architecture is nearly the same for all of them.
- Each browser consists of the following parts :
  1. A controller
  2. Client programs
  3. Interpreters.
- Fig. 7.11.4 shows the general architecture of a browser.




(G-665) **Fig. 7.11.4 : Browser architecture**

- The controller receives input from the keyboard or mouse. It then uses the client programs like HTTP, FTP etc to access the document.
- After accessing the document, the controller makes use of an interpreter such as HTML or Java (depending on type of document) and displays the accessed document on the screen.

### 7.11.5 Server :

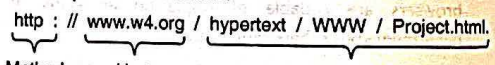
- All the information is stored in the form of web pages at the server. Whenever a client requests for one the corresponding document is sent to the client.

### 7.11.6 Uniform Resource Locator (URL) :

- The client accessing a web page needs an address.
- The HTTP uses the URL to facilitate the access of any document distributed over the world.
- The URL specifies any information on Internet by using four thing as shown in Fig. 7.11.5(a).


(G-660) **Fig. 7.11.5(a) : URL**

- They are as follows :
  1. Method or protocol
  2. Host computer
  3. Port
  4. Path.
- Method is the protocol used such as FTP, HTTP which helps retrieving the desired information. Host is the computer where the required information is located.
- The name of the computer begins with www but this is not mandatory. URL can optionally contain the server's port number.
- If the port is to be included then it should be inserted between host and path and it should be separated by a colon, as shown in Fig. 7.11.5(a).
- Path is the name of the file where the information is located. The port and path fields are separated from each other by a slash.
- **Version :** The latest version of HTTP is 1.1 but the versions 0.9 and 1 are also used.
- The example of URL is shown in Fig. 7.11.5(b). Note that the port is not included.




(G-1969) **Fig. 7.11.5(b) : Example of URL**

### 7.11.7 Cookies : User-Server Interaction :

- We know that the HTTP servers are stateless. The disadvantage of being stateless is that the server cannot identify the client.
- The meaning of statelessness is that the client server relationship gets over as soon as their communication terminates.
- But the advantage of statelessness is that the server design is simplified to a great extent and it permits the engineers to develop high performance web servers which can handle thousands of TCP connections at a time.
- But many a times it is necessary for a web site to identify users. In such cases HTTP uses **cookies.** Cookies are defined in RFC 2109 and they allow sites to keep track of users.





- Cookies are not used by all the sites but some of the prominent sites that use cookies are : Yahoo, Amazon etc.

**Components of cookie technology :**

- Following are the four components of the cookie technology :
  1. A cookie header line in HTTP response message.
  2. A cookie header line in the HTTP request message.
  3. A cookie file kept on the user's end system and managed by user's browser.
  4. A back end database at web site.

**Operating principle :**

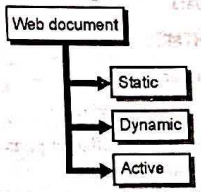
- If a new user X contacts a site (that uses cookies) for the first time, then that web site creates a unique identification number for this new user and then creates an entry in its back end data base.
- This entry is associated with the identification number of user X.
- The server will then respond to X's browser by including the header set-cookies : header, in the HTTP response number of user X.
- For example the header line can be :

  Set-cookie : 1 2 3 4 5 6 7

  Where, 1 2 3 4 5 6 7 is the identification number.
- When X's browser receives the HTTP response message, it reads the set cookie : header. The browser then appends a line to the special cookie file which is managed by the browser.
- This line will include the hostname of the server and the identification number 1 2 3 4 5 6 7. Next time when X visits this same site again, his browser will include the same identification number in each of his HTTP request.
- Thus it is now possible for the web site to track X's activities.
- It is then possible to know the areas of interest of X, which pages does he visit and at what time etc.
- Cookies simplify the internet shopping to a great extent but they remain highly controversial because they are thought as invasion in users privacy.
- It is possible to use cookies to gather personal information about X across a large number of websites.

## 7.12 Web Documents :

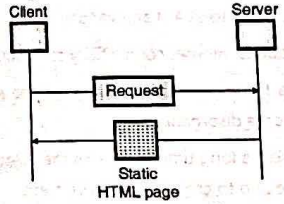
- The web documents can be classified into three categories as shown in Fig. 7.12.1.




(G-668) **Fig. 7.12.1 : Categories of web documents**

### 7.12.1 Static Documents :

- The contents of static documents are fixed. These contents are created and stored in a server. If required the client can get a copy of static document.
- The contents of the static document are determined when it is created.
- These contents cannot be changed when the static document is being used.
- It is possible to change the contents of static document at the server but the user cannot change them. The user can display the static document by using a browser as shown in Fig. 7.12.2.




(G-669) **Fig. 7.12.2 : Static document**

### 7.12.2 HTML (Hypertext Markup Language) :

- The web pages are created by using a language called HTML.
- It uses certain marks to format the text. For example if a part of text is required to be "boldface" then we can use the beginning and ending bold face tags (marks) in the text as shown below :

  <B> – Beginning of boldface

  </B> – End of boldface.
- Here <B> and </B> are the instructions for the browser.
- The browser will make the part of the text between these tags bold. HTML lets the user to use only ASCII characters for the main text as well as for formatting instructions.

- So every computer can receive the whole document as an ASCII document.
- The formatting instructions are used by the browser to format the data.

**Advantages of HTML :**

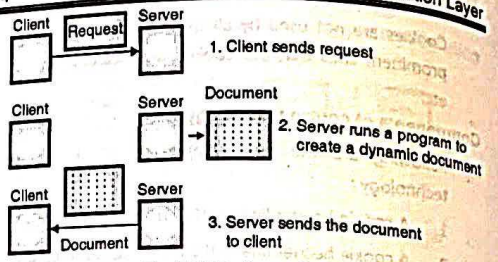
1. Any one can edit it.
2. It is easy to learn and use.
3. People located in different parts of world can work on the same document.
4. It widens the access to web publishing for non-technical users.
5. It is a very flexible tool which can be used for a number of applications.
6. It can be installed free of cost.
7. It is widely used and almost every browser supports it.
8. It is fast to download because the text is compressible.
9. It can be used to present almost any kind of data.

**Disadvantages of HTML :**

1. As anyone can edit, this may be too open for some applications (for example confidential documents).
2. It is open to SPAM and vandalism.
3. Requires Internet connectivity to collaborate.
4. Due to flexibility of its structure, the structure can become disorganized.
5. It takes a long time to choose the colour scheme of page and to create tables, forms etc.
6. It can only create static and plain pages. It is not useful to create dynamic pages.
7. Security features of HTML are not good.
8. It is not centralized. So all the web pages must be edited separately.
9. It has very limited styling capabilities.

## 7.12.3 Dynamic Document :

- The dynamic documents are not present in a predefined format, like static documents.
- A dynamic document is created by a web server on the request for the document from a browser.
- Refer Fig. 7.12.3 to understand how a dynamic document is created and passed on to the client.




(G-670) **Fig. 7.12.3 : Dynamic document**

- First the client sends a request to the web server. After receiving this request, the web server will execute an application program to create a dynamic document.
- The server returns the dynamic document as a response of the request to the client. The contents of a dynamic document will be different corresponding to every request.
- A simple example of a dynamic document is to get time and data from the server. A server follows the steps given below to handle dynamic documents :
  1. The server checks the URL in order to find if it has defined a dynamic document.
  2. If the URL has defined the dynamic document, then the server executes the program.
  3. The output of this program is the dynamic document. It is returned back to the client.

## 7.12.4 Common Gateway Interface (CGI) :

- CGI is the name of a technology which creates the dynamic documents and handles them too.
- CGI is in fact a set of standards. It defines the way in which a dynamic document should be written, the way in which input data be supplied to the program and how the output result be used.
- Note that CGI is not a new language. It allows the user to use the existing languages such as C, C++, Perl etc. However CGI defines rules and terms which are to be followed by the programmers.
- The word **common** in CGI shows that this standard defines some rules which are commonly applicable to any language or platform.
- The word **gateway** indicates that a CGI program is gateway for accessing other resources such as databases and graphic packages.
- Lastly the word **Interface** in CGI indicates the presence of a set of terms, calls and variables which can be used in any CGI program.



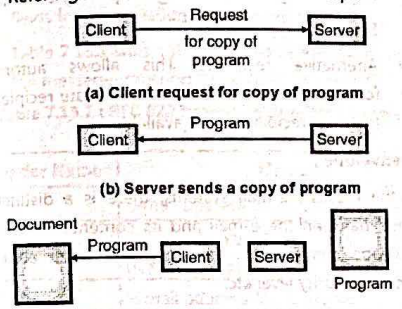


**CGI Program :**

- It is a code which is written in one of the languages that supports CGI (such as C, C ++, etc.).

## 7.12.5 Active Documents :

- Active document can be defined as the program, that is needed to be run at the client side.
- The examples of active documents are the programs creating animated graphics on the screen or the ones which help interaction with the user.
- Refer Fig. 7.12.4 to understand this concept.




(G-671) **Fig. 7.12.4 : Active document**

- It shows that whenever a browser requests for an active document, the server will send a copy of document in the form of byte code.
- The active document will then be run at the browser (client) site. The server stores the active document in the form of a binary code.
- The active document is stored on the server but it is not run on the server. The client receives the document and stores it, and can run it as many times as required without repeating the request.
- The server sends the active document to the client in the binary form. So it is possible to compress it at the server's site and then decompress it at the clients site.
- This will save the bandwidth as well as the transmission time.

**Steps in creation of an active document :**

- Refer Fig. 7.12.4 to understand the creation, compilation and execution of an active document.
  1. At the server, a program is written in source code and stored in a file.
  2. Then the program is compiled and binary code is creates and stored in a file at the server's site.
  3. A client (browser) requests for a copy of program as shown in Fig. 7.12.4(a). This program is transported from the server to the client in the compressed form.
  4. The client converts the received program from binary code into executable code using its own software.
  5. The client runs the program to create the desired result which can include animation or interaction with the user.

# 7.13 Electronic Mail :

- One of the most popular network services is electronic mail (e-mail).
- Simple Mail Transfer Protocol (SMTP) is the standard mechanism for electronic mail in the internet.
- The first e-mail systems simply consisted by file transfer protocols. But some of the limitations of this system were as follows :
  1. It is difficult to send a message to a group of people.
  2. Message did not have any internal structure. So its computer processing was difficult.
  3. The sender never used to know if a message arrived or not.
  4. It was not easy to handover one's e-mails to someone else for the purpose of managing them when one is out of town or country for sometime.
  5. The user interface with the transmission system is poorly integrated.
  6. It was not possible to create and send messages containing a text, drawing, facsimile and voice together.
- So more elaborate e-mail systems were proposed. ARPANET e-mail proposals were published as RFC 821 (transmission protocol) and RFC 822 (message format). These are used in Internet.

## 7.13.1 E-mail Architecture and Services :

- An e-mail system consists of two subsystems :
  1. User agents and
  2. Message transfer agents.

  **User agents :** They enable users to read and send e-mail.

  **Message transfer agents :** They move the messages from the sender to the receiver.

**Basic functions :**

- E-mail systems support five basic systems which are as follows :
  1. Composition
  2. Transfer
  3. Reporting
  4. Displaying and
  5. Disposition

**1. Composition :**

- The process of creating messages and to answer them is known as composition.
- The system can also provide assistance with addressing and a number of header fields attached to each message.

**2. Transfer :**

- It is the process of moving messages from the sender to the recipient.
- This includes establishment of a connection from sender to destination or some intermediate machine, transferring the message, and breaking the connection.

**3. Reporting :**

- The reporting system is designed to tell the sender about whether the message was delivered or rejected or lost.

**4. Displaying :**

- It is the process of displaying the incoming messages so that it can be read by the user. For this purpose simple conversions and formatting are required to be done.

**5. Disposition :**

- This is concerned with what the recipient does with the received message. Disposition is the final step in e-mail system.
- Some of the possibilities are as follows :
  1. Throw after reading
  2. Throw before reading
  3. Save messages
  4. Forward messages
  5. Process messages in some other way.

**Advanced features of E-mail systems :**

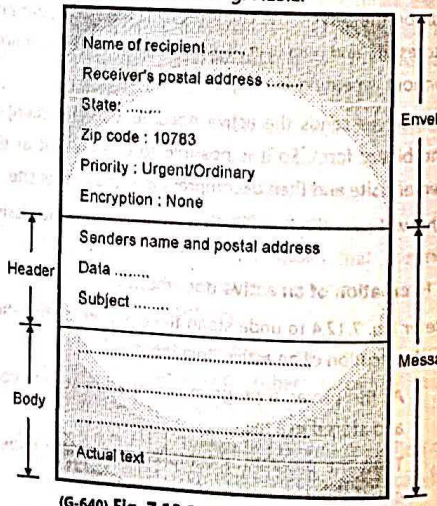
- Some of the advanced features included in addition to the basic functions are as follows :
  1. Forwarding an e-mail to a person away from his computer.
  2. Creating and destroying mailboxes to store incoming e-mail.
  3. Inspecting contents of mailbox, insert and delete messages from the mailboxes.
  4. Sending a message to a large group of people using the idea of mail list.
  5. To provide the facility of registered e-mail.
  6. Automatic notification of undelivered e-mails.
  7. Carbon copies
  8. High priority e-mail (setting the priority of e-mails)
  9. Secret (encrypted e-mail)
  10. Alternative recipient. This allows automatic forwarding of an e-mail to an alternate recipient if the main recipient is not available.

**E-mail envelope :**

- In the modern e-mail systems, there is a distinction made between the e-mail and its contents. An e-mail envelope contains the message, destination address, priority, security level etc.
- The message transport agents such as SMTP use this envelope for routing.

**Message :**

- The actual message inside the envelope is made of two parts :
  1. Header and 2. Body
- Header carries the control information while body contains the message contents. Envelopes and messages are shown in Fig. 7.13.1.




(G-640) Fig. 7.13.1 : Envelope and message





## 7.13.2 Message Formats :

- Let us now discuss the e-mail message formats.

**RFC 822 :**

- All the e-mail messages consist of an envelope, a few header fields, a blank line and then the message body.
- Each header field logically consists of a single line of ASCII text which consists of the field name, a colon and a field.
- Normally the user agent builds a message and passes it to the message transfer agent which uses some header fields for construction of an envelope.
- Table 7.13.1 shows the principle header fields related to the message transport. Let us discuss them one by one.

**Table 7.13.1 : RFC 822 header fields related to message transport**

| Header Name | Meaning |
|---|---|
| To : | E-mail address of primary recipients |
| Cc : | E-mail address of secondary recipients (Carbon copy) |
| Bcc : | E-mail address for blind carbon copies |
| From : | Originator of the message |
| Sender : | E-mail address of the person sending the message |
| Received : | Line added by each transfer agent along the route |
| Return – Path : | Can be used to identify the path back to the sender. |

**1. The To : field :**

- This field gives the DNS address of the primary recipient. It is allowed to have multiple recipients.

**2. The Cc : field :**

- This field gives the addresses of any secondary recipients. Cc stands for carbon copy.
- Whatever message and attachments are sent to the primary recipient the same are sent to the secondary recipient as well.

**3. The Bcc : field :**

- The long form of Bcc is blind carbon copy. This field is like Cc field, except that this is deleted from all the copies sent to the primary and secondary recipients.
- Thus a sender can send copies to third parties without primary and secondary recipients knowing about it.

**4. From : and Sender : fields :**

- These fields tell about who wrote the message and who actually sent the message respectively because the person who creates the message and the person who sends it can be different.
- The From : Field is necessary but the Sender : field can be omitted, if it is same as the From : field.
- These fields are required when the message cannot be delivered and is to be returned to the sender.

**5. Received : field :**

- A line containing Received : is added by each message transfer agent along the way. This line carries the agent's identity, date and time at which the message was received.
- It also contains some other information that can be used to find bugs in the routing system.

**6. The Return-Path : field :**

- This field is added by the final message transfer agent and it is intended to tell how to get back to the sender.
- This information can be obtained from all the received : headers.

**Other header fields :**

- In addition to the fields of Table 7.13.2, RFC 822 messages may contain many other header fields.
- These are used by either the user agents or human recipients some of them are shown in Table 7.13.2.

**Table 7.13.2 : Some fields in RFC 822 message header**

| Header | Meaning |
|---|---|
| Date : | The date and time of the message. |
| Reply-To | E-mail address to which the reply is to be sent |
| Message-Id : | Message identifying number |
| In-Reply-To : | Message-Id of the message to which this is a reply |
| References : | Other relevant message identifying numbers |
| Keywords : | Keywords chosen by user |
| Subject : | Summary of the message for the one line display. |

- The RFC 822 allows the users to invent new headers for their own private use but it is essential that these headers start with the string X-. For example X-Event of the week.

**Message body :**

- The message body comes after the header. The users can include anything that they want to send, in the message body.
- It is possible to terminate the messages with ASCII cartoons, quotations, political statements etc.

## 7.14 MIME – Multipurpose Internet Mail Extensions :

- In the early days, the e-mail used to consist of only the text messages in English and expressed in ASCII.
- RS 822 was sufficient for this environment. But in the worldwide internet environment, this approach is not adequate.
- Some problems are encounter in sending and receiving the following types of messages.
  1. Messages in certain languages that have accents such as French or Germans.
  2. Messages which do not contain text e.g. audio and video.
  3. Messages in the languages which do not have alphabets (e.g. Chinese and Japanese).
  4. Messages which contain some non-Latin alphabets such as Russian or Hebrew.
- The solution to these problems was MIME i.e. Multipurpose Internet Mail Extensions.
- It was proposed in the standard RFC 1341 and then updated in RFC 1521.

### 7.14.1 Principle of MIME :

- MIME uses the same RFC 822 format but it adds structure to the message body (In RFC 822 there is no structure to the message body).
- In addition to this, MIME defines encoding rules for non ASCII messages.
- It is possible to send MIME messages using the existing mail programs and protocols.
- The sending and receiving programs need to be changed to achieve this, which users can do themselves.

**New message headers :**

- Five new message headers are defined for MIME. They are listed in Table 7.14.1.

**Table 7.14.1 : New headers in MIME**

| Sr. No. | Header Name | Meaning |
|---|---|---|
| 1. | MIME – Version : | Indicates the MIME version |
| 2. | Content–Description : | Tells what is in the message |
| 3. | Content – Id : | Identifier |
| 4. | Content – Transfer – Encoding : | How is the body wrapped for transmission |
| 5. | Content – Type : | Type of the message |

**1. MIME-Version :**

- It tells the user agent that this message is a MIME message and it also specifies the version of MIME being used.

**2. Content-Description :**

- This header tells what the message is. It is in the form of ASCII string. This header is needed because the recipient will know whether it is worth decoding and reading the message.

**3. Content-Id :**

- This field identifies the contents. Its format is same as the format of standard Message- Id header.

**4. Content-Transfer-Encoding :**

- This field tells how the body is wrapped for transmission through a network. The network can object to most characters except for the letters, numbers and punctuation marks.
- Five schemes are provided for this purpose in addition to an escape to new schemes. These schemes are as follows :
  (a) The first scheme simply uses ASCII text. Each ASCII character uses 7-bits and can be carried directly if each line contains at the most 1000 characters.
  (b) The second scheme is similar to the first scheme except for one change. Instead of 7-bit characters, now 8-bit characters are used. The maximum line length remains unchanged.
  (c) Third is the messages using binary encoding. The problems with these messages are that they do not always use all the 8 bits and do not stick to the 1000 line per character limit. There is no guarantee that these messages arrive error free.
  (d) The fourth type is encoded binary messages using base 64 encoding which is also called as ASCII armour.





  (e) The fifth type is the messages with a special type of encoding called **quoted-printable encoding.**

**5. Content-Type :**

- This is last header in Table 7.14.1 and it is used to specify the type of the message body. RFC 1521 defines seven types with each one having one or more subtypes.
- The type and subtype are separated by a slash such as,
  Content – Type : Video / mpeg.
- The subtype must be given in the header. Table 7.14.2 gives the initial list of types and subtypes specified in RFC 1521.

**Table 7.14.2 : The MIME types and subtypes in RFC 1521**

| Type | Subtype | Description |
|---|---|---|
| Text | Plain | Text in the unformatted way |
| | Richtext | Text includes simple formatting commands |
| Image | Gif | Still pictures in GIF format |
| | Jpeg | Still pictures in JPEG format |
| Audio | Basic | Audio or sound content |
| Video | Mpeg | Movie (video) in MPEG format |
| Applications | Octet-stream | Byte sequence in uninterpreted form |
| | Post script | A printable document in Post script |
| Message | Rfc 822 | A MIME RFC 822 message |
| | Partial | Split message for transmission |
| | External body | Message itself should be fetched over the net |
| Multipart | Mixed | There are independent parts in the specified order |
| | Alternative | Same message in different formats |
| | Parallel | Parts must be viewed simultaneously |
| | Digest | Each part is a complete RFC 822 message |

- Note that many new types have been added to the basic list of Table 7.14.2 and the addition is still being made. Let us discuss the content types listed in Table 7.14.2.

**(a) Text :**

- The text type is for straight text. There are two subtypes namely plain and richtext. The text/plain combination represents the original messages without any encoding or further processing.
- The text/richtext allows simple formatting in the text. It allows the text with boldface, italics, small and large point sizes, indentation, subscripts, page layout etc.

**(b) Image :**

- This MIME type is used for transmitting still pictures. There are many formats used for storing and transmitting images with or without compression. The two subtypes are GIF and JPEG.

**(c) Audio and Video :**

- The audio type is for sound and video is for moving pictures. The video does not include any soundtrack. Only one video format defined is MPEG which is designed by the Moving Picture Experts Group (MPEG).

**(d) Applications :**

- This type is used for formats which require external processing and which is not covered by any other type.
- The octet stream is a sequence of uninterpreted bytes. When it is received a user agent should display it and suggest the user to copy it in a file for further processing.
- The other subtype is post script. It corresponds to the post script language produced by Adobe systems which is used for describing printed pages.

**(e) Message :**

- This type allows one message to be fully encapsulated inside the other message. This is useful in order to forward e-mails.
- The partial subtype allows to break an encapsulated message into pieces and send them separately. The external body subtype can be used for very long messages such as video films.

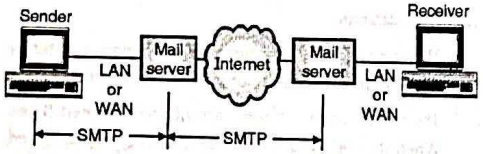
**(f) Multipart :**

- This is the last type of multipart. It allows a message to contain multiple parts in the same message. The beginning and end of each part is clearly demarked within a message.
- There are four subtypes. The mixed subtype allows each part to be different. In the alternative subtype each part should contain the same message expressed in a different medium or encoding.

- The alternative subtype can be used for multiple languages as well.
- The parallel subtype is used for viewing all parts simultaneously e.g. audio and video parts of a movie.
- The fourth subtype is digest. It is used when many messages are packed together to form a composite message.

## 7.15 Message Transfer Agent : SMTP :

- The actual mail transfer is carried out through the message transfer agent.
- A system should have the client MTA in order to send a mail and it should have a server MTA in order to receive one.
- SMTP is the protocol which defines MTA client and server in the Internet.
- As shown in Fig. 7.15.1, the SMTP is used twice, once between the sender and sender's mail server and then between the two mail servers.

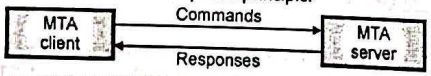



(G-641)Fig. 7.15.1 : SMTP range

- The job of SMTP is simply to define how commands and responses be sent back and forth. Each network can choose its software package for implementation.

### 7.15.1 Commands and Responses :

- As shown in Fig. 7.15.2, SMTP the transfer of messages between MTA client and MTA server takes place using the command and response principle.




(G-642)Fig. 7.15.2 : Commands and responses in HTTP

- Each command or response is terminated by a two character end of line token. The two characters used are carriage return and line feed.

### 7.15.2 SMTP (Simple Mail Transfer Protocol) :

- In internet the source machine establishes a connection to port 25 of the destination machine so as to deliver an e-mail.
- An e-mail daemon which speaks SMTP is listening to this port.
- This daemon is supposed to perform the following tasks :
  1. Accept the incoming connections, and copy messages from them into appropriate mailboxes.
  2. Return an error message to the sender, if a message is not delivered.
- SMTP is a simple ASCII protocol. Once a TCP connection between a sender and port 25 of the receiver is established, the sending machine operates as a client and the receiving machine acts as a server.
- The client then waits for the server to take initiative in communication. The server sends a line of text which declares its identity and announces its willingness/ unwillingness to receive mail.
- If the server is not prepared, the client will release the connection, wait for some time and try again later. But if the server is willing to accept e-mail, then the client announces the sender of e-mail and its recipient.
- If such a recipient exists at the destination, then the server tells the client to send the message. The client, then sends the message and the server sends back its acknowledgement.
- No checksums are generally required because TCP provides a reliable byte stream. If there are any more e-mail, then they can be sent now.
- After exchanging all the e-mail, the connection is released. SMTP uses numerical codes. The lines sent by the client are marked C : ; and those sent by the server are marked S : ;
- Some of the commands, useful for communication are :

  HELO, RCTP, DATA, QUIT etc.

- RCTP represents recipient. If only one command is used then the message is being sent to only one recipient. If the command is used many times, then it indicates that the message is sent to more than one recipients.
- In such a case each message is individually acknowledged or rejected.
- The syntax of four character commands for the clients are rigidly specified but the syntax for the replies are not that rigid.
- The SMTP protocol is well defined by RFC 821 but some problems are still present.

**Problems in SMTP :**

- Some of the problems in SMTP are as follows :





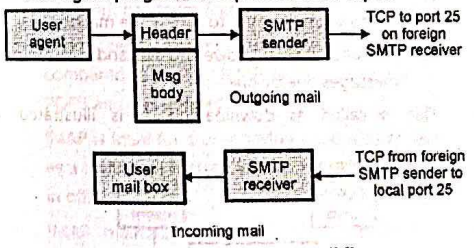
1. Some older versions of SMTP are not capable of handling messages longer than 64 kB.
2. If client and server have different time-outs, then one of them may give up when the other is still busy. This will terminate the connection unnecessarily.
3. In rate situations, infinite mailstorms can be triggered.

**Extended SMTP (ESMTP) :**

- Some of these problems can be solved by using the extended SMTP (ESMTP) which is defined in RFC 1425.

### 7.15.3 Components of E-mail System :

- The three main components of internet mail system are :
  1. User Agent (UA)
  2. SMTP sender
  3. SMTP receiver
- They are shown in Fig. 7.15.3. The mail is created by a user agent program in response to user input.




(G-643) Fig. 7.15.3 : SMTP mail flow

- Each created message consists of a header which includes the recipient's E-mail address and other information and the message body containing the message to be sent.
- These messages are lined up to form a queue and provided as input to an SMTP sender program.
- The SMTP sender takes messages from the queue and transmits them to the proper destination host via SMTP connection over one or more TCP connections to port 25.
- The SMTP protocol is used to transfer a message from the SMTP sender to SMTP receiver and it uses TCP connection for the same.
- The SMTP receiver accepts each arriving message and stores it in the user mail box.
- If the mail is to be forwarded then the SMTP receiver copies it to the outgoing mail queue.

### 7.15.4 SMTP Commands :

- The operation of SMTP consists of a series of commands and responses exchanged between the SMTP sender and receiver.
- The SMTP sender establishes the TCP connection to the receiver.
- After establishing the connection, the SMTP sender sends commands over the connections to the receiver.
- The SMTP receiver generates exactly one reply from the SMTP receiver. Table 7.15.1 shows the SMTP commands.
- Each command consists of a single line of text which begins with a four letter command code followed in some cases by an argument field.
- Most replies are a single line. However multiline replies also are possible.

**Table 7.15.1 : SMTP commands**

| Name | Description |
|---|---|
| HELO | Send identification of the sender. |
| MAIL | Identifies originator of mail. |
| RCPT | Identifies recipient of mail. |
| DATA | Transfer message text. |
| RSET | Abort the current mail transaction. |
| NOOP | No operation. |
| QUIT | Close TCP connection. |
| SEND | Send mail to terminal. |
| SOML | Send mail to the terminal if possible, otherwise to mailbox. |
| SAML | Send mail to terminal and mail box. |
| VRFY | Confirm user name. |
| EXPN | Return membership of mailing list. |
| HELP | Send system-specific documentation. |
| TURN | Reverse role of sender and receiver. |

### 7.15.5 SMTP Operation :

- The basic SMTP operation occurs in three phases :
  1. Connection setup
  2. Exchange of one or more command-response pairs
  3. Connection termination

**1. Connection setup :**

- The sender opens (i.e. creates) a TCP connection with the receiver. Once the connection is established, the receiver identifies itself with "220 Service Ready".

- The sender identifies itself with HELO command. The receiver accepts the sender's identification with "250 OK".

**2. Mail transfer :**

- Once the connection has been established, the SMTP sender may send one or more messages to SMTP receiver. There are three logical phases to transfer a message :
  1. A MAIL command identifies the originator of message.
  2. One or more RCPT commands identify the recipient for this message.
  3. A DATA command transfers the message text.

**3. Connection closing :**

- The SMTP sender closes the connection in two steps. First the sender sends a QUIT command and waits for a reply.
- Second step is to initiate a TCP close operation for the TCP connection. The receiver initiates its TCP close after sending its reply to the QUIT command.

