UNIT II NETWORKS, PHYSICAL AND DATALINK LAYERS 10 hrs.

Point to Point Networks, Routing and Flow Control, Packet Communication Technology, Packet Broadcasting, Terrestrial Networks. Physical Layer: Guided transmission media – Magnetic media, Twisted Pair, coaxial cable, fiber optic network. Data Link Layer: Design Issues, Error detection and correction, Elementary Data Link Protocols, Sliding Window Protocols, Protocol Verification, Example Data Link protocols.

2.1.Point to Point Networks

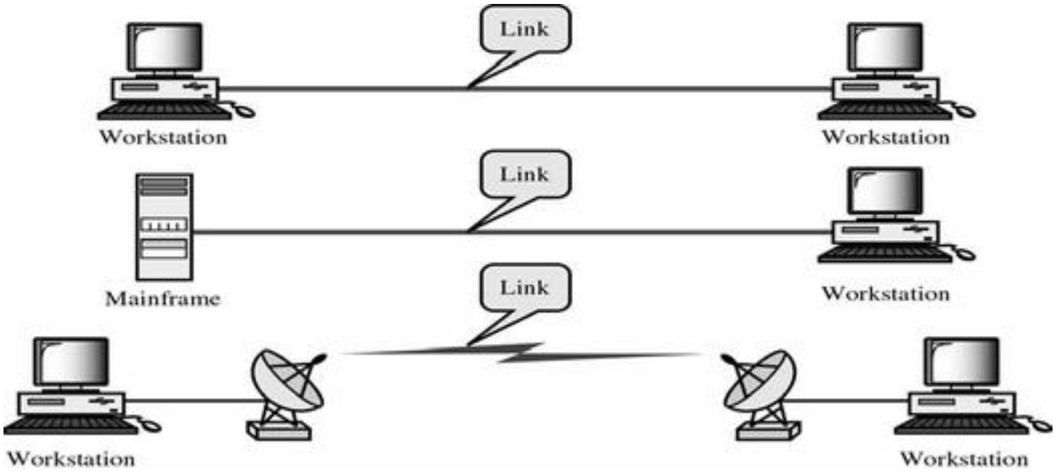


Fig 2.1 Point to Point Networks

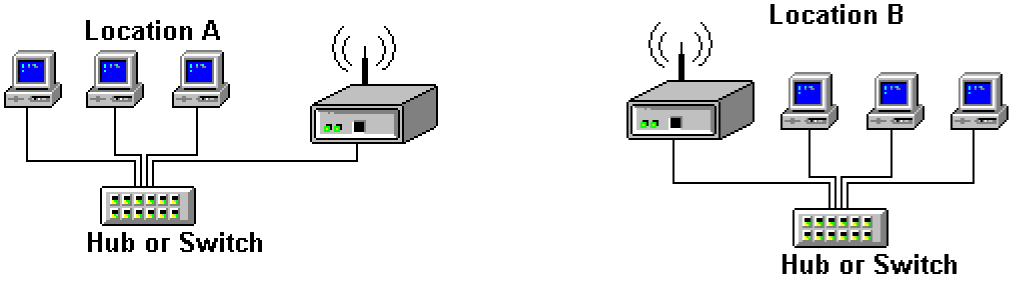


Fig 2.2 Point to Point Networks scenario

Point to point networks is used to connect one location to one other location. The above diagram shows two connected multi-user networks (using a Hub or Switch). Either location (A or B) may be configured as direct (without a Hub or Switch).

Application:

Internet, Intranet or Extranet configurations. ISP access networks (bridged or routed). LAN to LAN applications (bridged or routed see below). Remote data capture (Telemetry or SCADA). Remote Control. Remote Monitoring. Security.

Bridged or Routed

In a bridged connection the traffic sent from location A to Location B AND from Location B to Location A consists of:

- Traffic for a PC or system on the remote network
- Broadcast traffic (e.g. network management)
- Multicast traffic

In effect the two locations operate as a single, fully transparent LAN. Where both LANs consist of many systems the broadcast traffic can be considerable and consideration should be given to a routed network.

In a routed connection the traffic sent from location A to Location B AND from Location B to Location A consists of:

- Traffic for a PC or system on the remote network only

In this case the two LANs operate independently but communication is enabled between them.

2.2Routing and Flow Control: The two main functions performed by a routing algorithm are the selection of routes for various origin-destination pairs and the delivery of messages to their correct destination once the routes are selected. The second function is conceptually straightforward using a variety of protocols and data structures (known as routing tables), some of which will be described in the context of practical networks in Section 5.1.2. The focus will be on the first function (selection of routes) and how it affects network performance. There are two main performance measures that are substantially affected by the routing algorithm-throughput (quantity of service) and average packet delay (quality of service). Routing interacts with flow control in detaining these performance measures

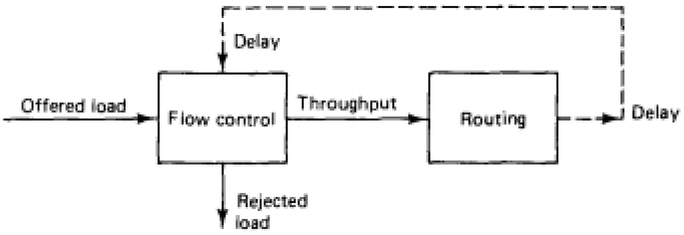


Figure 2.3 Interaction of routing and flow control. As good routing keeps delay low, flow control allows more traffic into the network.

by means of a feedback mechanism shown in Fig. 2.3. When the traffic load offered by the external sites to the subnet is relatively low, it will be fully accepted into the network, that is,

Throughput = offered load

When the offered load is excessive, a portion will be rejected by the flow control algorithm and throughput = offered load - rejected load The traffic accepted into the network will experience an average delay per packet that will depend on the routes chosen by the routing algorithm. However, throughput will also be greatly affected (if only indirectly) by the routing algorithm because typical flow control schemes operate on the basis of striking a balance between throughput and delay (i.e., they start rejecting offered load when delay starts getting excessive). Therefore, as the routing algorithm is more successful in keeping delay low, the flow control algorithm allows more traffic into the network. While the precise balance between delay and throughput will be determined by flow control, the effect of good routing under high offered load conditions is to realize a more favorable delay-throughput curve along which flow control operates, as shown in Fig. 2.4. The following examples illustrate the discussion above:

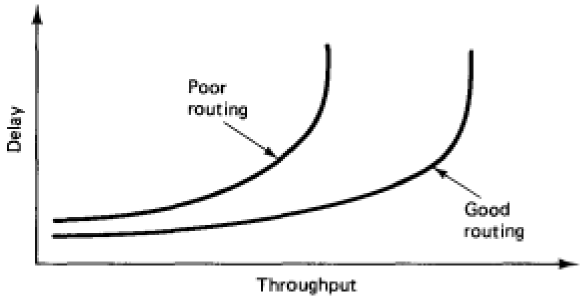


Figure 2.4 Delay-throughput operating curves for good and bad routing

Flow Control

Sender does not flood the receiver, but maximizes throughput

Sender throttled until receiver grants permission

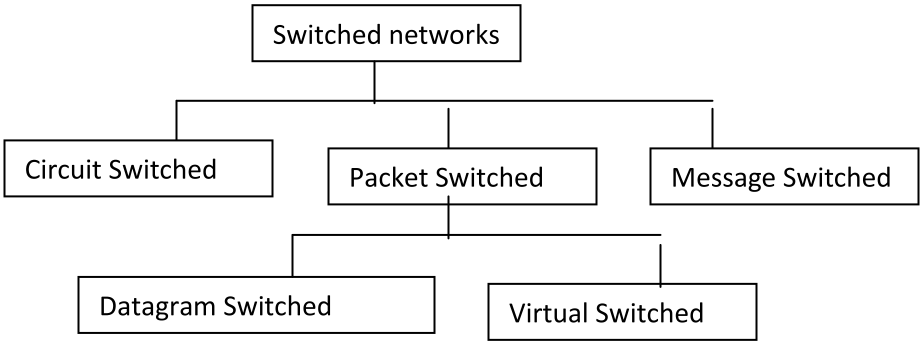
2.3Packet Switching Networks

What is a Computer Network? • Communication Networks: “Sets of nodes that are interconnected to allow the exchange of information such as voice, sound, graphics, pictures, video, text, data, etc…” • Telephone Networks: “ The first well established and most widely used communication networks which are used for voice transmission” – Telephone networks originally used analog transmission as a transmission technology for the information. However, digital transmission started to evolve replacing a lot of the analog transmission techniques used in telephone networks. • Computer Networks: “Collection of autonomous computers interconnected by a technology to allow exchange of information” A network is a series of connected devices. Whenever we have many devices, the interconnection between them becomes more difficult as the number of devices increases. Some of the conventional ways of interconnecting devices are a. Point to point connection between devices as in mesh topology.

b. Connection between central device and every other device – as in star topology

c. Bus topology-not practical if the devices are at greater distances. The solution to this interconnectivity problem is switching. A switched network consists of a series of interlinked nodes called switches. A switch is a device that creates temporary connections between two or more systems. Some of the switches are connected to end systems (computers and telephones) and others are used only for routing.

Taxonomy of switched networks



Circuit switching

- Traditional telephone networks operate on the basis of circuit switching

- In conventional telephone networks, a circuit between two users must be established for a communication to occur

- Circuit switched networks requires resources to be reserved for each pair of end users

- The resources allocated to a call cannot be used by others for the duration of the call

The reservation of the network resources for each user results in an inefficient use of bandwidth for applications in which information transfer is bursty or if the information is small

Packet Switching

- Packet switched networks are the building blocks of computer communication systems in which data units known as packets flow across the networks.

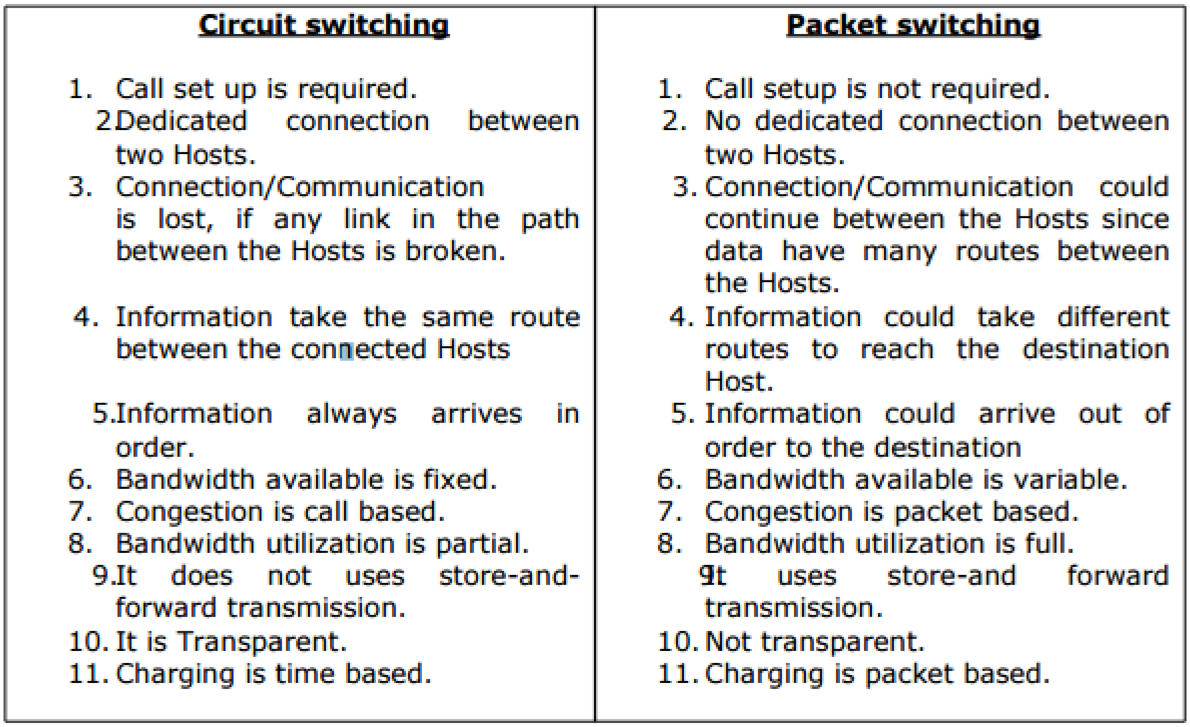
- It provides flexible communication in handling all kinds of connections for a wide range of applications e.g. telephone calls, video conferencing, distributed data processing etc...

- Packet switched networks with a unified, integrated data infrastructure known as the Internet can provide a variety of communication services requiring different bandwidths.

- To make efficient use of available resources, packet switched networks dynamically allocate resources only when required.

• The form of information in packet switched networks is always digital bits.