### 7.15.6 Comparison of HTTP and SMTP :

**Table 7.15.2 : Comparison of HTTP and SMTP**

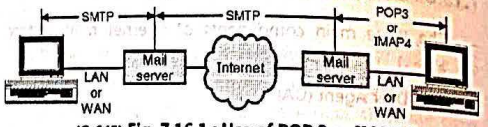
| Sr. No. | SMTP | HTTP |
|---|---|---|
| 1. | Message is transferred from client to server. | Message transfer is from client to server or the other way round. |
| 2. | Uses TCP. | Uses TCP. |
| 3. | Uses port 25 for transmission. | Uses port 80 for transmission. |
| 4. | SMTP messages are to be read by humans. | HTTP messages are to be read and understood by the HTTP servers and HTTP clients. |
| 5. | These messages are first stored and then forwarded. | These messages are immediately delivered. |

## 7.16 Message Access Agent : POP and IMAP :

- The SMTP is used in the first and second stages of mail delivery. But SMTP is not used in the third stage, because SMTP is a push protocol which is meant for pushing the message from client to server.
- The third stage needs a **pull** protocol because the client has to pull messages from the server. The bulk data gets transferred from the server to client.
- Therefore third stage uses a message access agent which is a pull protocol.
- The two message access agents available are :
  1. Post Office Protocol, version 3 (POP 3).
  2. Internet Mail Access Protocol (IMAP 4).

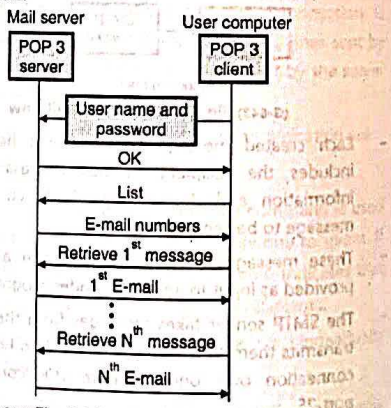
### 7.16.1 POP 3 :

- The POP3 consists of client POP3 software and server POP3 software.
- Out of these, the client POP3 software is installed on the receiving computer whereas the mail server gets the server POP3 software installed on it.
- When the user wants to download email from the mailbox on the email server, the events take place in the following sequence. Refer Fig. 7.16.1.




(G-645) **Fig. 7.16.1 : Use of POP 3 or IMAP 4**

  1. The client (user) establishes a connection with the server on TCP port 110.
  2. The client then sends its user name and password to the server in order to access the mailbox.
  3. The user is then allowed to list and get the mail messages one by one.
- This is called as downloading. It is illustrated in Fig. 7.16.2.




(G-647) **Fig. 7.16.2 : Downloading in POP3**

**Modes of POP 3 :**

- POP3 has two modes of operation :
  1. Delete mode and
  2. Keep mode.
- **Delete mode :** In this mode the mail is deleted from the mailbox after each retrieval.





- This mode is used when the user is working on his permanent computer because it is then possible for him to save and rearrange the received mail after reading it.
- **Keep mode :** If operated in this mode, the mail remains in the mailbox after retrieval.
- This mode is used when the user accesses mail away from the primary computer. The read mail can be organized later.

**Disadvantages of POP3 :**

1. POP3 does not allow organization of email on the server.
2. The user can not create different folders on the server. It can create them only on his own computer.
3. The user can not partially check the contents of E-mail before down loading.

### 7.16.2 IMAP4 :

- Internet Mail Access Protocol Version 4 (IMAP4) is another mail access protocol which is very similar to POP3 but has more features.
- This makes IMAP4 more powerful but more complex as compared to POP3. IMAP is more sophisticated than POP3 and it is defined in RFC 1064.
- IMAP is ideal for a user having multiple computers such as a laptop on the road, PC at home and a workstation in office.
- IMAP maintains a central repository which can be accessed from any machine. So IMAP does not copy e-mail to the user's personal machine.
- An important feature of IMAP is its ability to address mail not by arrival number but by using attributes. That means the mailbox is like a relational database system than a linear sequence of messages.

**Extra features of IMAP4 :**

1. It is possible for the user to check the header before down loading.
2. It is possible for the user to search for the contents of E mail before downloading.
3. It is possible to partially download E mail.
4. It is possible for the user to create, rename or delete mailboxes on the mail server.
5. It is possible for the user to create a hierarchy of mailboxes in a folder for storing e-mails.

### 7.16.3 Comparison of IMAP and POP 3 :

**Table 7.16.1 : Comparison of IMAP and POP 3**

| Sr. No. | Parameter | POP 3 | IMAP |
|---|---|---|---|
| 1. | Protocol is defined at | RFC 1939 | RFC 2060 |
| 2. | TCP port used | 110 | 143 |
| 3. | e-mail is stored at | User's PC | Server |
| 4. | e-mail is read | Off line | On line |
| 5. | Time required to connect | Small | Long |
| 6. | Use of server resources | Minimal | Extensive |
| 7. | Multiple mail boxes | Not possible | Possible |
| 8. | Who backs up mailboxes | User | ISP |
| 9. | For mobile users | Not good | Good |
| 10. | User control over download | Little | Great |
| 11. | Partial message downloads | No | Yes |
| 12. | Simplicity in implementation | Yes | No |
| 13. | Support | Wide spread | Increasing |

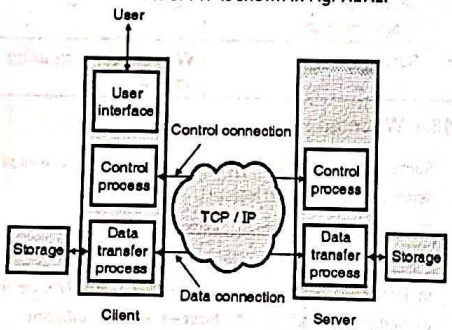
### 7.16.4 Webmail :

- Some Web sites like Hotmail and Yahoo provide e-mail service to anyone who wants it.
- These sites have normal message transfer agents that listens to port 25 for incoming SMTP connections.
- In Hotmail, you have to acquire their DNS MX (mail exchange) record by typing **host –a –v hotmail.com**.
- On a UNIX operating system, suppose the mail server is **mx10.hotmail.com**, then by typing **telnet mx10.hotmail.com 25** a TCP connection can be established over which SMTP we can send commands in the usual way.
- It may take several attempts to get a TCP connection accepted if big servers are busy.
- It is interesting to know how e-mail is delivered. Basically, when the user goes to the e-mail Web page, a form is presented, in which the web page asks the user for a login name and password.
- The login name and password are sent to the server when the user clicks on Sign In. The server then validates the login name and password.

- After a successful login, the server finds the user's mailbox and builds a listing formatted as a Web page in HTML.
- The Web page is then sent to the browser for display. There are many clickable items on the page, so that messages can be read, deleted, and so on.

## 7.17 File Transfer Protocol (FTP) :

- A standard mechanism provided by the Internet which helps in copying a file from one host to the other is known as the File Transfer Program (FTP).
- Some of the problems in transferring files from one system to the other are as follows :
  1. Two systems may use different file name conventions.
  2. Two systems may represent text and data in different ways.
  3. The directory structures of the two systems may be different.
- FTP provides a simple solution to all these problems. The basic model of FTP is shown in Fig. 7.17.1.




(G-648)**Fig. 7.17.1 : Basic model of FTP**

- FTP establishes two types of connections between the client and server. One of them is used for data transfer and the other is for the control information.
- The fact that FTP separates control and data makes it very efficient. The control connection uses simple rules of communication.
- Only one line of command or a line of response is transferred at a time. But the data connection uses more complex rules due to the variety of data types being transferred.
- FTP uses port 21 for the control connection and port 20 for the data connection. Both these are well known TCP ports.
- As shown in Fig. 7.17.1 the client is made of three blocks namely :
  1. User interface
  2. Control process and
  3. Data transfer process.
- The server has two blocks : the control process and data transfer process. The control connection connects the control processes while data connection connects the data transfer processes as shown in Fig. 7.17.1.
- The control connection is kept alive during the entire interactive FTP session.
- The data connection is first opened, file is transferred and data connection is closed. This is done for transferring each file.

**Control connection :**

- This connection is created in the same way as the other application programs described earlier. Control connection remains alive during the entire process.
- The IP uses minimize delay type service because this is an interactive connection between a user and a server.

**Data connection :**

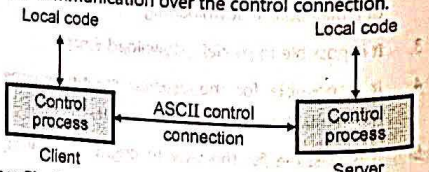
- Data connection uses the port 20 at the server site. This connection is opened when data to be transferred is ready and it is closed when transfer of data is over.
- The data connection does not remain open continuously like control connection. It is opened and closed many times as per requirement.

### 7.17.1 Communication in FTP :

- FTP operates in client – server environment. The two computers involved in communication may be different in terms of the operating systems, character sets, file structures and file formats etc.
- FTP can make them compatible. The approaches for communication over control connection and data connection are different from each other.

**1. Communication over control connection :**

- Refer Fig. 7.17.2 to understand the FTP's approach for the communication over the control connection.




(G-649)**Fig. 7.17.2 : Communication over control connection**

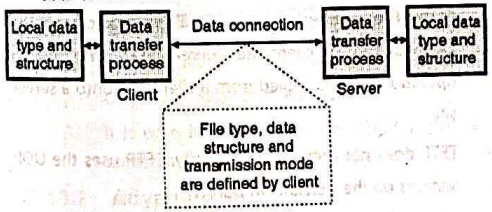




- Similar to SMTP, FTP uses a set of ASCII characters to communicate across the control connection. Communication is achieved through a process of commands and response. One command is sent at a time.
- Each command or response is only of one short line. So it is not necessary to think about file format or file structure.
- Each line is ended with a two character token. The two characters used in the token are carriage return and line feed.

**2. Communication over data connection :**

- The purpose of implementing a data connection is to transfer a file. For this the client has to define the following :
  1. Type of file being transferred.
  2. Structure of data in the file
  3. Mode of transmission.
- Before the transmission over data connection, the communication over control connection is performed. Refer Fig. 7.17.3 to understand communication over data connection.




(G-650)**Fig. 7.17.3 : Communication over the data connection**

- The problem of heterogeneity is solved by defining three attributes of communication : file type, data structure and transmission mode.

### 7.17.2 File Types :

- FTP can use one of the following file types for transfer of data over the data connection :
  1. ASCII file 2. EBCDIC file 3. Image file.
- ASCII file is a text file, EBCDIC file can be transferred if both ends use EBCDIC encoding.
- Image file is the default format for transferring the binary files. With ASCII or EBCDIC files one more attribute must be added for defining the printability of the file.
- This attribute is nonprint or TELNET.

### 7.17.3 Data Structure :

- FTP can use one of the following data structures :
  1. File structure (default)
  2. Record structure and
  3. Page structure.
- File has no structure. It is simply a continuous stream of bytes. In the record structure the file is divided into records.
- This data structure is suitable only for the text files. In page structure, a file is divided into pages which can be stored or accessed randomly or sequentially.

### 7.17.4 Transmission Mode :

- FTP uses one of the following modes to transfer a file :
  1. Stream mode
  2. Block mode and
  3. Compressed mode.

**1. Stream mode :**

- In this mode the data is delivered from FTP to TCP in the form of continuous stream of bytes. TCP chops this data into segments of appropriate size.
- Stream mode is the default mode of transmission.

**2. Block mode :**

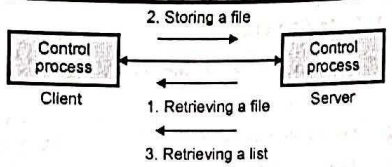
- In this mode, data delivery from FTP to TCP takes place in the form of data blocks. Each such block is preceded by a 3 byte header.

**3. Compressed mode :**

- For big files the data can be compressed. Generally a run length encoding is used for compression.

### 7.17.5 File Transfer :

- File transfer takes place over the data connection and the commands are sent over the control connection. The commands supervise the data transfer.
- But file transfer in FTP means one of the following :
  1. **Retrieving a file :** Server copies a file onto a client.
  2. **Storing of a file :** A file can be copied from client to the server.
  3. A server sends a list of directory or file names to the client. FTP treats such a list of directory also as a file.
- The file transfer has been illustrated in Fig. 7.17.4.




(G-651)Fig. 7.17.4 : File transfer

### 7.17.6 FTP Commands :

- The following commands are used for copying files using FTP.

**Table 7.17.1 : FTP commands to transfer files**

| Command | Explanation |
|---|---|
| Get | Copy a file from remote host to local host |
| M get | Copy multiple files from the remote host to local host |
| Put | Copy a file from local host to remote host |
| M put | Copy multiple files from the local host to remote host |

- FTP commands used to connect to a remote host are as shown in Table 7.17.2.

**Table 7.17.2 : FTP commands to connect to a remote host**

| Command | Explanation |
|---|---|
| Open | Select the remote host and initiate login session |
| User | Identify the remote user ID |
| Pass | Authenticate the user |
| Site | Send the information to the remote host. |

- FTP commands used to end an FTP session are as shown in Table 7.17.3.

**Table 7.17.3 : FTP command to terminate session**

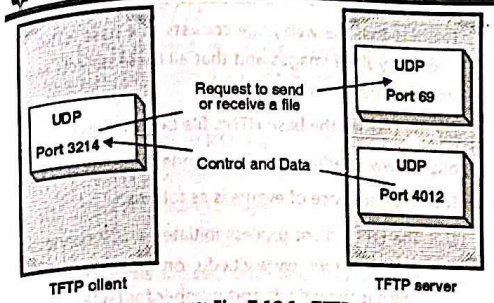
| Command | Explanation |
|---|---|
| Quit | Disconnect from the remote host and terminate FTP. |
| Close | Disconnect from the remote host but leave FTP client running. |

## 7.18 TFTP :

- The **Trivial File Transfer Protocol (TFTP)** is a minimal protocol for transferring files without authentication and without any separation of control information and data as in FTP.
- In certain situations, the user needs to just copy a file and does not need all the features provided by the FTP protocol.
- Take the example of booting of a discless work station or a rooter.
- For booting them, it is only necessary to download the bootstrap and configuration files without using any sophistication of FTP. We use TFTP in such situations.
- TFTP is frequently used by devices without permanent storage for copying an initial memory image (bootstrap) from a remote server when the devices are powered on.
- Due to the lack of any security features, the use of TFTP is generally restricted.
- TFTP uses the unreliable transport protocol UDP for the transportation of data.
- TFTP is an extremely simple protocol. For a diskless work station the TFTP can be stored in a ROM because it needs to use only the basic IP and UDP.
- TFTP can perform only two functions – either read a file or write a file.
- In the **file reading** operation, a file is copied from the server site to the client site whereas in the **file writing** operation, a file is copied from a client site onto a server site.
- TFTP does not provide any security. TFTP uses the UDP services on the well known port 69.
- Each TFTP message is carried in a separate UDP datagram.
- The first two bytes of a TFTP message specify the type of message, which can be a request to download a file, request to upload a file, a data message, or an acknowledgement or error message.
- At the beginning of a TFTP session a TFTP client sends a request to upload or download a file from a UDP port to the (well-known) UDP port 69 of an TFTP server.
- When the request is received the TFTP server picks a UDP port of its own and uses this port to communicate with the TFTP client.
- Thus, both client and server communicate using ephemeral ports as shown in Fig. 7.18.1.








(G-652) Fig. 7.18.1 : TFTP

- Since UDP does not recover lost or corrupted data, TFTP is supposed to maintain the integrity of the data exchange.
- TFTP transfers data in blocks of 512 bytes. A 2 byte long sequence number is assigned to each block and is transmitted in a separate UDP datagram.
- A block must be acknowledged before the next block can be sent.
- When an acknowledgment is not received before a timer expires, the block is retransmitted.
- When the receiver receives a block that is less than 512 bytes long, it assumes that the end of file has been reached.

### 7.18.1 Applications of TFTP :

1. TFTP is used for basic file transfer application.
2. Used to initialize bridges and routers.
3. It is used in conjunction with DHCP to obtain the contents of the configuration file.

### 7.18.2 Advantages :

1. It is a very simple protocol.
2. It needs a very small storage space and can be stored on a ROM.

### 7.18.3 Disadvantages :

1. TFTP does not have flow control.
2. There is no error control mechanism.
3. There is no provision for security.

### 7.18.4 Comparison of FTP and TFTP :

**Table 7.18.1 : Comparison of FTP and TFTP**

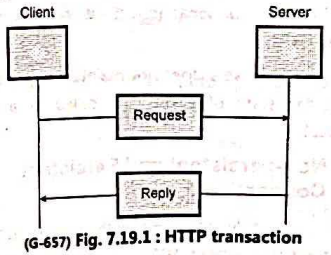
| Sr. No. | Parameter | FTP | TFTP |
|---|---|---|---|
| 1. | Operation | Transferring files | Transferring files |
| 2. | Authentication | Yes | NO |
| 3. | Control and data | Separated | Not separated |
| 4. | Protocol | TCP | UDP |
| 5. | Ports | 21 – control, 20 – data | Port 3214, 69, 4012 |
| 6. | Data transfer | Reliable | Unreliable |

## 7.19 HTTP (Hypertext Transfer Protocol) :

- The main function of HTTP is to access data on WWW. This protocol can access the data in various forms such as plaintext, hypertext, audio, video etc.
- The function of HTTP is equivalent to a combination of FTP and SMTP. It uses services of TCP. It uses only one TCP connection (port 80).
- There is no separate control connection like the one in FTP. Only the data transfer takes place between the client and server so there is only one connection and it is the data connection.
- The data transfer in HTTP is similar to SMTP. The format of the messages is controlled by MIME like headers.

### 7.19.1 Principle of HTTP Operation :

- The principle of HTTP is simple. A client sends a request. The server sends a response. The request and response messages carry data in the form of a letter with a MIME like format.
- Fig. 7.19.1 shows the HTTP transactions between client and server.




(G-657) Fig. 7.19.1 : HTTP transaction

- The client initializes the transaction by sending a request message and the server responds by sending a response.

### 7.19.2 The Web and HTTP :

- HTTP is the Web's application layer protocol. It is the heart of the Web.
- It has been defined in [RFC 1945] and [RFC 2616].

- HTTP is implemented in two programs :
  1. A client's program 2. A server's program.
- These programs are executed on different and systems and talk to each other by exchanging HTTP messages.
- HTTP defines how Web clients such as browsers request Web pages from Web servers and how servers transfer Web pages to clients.
- HTTP uses TCP as its under lying transport protocol (rather than using UDP). The HTTP client first initiates a TCP connection with the server.
- After establishing a connection, the browser and the server processes access TCP through their socket interface. TCP provides a reliable data transfer service to HTTP.
- That means each HTTP request message, transmitted by a client will eventually arrive intact at the server. Similarly each HTTP response message transmitted by the server will eventually arrive intact at the client, due to the reliable TCP connection.
- Due to this kind of layered architecture HTTP need not have to worry about the lost data or about the details of how TCP deals with the loss and retransmission of data. It is managed by TCP.

**Statelessness :**

- In HTTP, the server sends the files requested to the client without storing any state information about the client.
- So it may happen that the same client may ask the same information repeatedly to the server and the server would not even understand it. So it will keep resending those files.
- As the HTTP servers does not maintain any information about the state of client it is called as a stateless protocol.

### 7.19.3 Non-persistent and Persistent Connection :

- HTTP is capable of using both non-persistent and persistent connections. HTTP uses persistent connection in its default mode.
- But HTTP clients and servers can be configured to use the non-persistent connection as well.

**1. Non-persistent connections :**

- Let us discuss the step-by-step procedure followed for transferring a web page from server to client for a non-persistent connection.
- Imagine that the web page consists of a base HTML file and many JPEG images and that all these objects reside on the same server.
- Let the URL for the base HTML file be as follows :

  http://www.vit.edu/itdept/home.index
- Then the sequence of events is as follows :
  1. The HTTP client process initiates a TCP connection to the server www.vit.edu on port number 80, which is the default port number for HTTP.
  2. The HTTP client, sends an HTTP request message to the server via its socket associated with the TCP connection. This request message is of the following format :

     Path name/itdept/home.index.
  3. The HTTP server process receives the request message via its socket associated with the connection. It then retrieves the object.

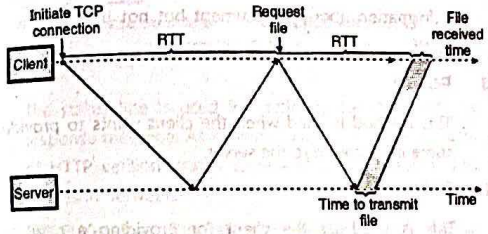
     /itdept/home.index from its storage.
- It then encapsulates this retrieved object in an HTTP response message and sends the response message to the client via its socket.
  4. The HTTP server process tells TCP to close the TCP connection.
  5. As soon as the HTTP client receives the response message, the TCP connection is terminated.
  6. The response message indicates that the encapsulated object is an HTML file. The client takes out the file from the response message and examines the HTML file. The client will find references to all the JPEG objects.
  7. The client follows the first four steps for each JPEG object.
- As the browser receives the web page, it displays the page. Different browsers can display the same web page differently.
- However HTTP is not concerned about this. Its specifications define only the communication between the HTTP client program and HTTP server program.
- The steps discussed earlier were for the **non-persistent** connection where each TCP connection is closed after the server sends the object.
- That means the TCP connection does not persist for other objects. Each TCP connection transports one request message and one response message.





**Round-Trip Time (RTT) :**

- The RTT is defined as the time taken by a small packet to travel from client to server and then back to the client.
- The components of RTT are :
  1. Packet propagation delays
  2. Packet queuing delays
  3. Packet processing delays.
- Now consider the sequence of events taking place when a user clicks on a hyperlink. These events are illustrated in Fig. 7.19.2.




(G-658) **Fig. 7.19.2**

1. The browser initiates a TCP connection between the browser and web server. This process makes use of a **three way handshake**.
- In the three way handshake, the client sends a small TCP segment to the web server. The server acknowledges and responds with another small TCP segment. Finally the client acknowledges back to the server.
  2. After completing the first two parts of the three way handshake the client sends the HTTP request message to the server.
  3. In response the server sends the HTML file to the client. The total response time as shown in Fig. 7.19.2 is equal to 2RTT plus the time taken by the server to transmit the file.

**Disadvantages of non-persistent connections :**

1. It is necessary to establish and maintain a new connection for each requested object.
2. For each connection TCP buffers need to be allocated and TCP variables need to be kept in both the client and server.
3. There is a delay of 2RTTs associated with the transfer of each object.

**2. Persistent connection :**

- The disadvantages of non-persistent connections can be overcome if persistent connection is used.
- With the persistent connection, the server leaves the TCP connection open after sending a response.
- All the requests and responses between the same client and server can be sent over the same connection. Hence the entire web page can be sent over a single persistent connection.
- It is also possible to send the multiple web pages residing on the same server to the same client over a single persistent TCP connection. The TCP connection is closed only after the time out interval by the HTTP server.

**Types of persistent connections :**

- The two versions of persistent connections are as follows :
  1. Without pipelining 2. With pipelining.

**1. Without pipelining :**

- For this version, the client has to issue a new request only when it receives the previous response. The delay of only one RTT is experienced by the client in order to request and receive each object.
- This is an improvement over the non-persistent connection which experiences a delay of 2RTT. This delay can be reduced by using pipelining.
- Another disadvantage of no pipelining is that the TCP connection becomes idle i.e. does nothing while it waits for another request after the server had sent an object.

**2. With pipelining :**

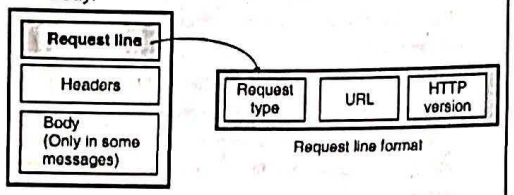
- This mode reduces the delay further. The default mode of HTTP uses persistent connection. With pipelining the HTTP client will issue a request as soon as it encounters a reference.
- This allows the HTTP to make back to back requests. It can make a new request before receiving the response.
- When the server receives back to back requests, it sends the objects back to back.
- With pipelining only one RTT will be expended for all the referenced objects. Another advantage is that the pipelined TCP connection remains idle for a very short time.

### 7.19.4 HTTP Messages :

- The HTTP messages are of two types :
  1. Request message
  2. Response message.
- The format of both these messages is almost the same.

### 7.19.5 Request Message :

- Fig. 7.19.3(a) shows the format of the request message. It consists of a request line, headers and sometimes a body.



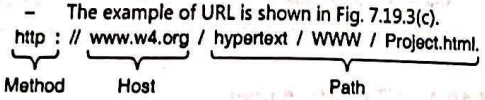

(G-659) Fig. 7.19.3(a) : HTTP request message

1. **Request line :**
   - The request line is used for defining the request type, resource (URL) and HTTP version as shown in Fig. 7.19.3(a).
   - **Request type :** Several request types are defined.
   - **Uniform Resource Locator (URL) :** The client accessing a web page needs an address. The HTTP uses the URL to facilitate the access of any document distributed over the world. The URL defines four thing as shown in Fig. 7.19.3(b). They are as follows :

     1. Method 2. Host computer
     3. Port 4. Path.


(G-660) Fig. 7.19.3(b) : URL

   - Method is the protocol used such as FTP, HTTP. Host is the computer where the required information is located. The name of the computer begins with www but this is not mandatory.
   - URL can optionally contain the server's port number. If the port is included then it should be inserted between host and path and it should be separated by a colon.
   - Path is the name of the file where the information is located.
   - **Version :** The latest version of HTTP is 1.1 but the versions 0.9 and 1 are also used.
   - The example of URL is shown in Fig. 7.19.3(c).




(G-1969) Fig. 7.19.3(c) : Example of URL

### 7.19.6 Methods (Request Type) :

- This is one of the fields in the request line format. It defines different types of messages referred to a request types or methods.
- The request method is a command or request issued by the client to the server.
- Following are some of the important methods (request types).

1. **GET :**

   The client uses this method for retrieving a document from the server. The address from where this document is to be obtained is defined in the URL.

2. **HEAD :**

   The client uses this method in order to obtain some information about a document but not the document itself.

3. **POST :**

   This method is used when the client wants to provide some information to the server.

4. **PUT :**

   This is used by the client for providing a new or replacement document to be stored on the server.

5. **PATCH :**

   This method is similar to PUT. But there is one change. The patch request contains a list of differences which should be implemented in the existing file.

6. **COPY :**

   This method is used to copy a file to another location.

7. **MOVE :**

   This method is used for moving a file to another location.

8. **DELETE :**

   It is used for removing a document on the server.

9. **LINK :**

   It is used for creating a link or a link from a document to another location. The location of the file is specified in the URL request line and the location of destination is specified in the entity header.

10. **UNLINK :**

    It is used for deleting the links created by the LINK method.

11. **OPTION :**

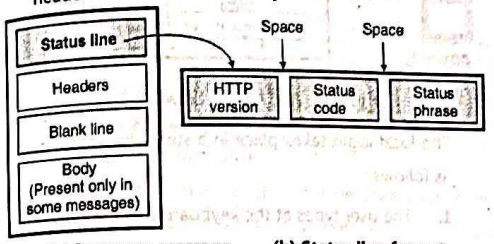
    It is used by the client to ask the server about various options that are available.





### 7.19.7 Response Message :

- Fig. 7.19.4(a) shows the format of the response message. A response message is made of a status line, a header and sometimes a body.




(a) Response message (b) Status line format

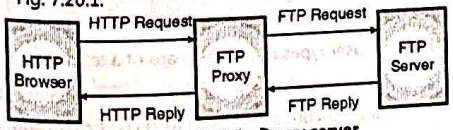
(G-662) Fig. 7.19.4

**Status line :**

- The status line is used for defining the status of the response message. As shown in Fig. 7.19.4(b) it consists of HTTP version, status code and status phrases with spaces in between.
- **HTTP Version :** This field indicates the version of HTTP being used. This field is same as the HTTP version field used in the request line.
- **Status Code :** It is a three digit field which is similar to those in FTP and SMTP protocols.
- **Status Phrase :** It is used for explaining the status code in the text form.

## 7.20 Proxy Server :

- All the servers cannot speak HTTP some of them use the FTP, Gopher or some other protocols. A large information is available on FTP and Gopher servers so it should be made available to web users.
- To do so, one solution can be to have a browser which can use the HTTP as well as FTP, Gopher and other protocols.
- But this makes the browser unnecessarily large. The other solution to this problem is proxy server, shown in Fig. 7.20.1.




(G-656) Fig. 7.20.1 : Proxy server

- Proxy server is basically a gateway which communicates using HTTP to the browser FTP, Gopher or some other protocol for communicating to the server.
- It receives HTTP requests from a browser, converts them in FTP or Gopher requests and sends them to the FTP/Gopher server as shown in Fig. 7.20.1.
- Proxy server can be a program running on the same machine working as a browser or it can be a separate machine.
- The users can configure their browsers with proxies for those protocols which the browser does not use for communication.
- The other important feature of a proxy server is caching. A caching proxy server collects and stores all the pages which pass through it.
- When a user asks for a page, the proxy server will first see if it has the page stored with it. If the page is there then it will see if the page is up to date.
- If the page is updated then, it passes the page to the user otherwise it will fetch a new copy of the page. A proxy server can be put inside a firewall. The user can access the web but he is not allowed the full Internet access.
- In such situation the user talks to the proxy server and the server communicates with obtains different sites and obtains pages on behalf of the user.