Differences between Circuit Switching and Packet Switching



Circuit switching	**Packet switching**
1. Call set up is required.	1. Call setup is not required.
2 Dedicated connection between two Hosts.	2. No dedicated connection between two Hosts.
3. Connection/Communication is lost, if any link in the path between the Hosts is broken.	3. Connection/Communication could continue between the Hosts since data have many routes between the Hosts.
4. Information take the same route between the connected Hosts	4. Information could take different routes to reach the destination Host.
5.Information always arrives in order.	5. Information could arrive out of order to the destination
6. Bandwidth available is fixed.	6. Bandwidth available is variable.
7. Congestion is call based.	7. Congestion is packet based.
8. Bandwidth utilization is partial.	8. Bandwidth utilization is full.
9.It does not uses store-and-forward transmission.	9It uses store-and forward transmission.
10. It is Transparent.	10. Not transparent.
11. Charging is time based.	11. Charging is packet based.

Packet networks can be viewed from two perspectives:

• External view of network :- It is Concerned with the services that the network provides to the transport layer

• Internal operation of the network.

Network service can be Connection-oriented service or connectionless service

Connectionless service:

• Connectionless service is simple with two basic interactions (1) a request to network layer that it send a packet (2) an indication from the network layer that a packet has arrived

• It puts total responsibility of error control, sequencing and flow control on the end system transport layer

Connection-oriented service:

The Transport layer cannot request transmission of information until a connection is established between end systems

Network layer must be informed about the new flow

Network layer maintains state information about the flows it is handling

During connection set up, parameters related to usage and quality of services may be negotiated and network resources may be allocated

Connection release procedure may be required to terminate the connection It is also possible for a network layer to provide a choice of services to the user of network like:

best-effort connectionless services

Low delay connectionless services

Connection oriented reliable stream services

Connection oriented transfer of packets with guaranteed delay and bandwidth

2.5Physical Layer

Physical layer is concerned with transmitting raw bits over a communication channel. The design issues have to do with making sure that when one side sends a 1 bit, it is received by the other side as 1 bit and not as 0 bit. In physical layer we deal with the communication medium used for transmission.

Types of Medium

Medium can be classified into 2 categories.

1. **Guided Media:** Guided media means that signals is guided by the presence of physical media i.e. signals are under control and remains in the physical wire. For eg. copper wire.
2. **Unguided Media:** Unguided Media means that there is no physical path for the signal to propagate. Unguided media are essentially electro-magnetic waves. There is no control on flow of signal. For eg. Radio waves.

Communication Links

In a network nodes are connected through links. The communication through links can be classified as

1. **Simplex :** Communication can take place only in one direction. eg. T.V broadcasting.
2. **Half-duplex :** Communication can take place in one direction at a time. Suppose node A and B are connected then half-duplex communication means that at a time data can flow from A to B or from B to A but not simultaneously. eg. two persons talking to each other such that when speaks the other listens and vice versa.
3. **Full-duplex :** Communication can take place simultaneously in both directions. eg. A discussion in a group without discipline.

Links can be further classified as

1. **Point to Point :** In this communication only two nodes are connected to each other. When a node sends a packet then it can be recieved only by the node on the other side and none else.
2. **Multipoint** : It is a kind of sharing communication, in which signal can be recieved by all nodes. This is also called broadcast.

Generally two kind of problems are associated in transmission of signals.

1. **Attenuation :** When a signal transmitts in a network then the quality of signal degrades as the signal travels longer distances in the wire. This is called attenuation. To improve quality of signal amplifiers are used at regular distances.
2. **Noise :** In a communication channel many signals transmits simultaneously, certain random signals are also present in the medium. Due to interference of these signals our signal gets disrupted a bit.

Bandwidth

Bandwidth simply means how many bits can be transmitted per second in the communication channel. In technical terms it indicates the width of frequency spectrum.

Transmission Media

- **Guided**
- **Transmission**
- **Media**

In Guided transmission media generally two kind of materials are used.

- Copper
- Coaxial Cable
- Twisted Pair
- Optical Fiber

1. **Coaxial Cable:** Coaxial cable consists of an inner conductor and an outer conductor which are seperated by an insulator. The inner conductor is usually copper. The outer conductor is covered by a plastic jacket. It is named coaxial because the two conductors are coaxial. Typical diameter of coaxial cable lies between 0.4 inch to 1 inch. The most application of coaxial cable is cable T.V. The coaxial cable has high bandwidth, attenuation is less.

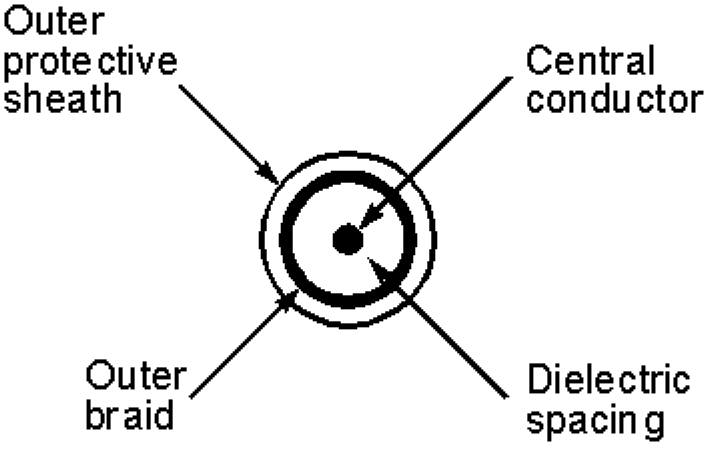


Fig 2.5 **Coaxial Cable**

2. **Twisted Pair:** A Twisted pair consists of two insulated copper wires, typically 1mm thick. The wires are twisted together in a helical form the purpose of twisting is to reduce cross talk interference between several pairs. Twisted Pair is much cheaper then coaxial cable but it is susceptible to noise and electromagnetic interference and attenuation is large.

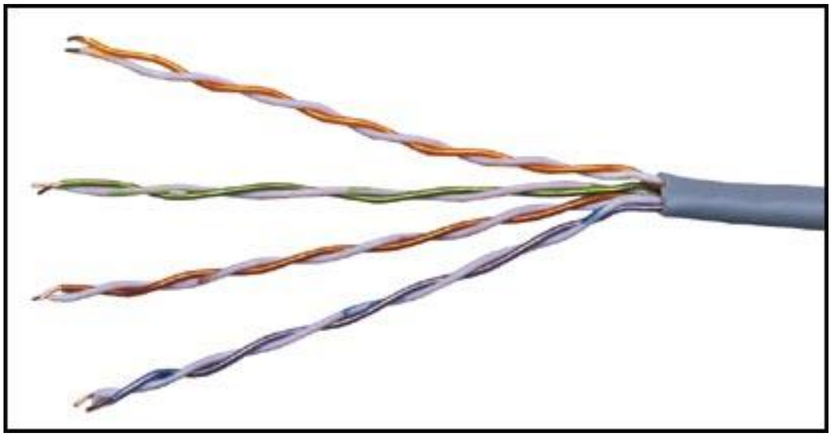


Fig 2.5 **Twisted Pair**

Twisted Pair can be further classified in two categories:
Unshielded twisted pair: In this no insulation is provided, hence they are susceptible to interference.
Shielded twisted pair: In this a protective thick insulation is provided but shielded twisted pair is expensive and not commonly used.

The most common application of twisted pair is the telephone system. Nearly all telephones are connected to the telephone company office by a twisted pair. Twisted pair can run several kilometers without amplification, but for longer distances repeaters are needed. Twisted pairs can be used for both analog and digital transmission. The bandwidth depends on the thickness of wire and the distance travelled. Twisted pairs are generally limited in distance, bandwidth and data rate.

3. **Optical Fiber:** In optical fiber light is used to send data. In general terms presence of light is taken as bit 1 and its absence as bit 0. Optical fiber consists of inner core of either glass or plastic. Core is surrounded by cladding of the same material but of different refractive index. This cladding is surrounded by a plastic jacket which prevents optical fiber from electromagnetic interference and harsh environments. It uses the principle of total internal reflection to transfer data over optical fibers. Optical fiber is much better in bandwidth as compared to copper wire, since there is hardly any attenuation or electromagnetic interference in optical wires. Hence there are fewer requirements to improve quality of signal, in long distance transmission. Disadvantage of optical fiber is that end points are fairly expensive. (eg. switches)

Differences between different kinds of optical fibers:

- Depending on material

- Made of glass
- Made of plastic.
- Depending on radius
- Thin optical fiber
- Thick optical fiber
- Depending on light source
- LED (for low bandwidth)
- Injection laser diode (for high bandwidth)

Wireless Transmission

1. **Radio:** Radio is a general term that is used for any kind of frequency. But higher frequencies are usually termed as microwave and the lower frequency band comes under radio frequency. There are many application of radio. For eg. Cordless keyboard, wireless LAN, wireless Ethernet. But it is limited in range to only a few hundred meters. Depending on frequency radio offers different bandwidths.
2. **Terrestrial microwave:** In terrestrial microwave two antennas are used for communication. A focused beam emerges from an antenna and is received by the other antenna, provided that antennas should be facing each other with no obstacle in between. For this reason antennas are situated on high towers. Due to curvature of earth terrestrial microwave can be used for long distance communication with high bandwidth. Telecom department is also using this for long distance communication. An advantage of wireless communication is that it is not required to lay down wires in the city hence no permissions are required.
3. **Satellite communication:** Satellite acts as a switch in sky. On earth VSAT(Very Small Aperture Terminal) are used to transmit and receive data from satellite. Generally one station on earth transmits signal to satellite and it is received by many stations on earth. Satellite communication is generally used in those places where it is very difficult to obtain line of sight i.e. in highly irregular terrestrial regions. In terms of noise wireless media is not as good as the wired media. There are frequency band in wireless communication and two stations should not be allowed to transmit simultaneously in a frequency band. The most promising advantage of satellite is broadcasting. If satellites are used for point to point communication then they are expensive as compared to wired media.

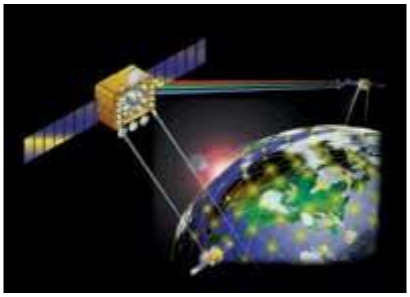


Fig 2.6 **Satellite communication**

2.6 Data Link Layer

2.6.1Data Link Layer Design Issues

This layer provides reliable transmission of a packet by using the services of the physical layer which transmits bits over the medium in an unreliable fashion. This layer is concerned with :

- Framing : Breaking input data into frames (typically a few hundred bytes) and caring about the frame boundaries and the size of each frame.
- Acknowledgment : Sent by the receiving end to inform the source that the frame was received without any error.
- Sequence Numbering : To acknowledge which frame was received.
- Error Detection : The frames may be damaged, lost or duplicated leading to errors.The error control is on **link to link** basis.
- Retransmission : The packet is retransmitted if the source fails to receive acknowledgment.
- Flow Control : Necessary for a fast transmitter to keep pace with a slow receiver.

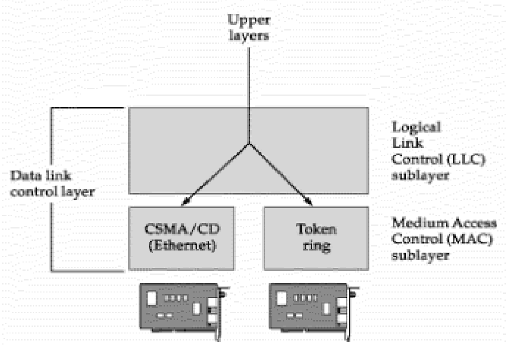

Upper layers
Data link control layer
Logical Link Control (LLC) sublayer
CSMA/CD (Ethernet)
Token ring
Medium Access Control (MAC) sublayer

Fig 2.7 Data Link Layer

Error Detecting Codes

Basic approach used for error detection is the use of redundancy, where additional bits are added to facilitate detection and correction of errors. Popular techniques are:

• Simple Parity check

• Two-dimensional Parity check

• Checksum

• Cyclic redundancy check

Simple Parity Checking or One-dimension Parity Check The most common and least expensive mechanism for error- detection is the simple parity check. In this technique, a redundant bit called parity bit, is appended to every data unit so that the number of 1s in the unit (including the parity becomes even). Blocks of data from the source are subjected to a check bit or Parity bit generator form, where a parity of 1 is added to the block if it contains an odd number of 1’s (ON bits) and 0 is added if it contains an even number of 1’s. At the receiving end the parity bit is computed from the received data bits and compared with the received parity bit, as shown in

Fig2.8This scheme makes the total number of 1’s even, that is why it is called even parity checking. Considering a 4-bit word, different combinations of the data words and the corresponding code words are given in Table