### 7.20.1 HTTP Security :

- As such HTTP does not provide any security. But it can be run over the secured socket layer (SSL). If so, the HTTP is called as HTTPS.
- The security features of HTTPS include confidentially, authentication of client and server and data integrity.

## 7.21 Remote Login : TELNET and SSH :

- The Internet and TCP/IP suite have been designed primarily to provide service to its users.
- The requirements of different users will be of different types and with increase in the number of users, the number of diversified demands will also be very large.
- It is practically impossible to write a specific client - server program for each demand.
- Therefore a general purpose client - server program should be developed which will help a user to access any application on a remote computer.
- That means a user will be allowed to log into a remote computer.

- Two of such general purpose client - server programs which allow remote login are : TELNET and SSH.

### 7.21.1 TELNET :

- The long form of TELNET is TErminal NETwork. It was proposed by ISO as a standard TCP/IP protocol for a virtual terminal service.
- TELNET enables a user to establish a connection to a remote system.

**Concepts related to TELNET :**

- Some of the important concepts related to TELNET are as follows :
  1. Time sharing environment.
  2. Login : Local or Remote.
  3. Network Virtual Terminal.

**Time sharing environment :**

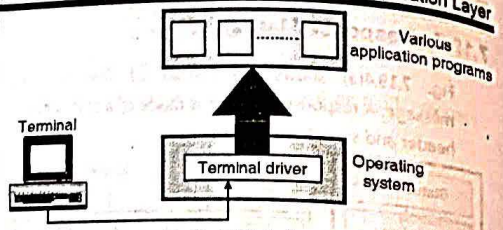
- TELNET was designed during those days when almost all the operating systems were operating on the time - sharing principle.
- In the time sharing environment there is a large central computer which supports all the users.
- All the processing is done by the central computer, and each user feels that it is a dedicated computer.
- The users can access all the common system resources, use all the programs or switch from one program to the other.

**Login :**

- In a system based on time sharing, every user must have an identification and a password for his authentication.
- Whenever a user wants to access the system he will log into the system with his user id and password.
- The system will check the password to allow only the authorized users to access the resources.
- The logic can be one of the following two types :
  1. Local login.
  2. Remote login.

**1. Local login :**

- The user login into a local time sharing system is called as local login. Fig. 7.21.1 illustrates the principle of local login.

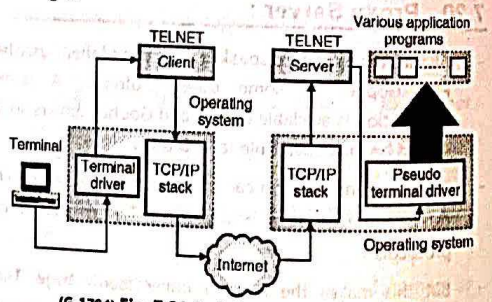



(G-1793) **Fig. 7.21.1 : Local login**

- The local login takes place in a step - by - step manner as follows :
  1. The user types at the keyboard of a terminal.
  2. The terminal driver accepts these keystrokes.
  3. It converts the keystrokes to characters.
  4. It passes the characters to operating system.
  5. The O.S. understands the combination of characters.
  6. It allows access of intended application to the user.

**2. Remote login :**

- The user will have to go for the remote login process when he wants to access an application program residing on a remote computer.
- He can do it using the TELNET client and server programs. Fig. 7.21.2 illustrates the principle of remote login.




(G-1794) **Fig. 7.21.2 : Principle of remote login**

- Remote login takes place in a step-by-step manner as follows :
  1. The user types at the keyboard of a terminal.
  2. The terminal driver at local O.S. accepts the characters but sends them to TELNET client without interpreting them.
  3. TELNET client converts them into NVT characters. NVT is Network Virtual Terminal. This is a universal character set.

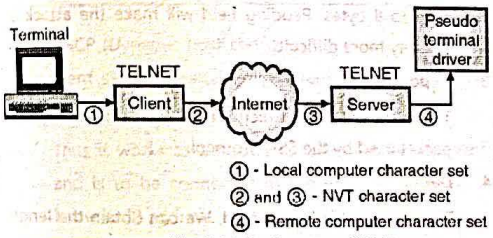




4. NVT characters are delivered to TCP/IP stack (local).
5. The NVT characters travel on the Internet and reach the TCP/IP stack of the remote machine.
6. The NVT characters are applied to the TELNET server which converts them appropriately so that the remote computer can understand them.
7. These characters are applied to a software called pseudo terminal driver.
8. The O.S. at the remote machine then passes the character to the intended application.

### 7.21.2 Network Virtual Terminal (NVT) :

- NVT character set is a universal interface defined by TELNET in order to ensure that a user can access any remote computer in this world. Fig. 7.21.3 illustrates the concept of NVT.




(G-1795) **Fig. 7.21.3 : Concept of NVT**

- The local computer character set is used for the communication between the user terminal and TELNET client.
- Then between the TELNET client and TELNET server the communication takes place using the NVT character set.
- And finally the remote computer character set is used for the communication between the TELNET server and the pseudo terminal driver as shown in Fig. 7.21.3.
- NVT has two sets of characters. One set is for the data and the other set is for control. Both have 8 bit characters.

### 7.21.3 Security Problems of TELNET :

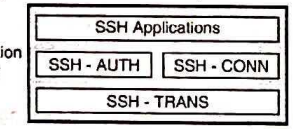
- TELNET is not a very secured system. It needs username and password for logging in. But it is not enough.
- A snooper software would be enough to capture the login name and password even if they are encrypted.

## 7.22 Secure Shell (SSH) :

- Secure Shell or SSH is another popular remote login application program.
- The underlaying transport program for SSH is TCP. This is similar to TELNET.
- However SSH has two advantages over TELNET :
  1. It is more secured than TELNET.
  2. It provides more services.
- There are two versions of SSH namely $SSH_1$ and $SSH_2$, out of which $SSH_2$ is being used. We will discuss $SSH_2$ in this section. Note that these two versions are not compatible to each other.

**SSH Components :**

- This is a proposed application layer protocol and as shown in Fig. 7.22.1, it has four components.




(G-1796) **Fig. 7.22.1 : SSH components**

- The four SSH components are :

| | |
|---|---|
| 1. SSH – TRANS. | 3. SSH – CONN. |
| 2. SSH – AUTH. | 4. SSH – Applications. |

**1. SSH – TRANS :**

- The long form is SSH – Transport Layer Protocol. TCP is not a secured protocol, therefore SSH makes use of a protocol which creates a secured channel on top of TCP.
- This new secured channel is an independent protocol called SSH – TRANS.
- When SSH is used, the client and server will first establish an unsecured TCP connection and then develop a secured layer over this by exchanging various security parameters.
- The SSH – TRANS protocol provides the following services :
  1. Confidentiality of the messages.
  2. Data integrity of the exchanged messages.
  3. Authentication of the server.
  4. Message compression.

**2. SSH – AUTH :**

- The second component of SSH is the SSH – AUTH i.e. SSH – Authentication protocol.
- This protocol is used to authenticate the client for the server after establishing a secure channel between client and the server.

**3. SSH – CONN :**

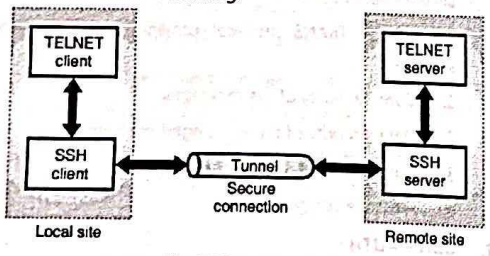
- The third component of SSH is SSH – CONN. i.e. SSH connection protocol.
- This piece of software is called for by the SSH once a secure connections has been established and authentication done.
- SSH – CONN performs the multiplexing as one of its services.
- It allows the client to create multiple logical channel over the secure channel established between the client and the server.

**4. SSH – Applications :**

- As soon as the connection establishment, authentication etc. is complete, the SSH connection can be used by multiple applications.
- Each application can create its own logical channel and make use of secure SSH connection. In addition to the remote login, the other applications that make use of SSH are : file transfer application. That is called as secure file transfer.

### 7.22.1 Port Forwarding :

- Port forwarding is one of the services provided by the SSH – protocol.
- The port forwarding mechanism can be used to access application programs which do not provide any security. e.g. TELNET or SMTP.
- Such application programs can use the secure channel created by SSH to create a tunnel to carry the messages as shown in Fig. 7.22.2. Therefore this mechanism is also called as SSH Tunneling.

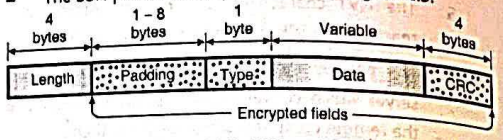



(G-1797) **Fig. 7.22.2 : Port forwarding**

- We can apply the port forwarding concept to change the insecure connection between TELNET client and TELNET server into a secure connection, as shown in Fig. 7.22.2.

### 7.22.2 SSH Packet Format :

- The SSH packet format is as shown in Fig. 7.22.3.




(G-1798) **Fig. 7.22.3 : SSH packet format**

- Description of various fields are as follows :

1. **Length :**

   This is a 4-byte long field which defines the length of the SSH packet which includes the type, the data and the CRC fields but does not include the length and the padding fields.

2. **Padding :**

   This is a variable length field. Its length can vary from 1-byte to 8-bytes. Padding field will make the attack on security more difficult.

3. **Type :**

   This is a 1-byte field which is used to specify the type of packet used by the SSH protocol.

4. **Data :**

   This is a variable length field. We can obtain the length of the data field by deducting the 5-bytes from the value of the length field.

5. **CRC :**

   This 4-bytes long field is used for error detection purpose.

### 7.22.3 Comparison of TELNET and SSH :

**Table 7.22.1 : Comparison of TELNET and SSH**

| Sr. No. | Parameter | TELNET | SSH |
|---|---|---|---|
| 1. | Port number | Uses TCP port number 23. | Uses TCP port number 22. |
| 2. | Security | Less secured than SSH. | Highly secured |
| 3. | Data format | Telnet sends the data in plain text. | SSH sends all the data in encrypted format. SSH uses secure channel to transfer data over the network. |



| Sr. No. | Parameter | TELNET | SSH |
|---|---|---|---|
| 4. | Authentication | No authentication mechanisms. | SSH uses public key encryption in order to authenticate the remote users. |
| 5. | Use of bandwidth | Low | High |
| 6. | Suitable for | Private networks | Public networks |
| 7. | Operating system | Used in linux and windows operating system | All popular operating system |

## 7.23 Host Configuration : DHCP :

- DHCP (Dynamic host configuration protocol) is the first client server application program that is used after a host is booted.
- Thus it works as a bootstrap when the host is booted and is to be connected to the Internet, but does not know its IP address.
- A computer that makes use of the TCP/IP suite must know its IP address.
- Along with its IP address it must also know the following information :
  1. Subnet mask of the computer.
  2. IP address of the router, so that it can communicate with other networks.
  3. IP address of the name server so that it can use the names instead of addresses.
- All this information can be saved in a configuration file and accessed by computer when booting takes place. This is known as host configuration process.
- But what will happen if the workstation is diskless or the computer is with a disc but it is being booted for the first time.
- If a computer is diskless, then it is possible to store the operating system and networking software in the ROM.
- But this information is not known to the manufacture and therefore cannot be stored in ROM.
- This information is dependent on the configuration of individual machine and it defines which network the machine is connected to.

### 7.23.1 Previously used Protocols :

- Now a days DHCP has become the formal protocol for host configuration. But the two protocols which were used earlier for the same purpose were RARP and BOOTP.
- RARP is Reverse Address Resolution Protocol and BOOTP stands for Bootstrap protocol.

### 7.23.2 DHCP :

- The Dynamic Host Configuration Protocol (DHCP) was devices by IETF in order to make the configuration automatic.
- Thus DHCP does not require an administrator to add an entry for each computer, to the database that a server uses.
- Instead, in DHCP a mechanism is provided for any computer to join a new network and obtain an IP address automatically with no manual intervention. This is known a plug and play networking.
- Thus DHCP allows the use of computers that, run server software as well as computers that run client software.
- When a computer that runs client software is shifted to a new network, it can use DHCP to obtain configuration information automatically.
- DHCP assigns a permanent address to a nonmobile computer that run server software.
- This address will not change when the computer **reboots.**
- To accommodate both type of computers, DHCP makes use of a client server approach.
- When a computer boots, it will broadcasts a DHCP Request. In response a server sends a DHCP Replay. An administrator can configure a DHCP server to have two types of addresses.
- First is the permanent address that are assigned to server computers, and second type is a pool of addresses which can be assigned on the basis of demand, when a computer boots and sends a request to DHCP.
- The DHCP find the configuration information by accessing its database If the database contains a specific entry for the computer then the server returns the information from the entry.

- However if there is no such entry exists for the computer, then the server chooses the next IP address from the pool and assigns it to the computer.

**What is DHCP ?**

- DHCP, as the name suggests, is a protocol used for dynamically configuring the hosts on a network, such as workstations, personal computers and printers.
- DHCP can help in assigning various types of information such as routing information, directory-services information and default web server and mail servers.
- However, the most important and commonly used information for which DHCP is used is the IP address and subnet mask information.
- DHCP was primarily designed for managing the network and the clients automatically. With DHCP, it is not necessary to configure the network and client information manually for individual hosts.
- In addition, DHCP can coexist with statically configured hosts with fixed IP addresses. DHCP can also carry out the allocation of certain configuration information to a host on a permanent basis.
- This protocol provides a four point information (IP address, subnet mask, IP address of router, IP address of name server) to a diskless computer or to a computer which is booted for the first time.
- It is a client / server protocol which is backward compatible to the BOOTP.

## 7.23.3 Advantages of DHCP :

- The use of DHCP on a network offers the following advantages :

1. It sets free the network administrator from the duties of setting up the configuration information, such as the IP address, the subnet mask, and the routing tables, manually. The DHCP simplifies network administration by doing these tasks automatically.
2. Avoids this and the sometimes the same IP address is assigned to two different hosts. The DHCP avoids this and the consequent malfunctioning of both the hosts from happening.
3. If the DHCP was not used, then the movement of computers from one network to another requires must be reconfigured. With DHCP, you can move the computers to different subnets or networks without the need to reconfigure them. In such situations, DHCP takes care of IP address assignment and other configuration details.
4. Mobile computers, such as laptops and palmtops, can easily get connected to different networks. They don't require reconfiguration any more as they get their configuration information from the DHCP server.
5. DHCP allocates IP addresses from a pool of IP addresses. In addition, when a computer gets disconnected, its released IP address is returned to the resource pool. Therefore, the possibility of having unused IP addresses are minimized.

## 7.23.4 Components of DHCP :

- The use of DHCP on a network requires the following three components :

1. **DHCP server :**
- It assigns the IP address and other information to the clients when they request for the information.

2. **DHCP client :**
- It communicates with the DHCP server to get the desired information regarding its configuration. This communication can take place when the computer starts.
- The user of the DHCP client can also initiate a DHCP client request to the DHCP server to renew its information.

3. **DHCP relay agent :**
- It is used to relay (forward) client requests to the DHCP server.
- This is required when the DHCP server is yet to assign the client an IP address.
- Without an IP address, a client cannot use IP routing on its own.
- A DHCP relay agent helps the client to communicate with the DHCP server when the client does not have an IP address.
- When a client starts, it has an IP address of 0.0.0.0. It sends a broadcast message containing its MAC address and the computer name.
- In response the DHCP server sends an offer message that contains the MAC address of the client, the IP address offered to that client, the lease period for which the IP address will remain valid and its own IP address.





- The lease period is the time duration for which a client can use the IP address that has been assigned to it by the DHCP server.
- You can configure a DHCP server to set the lease time.
- When the client receives the IP address, it accepts the offer and then broadcasts the message that it has accepted the offer.

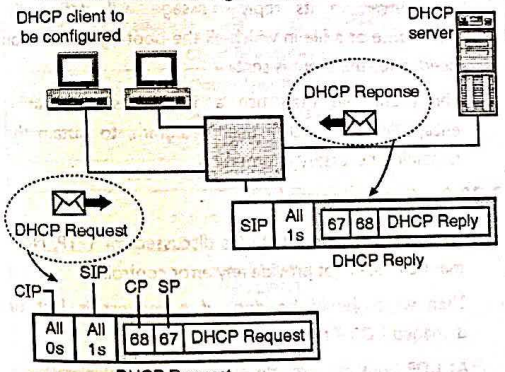
## 7.23.5 DHCP Operation :

- We will discuss the DHCP operation under two different operating conditions :

1. DHCP client and server on the same network.
2. DHCP client and server on different networks.

**Operation on the same network :**

- This situation is not a very common one. But sometimes the DHCP client and server happen to be on the same network as shown in Fig. 7.23.1.



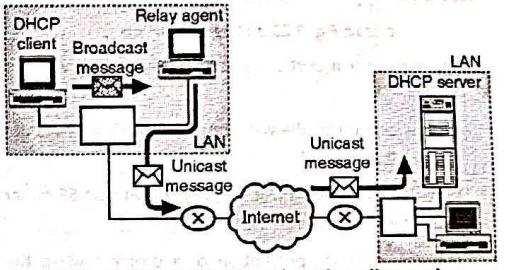

**(G-1789) Fig. 7.23.1 : Operation of DHCP when client and server are on the same network**

- The operation takes place as follows :

1. The DHCP server sends a passive open command on port 67 of UDP and waits for clients response.
2. The DHCP client sends an active open command on port 68 of UDP. This message is encapsulated in the UDP datagram with port 67 as destination port and port 68 as the source port. The UDP datagram is then encapsulated in an IP datagram. Note that the client at this time does not know its own IP address (i.e. the source address) and the server's IP address (destination address). Therefore the client uses an all zero address as source address and an all one address as destination address.
3. The server responds to this message by sending either a broadcast or a unicast message using port 67. It uses port 68 as the destination port. Broadcast address is used only for those system which do not allow the bypassing of ARP.

## 7.23.6 DHCP Operation on Different Networks :

- In this situation the DHCP client and server are on two entirely different networks, as shown in Fig. 7.23.2.



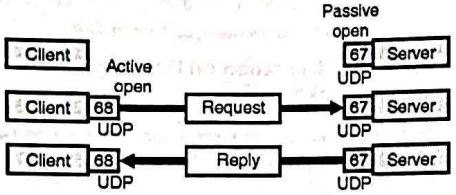

**(G-1790) Fig. 7.23.2 : DHCP operation when client and server are on different networks**

- In this situation a problem arises due to the broadcast nature of DHCP request. The client does not know the IP address of the server.
- Hence the DHCP request is a broadcast type (all 1s IP address).
- Any server does not allow the broadcast request to pass through it. So this request cannot reach the DHCP server.
- In order to solve this problem we can configure one of the hosts or router to operate as a relay agent as shown in Fig. 7.23.2.
- The relay agent knows the unicast address of the DHCP server. The relay will look for the broadcast request on port 67.
- As soon as it receives the broadcast request message, it encapsulates this message in a unicast datagram and sends it to the DHCP server.
- Such a unicast message is allowed to pass through by any router. Thus the request message reaches the DHCP server.
- The DHCP server sends its reply to the relay agent which in turn sends it to the DHCP client.

**Note :** In Fig. 7.23.2 only the message between the relay agent and client is it broadcast type. All the other messages are unicast types.

### 7.23.7 UDP Ports :

- The interaction between a client and DHCP server has been shown in Fig. 7.23.3.




**(G-1994) Fig. 7.23.3 : Use of UDP ports**

- The well known port 67 is used by the server, which is normal.
- But the client uses the well known port 68, which is not normal. It is unusual.
- Why does a client choose the well known port 68 rather than an ephemeral port ?
- The answer is for prevention of a problem when the reply from the server to client is of **broadcast** type. In order to understand the exact nature of the problem, let us assume that an **ephemeral port** is used instead of the well known port 68 and study its effect.
- Suppose host A on a network is using a DHCP client. It is using the ephemeral port say 2017 which we have chosen randomly.
- On the same network, there is another host B, which is using a DAYTIME client on ephemeral port 2017 which is accidentally the same.
- In this situation, the DHCP server sends a broadcast reply message with the destination port number 2017 and broadcast IP address $FFFFFFFF_{16}$.
- Every host has to open a packet which carries this destination IP address. Host A would find a message from an application program on ephemeral port 2017.
- Thus the DHCP client receives a **correct message** but the DAYTIME client receives an **incorrect message.**
- This confusion takes place due to the process of demultiplexing which is based on the **socket address.**
- Remember that a socket address is the combination of IP address and port number and both are same in this case.
- If a well known port (less than 1024) is used then the use of same two destination port numbers would be prevented.
- It would not be possible for host B to select port 68 as the ephemeral port due to the fact that ephemeral port numbers are greater than 1023.
- The final question is what happens if host B is also running the DHCP client ? The answer is that because of the same socket address, both the clients will receive the message.
- In order to handle such a situation, **the third identification number** is used to differentiate the clients.
- In DHCP this another number is called as the **transactional ID** and for each DHCP connection, it is chosen randomly. It is almost impossible that both the host choose the same transactional ID.

### 7.23.8 Using TFTP :

- Note that all the information needed by a client for booting purpose is not sent by the server.
- The server, in its reply message will define the **pathname** of a file in which all the booting information needed for the client is sent.
- The client can then use a TFTP message that is encapsulated in a UDP user datagram, to obtain the remaining necessary information.

### 7.23.9 Error Control :

- DHCP can use either UDP (as discussed) or TFTP. Note that UDP does not provide any error control.
- Then what should be done if a request is lost or damaged ? OR if the reply is damaged ?
- As UDP does not provide any error control, the DHCP should provide it.
- Two strategies could be used to achieve the goal of error control :
  1. Ask UDP to use checksum. The UDP has an option of using the checksum.
  2. Ask DHCP client to use timers alongwith the retransmission policy if DHCP request or reply gets damaged or lost.

### 7.23.10 Optimizations in DHCP :

- The DHCP protocol has following steps :
- The first step is that a computer **broadcasts** a DHCP discover message in order to find DHCP server, and the other step is that the computer selects one of the available DHCP servers that responds to its message and sends a request to that server.

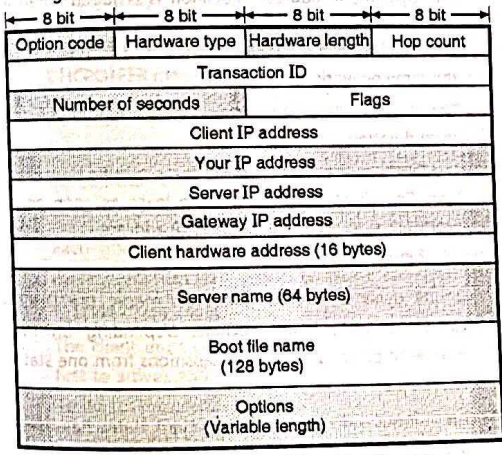




- To avoid a situation in which a computer follows both steps each time its boots or each time it needs to extend the lease, DHCP uses caching.
- When a computer discovers a DHCP server, the computer saves the address of that server in a cache on **permanent** storage (e.g. a disk file).
- Similarly, once an IP address has been allotted to it the computer saves the IP address in a cache. When a computer reboots, it uses the cached information to revalidate its former address.
- Doing so saves time and reduce network traffic.

### 7.23.11 Packet Format :

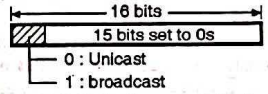
- The format of a DHCP packet has been shown in Fig. 7.23.4.




**(G-1995) Fig. 7.23.4 : DHCP packet format**

- Let us describe each field in the DHCP packet.

1. **Operation code :**
   - This is an 8 bit field which is used to define the type of DHCP packet. If this field contains (1) then the packet is **request type** and if this field contains (2) then the packet is **reply type.**
2. **Hardware type :**
   - This 8-bit field is used to define the type of physical network. An integer has been assigned to each type of network e.g. the value of this field is 1 for Ethernet.
3. **Hardware length :**
   - This is an 8-bit field which is used for defining the length of the physical address in bytes.
   - The value of this field is 6 for Ethernet because the physical address of Ethernet is 6 byte long.
4. **Hop count :**
   - This is an 8-bit field which is used for define the maximum number of hops a packet can travel.
5. **Transaction ID :**
   - This is a 32-bit or 4-byte long field which carries an integer in it.
   - The contents of this field are known as transaction **identification** and it is set by the client. This field is used for matching a reply with the request.
   - The same value is returned by the server in its reply packet.
6. **Number of seconds :**
   - This is a 16-bit field which is used to indicate the amount of time (in seconds) elapsed from the instant at which the client started to boot.
7. **Flag :**
   - This is a 16-bit long field, as shown in Fig. 7.23.5. Out of these 16 bits, only the leftmost bit is used and the remaining 15 bits are set to 0s.




**(G-1996) Fig. 7.23.5 : Format of the flagfield**

   - The leftmost bit is used to specify a forced broadcast reply (instead of unicast) from the server.
8. **Client IP address :**
   - This 4-byte long field is used to carry the client IP address. A "0" in this field indicates that the client does not have this information.
9. **Your IP address :**
   - This is also a 4-byte long field which is used to carry the clients IP address. This address is requested by the client and filled by the server in the reply message.
10. **Server IP address :**
    - This is also a 4-byte long field which contains the IP address of the server. This address is sent by the server in the reply message.
11. **Gateway IP address :**
    - This is a 4-byte or 32 bit long field that contains the IP address of a router which is filled in the reply message by the server.

**12. Client hardware address :**

- This is a 16-byte field which contains the physical address of the client.

**13. Server name :**

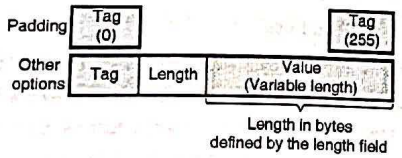
- This is a 64 byte long field which is filled on the optional basis by the server in a reply packet.
- This field consists of a null terminated string containing the domain name of the server.
- If no information about the server name is to be given, then the server should fill up this field with all zeros.

**14. Boot filename :**

- This is a 128-byte field which contains a null terminated string consisting of full pathname of the boot file.
- This path can be used by the client in order to obtain additional information about booting. This field is filled by the server in the reply message on the optional basis.
- If the server does not want to fill data in this field, then the entire field should be filled up with 0s.

**15. Options :**

- This is a 64-byte field which can be used for a dual purpose as follows :
  1. It is used to carry some additional information such as default router address or network mask.
  2. Or it is used to carry some specific information about the vendor.
- It is important to note that, the **options** field is used only in the **reply message.** The server makes use of a number called **magic cookie.**
- After finishing reading of the message the client searches for the magic cookie.
- If it is present, then the next 60 bytes data will correspond to **options.** Fig. 7.23.6 shows the format of the option. It consists of three fields as follows : a 1-byte **tag** field, a 1-byte **length** field and a variable length **value** field.




**(G-1997) Fig. 7.23.6 : Format of the options field**

- The function of the length field is to specify the length of the variable length **value** field and not of the whole option.

## 7.24 Configuration of DHCP :

- DHCP is capable of providing static and dynamic address allocation.

### 7.24.1 Static Address Allocation :

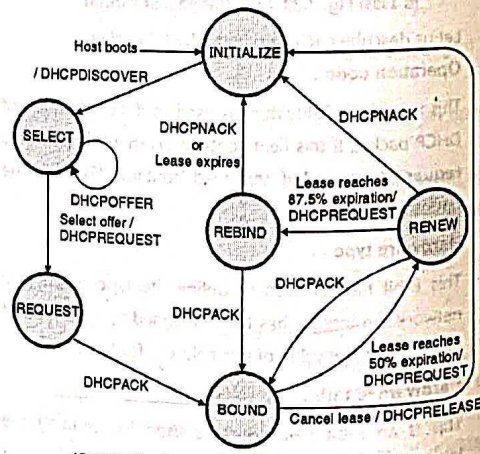
- In the static address allocation capacity, the database of DHCP server would bind the physical addresses to the IP addresses.

### 7.24.2 Dynamic Address Allocation :

- DHCP has another database in which a pool of available IP addresses is present. This database is used by DHCP when a client requests for a temporary IP address.
- In response to such a request the DHCP allots an unused IP address from the pool to the requesting client for a negotiable period of time.
- The dynamic IP address allocation is required when a host moves from one network to the other or when the host is frequently getting connected or disconnected to the same network.
- The dynamic IP address is assigned temporary for a short duration.
- The DHCP server issues a **lease** for a specific period of time. The client has to renew the lease as soon as it expires or stop using the assigned IP address.

### 7.24.3 Transition States :

- In the dynamic address allocation mode, the DHCP client acts as a state machine. Depending on the received message it will make transitions from one state to the other.
- The transition diagram with important states has been shown in Fig. 7.24.1.




**(G-582) Fig. 7.24.1 : State diagram of DHCP**





### 7.24.3.1 Address Acquisition States :

- When the DHCP is being used to obtain an IP address, a client is in one of six states. The state transition diagram in Fig. 7.24.1 shows events and messages that force a client to change state.
- When a client first boots, it enters the INITIALIZE state. In order to get an IP address, the client first contacts all DHCP servers in the local net.
- To do so, the client broadcasts a DHCPDISCOVER message and enters into the SELECT state.
- Because DHCP is an extension of BOOTP, the client sends the DHCPDISCOVER message in a UDP datagram with the destination port set to the BOOTP port (i.e., port 67).
- All DHCP servers which are connected to the local net receive the message, and those servers that have been programmed to respond to that particular client send a DHCPOFFER message.
- Thus, a client may receive zero response or multiple responses.
- When the client is in state SELECT, it collects DHCPOFFER responses from DHCP servers.
- Each offer contains configuration information for the client along with an IP address that the server is offering to lease to the client.
- The client must choose one of the responses (e.g., the first to arrive), and negotiate with the server for a lease. To do so, the client sends a DHCP REQUEST message to the server, and enters the REQUEST state.
- To acknowledge the request has been received and to start the lease, the server responds by sending a DHCPACK.
- When of the acknowledgement is received the client to moves to the BOUND state, where the client proceeds to use the address.
- To use DHCP, a host becomes a client by broadcasting a message to all servers on the local network.
- The host then collects lease offers from servers, selects one of the offers, and acknowledges the acceptance with the server.