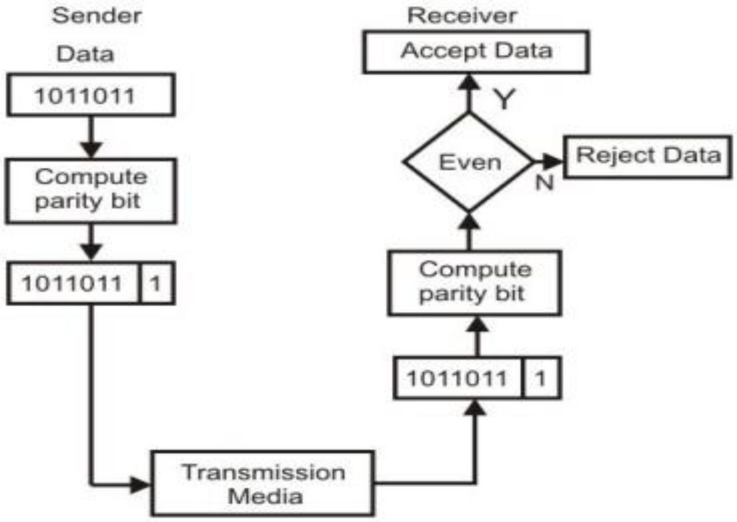


Fig 2.8Even-parity checking scheme

Table: Possible 4-bit data words and corresponding code words

Decimal value	Data Block	Parity bit	Code word
0	0000	0	00000
1	0001	1	00011
2	0010	1	00101
3	0011	0	00110
4	0100	1	01001
5	0101	0	01010
6	0110	0	01100
7	0111	1	01111
8	1000	1	10001
9	1001	0	10010
10	1010	0	10100
11	1011	1	10111
12	1100	0	11000
13	1101	1	11011
14	1110	1	11101
15	1111	0	11110

Note that for the sake of simplicity, we are discussing here the even-parity checking, where the number of 1’s should be an even number. It is also possible to use odd-parity checking, where the number of 1’s should be odd.

Performance

An observation of the table reveals that to move from one code word to another, at least two data bits should be changed. Hence these set of code words are said to have a minimum distance (hamming distance) of 2, which means that a receiver that has knowledge of the code word set can detect all single bit errors in each code word. However, if two errors occur in the code word, it becomes another valid member of the set and the decoder will see only another valid code word and know nothing of the error. Thus errors in more than one bit cannot be detected. In fact it can be shown that a single parity check code can detect only odd number of errors in a code word.

Error-Correcting Codes:

Network designers have developed two basic strategies for dealing with errors. One way is to include enough redundant information along with each block of data sent, to enable the receiver to deduce what the transmitted data must have been. The other way is to include only enough redundancy to allow the receiver to deduce that an error occurred, but not which error, and have it request a retransmission. The former strategy uses error-correcting codes and the latter uses error-detecting codes. The use of error-correcting codes is often referred to as forward error correction.

Each of these techniques occupies a different ecological niche. On channels that are highly reliable, such as fiber, it is cheaper to use an error detecting code and just retransmit the occasional block found to be faulty. However, on channels such as wireless links that make many errors, it is better to add enough redundancy to each block for the receiver to be able to figure out what the original block was, rather than relying on a retransmission, which itself may be in error.

To understand how errors can be handled, it is necessary to look closely at what an error really is. Normally, a frame consists of m data (i.e., message) bits and r redundant, or check, bits. Let the total length be n (i.e., n = m + r). An n-bit unit containing data and check bits is often referred to as an n-bit codeword.

Given any two code words, say, 10001001 and 10110001, it is possible to determine how many

Corresponding bits differ. In this case, 3 bits differ. To determine how many bits differ, just exclusive OR the two code words and count the number of 1 bits in the result, for example:

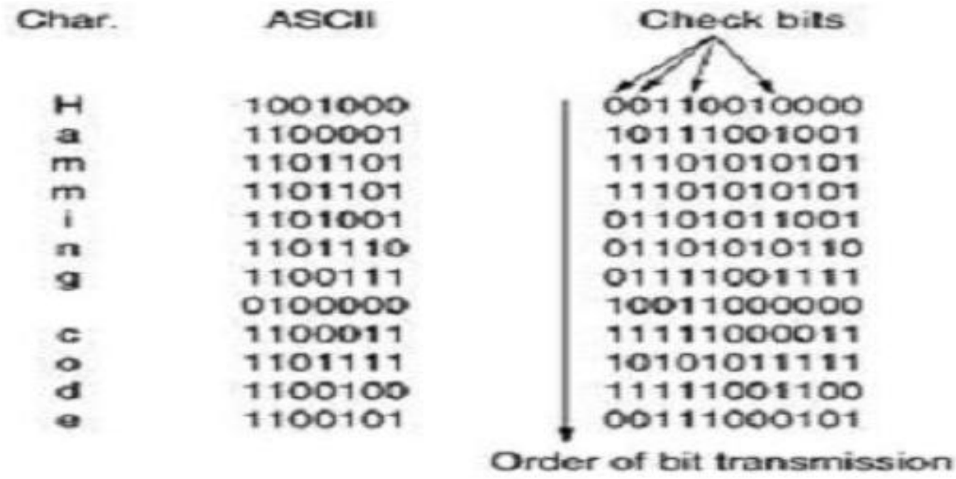
The number of bit positions in which two code words differ is called the Hamming distance. Its significance is that if two codeword's are a Hamming distance d apart, it will require d single-bit errors to convert one into the other.

In most data transmission applications, all 2m possible data messages are legal, but due to the way the check bits are computed, not all of the 2n possible code words are used. Given the algorithm for computing the check bits, it is possible to construct a complete list of the legal codeword's, and from this list find the two codeword's whose Hamming distance is minimum. This distance is the Hamming distance of the complete code. The error-detecting and error correcting properties of a code depend on its Hamming distance. To detect d errors, you need a distance d + 1 code because with such a code there is no way that d single-bit errors can change a valid codeword into another valid codeword. When the receiver sees an invalid codeword, it can tell that a transmission error has occurred. Similarly, to correct d errors, you need a distance 2d + 1 code because that way the legal codewords are so far apart that even with d changes, the original codeword is still closer than any other codeword, so it can be uniquely determined. As a simple example of an error-detecting code, consider a code in which a single parity b appended to the data. The parity bit is chosen so that the number of 1 bits in the codeword is even (or odd). For example, when 1011010 is sent in even parity, a bit is added to the end to make it 10110100. With odd parity 1011010 becomes 10110101. A code with a single parity bit has a distance 2, since any single-bit error produces a codeword with the wrong parity. It can be used to detect single errors.

As a simple example of an error-correcting code, consider a code with only four valid codewords: 0000000000, 0000011111, 1111100000, and 1111111111

This code has a distance 5, which means that it can correct double errors. If the codeword 0000000111 arrives, the receiver knows that the original must have been 0000011111. If, however, a triple error changes 0000000000 into 0000000111, the error will not be corrected properly.