### 7.24.3.2 Early Lease Termination :

- We think of the BOUND state as the normal state of operation; a client generally remains in the BOUND state when it is using the IP address allotted to it.
- If a client has secondary storage (e.g., a local disk), the client can store the IP address on it and request for the same address when it restarts again.
- In some cases, however, a client in the BOUND state may discover that it does not need an IP address anymore. For example, suppose a user attaches a portable computer to a network, uses DHCP to acquire an IP address and then uses TCP/IP to read electronic mail.
- The user may not know how long reading mail will require, or the portable computer may allow the server to choose a lease period.
- In any case, the minimum lease period specified by DHCP is of one hour. If after obtaining an IP address, the user discovers that no e-mail messages are waiting to be read, the user may choose to shutdown the portable computer and move to another location.
- If the clients does not needs a lease anymore, DHCP allows a client to terminate a lease and does not for the lease to expire.
- Early termination is especially important if only a small number of IP addresses are available at the server as compared to the number of computers that attach to the network. If each client terminates its lease as soon as the IP address is no longer needed, the server will be able to assign the address to another client.
- To terminate a lease early, a client sends a DHCPRELEASE message to the server. once an address is released the client is not allowed to use that address further.
- Thus, after transmitting the release message, the client must not send any other datagrams that use the address.
- As seen in the state transition diagram of Fig. 7.24.1 a host that sends a DHCPRELEASE leaves the BOUND state, and must start at the INITIALIZE a host state again before it can use IP.

### 7.24.3.3 Lease Renewal States :

- We have seen that when a client acquires an address, it moves to the BOUND state. After entering the BOUND state, the client sets three timers that control lease renewal, rebinding and expiration.
- A DHCP server can specify precise values for the timers when it allocates an address to the client; if the server does not specify timer values, the client uses the default values.

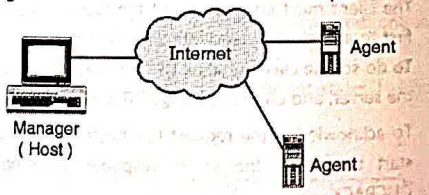
- The default value for the first timer is one-half of the total lease time. When the first timer expires, the client must attempt to renew its lease. To request a renewal, the client sends a DHCPREQUEST message to the server from which it had obtained the lease.
- The client then moves to the RENEW state and waits for a response. The DHCPREQUEST contains the IP address which the client is currently using, and asks the server to extend the lease time to use the same address.
- Similar to the initial lease negotiation, a client can request its preferred period for the extension, but the actual lease time allotment is controlled entirely by the server.
- A server can respond to a client's renewal request in one of two ways: it can instruct the client to stop using the address or it can allow the client to continued use.
- If it allows the client to continue then, the server sends a DHCPACK, which causes the client to return to the BOUND state and continue using the same IP address.
- The DHCPACK can also contain new values for the client's timers. If a server does not allow the client to continue using the same address then, the server sends a DHCPNACK (negative acknowledgement), which causes the client to stop using the address immediately and return to the INITIALIZE state.
- After sending a DHCPREQUEST message that requests an extension on its lease, a client remains in state RENEW and waits for a response from the server.
- If it does not receive any response then the server that granted the lease is either down or unreachable. To handle this situation, DHCP relies on a second timer, which was set when the client entered the BOUND state.
- The second timer expires after 87.5% of the lease period, and makes the client to move from state RENEW to state REBIND. When making the transition, the client assumes the old DHCP server is not available anymore and starts broadcasting a DHCPREQUEST message to any server on the local net.
- Any server configured to provide service to the client can respond positively (i.e. to extend the lease), or negatively (i.e. to deny further use of the same IP address).
- If it receives a positive response, the client returns to the BOUND state, and resets the two timers.
- If it receives a negative response, the client must move to the INITIALIZE state, must immediately stop using the IP address, and must acquire a new IP address before it can continue to use IP.
- After moving to the REBIND state, a client should have asked the original server and all servers on the local net for a lease extension. Sometimes a client does not receive any response from any server before its third timer, expires, the lease expires.
- The client must stop using the IP address, must move back to the INITIALIZE state, and begin acquiring a new address.

## 7.25 Simple Network Management Protocol (SNMP) :

- SNMP provides the frame work which is necessary for management of devices in an internet (any internet). It uses TCP/IP suite for the same.
- SNMP can be used to monitor and maintain an internet with the help of some fundamental operations.

### 7.25.1 Concept :

- SNMP is based on the concept of manager and agent. A manager is usually a host and agents are usually routers. The host controls and monitors the routers.
- Fig. 7.25.1 demonstrates the SNMP concept.




**(G-1531) Fig. 7.25.1 : SNMP concept**

- SNMP is an application-level protocol. It is designed in such a way that it can monitor devices made by different manufactures, used in different physical networks.
- SNMP can work successfully in a heterogeneous internet made up of LANs and WANs that are interconnected by different types of routers.

### 7.25.2 Managers and Agents :

- A **manager** is actually a management station that runs the SNMP client program.
- An agent is actually a managed station which is a router or a host that runs the SNMP server program.

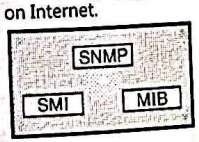




- The manager and agent simply interact with each other to achieve the management objective.
- The agent maintains a database which has its performance information. The manager can access these values in the database.
- The manager can also force the agent (router) to perform certain tasks. For example the manager can reboot the router remotely at any time by sending a packet to force a 0 value in the reboot counter of the router. (A router reboots itself when the contents of reboot counter go to zero).
- An agent can get involved in the management process by sending a trap (warning) message to the manager if it (agent) notices something wrong around it.
- Thus the SNMP manages on the basis of the following three basic ideas :
  1. A manager controls an agent by asking for information about behaviour of the agent.
  2. A manager can force an agent to perform certain actions.
  3. An agent can send warning messages (trap) to manager to report anything unusual around itself.

### 7.25.3 Management Components :

- SNMP uses two other protocols to perform its management tasks :
  1. Structure of Management Information (SMI)
  2. Management Information Base (MIB)
- So management on the Internet is performed through the simultaneous use of three protocols : SNMP, SMI and MIB Fig. 7.25.2 shows the components of network management on Internet.




**(G-1533) Fig. 7.25.2 : Components of network management on the Internet**

- These three protocols interact with each other in the following manner.

**1. Role of SNMP :**

- SNMP plays very specific roles in network management. Some of them are as follows :
  1. To define the format of the packet sent from manager to agent and vice versa.
  2. To interprete the results and create statistics by making use of other management software.
  3. To read and exchange the status (values) of objects (variables) in SNMP packets.

**2. Role of SMI :**

- The role of SMI is as follows :
  1. To define the general rules which can be used for naming objects, defining object types. These rules are required for using SNMP.
  2. To show how to encode objects and values.

**Note :** SMI does not define the number of objects that are to be managed by an entity. It also does not name the objects to be managed or define the relationship between objects and their values.

**3. Role of MIB :**

- The role of MIB is as follows :
  1. To define the number of objects and name them for each entity to be managed. This is to be done according to the rules defined by SMI.
  2. To associate a name to each numbered object.

### Review Questions

Q. 1 Explain in brief about the application layer.

Q. 2 Write a short note on providing services.

Q. 3 Explain about the standard and nonstandard protocols at the application layer.

Q. 4 Explain in brief client-server paradigm.

Q. 5 State the problems and applications of client-server paradigm.

Q. 6 Explain the P2P paradigm.

Q. 7 State the merits, demerits and applications of P2P paradigm.

Q. 8 Explain the term API and state its types.

Q. 9 Define a socket and state its role.

Q. 10 Draw and explain the structure of www.

Q. 11 Explain the non-persistent and persistent connections in HTTP.

Q. 12 Write a note on : HTTP messages.

Q. 13 What is FTP ? Explain the communication in FTP.

Q. 14 Write a note on E-mail.

Q. 15 Compare SMTP and HTTP.

Q. 16 Write a note on message access agents.

Q. 17 Briefly discuss the following terms, emphasis more on implementation details :

(a) DNS (b) Mail server

Q. 18 What is domain name system ? How does it work ?

Q. 19 Describe a typical resolution process in DNS.

Q. 20 Explain how file transfer protocol clients servers are configured. Discuss the various FTP and telnet commands.

Q. 21 Why do HTTP, FTP, SMTP, POP3 and IMAP run on top of TCP rather than UDP ?

**Ans. :**

– All these protocols require a reliable end to end connection oriented service which they can get only from TCP and not from UDP.

Q. 22 Explain SSH.

Q. 23 Explain TELENT.

Q. 24 Explain DHCP.

□□□

Unit VI

Chapter 8

# Medium Access Control

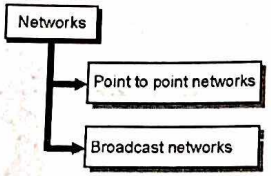
## Syllabus

**Channel allocation :** Static and dynamic, **Multiple Access Protocols :** Pure and slotted ALOHA, CSMA, WDMA, IEEE 802.3 standards and frame formats, CSMA/CD, Binary exponential back-off algorithm, Fast ethernet, Gigabit ethernet, IEEE 802.11a/b/g/n and IEEE 802.15 and IEEE 802.16 standards, Frame formats, CSMA/CA.

## Chapter Contents

## 8.1 Introduction :

- We can classify the networks into two categories as shown in Fig. 8.1.1.

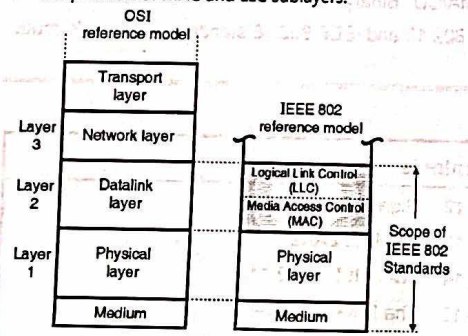



**(G-264) Fig. 8.1.1 : Classification of networks**

- In this chapter, we are going to discuss the broadcast networks and their protocols. The broadcast channels are also called as multi-access channels or random access channels.
- In the broadcast networks the most important point is the criteria by which we decide, who is allowed to use the common channel when more than one user want to use it. A protocol is used to make this decision.
- Such a protocol belongs to a sublayer of data link layer called the MAC (Medium Access Control) sublayer. The MAC sublayer is very important in LANs because it is a broadcast network.

### 8.1.1 MAC and LLC Sublayers :

- Fig. 8.1.2 shows the layered OSI model (partial) to show the position of MAC and LLC sublayers.




**(G-265) Fig. 8.1.2 : IEEE 802 protocol layers compared to OSI model**

- We will discuss the broadcast protocols corresponding to the lower layers (1 and 2) of the OSI model as shown in Fig. 8.1.2.
- Fig. 8.1.2 relates the LAN protocols with the OSI architecture.
- This architecture was developed by IEEE 802 committee and it has been accepted as LAN standard.
- It is called as IEEE 802 reference model. Let discuss this model layer by layer.

**Functions of Media Access Control (MAC) sublayer :**

- The functions of MAC sublayer are as follows :
  1. To perform the control of access to media.
  2. It performs the unique addressing to stations directly connected to LAN.
  3. Detection of errors.

**Functions of Logical Link Control (LLC) sublayer :**

- The functions of LLC sublayer are as follows :
  1. Error recovery.
  2. It performs the flow control operation.
  3. User addressing.

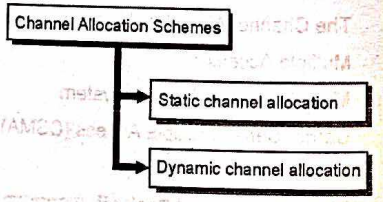
## 8.2 The Channel Allocation Problem :

**Channel allocation :**

- In a broadcast network, the single communication channel is to be allocated to only one transmitting user at a time.
- The other users connected to this medium should wait till the transmission medium becomes idle again, otherwise, the transmitted packets from multiple sources would collide with each other and lost.
- This is called as channel allocation.

**Channel allocation schemes :**

- There are two different schemes used for channel allocation as shown in Fig. 8.2.1.




**(G-266) Fig. 8.2.1 : Channel allocation schemes**

- The channel allocation schemes are :
  1. Static channel allocation and
  2. Dynamic channel allocation





### 8.2.1 Static Channel Allocation :

- The traditional way of allocating a single channel, among many users is by means of frequency division multiplexing (FDM).
- The Frequency Division Multiplexing (FDM) and Time Division Multiplexing (TDM) are the examples of static channel allocation.
- In these methods either a fixed frequency band or a fixed time slot is allotted to each user.
- Thus, either the entire available bandwidth or the entire time is shared.

**Problems in static allocation :**

- Some of the major problems with the static channel allocation schemes are as follows :

**1. Wastage of channel bandwidth :**

- The problem in the static channel allocation methods is that if all the N users are not using the channel, the channel bandwidth is wasted.

**2. Lack of channel bandwidth :**

- If there are more than N users who want to use the channel, they cannot do so for the lack of bandwidth.

**3. Low efficiency :**

- For a small number or users and light traffic the static FDM is an efficient method of allocation but its performance is poor for large number of users.

**4. Poor performance for bursty traffic :**

- The static channel allocation has a poor performance with bursty traffic.
- Therefore, generally dynamic channel allocation is used, for computer networks where the traffic is of bursty nature.

**Performance of static allocation schemes :**

- To see the poor performance of static channel, let us consider an example of a FDM system. Let the mean time delay be (T) for a channel of capacity C bps, with an arrival rate of $\lambda$ frames/sec.
- Each frame having a length drawn from an exponential probability density function with mean $1/\mu$ bits/frame is given as,

$$T = \frac{1}{\mu C - \lambda}$$

- If the single channel is divided into N independent sub-channels the above equation is modified as follows :

$$T_{FDM} = \frac{1}{\mu(C/N) - (\lambda/N)} = \frac{N}{\mu C - \lambda}$$

$$T_{FDM} = NT$$

- From the above equation, it is clear that the mean delay using FDM becomes worse with increase in the number of users N.
- The dynamic channel allocation is used, for computer networks where the traffic is of bursty nature.

### 8.2.2 Dynamic Channel Allocation :

- In this method either a fixed frequency or fixed time slot is not allotted to the user.
- The user can use the single channel as per his requirement.

**Assumptions :**

- Following assumptions are made for the implementation of dynamic channel allocation method :

**1. Station model :**

- This model consists of N independent stations such as a PC, computer etc. which can generate frames for transmission.

**2. Single channel :**

- A single channel is available for all communication.

**3. Collision :**

- If frames are transmitted at the same time by two or more stations, there is an overlap in time and the resulting signal is garbled.
- This is called as collision.

**4. Continuous or slotted time :**

- There is no master clock used to divide time into discrete time intervals. So frames can begin at any random instant.
- This is continuous time. For a slotted time, the time is divided into discrete time slots.

**5. Carrier or No carrier sense :**

- Stations sense the channel before transmission or they directly transmit without sensing the channel.

### 8.2.3 Comparison between Static and Dynamic Channel Allocation :

**Table 8.2.1 : Comparison between static and dynamic channel allocation**

| Sr. No. | Parameter | Static channel allocation | Dynamic channel allocation |
|---|---|---|---|
| 1. | Performance | Better under heavy traffic | Better under low/moderate traffic |
| 2. | Suitable | For large networks | For small / medium size networks |
| 3. | Flexibility | Low | High |
| 4. | Control | Centralized | Centralized, or distributed |
| 5. | Call set-up delay | Low | High |
| 6. | Application | Long duration voice calls | Voice calls of short duration data transmission |
| 7. | Signaling load | Low | High |
| 8. | Examples | FDM, TDM | CSMA/CA, CSMA/CD |

## 8.3 Multiple Access :

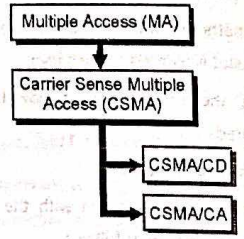
- When a number of stations (users) use a common link of communication system we have to use a multiple access protocol in order to coordinate the access to the common link.

**Types of Multiple Access Techniques :**

- The three techniques used to deal with the multiple access problem are as follows :
  1. Random Access
  2. Controlled Access
  3. Channelization

### 8.3.1 Random Access :

- In the random access technique there is no control station.
- Each station will have the right to use the common medium without any control over it.
- With increase in number of stations, there is an increased probability of **collision** or **access conflict.**
- The collisions will occur when more than one user tries to access the common medium simultaneously.
- As a result of such collisions some frames can be either modified (due to errors) or destroyed. In order to avoid collisions, we have to set up a procedure.
- The evolution of the random access methods is shown in Fig. 8.3.1.




(G-267) **Fig. 8.3.1 : Evolution of random access methods**

### 8.3.2 Evolution of Random Access Methods :

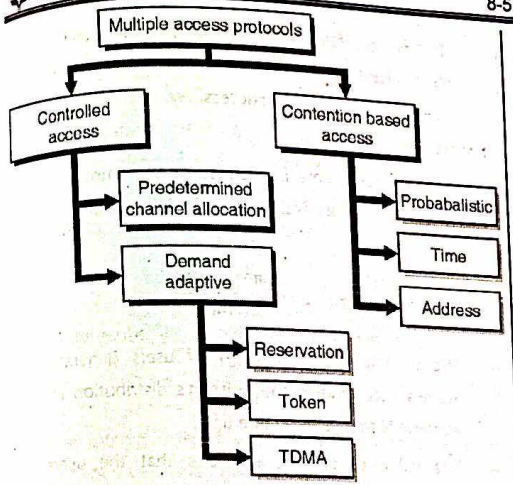
- The first method in the evolution ladder of Fig. 8.3.1, known as ALOHA used a simple procedure called multiple access (MA).
- It was improved to develop the carrier sense multiple access (CSMA).
- The CSMA further evolved into two methods namely CSMA/CD (CSMA with collision detection) and CSMA/CA (CSMA with collision avoidance) which avoids the collisions.

### 8.3.3 Classification of Multiple Access Protocols :

- Multiple access protocols, determine the allocation of channel among the stations.
- These protocols can be broadly divided into two classes :
  1. Controlled access
  2. Contention based access.
- Fig. 8.3.2 explains these two classes and their further sub-division.
- The controlled access protocols are characterized by a collision free access to the channel.
- That means the stations are co-ordinated in such a way that two or more stations never attempt to transmit simultaneously.








(G-1544) **Fig. 8.3.2 : Classification of Multiple Access Protocols**

- The controlled access protocols are further classified into two types :
  1. Demand adaptive
  2. Predetermined channel allocation.
- The PCA (TDMA) protocols allocate the channel to the stations in a static manner (on fixed time sharing basis).
- The demand adaptive protocols try to allocate the channel as per the demands of the stations.
- Both PCA and demand adaptive protocols are inefficient if there are large number of stations having bursty message transmissions.
- The second broad class of multiple access protocols is known as contention-based protocols.
- These are further classified into three types, probabilistic, them based and address based protocols.
- The ALOHA and CSMA protocols (Sections 8.4 and 8.5) belong to the contention based protocols.

## 8.4 Multiple Access ALOHA System :

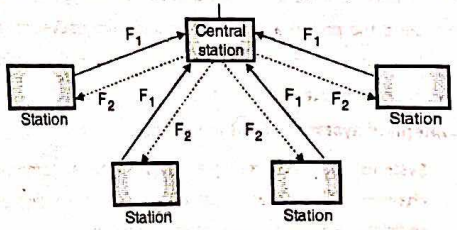
**ALOHA System :**

- Systems in which multiple users share a common channel in a way that can lead to conflicts are widely known as **contention systems.**
- The ALOHA system is a contention protocol which was developed at the University of Hawaii in the early 1970's by Norman Abramson and his colleagues.
- The ALOHA system has two versions :
  1. Pure ALOHA – Does not require global time synchronisation.
  2. Slotted ALOHA – Requires time synchronisation.

### 8.4.1 Pure ALOHA :

**Principle :**

- It works on a very simple principle. Essentially it allows for any station to broadcast at any time.
- If two signals collide, each station simply waits a random time and try again.
- Collisions are easily detected. As shown in the Fig. 8.4.1, when the central station receives a frame it sends an acknowledgement on a different frequency.



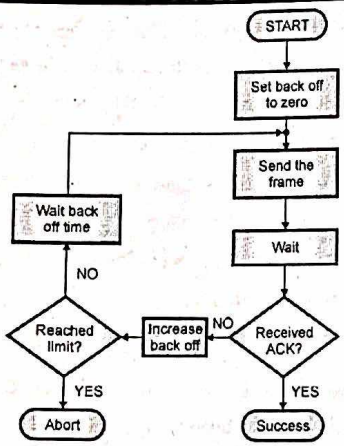

$F_1$ = Broadcast frequency from the individual stations.
$F_2$ = Broadcast frequency from the central station.

(G-268) **Fig. 8.4.1 : Pure ALOHA system**

- If a user station receives an acknowledgement it assumes that the transmitted frame was successfully received and if it does get an acknowledgement it assumes that collision had occurred and is ready to retransmit.
- The advantage of pure ALOHA is its simplicity in implementation but it's performance becomes worse as the data traffic on the channel increases.

**Protocol Flow Chart for ALOHA :**

- Fig. 8.4.2 shows the protocol flow chart for ALOHA.




**(G-269)Fig. 8.4.2 : Protocol flow chart for ALOHA**

**Explanation :**

- A station which has a frame ready for transmission will send it and waits for some time. If it receives an acknowledgement then the transmission is successful.
- Otherwise the station uses a **back-off** strategy, and will send the packet again. After sending the packet, many times if there is no acknowledgement then the station aborts the idea of transmission.

**Contention system and Retransmission :**

- Systems in which multiple users share a common channel in such a way that can lead to a conflict or collision are known as the contention systems.
- Whenever two frames try to occupy the channel at the same time, there is bound to be a collision and both will be garbled.
- Retransmission is essential for all the destroyed frames.

## 8.4.2 Efficiency of an ALOHA System :

- Efficiency of an ALOHA system is that fraction of all transmitted frames which escape collisions i.e. which do not get caught in collisions.
- Consider ∞ number of interactive users at their computers (stations).
- Each user is either typing or waiting. Initially all of them are in the typing state.
- When a user types a line, the user stops and waits. The station then transmits a frame containing this line and checks the channel to confirm the success.
- If it is successful then the user will start typing again, otherwise the user waits and its frame is retransmitted many time till it is sent successfully.

**Frame time :**

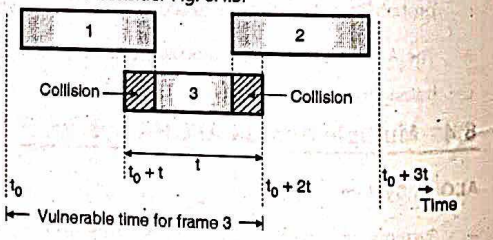
- Let the frame time be defined as the amount of time required to transmit the standard fixed length frame. Note that

$$\text{Frame time} = \frac{\text{Frame length}}{\text{Bit rate}}$$

- We assume that ∞ number of users generate new frames according to the Poisson's distribution with an average N frames per frame time.
- The value of $N > 1$ indicates that the users are generating frames at a rate higher than that can be handled by the channel.
- So most of the frames will face collision. Hence $0 < N < 1$ in order to reduce number of collisions. Let there be k transmission attempts (including retransmissions) per frame time.
- The probability of k transmissions per frame time is also Poisson.
- Let the mean of number of transmissions be G per frame time. So $G \geq N$.
- At low load $N \approx 0$ there will be less number of collisions so less number of retransmissions and $G \approx N$.
- With increase in load there are many collisions so $G > N$. Combining all these we can say that for all the loads the throughput is given by,

$$S = GP_0$$

- Where $P_0$ = Probability that a frame does not suffer a collision. Consider Fig. 8.4.3.




**(G-270) Fig. 8.4.3**

- What is the condition for frame 3 in Fig. 8.4.3 to arrive undamaged without collision ?





- Let t = Time required to send a frame. If frame 1 is generated at any instant between $t_0$ to $(t_0 + t)$ then it will collide with frame 3.
- Similarly any frame (2) generated between $(t_0 + t)$ and $(t_0 + 2t)$ also collides with frame 3. As per Poisson's distribution, the probability of generating k frames during a given frame time is given by,

$$P[k] = \frac{G^k e^{-G}}{k!}$$

- So the probability of generating zero frames i.e. k = 0 is

$$P_0 = \frac{G^0 e^{-G}}{0!} = e^{-G}$$

- If an interval is two frame time long, the mean number of frames generated during that interval is 2G.
- The probability that no other frame is transmitted during the Vulnerable period (time when collision can take place) is,

$$P_0 = e^{-2G}$$

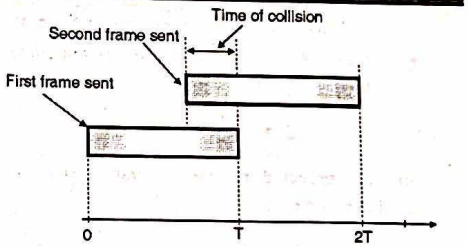
- But throughput $S = G P_0$

$$\therefore \quad S = Ge^{-2G}$$

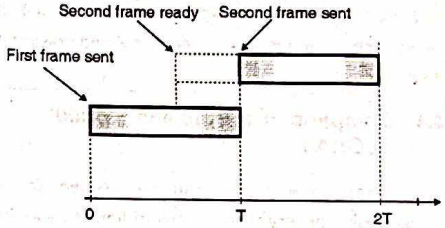
- Fig. 8.4.5 shows the relation between the offered traffic G and the throughput S. It shows that the maximum throughput occurs at G = 0.5 and $S_{max} = 0.184$. So the best possible channel utilization is on 18.4 percent.

## 8.4.3 Slotted ALOHA :

- To overcome the disadvantage of the pure ALOHA system (of low capacity) Robert published a method for doubling the capacity of traffic on the channel.
- In this method it was proposed that the time be divided up into discrete intervals and each interval correspond to one frame.
- This method requires that the users agree on the slot boundaries. In this method for achieving synchronization one special station emits a pip at the start of each interval, like a clock.
- This method is known as the slotted ALOHA system. Collisions occur if any part of two transmission overlaps.
- Suppose that T is time required for one transmission and that two stations must transmit.
- The total time required for both stations to do so successfully is 2T as shown in Fig. 8.4.4.




**(a) Transmission using pure ALOHA**




**(b) Transmission using slotted ALOHA**

**(G-271) Fig. 8.4.4**

- In case of pure ALOHA allowing a station to transmit at arbitrary times can waste time up to 2T.
- As an alternative, in the slotted ALOHA method the time is divided into intervals (slots) of T units each and require each station to begin each transmission at the beginning of a slot.
- In other words, even if station is ready to send in the middle of a slot, it must wait until the beginning of the next one as shown in Fig. 8.4.4(b).
- In this method a collision occurs when both stations become ready in the same slot. Slotted ALOHA is thus a discrete time system whereas pure ALOHA is a continuous time system.
- The Vulnerable period has been reduced to half that of pure ALOHA, the throughput for slotted ALOHA is given by,

$$S = Ge^{-G}$$

- The maximum throughput corresponds to G = 1 and it is given by $S_{max} = 1/e = 0.368$ as shown in Fig. 8.4.5.
- So for a slotted ALOHA with G = 1 the probability of success is 37%.
- The probability of empty slots is,

$$P(k) = \frac{G^k e^{-G}}{k!}$$

For G = 1 and k = 0 we get P (k = 0) = 0.368.

And the probability of collisions is 26 %.

- The probability of transmission requiring exactly k attempts (i.e. k – 1 collisions followed by one success) is given by,

$$P_k = e^{-G}(1-e^{-G})^{k-1}$$

- And the expected number of transmissions E per carriage return typed is

$$E = e^{G}$$

**Conclusion :** As E depends exponentially on G, with a small increase in G, there is a large increase in E and drastic fall in performance.

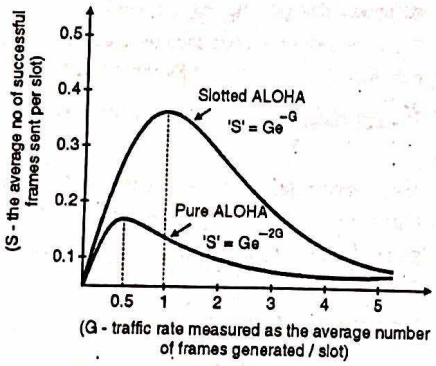
### 8.4.4 Comparison of Pure and Slotted ALOHA :

- A mathematical model can be created for the relationship between the number of frames transmitted and the number of frames transmitted successfully.
- Let G represent the traffic measured as the average number of frames generated per slot. Let S be the success rate measured as the average number of frames sent successfully per slot.
- The relationship between G and S for both pure and slotted ALOHA is given as follows :

  Pure ALOHA → $S = Ge^{-2G}$

  Slotted ALOHA → $S = Ge^{-G}$

  Where e is the mathematical constant = 2.718.
- From the above equation a success rate curve for pure and slotted ALOHA can be plotted as shown in Fig. 8.4.5.




**(G-272)Fig. 8.4.5 : Comparison of pure and slotted ALOHA**

- As seen in the Fig. 8.4.5 both graphs have the same shape.
- If G is small so is S, which means that if few frames are generated few frames will be transmitted successfully.
- As G increases so does S but upto a certain point. As G continues to increase S approaches to 0 which means that if more frames are generated there will be more collisions and the success rate will fall to 0.
- Similarly for pure ALOHA the maximum occurs at G = 0.5 for which S = 1/2e = 0.184 which means the rate of successful transmissions is approximately 18.4%.
- As seen from the graph the maximum for slotted ALOHA occurs at G = 1 for which S = 1/e = 0.368.
- In other words the rate of successful transmissions is approximately 0.368 frames per slot time or 37% of the time will be spent on successful transmissions.
- Hence the slotted ALOHA has a double throughput efficiency than the pure ALOHA system.
- The maximum utilization achievable using CSMA can be increased much beyond that obtainable using ALOHA or slotted ALOHA.
- The maximum utilization is dependent on length of the frame and on the propagation time. With increase in the length of the frame or reduction in the propagation time the utilization gets improved.

**Ex. 8.4.1 :** A group of N users share 56 kbps pure ALOHA channel. Each station outputs 1000 bits frame on an average of once 100 sec. Even if the previous has not yet been sent (buffered) what is maximum value of N.

**Soln. :**

**For pure ALOHA :**

- The maximum throughput = 0.184
- ∴ The maximum usable channel bandwidth is given by,

$$R = 0.184 \times 56 \text{ kbps} = 10.3 \text{ kbps}$$

$$\text{Transmission rate of stations} = \frac{1000 \text{ bits}}{100 \text{ sec}} = 10 \text{ bits/sec.}$$

- Let N be the number of stations that can use the channel.

$$\therefore N = \frac{R}{10 \text{ bits/sec}} = \frac{10.3 \text{ kbps}}{10}$$

$$\therefore N = 1030 \quad \text{...Ans.}$$

**Ex. 8.4.2 :** ALOHA protocol is used to share 56 kbps satellite channel. If each packet is 1000 bits long find maximum throughput in pockets/sec.





**Soln. :**

**Given :** Rate of transmission = 56 kbps = 56000 bps

Frame length = 1000 bits

**1. For pure ALOHA :**

$$\therefore \text{Number of frames/sec} = \frac{56000 \text{ bits}}{1000 \text{ bits/frame}}$$

$$= 56 \text{ frames/sec}$$

- The maximum throughput for pure ALOHA = 0.184

$$\therefore \text{Throughput} = 56 \times 0.184$$

$$= 10.304 \text{ frames/sec} \quad \text{...Ans.}$$

**2. For slotted ALOHA :**

Maximum throughput = 0.368

$$\therefore \text{Throughput} = 0.368 \times 56$$

$$= 20.608 \text{ frames/sec.} \quad \text{...Ans.}$$

**Ex. 8.4.3 :** Calculate maximum throughput possible for slotted ALOHA and pure ALOHA for a radio system with 9600 bps channel used for call setup request to base station. Take frame length of 200 bits.

**Soln. :**

**Given :** Rate of transmission = 9600 bps,

Frame length = 200 bits.

**To find :** Throughput for slotted ALOHA and pure ALOHA.

$$\therefore \text{Number of frames per second} = \frac{\text{Rate transmission}}{\text{Frame length}}$$

$$= \frac{9600}{200} = 48 \text{ frames/sec.}$$

- The maximum throughput for slotted ALOHA system is 0.68.

$$\therefore \text{Throughput} = 0.368 \times \text{Number of frames/sec.}$$

$$= 0.368 \times 48$$

$$= 17.664 \text{ frames/sec.} \quad \text{...Ans.}$$

- The maximum throughput for pure ALOHA system is 0.184.

$$\therefore \text{Throughput} = 0.184 \times \text{Number of frames/sec.}$$

$$= 0.184 \times 48$$

$$= 8.832 \text{ frames/sec.} \quad \text{...Ans.}$$

**Ex. 8.4.4 :** 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 1) 1000 frames per second, 2) 500 frames per second, 3) 250 frames per second.

**Soln. :**

1. In slotted ALOHA,

   for G = 1 maximum throughput = 0.368

   $\therefore$ Throughput = $0.368 \times 1000 = 368$ frames/sec.

2. In this case G = 500 frames/sec. i.e. $\frac{1}{2}$

$$S = Ge^{-4} = \frac{1}{2}e^{-(1/2)} = 0.3032$$

Throughput = $0.3032 \times 500 = 151.63$

$\cong$ 151 frames/sec.

3. In this case G = 250 frames/sec. i.e. $\frac{1}{4}$

$$\therefore S = Ge^{-4} = \frac{1}{4} \times e^{-(1/4)} = 0.1947$$

$\therefore$ Throughput = $0.1947 \times 250 = 48.67$

$\cong$ 49 frames/sec.

**Ex. 8.4.5 :** Measurement of slotted ALOHA channel with an infinite number of users show that 20 % slots are idle.

1. What is the channel load ?
2. What is the throughput ?
3. Is the channel underload or overload ? Show with graph.

**Soln. :**

**Given :** $P_0$ = 20 % i.e. 0.2, Type : Slotted ALOHA

**To find :** 1. Channel load G

2. Throughput S.

3. Decide the status of the channel

**1. Channel load (G) :**

For the slotted ALOHA, $P_0 = e^{-G}$

$$\therefore 0.2 = e^{-G}$$

$$\therefore G = 1.6094 \quad \text{...Ans.}$$

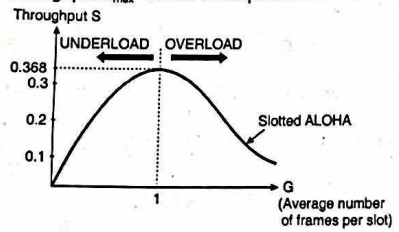
**2. Throughput (S) :**

$$S = Ge^{-G} = P_0 G = 0.2 \times 1.6094$$

$$\therefore \quad S = 0.3218 \qquad \text{...Ans.}$$

**3. Status of the channel :**

- From Fig. P. 8.4.5 it is evident that the maximum throughput $S_{max} = 0.368$ corresponds to G = 1.




**(G-338) Fig. P. 8.4.5 : Graph for slotted ALOHA**

- Since the value of G = 1.6094 which is greater than 1, the channel is overloaded.

## 8.5 Carrier Sense Multiple Access (CSMA) :

- The CSMA protocol operates on the principle of carrier sensing.
- In this protocol, a station listens to see the presence of transmission (carrier) on the cable and decides to act accordingly.

**Non-Persistent CSMA :**

- In this scheme, if a station wants to transmit a frame and it finds that the channel is busy (some other station is transmitting) then it will wait for fixed interval of time.
- After this time, it again checks the status of the channel and if the channel is free it will transmit.

**1-Persistent CSMA :**

- In this scheme the station which wants to transmit, continuously monitors the channel until it is idle and then transmits immediately.
- The disadvantage of this strategy is that if two stations are waiting then they will transmit simultaneously and collision will take place. This will then require retransmission.

**P-Persistent CSMA :**

- The possibility of such collisions and retransmissions is reduced in the p-persistent CSMA.
- In this scheme all the waiting stations are not allowed to transmit simultaneously as soon as the channel becomes idle.
- A station is assumed to be transmitting with a probability "p".
- For example if p = 1/6 and if 6 stations are waiting then on an average only one station will transmit and others will wait.

### 8.5.1 Carrier Sense Multiple Access/Collision Detection (CSMA/CD) :

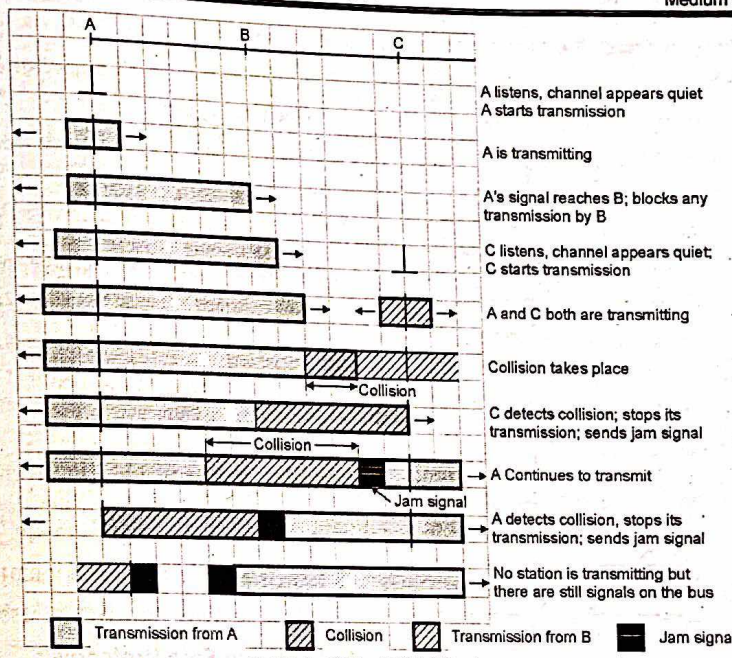
- The CSMA/CD specifications have been standardized by IEEE 802.3 standard. It is a very widely used MAC protocol.

**Media access control :**

- The problem in CSMA explained earlier is that a transmitting station continues to transmit its frame even though a collision occurs.
- The channel time is unnecessarily wasted due to this.
- In CSMA/CD, if a station receives other transmissions when it is transmitting, then a collision can be detected as soon as it occurs and the transmission time can be saved.
- As soon as a collision is detected, the transmitting stations release a jam signal.
- The jam signal will alert the other stations.
- The stations then are not supposed to transmit immediately after the collision has occurred.
- Otherwise there is a possibility that the same frames would collide again.
- After some "back off" delay time the stations will retry the transmission.
- If again the collision takes place then the back off time is increased progressively.
- A careful design can achieve efficiencies of more than 90% using CSMA/CD.
- This scheme is as shown in Fig. 8.5.1



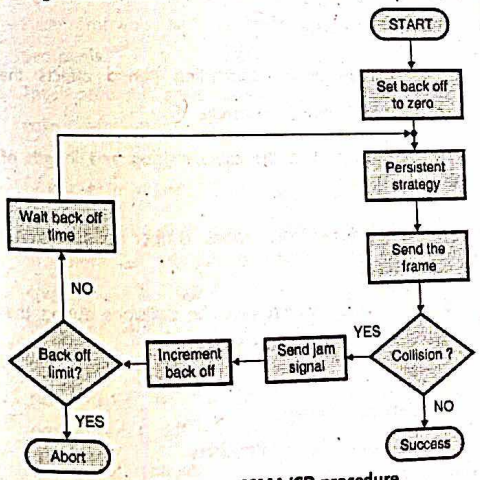





**(G-273) Fig. 8.5.1 : CSMA/CD scheme**

### 8.5.2 CSMA/CD Procedure :

- Fig. 8.5.2 shows a flow chart for the CSMA/CD protocol.




**(G-276) Fig. 8.5.2 : CSMA/CD procedure**

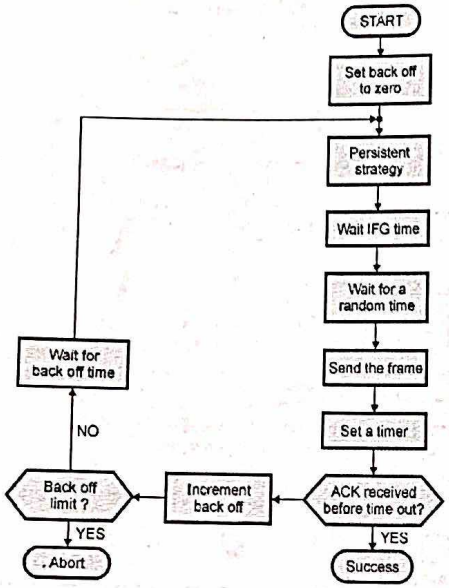
**Explanation :**

- The station that has a ready frame sets the back off parameter to zero.
- Then it senses the line using one of the persistent strategies.
- It then sends the frame, if there is no collision for a period corresponding to one complete frame, then the transmission is successful.
- Otherwise (in the event of collision) the station sends the jam signal to inform the other stations about the collision.
- The station then increments the back off time and waits for a random back off time and sends the frame again.
- If the back off has reached its limit then the station aborts the transmission. CSMA/CD is used for the traditional Ethernet.
- CSMA/CD is an important protocol. IEEE 802.3 (Ethernet) is an example of CSMA/CD. It is an international standard.
- The MAC sublayer protocol does not guarantee reliable delivery.
- Even in absence of collision the receiver may not have copied the frame correctly.

### 8.5.3 CSMA / CA :

- The long form of CSMA/CA is CSMA protocol with collision avoidance.

- Fig. 8.5.3 shows the flow chart explaining the principle of CSMA / CA.




**(G-277)Fig. 8.5.3 : CSMA/CA procedure**

- The station ready to transmit, senses the line by using one of the persistent strategies. As soon as it find the line to be idle, the station waits for a time equal to an IFG (Inter-frame gap).
- It then waits for some more random time and sends the frame.
- After sending the frame, it sets a timer and waits for the acknowledgement from the receiver. If the acknowledgement is received before expiry of the timer, then the transmission is successful.
- But if the transmitting station does not receive the expected acknowledgement before the timer expiry then it increments the back off parameter, waits for the back off time and senses the line again. CSMA/CA completely avoids the collision.

**Ex. 8.5.1 :** Consider building a CSMA/CD network running at 1 Gbps over a 1 km cable with no repeaters. The signal speed in the cable is 2,00,000 km/sec, what is the minimum frame size ?

**Soln. :**

**Given :** Bit rate $R = 1 \times 10^9$ bits/sec, No repeaters used.

Length $L = 1$ km $= 1 \times 10^3$ m

Speed $v = 2{,}00{,}000$ km/s $= 2 \times 10^8$ m/s

**To find :** Minimum frame size

- Let the time for a signal to propagate between two farthest stations be $\tau$.
- The contention interval is such that width of each slot is $2\tau$.
- On a 1 km long cable $\tau \approx 5$ µsec. $\therefore 2\tau = 10$ µsec.
- To make CSMA/CD work, it must be ensured that the minimum frame size should be equal to $2\tau = 10$ µsec.

$$\text{But} \quad R = 1 \times 10^9 \text{ bits/sec}$$

$$\therefore \quad 1 \text{ sec} \equiv 1 \times 10^9 \text{ bits}$$

$$\therefore \quad 10 \times 10^{-6} \text{ sec} = ? \text{ bits}$$

$$\therefore \quad \frac{1}{10 \times 10^{-6}} = \frac{1 \times 10^9}{x}$$

$$\therefore \quad x = 1 \times 10^9 \times 10 \times 10^{-6}$$

$$= 10 \times 10^3 = 10{,}000 \text{ bits.}$$

$\therefore$ Minimum frame size = 10,000 bits or 1250 bytes.

## 8.6 Collision Free Protocols :

- As we have seen that almost all collisions can be avoided in CSMA/CD, they can still occur during the contention period.
- The collision during contention period affects the systems performance adversely.
- This happens when the cable is long and length of frames is short.
- This problem becomes serious as fiber optic networks come into use.
- Some protocols that resolve the collisions during the contention period are as follows :
  1. Bit map protocol
  2. Binary Countdown
  3. Limited Contention Protocols

### 8.6.1 Limited Contention Protocols :

**Meaning of contention system :**

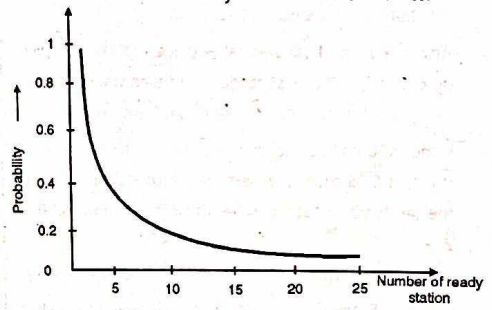
- The systems in which multiple users share a common channel in such a way that results in conflicts (collisions) are known as **contention systems.**





**Contention protocols :**

- Till now we have considered two different techniques for the channel allocation namely :
  1. Contention (such as CSMA) protocols
  2. Collision free methods.
- The performance of these techniques can be judged based on two performance parameters namely delay at light loads and efficiency at heavy loads.
- As the load of the channel increases, contention based schemes (protocols) becomes increasingly less attractive, because the overhead associated with channel arbitration will increase, and reduce the efficiency.
- Now consider the collision-free protocols. At low load, they have high delay, (bad performance) but as the load increases, the channel efficiency improves.
- Therefore, it would be an ideal thing to do if we could combine the best properties of the contention and collision-free protocols, to create at a new protocol that uses the contention at low loads to provide short delay, but uses a collision-free technique at heavy load to ensure good channel efficiency.
- Such protocols are called as **limited contention protocols.**
- These protocols are a combination of contention and collision–free protocols, because contention protocols provide a low delay at low loads and collision–free protocols provide good channel efficiency at high loads.
- The contention protocols like CSMA/CD are symmetric in nature i.e. each station attempts to acquire the channel with the same probability P. But this degrades the performance.
- In case of limited contention protocols the overall performance is improved by assigning different probabilities to different stations.
- In case of symmetric protocols for small number of stations, the chance of success are good but it becomes worse as the number of stations increases.
- In case of limited contention protocols which are asymmetric in nature the probability of some station acquiring the channel can be increased only by decreasing the amount of competition.
- In this method the stations are first divided up into groups.
- Only the members of group 0 are permitted to compete for slot 0. This reduces the competition.
- If one of them succeeds it acquires the channel and transmits its frame. If the slot lies empty or if there is a collision, the members of group 1 compete for slot 1 and so on.
- Thus by making a correct division of stations into groups, the amount of contention (competition) for each slot (competition) can be reduced.
- Fig. 8.6.1 shows the graph of probability plotted against number of stations ready to transmit their frames.

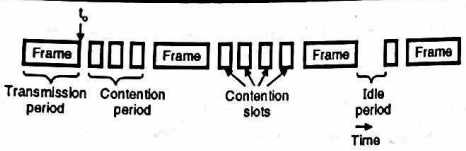



**(G-280)Fig. 8.6.1**

- The important question is how to assign stations to slots ?
- At one extreme we have one member per group whereas on the other side a single group will contain all the stations (slotted ALOHA).
- What is required is a way to assign stations to slots dynamically depending on load.
- When the load is low, many stations should be assigned per slot whereas when the load is high few (or even one) station per slot should be assigned.

## 8.7 Binary Exponential Back off Algorithm :

- In order to understand the principle of operation of binary exponential back off algorithm, refer the model shown in Fig. 8.7.1.



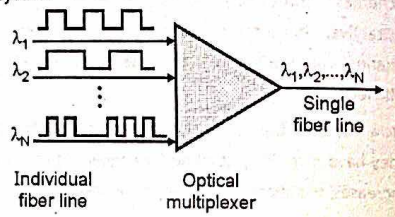

(G-1275)Fig. 8.7.1

- After a collision, the time is divided into discrete slots whose length is equal to the worst-case round trip propagation time on the Ether (2τ).
- In order to accommodate the longest path allowed by the Ethernet, the slot time is fixed to be equal to 512 bit times or 51.2 μsec.
- After the first collision takes place, each station will wait either 0 or 1 slot times before trying again. If two stations collide and each one picks up the same random number, then they will collide again.
- After the second collision takes place, each station picks up either 0, 1, 2 or 3 at random and then waits for those many number of slots (i.e. from 0 to $2^2 - 1$ slots).
- If the third collision takes place the probability of which is about 0.25 then, the next time each station waits for the randomly chosen number of slots from interval 0 to $2^3 - 1$.
- In general after "n" collisions, a random number between 0 and $2^n - 1$ is chosen and those many slots are skipped.
- But after 10 collisions the randomization interval is restricted to a maximum of 1023 slots.
- This algorithm is called as **binary exponential backoff.** It was selected to dynamically adapt the number of stations trying to send.
- If the randomization interval for all the collisions was 1023 i.e. constant, then the chance of second collision between two stations would be negligible.
- But then the average waiting time after the collision will also be very long (hundreds of slot times). This will introduce a lot of delay.
- In the binary exponential algorithm, the randomization interval increases exponentially as more and more consecutive collisions take place.
- This ensures that the delay is kept low when only a few stations collide and the collisions are resolved in a reasonable amount of time, when many stations collide.

### 8.7.1 Wavelength Division Multiplexing (WDM) :

**Principle :**

- The WDM works on the principle of multiplexing different sources operating at slightly different wavelengths and transmit several independent information streams simultaneously over the same fiber.
- Conceptually, the WDM system is the same as frequency division multiplexing (FDM).
- Fig. 8.7.2 shows the block diagram of the basic WDM system.

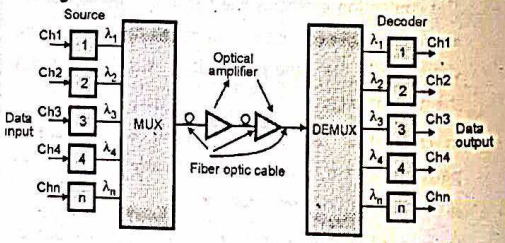



(O1199) **Fig. 8.7.2 : Basic concept of WDM**

- Here N independent optical information streams, with different wavelengths, are combined by means of an optical multiplexer and sent over the same fiber.
- Note that each stream could be at a different data rate.
- Individual fiber lines Optical multiplexer.
- Each information stream maintains its individual data rate after multiplexing with the other traffic streams, and still operates at its unique wavelength.

**Block diagram :**

- A simple block diagram of WDM transmitter and receiver system with different channels is as shown in Fig. 8.7.3.




(G-166) **Fig. 8.7.3 : WDM system**

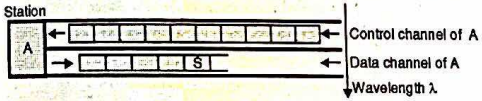
### 8.7.2 Wavelength Division Multiple Access Protocols (WDMA) :

- A different approach to channel allocation is to divide the channel into sub channels using FDM, TDM, or both, and dynamically allocate them as needed.





- Schemes like this are commonly used on fiber optic LANs in order to permit different conversations to use different wavelengths at the same time.
- A simple way to build an all-optical LAN is to use a passive star coupler.
- In effect, two fibers from each station are fused to a glass cylinder. One fiber is for output to the cylinder and one for input from the cylinder. Passive stars can handle hundreds of stations.
- To allow multiple transmissions at the same time, the spectrum is divided up into channels (wavelength bands).
- In this protocol, WDMA (Wavelength Division Multiple Access), each station is assigned two channels.
- A narrow channel is provided as a control channel to signal the station, and wide channel is provided so the station can output data frames.
- Each channel is divided into groups of time slots as shown in Fig. 8.7.4.




(G-1244) **Fig. 8.7.4 : WDMA principle**

- The control channel of station A is used by the other stations to contact station A. Whereas the wide data channel is used by station A to send its data to other stations.
- All the channels are synchronized by a single global clock.
- This protocol can be used for three types of traffic as follows :
  1. Constant data rate connection-oriented traffic, such as uncompressed video.
  2. Variable data rate connection-oriented traffic, such as a file transfer.
  3. Datagram traffic, such as UDP packets (User Datagram Protocol).
- For the two connection oriented protocols, if station A is to communicate with B then it should first insert a connection request frame on B's control channel.
- If B accepts this request, then A should send data on the data channel of B.
- Each station has two transmitters and two receivers, as follows :
  1. A fixed-wavelength receiver for listening to its own control channel.
  2. A tunable transmitter for sending on other station's control channel.
  3. A fixed-wavelength transmitter for outputting data frames.
  4. A tunable receiver for selecting a data transmitter to listen to.
- Every station will listen to its own control station in order to know about the incoming requests but it has to tune to the transmitter's wavelength to acquire data.
- Various types of WDMA protocol are possible. One of the variations can be to give each station a slot in a common control channel instead of giving a separate control channel.
- Another variation is to use a single tunable transmitter and a single tunable receiver per station with each station's channel being divided up into m control slots followed by (n + 1) data slots.
- When a large number of frequencies are being used, the system is sometimes called **DWDM** (Dense Wavelength Division Multiplexing).

## 8.8 Wired LANs : Ethernet Protocol :

- We know that for physical layer and data-link layer, TCP/IP protocol suite does not define any protocol. i.e. at these two layers any protocol can be accepted by TCP/IP which can provide service to the network layer.
- We know that for a limited geographic area such as a campus or building, local area networks (LANs) are designed.
- In an organization to connect computers, LAN is used as an isolated network.
- Now a days LANs are linked to a internet or WANs. Many different types of LANs were used in the 1980's and 1990's
- For resolving the problem of sharing the media, media access method was used by all these LANs.
- The approach used in the Ethernet is CSMA/CD. Whereas token-passing approach is used in the token bus, token ring and FDDI.

- During this period, in the market new LAN technology ATM LAN comes into existance which deployed the high speed WAN (wide area networks) technology.
- From the marketplace except Ethernet almost every LAN has disappeared. To meet the requirements of time, Ethernet was able to update itself.
- With the demand for high transmission rate, Ethernet could evolve for this reason the Ethernet protocol was designed.
- In the past, the organization that has used an ethernet LAN, now if they want higher data rate, instead of switching to another technology they would update to the new generation. It may cost more.
- In the following sections we will discuss wired LAN i.e. Ethernet.
- In the 1970s, the Ethernet LAN was developed by Robert Metcalfe and David Boggs..

### 8.8.1 Ethernet :

- Both Internet and ATM were designed for wide area networking.
- But in many applications, a large number of computers are to be connected to each other.
- For this the local area network (LAN) was introduced. The most popular LAN is called **Ethernet.**
- The IEEE 802.3 standard is popularly called as Ethernet. It is a bus based broadcast network with decentralized control.
- It can operate at 10 Mbps or 100 Mbps or even above 1 Gbps. Computers on an Ethernet can transmit whenever they want to do so.
- If two or more machines transmit simultaneously, then their packets collide.
- Then the transmitting computers just wait for an arbitrary time and retransmit their signal.
- There are various technologies available in the LAN market but the most popular one of them is **Ethernet.**
- In this section we are going to discuss three generations of Ethernet :
  1. Traditional Ethernet (10 Mbps)
  2. Fast Ethernet (100 Mbps)
  3. Gigabit Ethernet (1000 Mbps)

**Why is it called Ethernet ?**

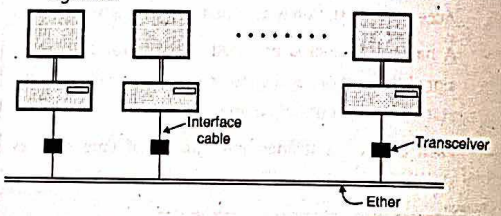
- This system is called as Ethernet after the luminiferous ether through which the electromagnetic radiation was once thought to propagate.

**Transmission medium :**

- The transmission medium is thick co-axial cable (called ether) upto 2.5 km long. Repeaters are placed after every 500 meters.
- Upto 256 machines can be attached to the multidrop cable.

**Architecture of Ethernet :**

- The architecture of the original Ethernet is shown in Fig. 8.8.1.




**(G-293) Fig. 8.8.1 : Architecture of original Ethernet**

- The original Ethernet was standardized as IEEE 802.3 standard.
- The committee also standardized a token bus (802.4) and token ring (802.5) standards which were not as popular as Ethernet.

**Computer connected to internet via LAN :**

- When a computer is connected to Internet via LAN, it has to use all the five layers of the internet model.
- The three upper layers (network, transport and application) are common to all the LANs.
- The data link layer is divided into two sublayers namely the logical link control (LLC) and the medium access control sublayer (MAC).
- The LLC sublayer is designed to be the same for all the LANs so that all the LANs can be connected to each other and operate without any problem.
- This means that only the MAC sublayer and physical layer of various LANs will be different from each other.
- If we compare different types of Ethernets then it is observed that, the MAC sublayer is slightly different but the physical sublayer is almost the same.





### 8.8.2 Traditional Ethernet :

- The traditional Ethernet is the oldest version of Ethernet created in 1976 which is designed to operate at the maximum data rate of 10 Mbps.
- The access to the network by a device is through the CDMA/CD i.e. the MAC uses CSMA/CD and the media is shared between all the hosts connected in LAN.

**Why Ethernet has been so successful ?**

- First, an Ethernet is extremely easy to administer and maintain.
- There are no switches, which can fall, no routing or bath tables that have to be kept up-to-date.
- We can add new host easily to this network second, it is inexpensive, cable is cheap, only network adapter is little costly.

### 8.8.3 Fast Ethernet :

- Fast Ethernet is the protocol designed to work at higher data rates than the traditional one. Typically it can support the data rates upto 100 Mbps.
- The traditional Ethernet can operate only up to 10 Mbps.
- Hence for higher data rates fast Ethernet has been developed.

**Autonegotiation :**

- This is the new feature of the fast Ethernet. The autonegotiation will make it possible to negotiate on the mode or data rare of operation between the communicating devices.

### 8.8.4 Gigabit Ethernet :

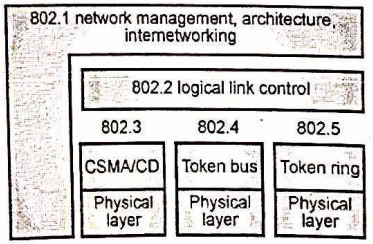
- The gigabit Ethernet protocol has been designed in order to operate at data rates upto 1000 Mbps or 1 Gbps.
- This is the highest bit rate of all the types. The MAC layer was supposed to remain unchanged for all the versions of the Ethernet but it does not remain so when such a high data rate is to be supported.
- The Gigabit Ethernet is capable of operating in either half duplex or full duplex modes.