Imagine that we want to design a code with m message bits and r check bits that will allow all single errors to be corrected. Each of the 2m legal messages has n illegal codewords at a distance 1 from it. These are formed by systematically inverting each of the n bits in the n-bit codeword formed from it. Thus, each of the 2m legal messages requires n + 1 bit patterns dedicated to it. Since the total number of bit patterns is 2n, we must have (n + 1)2m ≤ 2n. Using n = m + r, this requirement becomes (m + r + 1) ≤ 2r. Given m, this puts a lower limit on
the number of check bits needed to correct single errors. This theoretical lower limit can, in fact,
be achieved using a method due to Hamming (1950). The bits of the codeword are numbered consecutively, starting with bit 1 at the left end, bit 2 to its immediate right, and so on. The bits that are powers of 2 (1, 2, 4, 8, 16, etc.) are check bits. The rest (3, 5, 6, 7, 9, etc.) are filled up with the m data bits. Each check bit forces the parity of some collection of bits, including itself, to be even (or odd). A bit may be included in several parity computations. To see which check bits the data bit in position k contributes to, rewrite k as a sum of powers of 2.For example, 11 = 1 + 2 + 8 and 29 = 1 + 4 + 8 + 16. A bit is checked by just those check bits occurring in its expansion (e.g., bit 11 is checked by bits 1, 2, and 8). When a codeword arrives, the receiver initializes a counter to zero. It then examines each check bit, k (k = 1, 2, 4, 8 ...), to see if it has the correct parity. If not, the receiver adds k to the counter. If the counter is zero after all the check bits have been examined (i.e., if they were all correct), the codeword is accepted as valid. If the counter is nonzero, it contains the number of the incorrect bit. For example, if check bits 1, 2, and 8 are in error, the inverted bit is 11, because it is the only one checked by bits 1, 2, and 8. Figure 4.1 shows some 7-bit ASCII characters encoded as 11-bit code words using a Hamming code. Remember that the data are found in bit positions 3, 5, 6, 7, 9, 10, and 11.



Char.	ASCII	Check bits
H	1001000	00110010000
a	1100001	10111001001
m	1101101	11101010101
m	1101101	11101010101
i	1101001	01101011001
n	1101110	01101010110
g	1100111	01111001111
	0100000	10011000000
c	1100011	11111000011
o	1101111	10101011111
d	1100100	11111001100
e	1100101	00111000101

Order of bit transmission

Fig 2.9: Use of a Hamming code to correct burst errors

Hamming codes can only correct single errors. However, there is a trick that can be used to permit Hamming codes to correct burst errors. A sequence of k consecutive code words is arranged as a matrix, one codeword per row. Normally, the data would be transmitted one codeword at a time, from left to right. To correct burst errors, the data should be transmitted one column at a time, starting with the leftmost column. When all k bits have been sent, the second column is sent, and so on, as indicated in Fig2.9 When the frame arrives at the receiver, the matrix is reconstructed, one column at a time. If a burst error of length k occurs, at most 1 bit in each of the k codeword will have been affected, but the Hamming code can correct one error per codeword, so the entire block can be restored. This method uses kr check bits to make blocks of km data bits immune to a single burst error of length k or less.

Error-Detecting Codes

Error-correcting codes are widely used on wireless links, which are notoriously noisy and error prone when compared to copper wire or optical fibers. Without error-correcting codes, it would be hard to get anything through. However, over copper wire or fiber, the error rate is much lower, so error detection and retransmission is usually more efficient there for dealing with the occasional error. As a simple example, consider a channel on which errors are isolated and the error rate is 10-6 per bit. Let the block size be 1000 bits. To provide error correction for 1000-bit blocks, 10 check bits are needed; a megabit of data would require 10,000 check bits. To merely detect a block with a single 1-bit error, one parity bit per block will suffice. Once every 1000 blocks, an extra block (1001 bits) will have to be transmitted. The total overhead for the error detection + retransmission method is only 2001 bits per megabit of data, versus 10,000 bits for a Hamming code.

If a single parity bit is added to a block and the block is badly garbled by a long burst error, the probability that the error will be detected is only 0.5, which is hardly acceptable.

The odds can be improved considerably if each block to be sent is regarded as rectangular matrix n bits wide and k bits high, as described above. A parity bit is computed separately for each column and affixed to the matrix as the last row. The matrix is then transmitted one row at a time. When the block arrives, the receiver checks all the parity bits. If any one of them is wrong, the receiver requests a retransmission of the block. Additional retransmissions are requested as needed until an entire block is received without any parity errors. This method can detect a single burst of length n, since only 1 bit per column will be changed. A burst of length n + 1 will pass undetected, however, if the first bit is inverted, the last bit is inverted, and all the other bits are correct. (A burst error does not imply that all the bits are wrong; it just implies that at least the first and last are wrong.) If the block is badly garbled by a long burst or by multiple shorter bursts, the probability that any of the n columns will have the correct parity, by accident, is 0.5, so the probability of a bad block being accepted when it should not be is 2-n. Although the above scheme may sometimes be adequate, in practice, another method is in widespread use: the polynomial code, also known as a CRC (Cyclic Redundancy Check).

Polynomial codes are based upon treating bit strings as representations of polynomials with coefficients of 0 and 1 only. A k-bit frame is regarded as the coefficient list for a polynomial with k terms, ranging from xk-1 to x0. Such a polynomial is said to be of degree k - 1. The high order (leftmost) bit is the coefficient of xk-1; the next bit is the coefficient of xk-2, and so on. For example, 110001 has 6 bits and thus represent a six-term polynomial with coefficients 1, 1, 0, 0, 0, and 1: x5 + x4 + x0.

Polynomial arithmetic is done modulo 2, according to the rules of algebraic field theory. There are no carries for addition or borrows for subtraction. Both addition and subtraction are identical to exclusive OR. For example: Long division is carried out the same way as it is in binary except that the subtraction is done modulo 2, as above. A divisor is said "to go into" a dividend if the dividend has as many bits as the divisor. When the polynomial code method is employed, the sender and receiver must agree upon a generator polynomial, G(x), in advance. Both the high- and low-order bits of the generator must be 1. To compute the checksum for some frame with m bits, corresponding to the polynomial M(x), the frame must be longer than the generator polynomial. The idea is to append a checksum to the end of the frame in such a way that the polynomial represented by the check summed frame is divisible by G(x). When the receiver gets the check summed frame, it tries dividing it by G(x). If there is a remainder, there has been a transmission error.