## 8.9 IEEE Standards :

- The Institution of Electrical and Electronics Engineers (IEEE) has developed the layered architecture and other standards of LAN, under their project 802 set up in 1980.
- Some of the important IEEE 802 standards are as follows :

| | |
|---|---|
| 802.1 | Architecture, Management and Internetworking |
| 802.2 | Logical Link Control (LLC) |
| 802.3 | Carrier Sense Multiple Access/Collision Detect (CSMA/CD) |
| 802.11 | Wireless LAN Working Group |
| 802.15 | Wireless Personal Area Networking Group |
| 802.16 | Broadband Wireless Access Study Group. |

- In LANs, all the stations share the common cable (i.e. media).
- Therefore IEEE adopted three mechanisms of media access control namely :
  1. Carrier sense multiple access/collision detection (CSMA/CD)
  2. Token bus and
  3. Token ring
- Thus there are three protocols for the MAC sublayer.
- The IEEE standard 802.3 (CSMA/CD), 802.4 (Token bus), 802.5 (Token ring) are associated with these protocols as shown in Fig. 8.9.1.




**(G-295) Fig. 8.9.1 : IEEE LAN and related standards**

- The physical layer protocols do the job of signal encoding, data rate control and interfacing to the transmission medium.
- The Logical Link Control layer (LLC) specifications are given in IEEE 802.2.

## 8.10 Traditional Ethernet (IEEE 802.3) :

- The traditional Ethernet is the oldest version of Ethernet created in 1976 which is designed to support data rates up to 10 Mbps.
- The access to the network by a device is through the CDMA/CD i.e. MAC uses CSMA/CD and the media is shared between all the hosts connected on the Ethernet.

**Medium access control sublayer :**

- The MAC layer controls the operation of the access method which is CSMA/CD.
- It receives the data from the upper layer, frames it and passes it to the PLS sublayer for encoding.
- The access method used is 1-persistent CSMA/CD.

### 8.10.1 Traditional Ethernet Frame :

- Fig. 8.10.1 shows the frame format of traditional Ethernet.

| Preamble | SFD | Destination address | Source address | Length PDU | Data and padding | CRC |
|---|---|---|---|---|---|---|
| 7 bytes | 1 byte | 6 bytes | 6 bytes | 2 bytes | 0 – 46 bytes | 4 bytes |

**(G-305) Fig. 8.10.1 : Traditional Ethernet frame**

**Frame format :**

- The 64-bit (8 bytes) preamble allows the receiver to synchronize with the signal, it is a sequence of alternating 0's and 1's.

**DA and SA :**

- Both the source and destination hosts are identified with a 48-bit (6 bytes) address.
- These are indicated by the 6 byte number entered in the destination address (DA) and source address (SA) fields of the frame.
- The packet type field serves as the de-multiplexing key.

**Data :**

- Each frame contains upto 1500 bytes of data. The minimum size of a frame is 46 bytes of data, the reason for this is that the frame must be long enough to detect a collision.
- Each frame includes 32 bit (4 bytes) checksum. CRC is the last field in the Ethernet frame.
- The Ethernet is a bit-oriented framing protocol. An Ethernet frame has 14-byte header, two 6-bytes addresses and 2-byte type field.
- The sending adapter attaches the preamble, CRC and postamble before transmitting and the receiving adapter removes them.

**Start Frame Delimiter (SFD) :**

- This is the second field in the Ethernet frame and it is of 1 byte length. The byte stored at this field is 10101011.
- This field signals the beginning of the frame. The SDF is used to communicate to the station that this is the last chance for synchronization.
- The last two bits 11 alert the receiver that the next field in the frame contains the destination address.

### 8.10.2 Frame Length :

- There is a restriction imposed on the minimum and maximum length of the frame of the Ethernet.
- The minimum frame length is 512 bits or 64 bytes and the maximum frame length is 12,144 bits or 1518 bytes.
- The format of the minimum length frame is shown in Fig. 8.10.2(a) and that of the maximum length frame is shown in Fig. 8.10.2(b).

| Destination address | Source address | Length PDU | Data and padding | CRC |
|---|---|---|---|---|
| 6 bytes | 6 bytes | 2 bytes | 46 bytes | 4 bytes |

64 bytes

**(a) Minimum length frame**

| Destination address | Source address | Length PDU | Data and padding | CRC |
|---|---|---|---|---|
| 6 bytes | 6 bytes | 2 bytes | 1500 bytes | 4 bytes |

1518 bytes

**(b) Maximum length frame**

**(G-306) Fig. 8.10.2 : Minimum and Maximum length frame formats of traditional Ethernet**

- The restriction on the minimum length is to ensure correct operation of CSMA/CD, whereas the restriction on the maximum length is just out of some historical reasons.

### 8.10.3 Addressing :

- There can be various types of stations connected on an Ethernet network such as PC on workstation or printer.
- Each station has its own network interface card (NIC) which fits inside the station to contain the 6 byte physical address of the station.
- Fig. 8.10.3 shows a 6-byte Ethernet address in the hexadecimal notation.





**04 – 02 – 01 – 06 – 1C – 5B**

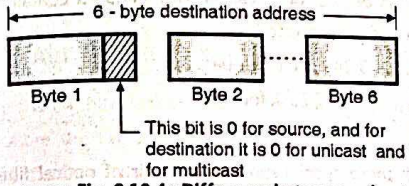
**(G-307) Fig. 8.10.3 : Ethernet address**

### 8.10.4 Types of Addresses :

- A source address is only unicast address. This is because the frame comes from only one source.
- The destination address can be one of the following three types :

1. Unicast 2. Multicast 3. Broadcast

- Fig. 8.10.4 shows how to differentiate between the unicast address and multicast address.




**(G-308) Fig. 8.10.4 : Difference between unicast and multicast addresses**

**1. Unicast destination address :**

- Uni means one. So this type of address defines only one destination and the relation between the sender and the receiver is one-to-one.
- The frame sent by the sender is meant only for one particular receiver.

**2. Multicast destination address :**

- Multi means many. So this type of address defines a group of destination addresses to which the same message is to be delivered.
- Thus the sender-receiver relation is one to many.

**3. Broadcast address :**

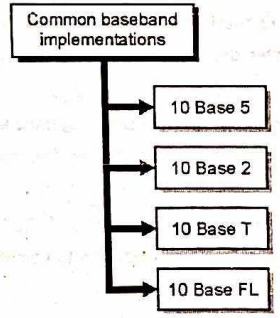
1. Broadcasting process is the process in which the sender transmit and all others receive or listen.
2. This type of destination address is a special case of multicast address in which all stations are destinations.

### 8.10.5 Physical Properties of Ethernet :

- Let us see some physical properties of Ethernet.
- An Ethernet segment is implemented on a coaxial cable of upto 500 m.
- A **transceiver**, which is a small device directly attached to the tap, detects when the line is idle and drives the signal when the host is transmitting. Tap must be at least 2.5 m apart.
- Transceiver also receives incoming signals. It is in turn, connected to an Ethernet adapter, which is plugged into the host. All the power of Ethernet is in adapter.
- Multiple Ethernet segments can be joined together by repeaters. A **repeater** is a device that forwards digital signals.
- Note that, no more than four repeaters may be positioned between any pair of hosts.
- Ethernet has a total reach of only 2500 m and it is limited to supporting a maximum of 1024 hosts with 100 base T, twisted pair.
- The common configuration have several point-to-point segments coming out of a multi-way repeater, called a hub, multiple 100-Mbps Ethernet segments can also be connected by a hub.

### 8.10.6 Physical Layer Implementation of Standard Ethernet :

- The standard has defined four different implementations for the baseband (digital) 10 Mbps Ethernet as shown is Fig. 8.10.5.




**(G-312) Fig. 8.10.5 : Categories of traditional Ethernet**

**IEEE 802.3 10 Mbps Specifications (Ethernet) :**

- IEEE 802.3 committee defines alternative physical configurations. Various defined options are as follows :
  1. 10 BASE 5
  2. 10 BASE 2
  3. 10 BASE – T (T stands for twisted pair)
  4. 10 BASE – FL (F stands for optical fiber)
- All the four options stated above are for the 10 Mbps Ethernet.

**1. 10 Base 5 : Thick Ethernet :**

- The first implementation of the traditional Ethernet is called 10 Base 5 or thick Ethernet or thicknet.
- This was the first Ethernet technology.
- The name thicknet is due to the use of thick coaxial cable. The thicknet uses the bus topology.
- It is the original 802.3 medium specification and is based directly on Ethernet. A 50 Ω coaxial cable is used.
- The data is converted into Manchester digital signalling. Maximum length of cable segment is 500 m. We have to use repeaters if the length is to be increased further.
- At the most four repeaters are allowed to be used. Hence the effective length of the medium is 2.5 km because there will be 5 segments of 500 m each with 4-repeaters.

**2. 10 Base 2 : Thin Ethernet :**

- This is second implementation of the traditional Ethernet, and it is also known as cheapernet. It uses a comparatively thin coaxial cable and bus topology.
- This is a low cost system than 10 BASE 5 and used for the personal computer LANs. This specification as well uses 50 Ω coaxial cable and the data is converted into Manchester digital signalling before putting it on the cable.
- Thin Ethernet uses a thin cable, supports less number of users and specified for an effective length of 185 metres only.
- The data rate is same as that of 10 BASE 5 specification i.e. 10 Mbps hence it is possible to combine them in a network.
- Note that the 10 BASE 2 should not be used to connect two segments of 10 BASE 5 cable.

**3. 10 Base-T : Twisted pair Ethernet :**

- This is the third physical layer implementation of traditional Ethernet. It makes use of a physical star topology.
- The twisted pair cable of unshielded type is used instead of coaxial cable as the common medium. The data is converted into Manchester digital signaling before putting it on the cable.
- The maximum segment length is reduced to only 100 m. It is much less than the 10 BASE 5 specification.
- The advantage of this type is that the twisted pair wire is easily available in any building (due to the existing telephone connection).
- As an alternative an optical fiber link can be used. Then the maximum length becomes 500 m.

**4. 10 Base FL : Fiber Link Ethernet :**

- This is the fourth physical layer implementation of traditional Ethernet. It makes use of the star topology for connecting stations to a hub.
- The transceiver is connected to the hub by using two pairs of fiber optic cables. This standard contains three specifications as follows :
  1. 10 BASE FP (P for passive).
  2. 10 BASE FL (L for link).
  3. 10 BASE FB (B for backbone).
- All these specifications use a pair of optical fibers for each transmission link.
- The data is converted into the Manchester code and then the Manchester signal is converted into light signal (off for 0 and on for 1).
- Hence the frequency of the Manchester bit stream actually needs to be 20 Mbps on the fiber.

## 8.11 Fast Ethernet :

- Fast Ethernet is the protocol designed to work upto 100 Mbps and it is compatible with the standard Ethernet.
- The traditional Ethernet can operate only upto 10 Mbps. Hence for higher data rates fast Ethernet has been developed.

**MAC sublayer :**

- In the evolution of Ethernet, care has been taken to keep the MAC sublayer untouched. So MAC sublayer of the fast Ethernet is same as that of the traditional Ethernet.
- For the standard Ethernet the bus and star topologies were used. But the fast Ethernet uses only the star topology.

**Access method :**

- The access method also remains the same. It is CSMA/CD.
- However the fast Ethernet is a full duplex protocol and does not need the CSMA/CD.





- But the CSMA/CD is used for backward compatibility, with the traditional Ethernet.

**Frame format :**

- Frame format of fast Ethernet is same as that of the traditional Ethernet.

**Minimum and maximum frame lengths :**

- Minimum and maximum frame lengths of the fast Ethernet frame are same as those of traditional Ethernet.

**Addressing :**

- Addressing is also same a that for the traditional Ethernet.

### 8.11.1 Autonegotiation :

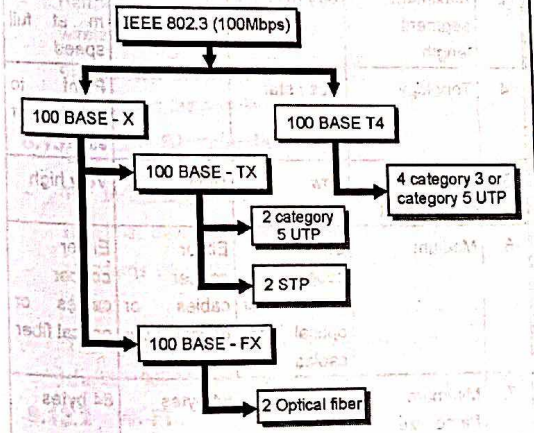
- This is the new feature of the fast Ethernet. Due to this feature the two stations can make the negotiation on the mode or data rare of operation.

**Features of the autonegotiation :**

- The important features of autonegotiation are :
  1. The non-compatible devices can be connected to each other.
  2. One device can be allowed to have multiple capabilities.
  3. A station can check hub's capabilities.

### 8.11.2 Physical Layer Implementation :

- Fig. 8.11.1 shows the various types of cables used for the fast Ethernet.




**(G-318) Fig. 8.11.1 : IEEE 802.3 100 BASE – T options**

- As shown, it can be either a two wire or four wire implementation.

**The 100 Base-X :**

- The 100 Base X is a two wire implementation. It can be either a twisted pair cable (100 Base – TX) or fiber optic cable (100 Base – FX).

**The 100 Base T4 :**

- The 100 Base T4 is a four wire specification and it is designed only for the twisted pair cable.
- All of the 100 BASE – T options shown in Fig. 8.11.1 use the IEEE 802.3 MAC protocol and frame format.
- 100 BASE – X indicates the options which use the medium specifications defined by FDDI.
- All the 100 BASE – X types use two physical links between any two nodes, one of them is used for transmission and the other one for the reception.
- Refer Fig. 8.11.1. The 100 BASE – TX uses either the shielded twisted pair (STP) or a high quality (category 5) unshielded twisted pair (UTP). Whereas 100 BASE-FX uses optical fiber.
- There is a disadvantage of using any of the 100 BASE-FX option because a new cable needs to be installed.
- So 100 BASE-T4 provides a low cost option because it uses category 3 voice grade UTP or a higher quality (category 5) UTP.
- 100 BASE – T4 uses four twisted pair lines between any two nodes in order to achieve 100 Mbps data rate over a low quality cable.
- For all the 100 BASE – T options, the star topology is used.

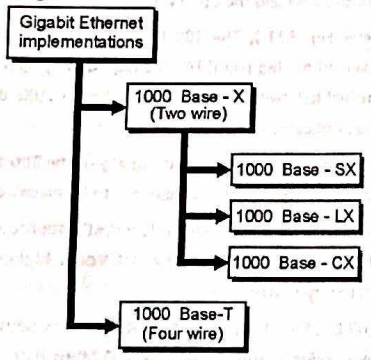
## 8.12 Gigabit Ethernet :

- The Gigabit Ethernet protocol has been designed in order to support the data rates upto 1000 Mbps or 1 Gbps.
- The MAC layer was supposed to remain unchanged throughout the evolution of the Ethernet but it does not remain so when the rate of 1 Gbps is to be supported.
- The Gigabit Ethernet is capable of operating in either half duplex or full duplex modes.
- If it operates in the half duplex mode, then the access method used is CSMA/CD.
- But if the full duplex mode is used then CSMA/CD is not required.

- Almost all the implementations in Gigabit Ethernet use the full duplex mode.
- The half duplex mode is used only for the backward compatibility with the standard and fast Ethernets.

### 8.12.1 Physical Layer Implementation :

- We can categorize the Gigabit Ethernet as either a two wire or a four wire implementation.
- The two wire implementation is known as 1000 Base X and the four wire implementation is known as 1000 Base-T. The four wire implementation uses twisted pair cable.
- Fig. 8.12.1 shows the physical layer implementations for the Gigabit Ethernet.




(G-323) **Fig. 8.12.1 : Physical layer Implementations of Gigabit Ethernet**

**Encoding :**

- The Gigabit Ethernet cannot use the Manchester encoding due to its high bit rate.
- Hence the 8B/10B block encoding followed by NRZ encoding is used for all the two wire implementations.

### 8.12.2 Ten Gigabit Ethernet :

- The next step of Gigabit Ethernet is ten gigabit Ethernet. The IEEE committee calls this Ethernet as standard 802.3ae.
- The goals of 10GB Ethernet are as follows :
  1. Data rate is to be upgraded to 10 Gbps.
  2. This Ethernet should be downward compatible to the standard, fast and gigabit Ethernet.
  3. Frame format and 48-bit address should be same as the older versions.
  4. Minimum and maximum frame lengths should remain same.
  5. This Ethernet should be connectable to the existing LAN, WAN and MAN.
  6. This Ethernet should be mode compatible with the technologies like Frame Relay and ATM.

**MAC sublayer :**

- This Ethernet operates only in the full duplex mode.
- Hence there is no possibility of contention. So CDMA/CD is not used in this Ethernet.

**Physical layer :**

- The physical layer of this Ethernet is designed to work with the optical fiber cable.
- The three commonly used implementations are :
  1. 10 G Base-S
  2. 10 G Base-I
  3. 10 G Base-E

### 8.12.3 Comparison of Ethernets :

**Table 8.12.1 : Comparison of Ethernets**

| Sr. No. | Parameter | Standard Ethernet | Fast Eternet | Gigabit Ethernet |
|---|---|---|---|---|
| 1. | Maximum speed | 10 Mbps | 100 Mbps | 1 Gbps |
| 2. | MAC technology | CSMA/CD | CSMA/CD | CSMA/CD |
| 3. | Maximum segment length | 500 m | | 25 m to 70 m at full speed |
| 4. | Topology | Bus / star | | Point to point or star |
| 5. | Bandwidth requirement | Low | High | Very high |
| 6. | Medium | Either copper cables or optical fiber cables | Either copper cables or optical fiber cables | Either copper cables or optical fiber |
| 7. | Minimum frame size | 64 bytes | 64 bytes | 64 bytes |
| 8. | Mode | Half duplex or full duplex | Full duplex | Full duplex and half duplex |





## 8.13 Wireless LANs :

- We all know wired local area networks (LANs) very well.
- In order to get rid of the wiring associated with the interconnections of PCs in LANs, researchers have tried to use radio waves or infrared light as a replacement to the wires. This is how the wireless LANs i.e. WLANs got evolved.

**Why Wireless LANs :**

- Standard LAN protocols such as Ethernet can operate at high speed using inexpensive connection hardware.
- However, the problem with these LANs is that they are limited to the physical, hard-wired infrastructure of the building and the network nodes are limited to access only through wired landline connections.
- Many mobile users of the network in businesses find many advantages from the added capabilities of wireless LANs.
- The WLANs are gaining popularity because they have certain advantages over the wired LANs such as : increased mobility and flexibility, cost effectiveness, ease of installation, ease of adding new members etc.
- Earlier the wireless LANs were costly, they could support only low data rates and a license was required to build and operate them.
- Hence there were limitations on the practical utility of wireless LANs.
- But all these problems are being addressed now which is increasing the popularity of wireless LANs day by day.

### 8.13.1 IEEE Standards :

- The Institution of Electrical and Electronics Engineers (IEEE) has developed the layered architecture and other standards of LAN, under their project 802 set up in 1980.
- The IEEE 802.3 standard is for the wired LAN whereas the IEEE 802.11 standard is for the wireless LANs.

### 8.13.2 Wi-Fi :

- Wi-Fi is a popular technology which allows an electronic device to exchange data or to connect to the Internet using radio waves.
- We can define Wi-Fi as any wireless local area network (WLAN) product that are based on the IEEE 802.11 standards.

### 8.13.3 ISM Band :

- Internationally the ITU has designated some frequency bands called as ISM bands for unlimited usage. The long form of ISM is Industrial, Scientific, Medical band.
- These frequency bands are located around 2.4 GHz and used for the wireless LAN and PAN applications. The wireless networks use the ISM frequency bands for their operation.
- These bands are as follows :
  1. 902-928 MHz,
  2. 2.4-2.4835 GHz and
  3. 5.725-5.85 GHz.
- No license is required for operating in this band. Most of the wireless LAN products operate within the unlicensed ISM bands.

## 8.14 Wi-Fi (IEEE 802.11) :

- Wi-Fi is a popular technology which allows an electronic device to exchange data or to connect to the Internet using radio waves.
- We can define Wi-Fi as any wireless local area network (WLAN) product that are based on the IEEE 802.11 standards.
- The devices which can use Wi-Fi are personal computers, video game consoles, smart phones, some digital cameras, Tablet computers etc.
- Wireless communication is one of the fastest growing technologies.
- The wireless LANs are used in the following applications :
  1. Office buildings
  2. Colleges 3. Public areas
- The 802.11 is the specifications for the wireless LANs, defined by IEEE.
- This specification defines the physical and data link layers. It is sometimes called as **Wireless Ethernet.**
- Generally the term **Wi-Fi** (Wireless fidelity) is used as a synonym for wireless LAN.

- However in reality, Wi-Fi is a wireless LAN which is certified by the **Wi-Fi Alliance** a global industry association.

### 8.14.1 Classification of WLANs :

- We can classify the WLANs into the following two categories :
  1. Infrastructure networks.
  2. Ad-hoc LANs.

**1. Infrastructure networks :**

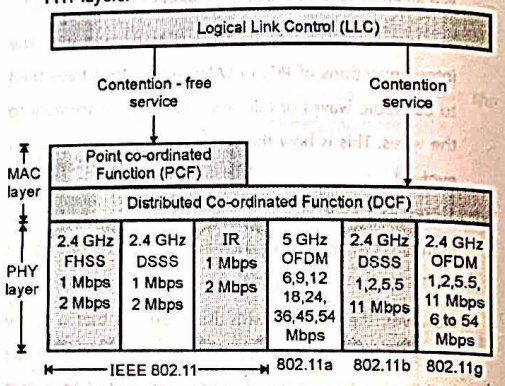
- These WLANs contain special nodes called Access Points (APs) via existing networks.
- APs can interact with wireless nodes as well as wired networks.
- The other wireless nodes known as mobile stations (STAs) communicate via APs. The APs can also work as bridges with other networks.

**2. Ad-hoc LANs :**

- These WLANs do not have any fixed architecture. They can be set up at any place. It is a peer-to-peer network without any centralized server.
- The Ad-hoc LAN is set up temporarily to meet some immediate requirements. Such as a group of people with laptops conferencing with each other in a room.
- The ad-hoc LAN does not have any infrastructure. It is just a collection of few stations within range of each other dynamically configuring themselves into a temporary network.

### 8.14.2 The IEEE 802.11 Protocol Stack :

- The IEEE approved the 802.11 standard (sometimes also referred to as Wi-Fi - for Wireless Fidelity) for WLANs in June 1997 and in July 1997, it was adopted as a worldwide International Standards Organization (ISO) standard.
- The WLAN standard defines three possible physical (PHY) layer implementations and one MAC layer for supporting data rates of 1 Mbps or 2Mbps.
- The IEEE 802.11 standard has different physical layer variants from 802.11a to 802.11g.
- In this chapter we concentrate more on the physical layer specifications of the IEEE 802.11 a/b standards.
- Fig. 8.14.1 shows the 802.11 protocol stack with various PHY layers.




**(O-899) Fig. 8.14.1 : 802.11 protocol stack**

- This shows that the MAC layer protocols are common for all standards while they are not always compatible at the PHY layer.
- Due to all the enhancements, along with ease of use and customer satisfaction, the IEEE 802.11 standard has become the most widely used WLAN standard.
- Table 8.14.1 enlists the six subgroups in IEEE 802.11 standard.

**Table 8.14.1 : IEEE 802.11 subgroups**

| | |
|---|---|
| 802.11a | High speed physical layer in 5 GHz band |
| 802.11b | Higher speed physical layer extension of wireless in 2.4 GHz band |
| 802.11d | Local and metropolitan area wireless |
| 802.11g | Broadband wireless |
| 802.11i | Security |
| 802.11n | Wideband service |

- The data rates of the indoor WLANs are in the range of 11 Mbps to 54 Mbps with the IEEE standards 802.11a and 802. 11g, that is considerably high.
- A high throughput amendment to the IEEE 802.11 standard is currently in the final stages with the 802.11n task group working on it.
- It uses multiple-input multiple output (MIMO) techniques that can support data rates over 100 Mbps.





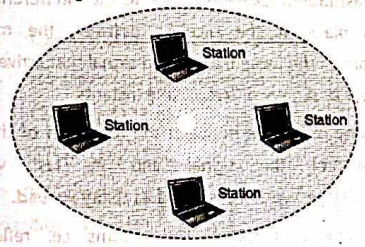
- The 802.11 MAC layer is considered as the "brains" of the network and it uses an 802.11 PHY layer, such as 802.11a/b/g etc to perform the tasks such as carrier sensing, transmission and receiving of 802.11 frames.
- The MAC layer functional specifications are essentially the same for all the PHY layer specifications with some minor differences.

### 8.14.3 802.11 Network Architecture :

- IEEE 802.11 is the most popular WLAN standard. It defines the specifications for the physical and MAC layers.
- We can use a WLANs to either replace a wired LANs, or as an extension of the wired LAN infrastructure.
- IEEE 802.11 defines two types of services :
  1. Basic Service Set (BSS)
  2. Extended Service Set (ESS)

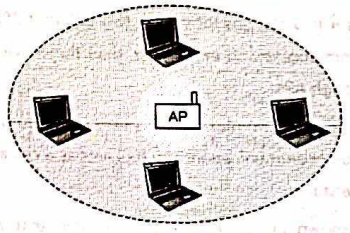
**1. Basic Service Set (BSS) :**

- As per IEEE 802.11 the BSS has been defined as the basic building block of wireless LAN.
- A BSS consists of stationary or moving wireless stations and a central base station which is optional called as the Access Point (AP).
- Thus a BSS can be either without AP or with AP as shown in Figs. 8.14.2(a) and (b).




**(G-380) Fig. 8.14.2(a) : BSS without AP**

- The BSS without AP cannot send data to another BSS. So no data exchange can take place outside that BSS hence it is known as a standalone network or **ad hoc BSS.**
- However all the stations inside a BSS can exchange data among themselves.
- A BSS with AP is as shown in Fig. 8.14.2(b). It can however communicate with the other BSS via the access point AP.

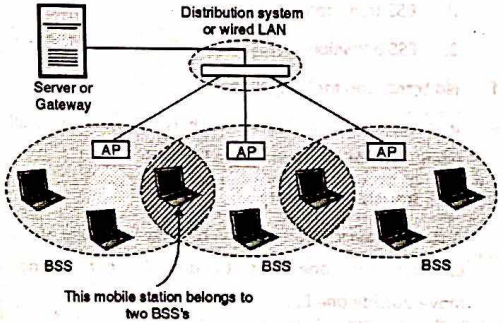



**(G-380) Fig. 8.14.2 (b) : BSS with an AP**

- The BSS with AP is also called as **Infrastructure BSS.**

**2. Extended Service Set (ESS) :**

- An Extended Service Set (ESS) consists of multiple BSSs with APs. The BSSs in this system are connected to each other via a **distribution system** or a wired LAN as shown in Fig. 8.14.3.




**(G-381) Fig. 8.14.3 : ESS (Extended Service Set)**

- The APs are connected to each other via the distribution system as shown. The distribution system can be any type of LAN such as Ethernet.
- The ESS contains two types of stations :
  1. Mobile stations which can move and change location.
  2. Stationary of non-moving stations.
- Out of these, the non-moving stations are the APs which are a part of the wired LAN. Whereas the mobile stations are those contained in the BSS. Fig. 8.14.3 shows the structure of an ESS.
- The BSSs are connected to each other to form a network called **infrastructure network.** In such networks the stations close to each other can communicate without taking help of AP.

- But if two stations located in two different BSS wish to communicate with each other, than they have to do so through APs.
- This type of communication is very similar to that in the cellular communication. The BSS acts as a cell and AP as base station.
- As shown in Fig. 8.14.3 it is possible that a mobile station can belong to more than one BSSs simultaneously.

### 8.14.4 Types of Stations :

- Three types of stations are defined by IEEE 802.11 depending on their mobility in the wireless LAN as :
  1. No transition
  2. BSS transition
  3. ESS transition

**1. No transition mobility :**

- It is defined as a station which is not moving at all (stationary) or moving inside a BSS only.

**2. BSS transition mobility :**

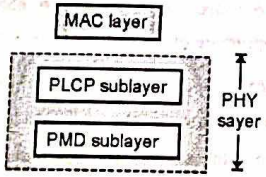
- A station having BSS transition mobility is the one which can move from one BSS to the other BSS but does not move outside one ESS.

**3. ESS transition mobility :**

- A station having ESS transition mobility is the one which can move from one ESS to any other ESS.
- But IEEE 802.11 does not guarantee a continuous communication when the station is moving.

## 8.15 The Physical Layer :

- The PHY layer acts as the interface between the MAC and wireless media, which transmit and receive data frames over a shared wireless media as shown in Fig. 8.15.1.




**(O-900) Fig. 8.15.1 : Sublayers within PHY**

- There are three important functions of the PHY as follows :
  1. To provide a frame exchange between the MAC and PHY that is controlled by the physical layer convergence procedure (PLCP) sublayer.
  2. To use signal carrier and spread spectrum modulation to transmit data frames over the media that is controlled by the physical medium dependent (PMD) sublayer.
  3. To provide a carrier sense indication back to the MAC to verify activity on the media.