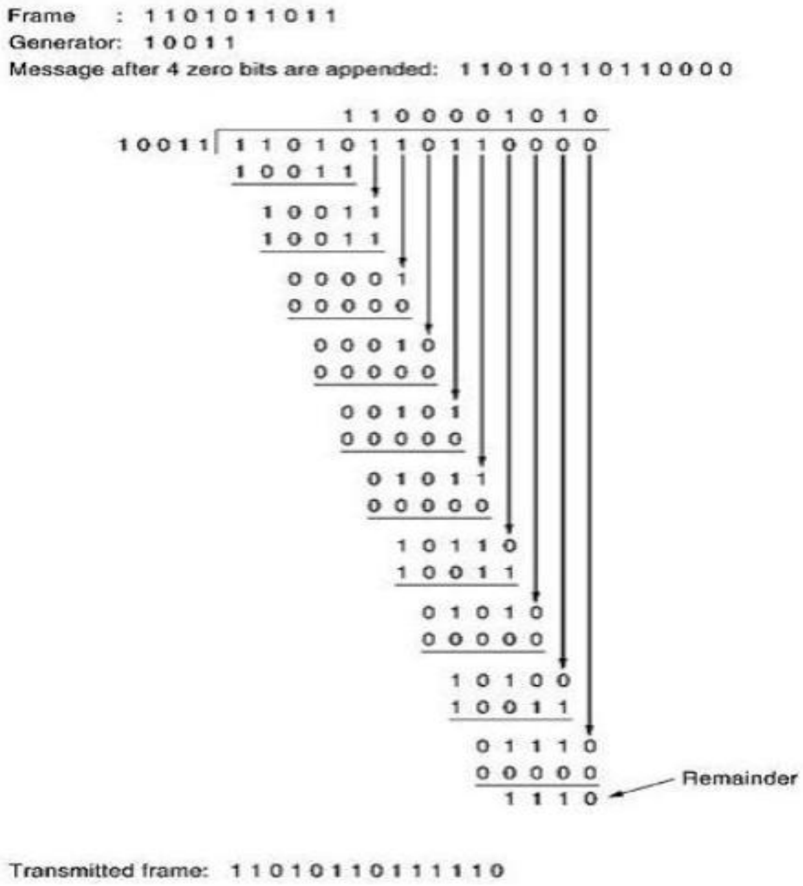
The algorithm for computing the checksum is as follows:

1. Let r be the degree of G(x). Append r zero bits to the low-order end of the frame so it now contains m + r bits and corresponds to the polynomial xr M(x).

2. Divide the bit string corresponding to G(x) into the bit string corresponding to xr M(x), using modulo 2 divisions.

3. Subtract the remainder (which is always r or fewer bits) from the bit string corresponding to xr M(x) using modulo 2 subtractions. The result is the check summed frame to be transmitted. Call its polynomial T(x).

Figure 2.10 illustrates the calculation for a frame 1101011011 using the generator G(x) = x4 + x+ 1.



Frame : 1 1 0 1 0 1 1 0 1 1
Generator: 1 0 0 1 1
Message after 4 zero bits are appended: 1 1 0 1 0 1 1 0 1 1 0 0 0 0

```
               1 1 0 0 0 0 1 0 1 0
1 0 0 1 1 | 1 1 0 1 0 1 1 0 1 1 0 0 0 0
            1 0 0 1 1
              1 0 0 1 1
              1 0 0 1 1
                0 0 0 0 1
                0 0 0 0 0
                  0 0 0 1 0
                  0 0 0 0 0
                    0 0 1 0 1
                    0 0 0 0 0
                      0 1 0 1 1
                      0 0 0 0 0
                        1 0 1 1 0
                        1 0 0 1 1
                          0 1 0 1 0
                          0 0 0 0 0
                            1 0 1 0 0
                            1 0 0 1 1
                              0 1 1 1 0
                              0 0 0 0 0
                                1 1 1 0  <-- Remainder
```

Transmitted frame: 1 1 0 1 0 1 1 0 1 1 1 1 1 0

Fig 2.10 Calculation of the polynomial code checksum

Elementary Data Link Layer Protocols

An Unrestricted Simplex Protocol:

As an initial example we will consider a protocol that is as simple as it can be. Data are transmitted in one direction only. Both the transmitting and receiving network layers are always ready. Processing time can be ignored. Infinite buffer space is available. And best of all, the communication channel between the data link layers never damages or loses frames. This thoroughly unrealistic protocol, which we will nickname "utopia" .

The protocol consists of two distinct procedures, a sender and a receiver. The sender runs in the data link layer of the source machine, and the receiver runs in the data link layer of the destination machine. No sequence numbers or acknowledgements are used here, so MAX_SEQ is not needed. The only event type possible is frame arrival (i.e., the arrival of an undamaged frame).

The sender is in an infinite while loop just pumping data out onto the line as fast as it can. The body of the loop consists of three actions: go fetch a packet from the (always obliging) network layer, construct an outbound frame using the variable s, and send the frame on its way. Only the info field of the frame is used by this protocol, because the other fields have to do with error and flow control and there are no errors or flow control restrictions here. The receiver is equally simple. Initially, it waits for something to happen, the only possibility being the arrival of an undamaged frame. Eventually, the frame arrives and the procedure wait_for_event returns, with event set to frame_arrival (which is ignored anyway). The call to from_physical_layer removes the newly arrived frame from the hardware buffer and puts it in the variable r, where the receiver code can get at it. Finally, the data portion is passed on to the network layer, and the data link layer settles back to wait for the next frame, effectively suspending itself until the frame arrives.

A Simplex Stop-and-Wait Protocol:

- **Stop and Wait**

  This flow control mechanism forces the sender after transmitting a data frame to stop and wait until the acknowledgement of the data-frame sent is received shown in Fig 2.11

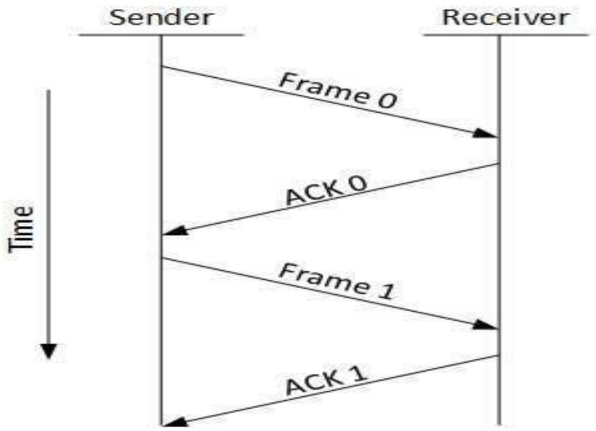


Fig 2.11 Stop and wait protocol

- **Sliding Window**

In this flow control mechanism, both sender and receiver agree on the number of data-frames after which the acknowledgement should be sent. As we learnt, stop and wait flow control mechanism wastes resources, this protocol tries to make use of underlying resources as much as possible.