**Fading :**

- Fading is defined as the phenomenon in which the signal strength at the receiver fluctuates.
- It is an important limiting factor for the high speed network performance.

**Types :**

- Fading can be classified into two types :
  1. Fast fading or small scale fading.
  2. Slow fading or large scale fading.

**1. Fast fading :**

- The rapid fluctuations in the amplitude phase or delay of the received signal is referred to as **fast fading, or small scale fading.**
- The fast fading takes place due to the interference that takes place at the receiver between the multiple versions of the same transmitted signal that arrive at the receiver at slightly different time instants.
- The time difference between the reception of the first version of the signal and its last version (most delayed version) is called as **delay spread.**
- The three propagation mechanisms i.e. reflection, diffraction and scattering are responsible for the multipath propagation of the transmitted signal which causes the fast fading.
- The variation in the received signal takes place because the multipath signals may sometimes add together to increase the signal power but they may subtract at some other time to reduce the signal power.
- Two commonly used techniques to overcome frequency selective fading are Spread Spectrum (e.g., FHSS or DSSS) and OFDM.

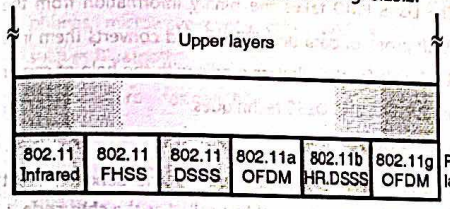




- The types of modulation schemes used by this standard are GFSK, DBPSK and DQPSK.
- Both the FHSS and DSSS modes are specified for operation in the 2.4 GHz ISM band, that is used by most electronic products.
- The third physical layer alternative is an infrared system using near-visible light in the 850 nm to 950 nm ranges as the transmission medium. But this is rarely used.

**Various PHY Specifications :**

- IEEE 802.11 has defined the specification for converting bits to a signal in the physical layer.
- One of them is in the infra-red frequency spectrum and the other five specifications are in RF range as shown in the partial 802.11 protocol stack of Fig. 8.15.2.




**(G-382) Fig. 8.15.2 : Part of 802.11 protocol stack**

- The specifications in RF range are :
  1. FHSS - Frequency Hopping Spread Spectrum (802.11).
  2. DSSS - Direct Sequence Spread Spectrum (802.11).
  3. OFDM - Orthogonal Frequency Division (802.11 a).
  4. HR-DSS-High Rate-DSSS (802.11 b).
  5. OFDM (802.11 g).
  6. OFDM (802.11 n).
- In this section we are going to discuss the six specifications listed in Table 8.15.1.

**Table 8.15.1 : Physical layer specifications**

| IEEE standard | Technique used | Frequency band | Modulation type | Data rate (Mbps) |
|---|---|---|---|---|
| 802.11 | FHSS | 2.4 - 4.835 GHz | FSK | 1 and 2 |
| | DSSS | 2.4 - 4.835 GHz | PSK | 1 and 2 |
| | None | Infrared | PPM | 1 and 2 |
| 802.11 a | OFDM | 5.725 – 5.85 GHz | PSK or QAM | 6 to 54 |
| 802.11 b | DSSS | 2.4 – 4.835 GHz | PSK | 5.5 and 11 |
| 802.11 g | OFDM | 2.4 – 4.835 GHz | Different | 2.2 and 54 |
| 802.11 n | OFDM | 5.725 – 5.85 GHz | Different | 6000 |

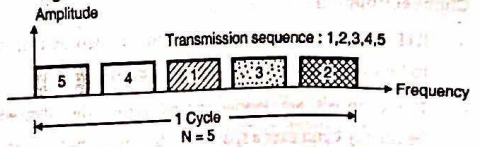
- All these physical layer implementations except the infrared, work in the ISM frequency band.
- As defined earlier the ISM band is the industrial, scientific and medical band. Three unlicensed frequency bands are defined under the ISM band.
- They are as follows :
  1. 902 - 928 MHz
  2. 2.400 – 4.835 GHz
  3. 5.725 – 5.850 GHz

### 8.15.1 IEEE 802.11 FHSS :

- The IEEE 802.11 describes a method called FHSS i.e. Frequency Hopping Spread Spectrum for conversion of bits into a signal.
- The frequency band used for this is 2.4 GHz ISM band.
- In FHSS, the FHSS PMD sub-layer controls the data transmission over the media as pet the directions given by the FHSS PLCP sub-layer.
- The FHSS PMD receives the binary information from the whitened PLCP service data unit (PSDU) and converts it into RF signals for the wireless media with the help of carrier modulation and FHSS techniques.

**Principle of FHSS :**

- In FHSS, the sender sends one carrier frequency for a short period of time.
- Then hops to another carrier frequency and transmits it for the same amount of time.
- Then it hops again to a new carrier frequency and transmits it for the same duration and so on. In all there are N-such hoppings in one cycles as shown in Fig. 8.15.3.




**(G-383) Fig. 8.15.3 : Principle of FHSS**

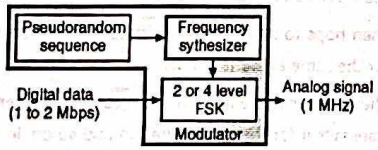
- The cycle repeats itself after N-hoppings. The bandwidth of FHSS signal is equal to NB Hz where B is the bandwidth of the original signal that is being converted to FHSS signal.
- The techniques of spreading used in FHSS makes it difficult for an unauthorized person to understand the transmitted data.
- The frequency of allocated bands in FHSS is decided with mutual agreement between the sender and the receiver.
- The frequency band by FHSS is 2.4 GHz ISM (industrial, scientific and medical). This band is divided in 79 equal sub-bands of 1 MHz each.
- The hopping frequency is selected by a pseudorandom number generator.

**PSDU data whitening :**

- Data whitening is applied to the PSDU before transmission in order to minimize the dc bias on the data if it contains long strings of 1's or 0's.
- The PHY achieves this by stuffing a special symbol after every 4 octets of the PSDU in a PPDU frame.
- A 127-bit sequence generator is employed using the polynomial $S(x) = x^7 + x^4 + 1$ and 32/33 bias-suppression encoding algorithm to randomize and whiten the data.

**Modulation :**

- The modulation technique used for this specification is either two level FSK of four level FSK, with 1 or 2 bits/baud. Therefore the data rates for this specification can be upto 1 to 2 Mbps.
- Fig. 8.15.4 shows the physical layer of IEEE 802.11 FHSS.




(G-2156) **Fig. 8.15.4 : Physical layer of IEEE 802.11 FHSS**

**Channel Hopping :**

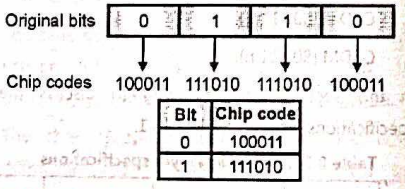
- IEEE 802.11 has defined a set of hop sequences that is to be used in the 2.4 GHz frequency band.
- The channels are evenly spaced across the allotted frequency band over a span 83.5 MHz. Hop channels are different in different countries.
- The FHSS PMD controls the process of channel hopping.
- It transmits the whitened PSDU by hopping from one channel to the other in a pseudorandom manner by using one of the hopping sequences.

## 8.15.2 IEEE 802.11 DSSS :

- IEEE 802.11 has defined the DSSS i.e. **direct sequence spread spectrum** technique in order to convert the bits into signal.
- The DSSS also uses the same frequency band as that of the FHSS i.e. 2.4 GHz ISM band.
- The DSSS PMD sub-layer controls the data transmission over the media as per the directions given by the DSSS PLCP sub-layer.
- The DSSS PMD takes the binary information from the PLCP protocol data unit (PPDU) and converts them into RF signals for the wireless media with the help of carrier modulation and DSSS techniques.

**Principle of DSSS :**

- In DSSS, each bit being sent by the sender is first converted into a group of bits called as the **chip code.**
- The time required to send each chip code should be equal to the time period of the original bit in order to avoid the buffering.
- Let N represent the number of bits in each chip code.
- Then the data rate of DSSS would be equal to N times the data rate of the original signal.
- Fig. 8.15.5 demonstrates the principle of DSSS.




(G-384) **Fig. 8.15.5 : Principle of DSSS**

- DSSS eventhough similar to CDMA is not a multiple access method.
- The bit sequence used in DSSS uses the entire frequency band of 2.4 GHz (ISM band).

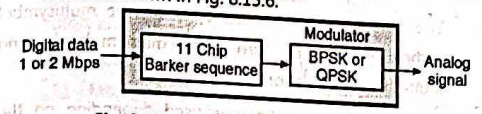
**Modulation :**

- DSSS uses the BPSK (Binary PSK) or QPSK as its modulation techniques. The data rate of this system can reach 1 to 2 Mbps.





- The physical layer implementation of IEEE 802.11 DSSS has been shown in Fig. 8.15.6.

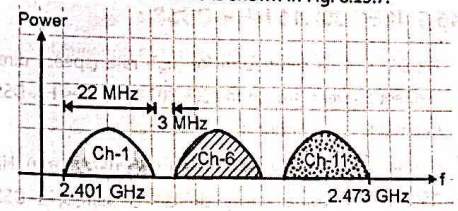



(G-2157) **Fig. 8.15.6 : Physical layer of IEEE 802.11 DSSS**

- The IEEE 802.11 implements DSSS to fight frequency-selective fading. DSSS is used by the 802.11 as well as by 802.11b standard.
- The data rates for 802.11 are 1 Mbps and 2 Mbps whereas those for 802.11b are 5.5 Mbps and 11 Mbps.
- IEEE 802.11b operates in the 2.4 GHz ISM band (2.40 to 2.4835 GHz), which is highly populated because it accommodates various other products, such as cordless phones, microwave ovens, other WLANs and WPANs such as Bluetooth.
- This band provides only 83 MHz of spectrum to all these products.
- Therefore these products are susceptible to interference effects.

**Operating channels and power requirements :**

- Each DSSS PHY channel occupies 22 MHz of bandwidth.
- There is a guard band of 3 MHz between the adjacent channel spectrums.
- This allows for three non overlapping channels spaced 25 MHz apart in the 2.4 GHz ISM frequency band. This DSSS channel scheme is shown in Fig. 8.15.7.




(O-901) **Fig. 8.15.7 : DSSS non-overlapping channels**

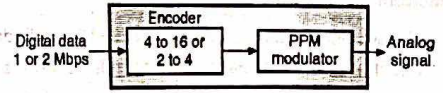
- Another important parameter that is regulated worldwide is the amount of transmitted power.
- The maximum allowable radiated power for the DSSS PHY is different for different regions.
- Nowadays, the nominal RF transmit power level selected by the wireless manufacturers is 100 mW.

**The PN sequence :**

- In IEEE 802.11, the PN sequence chosen for the DSSS PHY layer is the 11-chip barker sequence [1, 1, 1, – 1, – 1, – 1, 1, – 1, – 1, 1, – 1].
- This sequence is selected because it has some very interesting properties regarding its autocorrelation.
- The autocorrelation shows some very sharp peaks when the transmitter and the receiver are synchronized.
- The receiver can use these peaks to lock on the strongest received signal, to overcome the 'echo' signals due to the multipath channel, and reduce the probability of error.
- The DSSS is implemented by IEEE 802.11b in an improved way with an 8-chip Complementary Code Keying (CCK) modulation scheme instead of the barker codes.
- For all the IEEE 802.1 Ib payloads having different data rates, the preamble and header are sent at the 1 Mbps to maintain compatibility with earlier versions.

## 8.15.3 IEEE 802.11 Infrared :

- This specification of IEEE 802.11 uses the infrared light in the range or 800 to 950 nm as the connecting medium instead of wires.
- It uses the **pulse position modulation (PPM)** as the modulation technique.
- The physical layer implementation for IEEE 802.11 infrared is as shown in Fig. 8.15.8.




(G-2158) **Fig. 8.15.8 : Physical layer of IEEE 802.11 Infrared**

- An encoder first maps a 4 bit sequence into a 16 bit sequence. The mapped sequences are then converted into the infrared signals.
- A logical 1 is represented by the presence of light while a logical 0 is represented by its absence.

## 8.15.4 IEEE 802.11 a OFDM :

- OFDM stands for orthogonal frequency division multiplexing.

- It is used by IEEE 802.11a as the signal conversion technique.
- The IEEE 802.11a standard was approved in September 1999.
- But the development of new products has proceeded much more slowly than IEEE 802.11b due to the cost and complexity of implementation.
- This standard operates in the 5 GHz unlicensed national information infrastructure (UNII) band with a bandwidth of 300 MHz.

**Principle of OFDM :**

- The basic principle of OFDM is same as that of FDM. But the major difference between them is that in OFDM all the frequency sub-bands are used by one source at a given time.
- OFDM uses the 5 GHz ISM band for its operation. This band is subdivided into 52 sub-bands.
- Out of these 52 sub-bands, 48 sub-bands are used for sending 48 groups of bits at a time and the remaining 4 subbands are used for sending the control information.
- These subbands can be used randomly in order to increase the security of transmitted data.
- The type of modulation used by OFDM is BPSK and QAM (Quadrature Amplitude Modulation).
- The data rate with BPSK is 18 Mbps and that with QAM is 54 Mbps.

**Spectrum :**

- The available frequency spectrum of 300 MHz is divided into three "domains," each having a width of 100 MHz.
- Each domain has restrictions on the maximum allowed output power.
- The maximum output power in the first 100 MHz is restricted to 50 mW.
- The maximum output power in the second 100 MHz has a higher 250 mW maximum, whereas the third 100 MHz is mainly used for outdoor applications and it has a maximum power output of 1.0 W.
- OFDM is used in 802.11a and in 802.11g. It combines multicarrier, multi symbol and multi rate techniques, which require smart digital signal processing.

**Data rates :**

- The data rate is increased by using the multisymbol technique that in turn uses multiamplitude and multiphase modulation.
- Four modulation techniques used depending on the data rate to be supported are BPSK, QPSK, 16-QAM and 64-QAM.
- There is another approach to increase the data rate that uses a multirate modem, which provides one or more "fallback" modes of operation.
- The principle of the multirate technique is that if the modulation efficiency is increased by increasing the number of bits per symbol then the required signal-to-noise ratio (SNR) at the receiver also increases.
- That means, as the user moves away from the AP, the SNR reduces and the modem starts operating at a lower data rate, to provide a reasonable error rates at lower values of the SNR.

**Properties of OFDM :**

- Some of the important properties of OFDM are as follows :
  1. OFDM can eliminate the intersymbol interference (ISI) without increasing the bandwidth.
  2. It does not require very complex signal processing.
  3. OFDM is very sensitive to frequency offsets and timing jitter and it needs to use some additional mechanisms to address these issues.

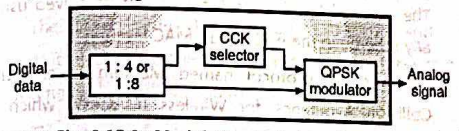
## 8.15.5 IEEE 802.11 b HR-DSSS :

- HR-DSSS is the short form for high rate (speed) direct sequence spread spectrum. IEEE 802.11 b uses HR-DSSS method for signal generation.
- It uses the frequency band of 2.4 GHz (ISM band). HR-DSSS is conceptually very similar to that of a DSSS techniques. The only difference is about the encoding method used.
- HR-DSSS uses a method called Complementary Code Keying (CCK) which encodes 4 or 8 bits of original data into one CCK symbol.
- The HR-DSSS needs to be backward compatible with DSSS. Hence HR-DSSS defines four data rates : 1, 2, 5.5 and 11 Mbps.





- The 1 and 2 Mbps data rates are allocated for the same type of modulation techniques as used for DSSS i.e. BPSK and QPSK.
- The modulation used for 5.5 Mbps is BPSK whereas the 11 Mbps data rate is assigned for QPSK type modulation. The modulation technique for this standard is shown in Fig. 8.15.9.




(G-2159) **Fig. 8.15.9 : Modulation technique for IEEE 802.11 b**

## 8.15.6 IEEE 802.11 g OFDM :

- It is a new specification which used ODFM and defines the forward error correction. The frequency band used is 2.4 GHz ISM band with a complex modulation technique used.
- It is possible to achieve data rate of 22 Mbps or 54 Mbps.
- This specification is backward compatible with IEEE 802.11 b, but the modulation technique it uses is OFDM and not PSK or QPSK like IEEE 802.11 b.

## 8.15.7 IEEE 802.11 n OFDM :

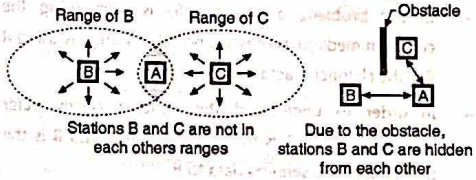
- 802.11 n is the upgraded version of 802.11 project. It is also known as the next generation of wireless LAN.
- The goal of designing 802.11 n specification is to increase the throughput of 802.11 wireless LAN.
- In this new upgraded standard, achieving a higher bit rate has been emphasized alongwith reduction or elimination of some unnecessary overheads.
- The noise problem present in the conventional wireless LANs can be overcome in the new standard by using what is called as **MIMO (Multiple Input Multiple Output Antenna).**
- For some implementations of this project, the data rates as high as 600 Mbps have been achieved.

# 8.16 Problems in Wireless LAN :

- If we try to use CSMA (the access method used for wired LANs) for the wireless LAN, then it uses the principle of simply listening to other transmission and only transmit if no one else is transmitting.
- But there are two problems in using CSMA. They are hidden station problem and exposed station problem.

## 8.16.1 Hidden Terminal Problem :

- The hidden station problem occurs when a station may not be aware that some other station is transmitting because of either range problem or some obstacle.
- In this situation collision may occur but may not be detected.
- The hidden station problem is illustrated in Fig. 8.16.1.




(G-2098) **Fig. 8.16.1 : Hidden station problem**

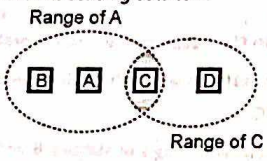
- Refer Fig. 8.16.1(a) which shows three wireless stations A, B and C.
- The transmission ranges of stations-B and C have been shown by the two ovals on left and right respectively which shows that station-C is not in the range of B and B is not in the range of C.
- However station-A is in the range of both B and C. So A can hear signals transmitted by B and C.
- Refer Fig. 8.16.1(a) where station-B is transmitting to station A.
- Now if station-C checks the medium to see if anyone is transmitting, it will not hear station B because it is out of range.
- So station-C will come to a wrong conclusion that no one is transmitting and so it can start transmitting to station A.
- If station-C starts transmitting, it will create a collision at station-A and will wipe out the frames from station-B.
- This problem in which a station is not able to detect an already transmitting other station which is too far away is called as the **hidden station problem.**
- In this example it is said that stations-B and C are hidden from each other with respect to station-A.

- Now consider Fig. 8.16.1(b) which shows the hidden station problem occurring due to an obstacle.

**Note :** Due to hidden station problem, the possibility of collision increases and the capacity network will reduce.

### 8.16.2 Exposed Station Problem :

- Earlier in this chapter we have discussed the problem of hidden station. The **exposed station problem** is a similar problem.
- In this problem, a station refrains from using the common medium even when no other station is using it (i.e. the channel is actually free).
- In order to understand this concept clearly, refer Fig. 8.16.2 where A is the sending station and B is the destination. A is sending data to B.




**(O-918) Fig. 8.16.2 : Exposed station problem**

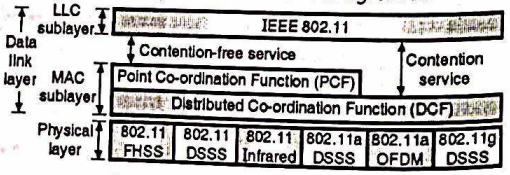
- Station C wants to send its data to station D and it is possible to do so without interfering in the communication between A and B.
- As shown in Fig. 8.16.2, station C is in the range of station A. In other words C is exposed to A.
- Therefore C listens to what A is transmitting and decides to refrain itself from sending its message to D.
- This causes wastage of channel capacity.

**Access control algorithm for wireless LAN :**

- Note that the carrier sense (CSMA/CD) protocol can provide the information about potential collisions at the sender, but not at the receiver.
- Therefore this information can be misleading for a distributed configuration because all stations are not within range of each other.
- The Medium Access with Collision Avoidance (MACA) protocol proposes the solution to the hidden terminal problem.
- The solution suggested by this protocol is to transmit Request-to-Send (RTS) and Clear-to-Send (CTS) packets between the nodes that wish to communicate.
- These RTS and CTS packets contain information about the duration of the data transfer of the communicating nodes.
- For this duration the neighbouring stations that do not participate in communication but overhear either of these packets will keep quiet.
- The exposed terminal problem cannot be solved using any scheme in the IEEE 802.11 MAC layer.
- However the protocol named Medium Access with Collision Avoidance for Wireless (MACAW) which is based on MACA solves this problem.
- In MACAW protocol the source transmits a **data sending** control packet that will alert exposed nodes of the impending arrival of an ACK packet.

## 8.17 MAC Sublayer :

- MAC protocol has the responsibility to arbitrate the accesses to a shared medium among several end systems.
- The IEEE 802.11 standard does this via an Ethernet-like stochastic and distributed mechanism : CSMA/CA.
- According to IEEE 802.11 protocol a multiple access network is defined as the network where all the devices have to compete with each other to get access to the wireless channel.
- IEEE 802.11 specifies two medium access control protocols, Point Coordination Function (PCF) and Distributed Coordination Function (DCF).
- Out of these DCF is a fully distributed scheme which is suitable for the adhoc networking, whereas PCF is an optional centralized scheme.
- In IEEE 802.11 protocol stack these two protocols are represented by two MAC sublayers.
- The relation between DCF, PDF, the LLC sublayer and the physical layer has been shown in Fig. 8.17.1.




**(G-2099) Fig. 8.17.1 : MAC layers in 802.11 standard**

- We have already discussed the physical layer implementations.





### 8.17.1 RTS and CTS Mechanism :

- IEEE 802.11 MAC addresses the hidden station problem by adding two additional frames, the RTS (request to send) and CTS (clear to send).
- Here, the source sends a RTS and the destination replies with a CTS.
- The other nodes that overhear the RTS and CTS messages will suspend their transmissions for a certain time period indicated in the RTS/CTS frames.
- The source station retransmits the RTS frame if the RTS/CTS handshake fails.
- The system treats this as a collision and retransmission occurs as per rules that are described later in the section on DCF.
- For avoiding faulty consecutive retransmissions, the **retry counters and timers** are employed to limit the lifetime of a frame.

**Disabling RTS / CTS Mechanism :**

- It is possible to disable the RTS/CTS mechanism by an attribute in the IEEE 802.11 management information base (MIB).
- The length of a frame that is required to be preceded by the RTS and CTS frames is defined by the value of the dot 11RTS threshold attribute.
- The RTS/CTS frames are employed if the frame size is larger than this threshold, otherwise, the frame can be transmitted directly.
- The RTS/CTS mechanism can also be disabled in the following situations :
  1. If the bandwidth demand id low;
  2. If all the stations are concentrated in an area such that all of them able to hear the transmissions of every other stations;
  3. If the contention level for the channel is low.
- The default value for this threshold is 128 and, by definition, an AP is heard by all stations in its BSS and will never have a hidden node.

### 8.17.2 Different Time Intervals :

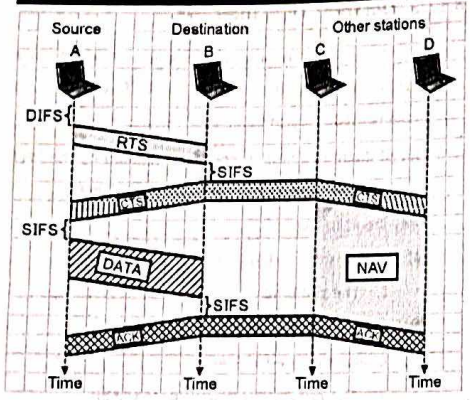
- In IEEE 802.11 standard there are five different time intervals are defined and both the DCF and PCF are implemented through them.
- Out of them, two time intervals are defined by the PHY layer and the remaining three by the MAC layer
- They are as follows :
  (a) Intervals defined in PHY :
    1. The slot time
    2. The short interframe space (SIFS)
  (b) Intervals defined in MAC :
    3. The priority interframe space (PIFS);
    4. The distributed interframe space (DIFS);
    5. The extended interframe space (EIFS).
- The basic function of inter frame spaces (IFSs) is to provide priority levels for channel access.
- The values of IFSs in IEEE 802.11b are as follows :
- The SIFS has the smallest value equal to 10 μsec, followed by the slot time which has a value of 20 μsec.
- The PIFS time period is equal to the sum of SIFS and one slot time whereas the DIFS is equal to the sum of SIFS and two slot times.
- The EIFS interval is much longer than any other interval and it is used by a station to set its NAV when it receives an erroneous frame.
- The values of these intervals may change from standard to standard.

### 8.17.3 Distributed Co-ordination Function (DCF) :

- IEEE has defined two protocols at the MAC sublayer. One of these two protocols is called as the distributed co-ordination function (DCF).
- The access method used by DCF is CSMA/CA.

**Frame Exchange Time Line :**

- The exchange of control and data frames with time has been shown Fig. 8.17.2.




(G-2100) **Fig. 8.17.2 : CSMA/CA and NAV**

- We assume that there are four wireless stations A, B, C and D present in a wireless LAN.
- A is a source and B is the destination. Therefore C and D are referred to as other stations.
- The sequence of control and data exchange is as given below :

1. The source station A senses the medium for its idleness before sending a frame. It does the media sensing by checking the energy level at the carrier frequency.
   (a) A persistence strategy is used with back of until the channel is found to be idle.
   (b) Once the channel is found to be idle, the source station A waits for a specific amount of time called as the Distributed Interframe Space (DIFS). After this waiting time the station A sends a control frame called as Request to Send (RTS) as shown in Fig. 8.17.2.
2. After receiving the RTS, the destination station B waits for a specific amount of time called the Short Interframe Space (SIFS) and then sends a control frame Clear to Send (CTS) back to the source station A. The CTS frame is an indication that the destination station is ready for receiving the data.
3. The source station receives the CTS frame, waits for a duration of SIFS and then sends the data to the destination station.
4. The destination station receives the data, waits for a duration of SIFS and sends the acknowledgement (ACK) frame to indicate that it has received the data frame.

- Note that in the CSMA/CA protocol, the acknowledgement (ACK) is needed because otherwise the source station does not have any means to know that the data has been received by the destination station.
- In CSMA/CD the ACK is not needed because the lack of collision itself is treated as an acknowledgement of data being received successfully.

**Network Allocation Vector (NAV) :**

- The question here is how do other stations restrain from sending their data when one channel is already transmitting ?
- In other words how is the collision avoidance is practically accomplished ?
- The answer to both these questions is a special feature called as NAV.
- The concept of NAV i.e. Network Allocation vector is as follows :
- When station A sends an RTS frame (see Fig. 6.10.3), which consists of the time duration for which A needs to use the channel, the stations which are affected by this transmission create a timer called as NAV.
- The NAV will indicate the amount of time that must pass before these stations can check again, whether the cannel has again become idle.
- This happens every time when a station sends its RTS frame, the other stations will initiate their NAV.
- During the NAV interval no other station will initiate its transmission.
- In this way the collision avoidance aspect of the CSMA / CA protocol is accomplished.
- The other stations check the channel for idleness only after the expiry of their NAV.

**Collision During Handshaking :**

- If the collision takes place when the RTS and CTS frames are in transition, then it is called as collision during handshaking.
- In such a situation, two or more stations try to send the RTS frame at the same time, which may collide with each other.
- But in CSMA/CA there is no mechanism to detect such collisions.





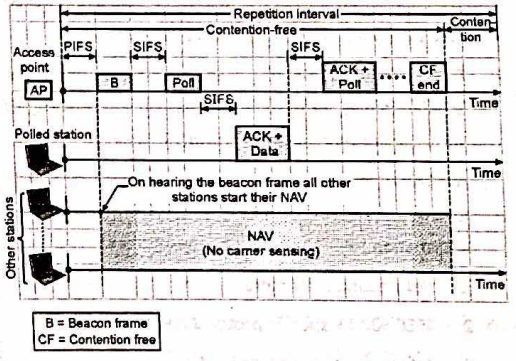
- So the collision of RTS frames also will go undetected.
- Then how will the sources know that a collision has taken place ?
- Well, a source will assume that the collision has taken place if it does not receive the **CTS** frame from the receiver in response to RTS.
- In such events, the sender applies the **back-off** strategy and tries after sometime. This is how the collision during handshaking is handled by CSMA/CA in the wireless environment.

## 8.17.4 Hidden Station Problem :

- Let us see now, how CSMA/CA avoids the hidden station problem.
- Refer Fig. 8.17.2. Actually the RTS and CTS frames (handshake frames) are used to solve the hidden station problem.
- As shown in Fig. 8.17.2, the RTS message from A can reach B but not C (because C is out of range of A).
- In response to this, station B sends the CTS frame. Since both A and C are in the range of B, the CTS frame will reach stations A as well as C.
- Due to this CTS frame station C will understand that some hidden station (A in this case) is already using the channel.
- Therefore C will refrain from transmitting. This will avoid the possible collision.

## 8.17.5 Point Co-ordinate Function (PDF) :

- An optional access method which can be implemented in the infrastructure network (a wireless LAN with AP) but not in the ad-hoc network (WLAN without and AP) is called as the Point Co-ordination Function (PDF).
- Note that the PCF is implemented on top of the DCF and used mostly for those applications that are time sensitive.
- The access method used by PCF is the centralized, contention free polling access method. The polling for stations that can be polled is performed by the A.P.
- These stations when polled in a sequential manner will send their data to AP on one by one basis.
- One more inter-frame space called PIFS has been defined to give priority to PCF over DCF. This inter-frame space PIFS (PCF IFS) is shorter than DIFS.
- The meaning of this arrangement is that if at the same time, a station wants to use DCF and an AP wants to use PCF, then the priority will be given to the AP.
- But the effect of this higher priority of PCF over DCF, is that the stations using DCF may never get the access to the common medium.
- But this should be avoided. Hence a **repetition interval** has been designed in order to cover the PCF (contention-free) as well as DCF (contention-based) traffic.
- The **repetition interval** always starts with a special type of frame called **beacon frame** as shown in Fig. 8.17.3.




(G-2101) **Fig. 8.17.3 : Example of repetition interval**

- The repetition frame is repeated continuously.
- As shown in Fig. 8.17.3, on hearing the beacon frame, the stations start their NAV for the duration of the contention free period of repetition interval.
- An example of repetition interval has been shown in Fig. 8.17.3.
- The PC (Point controller) can perform the following operations or any combination of them, during the repetition interval. 802.11 uses the concept of piggybacking.
- The PC (point controller) sends a **CF end** (contention - free end) frame at the end of the contention free period, so that the contention based (DCF traffic) stations can use the common medium.

### 8.17.6 Fragmentation :

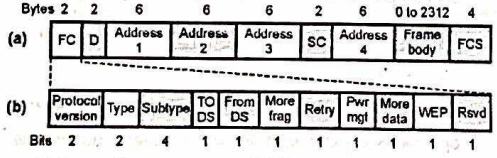
- In the wireless communication, the frames often get corrupted because the wireless environment is extremely noisy.
- The source has to retransmit the corrupt frame.
- Therefore the fragmentation process which is the process of dividing a large frame into small ones is recommended by the protocol.
- This is because in the event of corruption and retransmission, it is always better to resend a small frame than a big one.

## 8.18 802.11 Frame Format :

- The MAC layer accepts MAC Service Data Units (MSDUs) from higher layers and adds to it its headers and trailers to create MAC Protocol Data Units (MPDU).
- Optionally, the MAC may fragment MSDUs into many smaller frames, in order to increase the probability of successful delivery of each individual frame.
- A MAC frame contains the header followed by MSDU followed by the trailer contain the following information :
  1. Addressing information
  2. IEEE 802.11-specific protocol information
  3. Information for setting the NAV
  4. Frame check sequence.

**General frame format :**

- The general format of a MAC layer frame is as shown in Fig. 8.18.1. It consists of nine fields.




**(a) Frame format** **(b) Frame control (FC) field**

**(G-2149) Fig. 8.18.1**

- Various important fields in this frame are as follows :

**1. FC (Frame Control) :**

- This 2 byte long field is used for defining the type of the MAC frame. It also defines some control information.
- As shown in Fig. 8.18.1, the FC field has been subdivided into 11 subfields. Table 8.18.1 describes these subfields in short.

**Table 8.18.1 : Subfields in FC field**

| Field | Bits | Description |
|---|---|---|
| Version | 2 | Current version is 0. |
| Type | 2 | Type of information : Management (00) central (01) or data (10). |
| Subtype | 4 | Defines subtype of each type of frame (see Table 8.18.2). |
| To DS | 1 | Defined later. |
| From DS | 1 | Defined later. |
| More frag | 1 | If this is 1, it means more fragments. |
| Retry | 1 | If this is 1, it means retransmitted frame. |
| Pwr mgt | 1 | If it is 1, it means station is in power management mode. |
| More Data | 1 | If it is 1, it means station has more data to send. |
| WEP | 1 | Wired Equivalent Privacy (Encryption implemented). |
| Rsvd. | 1 | Reserved. |

**2. Duration ID (D/ID) :**

- This is a 2-byte long field which is used to define the transmission which is used to set the value of NAV.
- In one control frame, it is also used to define the ID of the frame.

**3. Addresses Field :**

- As shown in Fig. 8.18.1, there are four address fields from address – 1 to address – 4 and each field is of 6 - byte length.
- As will be discussed later on the values of **To DS** and **From DS** sub fields will decide the meaning of each address field.

**4. SC (Sequence Control) :**

- The sequence control or SC field is 2 byte or 16 bit long.
- Out of these 16 bits, the first four bits are used for defining the fragment number.





- The sequence number which is same for all the fragments is defined by the remaining 12-bits in SC.
- The four bit long fragment number sub-field is assigned to each fragment of an MSDU.
- The field for the first fragment is set to zero while subsequent fragments are incremented sequentially.
- The 12 bit long sequence number sub-field has a constant number for each MSDU which is incremented for each following MSDUs.

**5. Frame body :**

- This is a field with variable length up to 2304 bytes and 2312 bytes when encrypted. The information contained in this field, is specific to the particular data or management frame.

**6. FCS :**

- The frame check sequence is a 4-byte long field and it carries the CRC-32 error detection sequence.

### 8.18.1 Comparison of Ethernet and WLAN :

**Table 8.18.1 : Comparison of Ethernet and WLAN**

| Sr. No. | Ethernet | Wireless network |
|---|---|---|
| 1. | IEEE standard 802.3 | IEEE standard 802.11 |
| 2. | Communication medium is coaxial cable | Infrared or radio frequencies act as medium |
| 3. | Spread spectrum is not used. | Spread spectrum is used |
| 4. | Uses MAC | Uses two MAC sublayers |
| 5. | Uses CDMA/CD | Uses CSMA/CA |
| 6. | Efficiency is high | Efficiency is low |
| 7. | Addressing is simpler | Addressing is complicated |
| 8. | Large range | Short range. |

### 8.18.2 Advantages of WLAN :

- The following are a few advantages of deploying WLANs :
  1. The advantages of wireless LANs are the mobility and flexibility they provide so that a network user can move around without any restrictions and still remain connected to the network.
  2. In addition to increased mobility, wireless LANs offer increased flexibility. It is possible to establish or tear down a WLAN in a very short time.
  3. It is also very easy to add new members to a pre-existing WLAN.
  4. Sometimes, it may even be economical to use a wireless LAN.
  5. Cost-effective network setup for hard-to-wire locations such as older buildings.
  6. Reduces the cost of ownership.

### 8.18.3 Limitations of WLAN :

- Following are some of the major limitations of WLANs :
  1. Spectrum assignment and operational conditions are not same worldwide.
  2. Radiated power is limited to 100 mW. So the range will be limited.
  3. Wi-Fi networks have a limited range typically 35 m or 120 ft indoor and 100 m or 300 ft outdoor.
  4. There are data security risks. Wi-Fi networks are not protected thoroughly.
  5. Wi-Fi connections can be easily disrupted.

### 8.18.4 Applications of Wireless LAN :

- Due to flexibility and possibility to configure in a variety of topologies, WLANs can be used in a number of varied applications.
- Some of them are as follows :
  1. For accessing the Internet, checking E-mails and receive/send instant messages when the user is moving.
  2. WLANs can set up networks in the locations affected by earthquakes or other disasters where no suitable infrastructure is available and wired networks have been destroyed.
  3. In places of historic importance, where wiring may not be permitted, the WLAN can be used easily and effectively.

## 8.19 Wireless PAN (WPAN) IEEE 802.15 :

**Definition :**

- A **WPAN** is simply a short-distance network that allows many devices within a small area to connect to one another with wireless links.
- WPAN is a Wireless Personal Area Network. It is one-step down from WLANs.

- The WPANs cover smaller areas as compared to WLANs and need to use less power for transmission.
- WPANs are used for networking of portable and very small computers, cell phones, printers, speakers, microphones, etc.
- They are short-range to very short-range wireless networks. WPANs are used to exchange data between devices within the reach of an individual.

**Examples of WPAN :**

- The best example of WPAN is Bluetooth. Bluetooth can be found in many consumer electronics devices such as mobile phones, PDAs and wireless headsets etc.
- Some other examples are Zigbee (IEEE 802.15.4) and Ultra Wide Band or UWB (IEEE802.15.3).

**IEEE standard :**

- The IEEE 802 has established the IEEE 802.15 standard for WPANs. In wireless technologies, inter-working to generate heterogeneous wireless environment is an issue of WPANs.
- For example WPANs and WLANs can allow an extension of the 3G cellular networks (UMTS and cdma2000) into the devices without direct cellular access.
- The devices interconnected in a WPAN can make use of a combination of 3G access and WLAN access by selecting the type of access, which is best at a given time.
- In such networks, 3G, WPAN and WLAN technologies will not fight, instead they complement each other and allows users to choose the best connectivity for his/her purpose.
- This type of network allows users to tether computers, smart phones, tablets, printers and other wireless enabled devices for data sharing.

## 8.20 Bluetooth (WPAN) (IEEE 802.15.1) :

- WLANs can be preferred for connecting infrastructure based services via a wireless carrier provider. But it does not allow the personal devices to communicate wirelessly among each other.

Therefore a Personal Area Network **(PAN)** has been developed to fulfill this need.

- Bluetooth is the name given to a new technology which uses short-range radio links, that can replace the cable(s) connecting portable and/or fixed electronic devices.
- It is advantageous that it will allow the replacement of the many propriety cables that connect one device to another with one universal radio link.
- Its key features are robustness, low complexity, low power and low cost. Bluetooth has been designed to operate in noisy frequency environments.
- Therefore it uses a fast acknowledgement and **frequency-hopping** scheme to make the link robust.

**Frequency band :**

- Bluetooth radio modules operate in the unlicensed ISM band at 2.4 GHz, and avoid interference from other signals by hopping to a new frequency after transmitting or receiving a packet.
- Compared with other systems in the same frequency band, the Bluetooth radio hops faster and uses shorter wavelengths.
- Thus Bluetooth is a wireless LAN technology designed to connect devices of different functions such as telephones, computers, printers, cameras, etc.
- A Bluetooth LAN is an Ad hoc network. It is possible to connect the Bluetooth LAN to the Internet.
- This technology is implemented using the IEEE 802.15 standard. The power consumption of Bluetooth technology is low and it offers a range of operation up to ten meters.
- These features have made it find many applications.
- A Bluetooth LAN is small by nature. Therefore there will be a chaos if many gadgets try to connect at a time.
- According to the IEEE 802.15 standard, a wireless personal area network (PAN) can operate successfully in the area of the size of a room or a hall.

### 8.20.1 Features of Bluetooth :

- Following are the features of Bluetooth :
  1. It is implemented using IEEE 802.15 standard.
  2. Bluetooth used the ISM band at 2.4 GHz.
  3. It uses the FHSS technique.





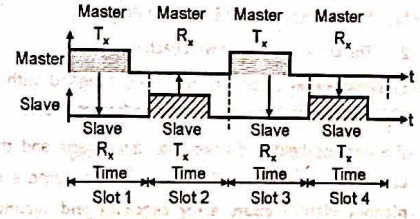
4. The specified rate of data transfer is 1 Mbps.
5. Its maximum range is 10 m.
6. It is a wireless PAN standard which uses radio links instead of wires.
7. Bluetooth uses a form of TDMA called as TDD-TDMA.
8. Type of modulation used : GFSK.
9. Bluetooth does not have a very good security.
10. It uses 1600 hops/sec.
11. It can be used for the transfer of data, or photographs.

### 8.20.2 Bluetooth Devices and Frequency Band :

- Bluetooth radio modules operate in the unlicensed **ISM** band at 2.4 GHz, and avoid interference from other signals by hopping to a new frequency after transmitting or receiving a packet.
- Every Bluetooth device consists of a built in short range **radio transmitter.**
- The current data rate is 1 Mbps and the bandwidth is 2.4 GHz.
- So an interface between the IEEE 802.11 wireless LAN and Bluetooth LAN is possible.
- The Bluetooth specification standard defines a short-range (10-meter) radio link.
- The devices carrying Bluetooth-enabled chips can easily transfer data at a rate of about 1 Mbps (Megabits per second) within 10 meters (33 feet) of range through walls, clothing and luggage bags.
- The interaction between devices occurs by itself without direct human intervention whenever they are within each other's range.
- In this process, the software technology embedded in the Bluetooth transceiver chip triggers an automatic connection to deliver and accept the data flow.
- Since Bluetooth is of short range with limited speed and low-power technology.
- It is less attractive to corporate wireless local area networks that are generally powered with the 802.11 wireless LAN technologies.
- Each Bluetooth-enabled device contains a 1.5-inch square transceiver chip operating in the ISM (Industrial, Scientific, and Medical) radio frequency band of 2.40 GHz to 2.48 GHz.
- This frequency is generally available worldwide for free without any licensing restrictions. The ISM band is divided into 79 channels with each carrying a bandwidth of 1 MHz.

### 8.20.3 Principle of Bluetooth :

- It is possible to use up to eight active Bluetooth devices in a small network, that is also called as a **piconet.**
- Inside a piconet the communication takes place between a predominant device called as the master and all other devices, called slaves. After establishing a connection, a slave synchronizes its time and frequency hopping to the masters time and frequency and waits to be polled by the master to transmit.
- Therefore a packet sent by a master is always followed by the master's packet.
- Refer Fig. 8.20.1 to understand the transmission of one-slot-type packet between a master and a slave.




**(G-2656) Fig. 8.20.1 : Principle of Bluetooth. $R_x$ : receiver; $T_x$ : transmitter**

- Here the master transmits a packet in a time slot and the slave sends its data to master in next time slot. The time slots must be synchronized between the master and the slave.

## 8.21 Bluetooth Architecture :

- In Bluetooth two different types of networks are defined. They are as follows :

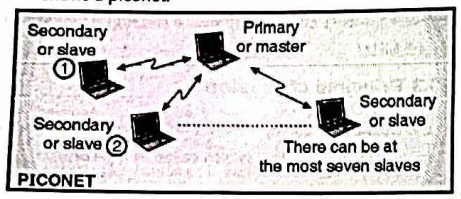
  1. Piconets. 2. Scatternets.

- Let us discuss them one by one.

### 8.21.1 Piconets :

- The first type of Bluetooth network is called as a **piconet** or a **small net.** It can have at the most eight stations. One of them is called as a **master** and all others are called as **slaves.**
- All the slave stations are synchronized in all aspects with the master.

- A piconet can have only one master station. Fig. 8.21.1 shows a piconet.

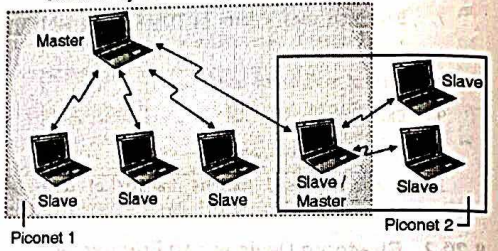



(G-2160) **Fig. 8.21.1 : A piconet**

- A master can also be called as a primary station and slaves are secondary station.
- The communication between a master and slaves can be one-to-one or one-to-many. Note that the communication takes place between the master and slaves but no direct communication takes place between the slaves.
- The formation of a piconet is governed by two factors :
  1. The address of each Bluetooth device.
  2. The clock associated with each device.
- Every device in a piconet has been assigned with a 48 bit address which is similar to an Ethernet address.
- The address field is divided into three parts and the Lower Address Part (LAP) is used for the purpose of **piconet identification, error checking and security checks.** Every device has a 28-bit clock which is called as **native clock.**
- Its frequency is 3200 pulses per second i.e. once in 312.5 μs, because this is exactly twice the normal hopping rate of 1600 hops/second.
- A piconet can have upto **seven active** slaves at any given instant of time. In order to identify a slave, each one is assigned a locally unique active member address AM-ADDR.
- Eventhough the maximum number of secondaries in a piconet is seven, there can be additional secondaries in the **parked state.**
- When a secondary is in the parked state, it is synchronized with the primary but cannot take part in any type of communication unless we move it to its **active state.**
- As the total number of active stations in a piconet can at the most be equal to eight, if a secondary is brought to the **active state** from **parked state,** some other secondary should go to the **parked state.**

## 8.21.2 Scatternets :

- Many piconets may exist simultaneously in a given area and they may even overlap each other. A scatternet is obtained by combining piconets as shown in Fig. 8.21.2.




(G-2161) **Fig. 8.21.2 : Scatternet**

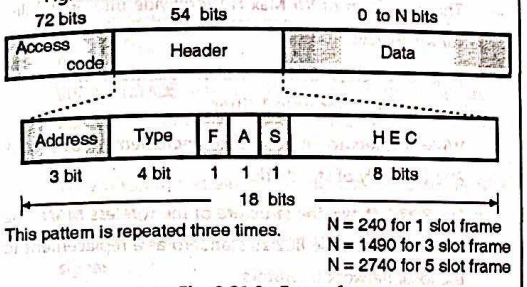
- Fig. 8.21.2 shows a scatternet consisting of two piconets. A slave in the first piconet can act as a master in the second piconet.
- It will receive the messages from the master in the first piconet by acting as a slave and then delivers the message to the slaves in the second piconet as shown in Fig. 8.21.2.
- So the same device acts as a slave in the first piconet and as master in the second piconet.
- With increase in the number of piconets, the possibility of collision increases. This will result in degradation of performance.
- Therefore a device can participate in two or more piconets by means of the time sharing process.
- To do so it must use the associated master's address and proper clock offset.
- A Bluetooth device can act as a master in only one piconet but it can work as a slave in multiple piconets. One of the important issues in a scatternet is that the utilization of bandwidth is not optimal.
- This happens because a device changes its role and takes part in different piconets. Another important issue is the timing that a device would be missing when it participates in more than one piconets.
- If a master of one piconet temporarily becomes slave in some other piconet then it will be missing from its own piconet for that much time. This reduces the quality of the Bluetooth link.





## 8.21.3 Frame Format :

- A frame in the baseband layer can be one of the three types :
  1. One slot 2. Three slot 3. Five slot.
- The frame format of the three frame types is shown in Fig. 8.21.3.




(G-391)**Fig. 8.21.3 : Frame format**

- The description of important fields is as follows :

**1. Access code :**

- It is a 72 bits field which contains the synchronization bits. It also contains the identifier of the master so as to distinguish the frame of one piconet from another.

**2. Header :**

- It is a 54 bits field which contains an 18 bits pattern repeated three times. (see Fig. 8.21.3).
- Each such 18 bits pattern consists of the following fields.
- **Address :** It is a three bit field. So it can define upto seven slaves (1 to 7). The 000 address is reserved for the broadcast communication between a master and the slaves. The other addresses from 001 to 111 define seven slaves.
- **Type :** This is a four bit subfield used for defining the type of data, coming from the upper layers.

**F :** This bit is used for the flow control. F = 1 is an indication of buffer full, that means the device can not receive more frames.

**A :** This bit is for acknowledgement. Since Bluetooth uses the stop-N-wait ARQ only 1 bit is sufficient to send an acknowledgement.

**S :** This bit is used to hold the sequence number.

**HEC :** This is an eight bit header error correction subfield. It contains the checksum for detection of errors in the 18 bit header section.

**3. Data (Payload) Field :**

- This field contains the data or control bits. It can be 0 to 2740 bits long. It contains data or control bits coming from the upper layers.

## 8.21.4 Bluetooth Advantages :

1. One can create a personal area network at home or on the road with Bluetooth-enabled devices such as keyboard, mouse, scanner, PDA, laptop, cell phone, etc.
2. This network can automatically help synchronize notes, calendar, address book and also print pictures, receive emails, access cell phones messages, etc. It can even help consumers pay bills with credit card through Bluetooth cash register if a Bluetooth PDA stores the card information

## 8.21.5 Bluetooth Limitations :

1. Bluetooth communication does not support routing.
2. The issues of handoffs have not been addressed.
3. Due to master slave configuration, many times performance degradation takes place due to bottlenecking at the master.
4. Interference with WLAN is essential as WLAN and Bluetooth both operate in the same ISM frequency band.

## 8.21.6 Applications of Bluetooth :

- Some applications of this technology are as follows :
  1. Ad-hoc network of laptops for interactive conference.
  2. Transferring data, photographs from one cell phone to other cell phones or computers and vice versa.
  3. Connecting a digital camera wirelessly to a mobile phone.
  4. Three in one phone where the same phone functions as an intercom, a cordless phone and a mobile phone.
  5. Mouse, printer, keyboards etc can be connected to a computer without using wires.
  6. Wireless head phones for mobile phones.
  7. Wireless interface between a computer and printer.

### 8.21.7 Comparison of WPAN and WLAN :

- Table 8.21.1 gives the comparison between WPAN and WLAN.

**Table 8.21.1 : Comparison of WPAN and WLAN**

| Sr. No. | Parameter | WPAN | WLAN |
|---|---|---|---|
| 1. | Protocol | 802.15 | 802.11 |
| 2. | Standards | Bluetooth | WiFi |
| 3. | Coverage | Within reach of a person | Within a building or campus |
| 4. | Performance | Moderate | High |
| 5. | Frequency range | 2.4 – 2.483 GHz | 5.15 – 5.35 GHz |
| 6. | Cell radius | 1 – 10 m | 1 – 500 m |
| 7. | Modulation | FHSS | OFDM, DSSS |
| 8. | Application | Cable replacement for peripheral devices. | Mobile extension of wired networks. |

### 8.21.8 Comparison of B.T. and WLAN :

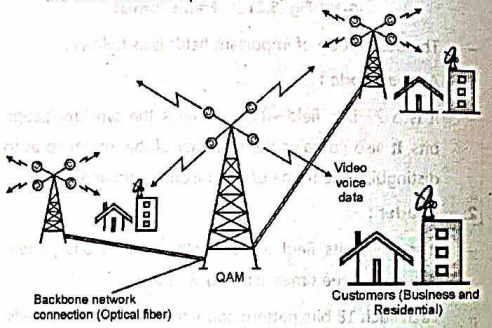
- The comparison between Bluetooth and wireless LAN is as given in Table 8.21.2.

**Table 8.21.2 : Comparison of Bluetooth and WLAN**

| Sr. No. | Parameter / Characteristics | WLAN | Bluetooth |
|---|---|---|---|
| 1. | Standard | IEEE 802.11 X | 802.15 |
| 2. | Number of hops / second | 2.5 hops/s | 1600 hops/s |
| 3. | Data transfer rate | 11 Mbps | 1 Mbps |
| 4. | Transmission range | Indoor : 50 – 150 m<br>Outdoor : 300 m | 10 m |
| 5. | Network contains | Laptops and other mobile devices | Smart phones |
| 6. | Modulation | CCK (Complementary Code Keying) | GFSK |
| 7. | Type of standard | LAN standard for a long time network | Standard for short time network |
| 8. | Data rate | Slow data rate of 1 Mbps | 24 to 54 Mbps |
| 9. | Power consumption | High | Low |
| 10. | Cost | Higher | Lower |
| 11. | Noise | Less noise resistant | More noise resistant |

## 8.22 Wi-Max ( IEEE 802.16) :

- The long form of Wi-Max is Worldwide Interoperability for Microwave Access.
- It is a wireless communication standard which can provide data rates upto 1 Gbps.
- Wi-Max refers to interoperable implementations of IEEE 802.16 family of standards.
- Fig. 8.22.1 shows the structure of the wireless MAN. The IEEE developed the 802.16 standard as a replacement to the local network operators.




(L-430) **Fig. 8.22.1 : Broadband wireless MAN IEEE 802.16**

- The 802.16 WMAN provides a standard solution for a cable free telecommunication service market.
- The IEEE 802.16 standard was published in 2002. Before this time, there were two other wireless MAN solutions already existing.
- They were :
  1. Multichannel Multipoint Distribution System (MMDS)
  2. local Multipoint Distribution System (LMDS)
- These two operated at different frequencies in the millimeter wavelength range.
- Both these WMAN solutions suffered due to lack of standardization. So their use remained restricted.
- The MMDS solution works at 2.4 GHz or 5 GHz band and has a range of upto 50 km. MMDS was designed originally for the wireless CATV solution, without a backward channel from the customer





- But in order to make as much revenue as possible the MMDS solutions are used for voice and internet services.
- The data rates for such applications are 128 kbps upto 3 Mbps downstream from base station to subscriber and 128 kbps from subscriber to the base station.
- The subscribers can access telephony and ISP through a modem.
- 802.16 MMDS and LMDS use the frequencies in the band 2 GHz to 60 GHz. With increase in frequency these waves become more and more directional.
- They travel in a straight line and can be easily absorbed by water. So rain, snow, trees absorb these electromagnetic waves and create errors in the received signal.
- In order to overcome this problem, the signal produced by the base station and the customer stations are encoded using Hamming codes.

### 8.22.1 IEEE Project 802.16 :

- Wi-Max is the outcome of the IEEE 802.16 project. This project was taken up as an effort to standardize the broadband wireless system in 2002. The other name for this standard is **wireless local loop.**
- We will first compare the 802.16 project with the 802.11 project, which is the wireless LAN standard.
- In the 802.11, the base station and a host are separated by a short distance but in 802.16 this distance can be long (typically tens of kilometers).
- 802.11 project defines a connectionless service but 802.16 project is designed for a connection oriented service.

**New standards :**

- IEEE 802.16 was revised later and the following two new standards were created. These standards do not alter the basic principle of original 802.16. Instead they concentrate on the nature of two services :
  1. IEEE 802.16 d - Concentrates on fixed Wi-Max.
  2. IEEE 802.16 e - Defines the mobile Wi-Max.

### 8.22.2 Wi-Max Services :

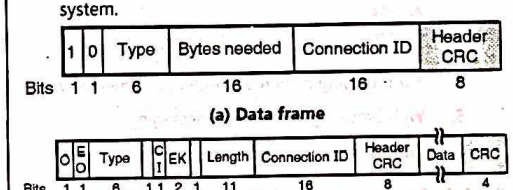
- The following two types of services are provided by Wi-Max to its subscribers :
  1. Fixed Wi-Max services.
  2. Mobile Wi-Max services.

### 8.22.3 Spectrum Allocation :

- There is no uniform global licensed spectrum for Wi-Max.
- However the three licensed spectrum profiles published by the Wi-Max forum are : 2.3 GHz, 2.5 GHz and 3.5 GHz.

### 8.22.4 802.16 Frame Format :

- Fig. 8.22.2 shows the frame formats of 802.16 WMAN system.




(L-432) **Fig. 8.22.2 : 802.16 Frame formats**

- There are two frames, namely the data frame and the control frame.
- In Fig. 8.22.2(a) and (b),

  EO = Encryption payload.

  Type = Type of frame

  CI = Check sum indication

  EK = Which of the 4 encryption keys used

  Length = Including Header

  Header CRC = $X^8 + X^2 + X + 1$
- The control frame is as shown in Fig. 8.22.2(b). It is used to request time slots for different types of services.
- The control frame identifies the terminal through the connection ID.
- The "type" field which is 6 bit long identifies the type of control packet.
- Bytes needed field identifies how much data (in terms of number of bytes) the terminal wants to transmit.
- Fig. 8.22.2(a) shows the data frame format. The first bit in the frame decides whether this frame is a data frame or control frame.

  0 – Data frame

  1 – Control frame

- The 6 bit type field identifies the type of frame connection ID and Header CRC fields are same as those in the control frame.

### 8.22.5 Applications of Wi-Max :

- The Wi-Max can be used in the following applications :
  1. To provide portable mobile broadband connectivity.
  2. It can be used as an alternative to cable, Digital Subscriber Line (DSL) for providing a broadband access.
  3. To provide services such as Voice on IP (VoIP).
  4. For providing a source of Internet connectivity.
  5. Web browsing and instant messaging
  6. Wireless telephone services

### 8.22.6 Comparison of WLAN and Wi-Max :

**Table 8.22.1 : Comparison of WLAN and Wi-Max**

| Sr. No. | Parameter / Characteristics | IEEE 802.11 | IEEE 802.16 |
|---|---|---|---|
| 1. | Type of standard | This is designed for wireless LANs | Designed for wireless WAN, MAN. |
| 2. | Distance between BS and subscriber | Very short | Very long (few tens of km). |
| 3. | Type of service | Connectionless | Connection oriented |
| 4. | Number of users | Few | Large |
| 5. | Bandwidth per user | Small | Large |
| 6. | Frequency band | ISM band | Millimeter wave band and microwave band |
| 7. | QoS | Not guaranteed | All transmissions are QoS guaranteed |

## Review Questions

Q. 1 Explain the layered architecture of LAN explaining the function of the LLC and MAC sublayer.

Q. 2 What is static and dynamic channel allocation ?

Q. 3 Compare and explain the pure and slotted ALOHA system.

Q. 4 Explain the different CSMA protocols.

Q. 5 What is CSMA with collision detection ?

Q. 6 Why there is no need of CSMA/CD for a full duplex Ethernet LAN ?

Q. 7 Explain CSMA/CD.

Q. 8 What is CSMA/CA ?

Q. 9 Define : WPAN.

Q. 10 Write a short note on wireless PAN.

Q. 11 What are the needs of WPAN ?

Q. 12 Explain in brief features and specification of Bluetooth.

Q. 13 Write a note on Bluetooth architecture.

Q. 14 Explain frame format in baseband layer.

Q. 15 Explain single secondary communication and multiple secondary communication in TDMA.

Q. 16 Write a short note on piconet synchronization.

Q. 17 Write a note on advantages, disadvantages and applications of Bluetooth.

Q. 18 What are the issues in Bluetooth interference ?

Q. 19 Compare WLAN and Bluetooth.

❑❑❑