Error Control

When data-frame is transmitted, there is a probability that data-frame may be lost in the transit or it is received corrupted. In both cases, the receiver does not receive the correct data-frame and sender does not know anything about any loss.In such case, both sender and receiver are equipped with some protocols which helps them to detect transit errors such as loss of data-frame. Hence, either the sender retransmits the data-frame or the receiver may request to resend the previous data-frame.

Requirements for error control mechanism:

- **Error detection** - The sender and receiver, either both or any, must ascertain that there is some error in the transit.
- **Positive ACK** - When the receiver receives a correct frame, it should acknowledge it.
- **Negative ACK** - When the receiver receives a damaged frame or a duplicate frame, it sends a NACK back to the sender and the sender must retransmit the correct frame.
- **Retransmission:** The sender maintains a clock and sets a timeout period. If an acknowledgement of a data-frame previously transmitted does not arrive before the timeout the sender retransmits the frame, thinking that the frame or it’s acknowledgement is lost in transit.

There are three types of techniques available which Data-link layer may deploy to control the errors by Automatic Repeat Requests (ARQ):

- Stop-and-wait ARQ

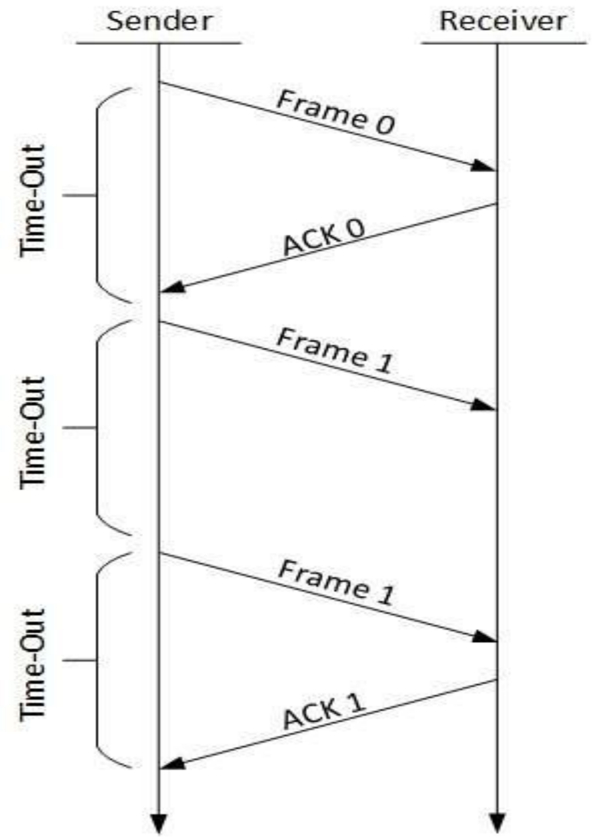


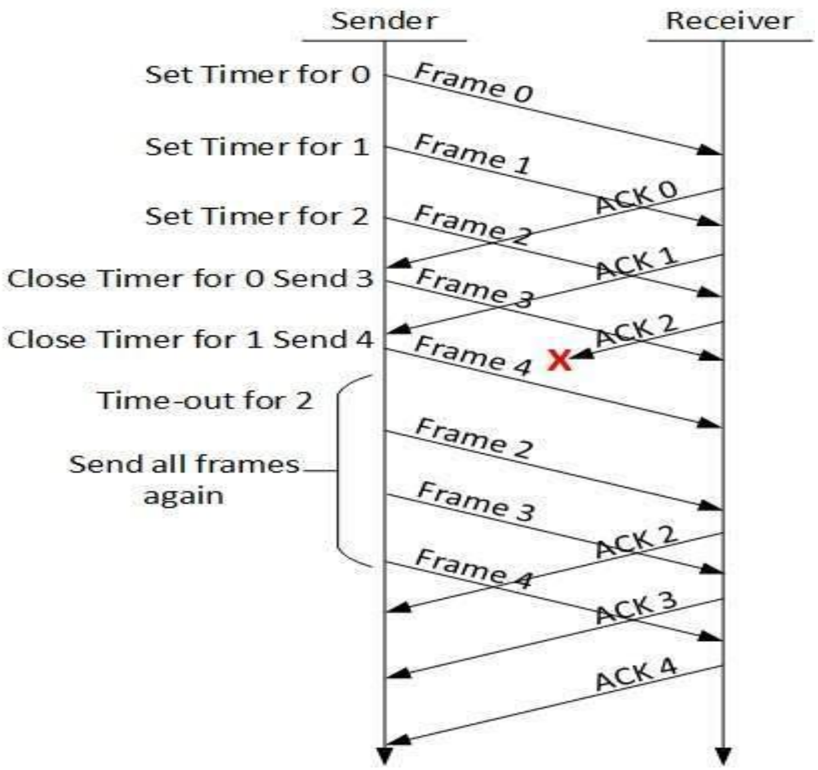
Fig 2.12: Stop-and-Wait ARQ

The following transition may occur in Stop-and-Wait ARQ as shown in fig 2.12

- The sender maintains a timeout counter.
- When a frame is sent, the sender starts the timeout counter.
- If acknowledgement of frame comes in time, the sender transmits the next frame in queue.
- If acknowledgement does not come in time, the sender assumes that either the frame or its acknowledgement is lost in transit. Sender retransmits the frame and starts the timeout counter.
- If a negative acknowledgement is received, the sender retransmits the frame.

- Go-Back-N ARQ
- Stop and wait ARQ mechanism does not utilize the resources at their best.When the acknowledgement is received, the sender sits idle and does nothing. In Go-Back-N ARQ method as shown in fig Go-Back-N ARQ

, both sender and receiver maintain a window.



- Fig 2.13 Go-Back-N ARQ

The sending-window size enables the sender to send multiple frames without receiving the acknowledgement of the previous ones. The receiving-window enables the receiver to receive multiple frames and acknowledge them. The receiver keeps track of incoming frame's sequence number.

When the sender sends all the frames in window, it checks up to what sequence number it has received positive acknowledgement. If all frames are positively acknowledged, the sender sends next set of frames. If sender finds that it has received NACK or has not receive any ACK for a particular frame, it retransmits all the frames after which it does not receive any positive ACK.

- Selective Repeat ARQ

In Go-back-N ARQ, it is assumed that the receiver does not have any buffer space for its window size and has to process each frame as it comes. This enforces the sender to retransmit all the frames which are not acknowledged.

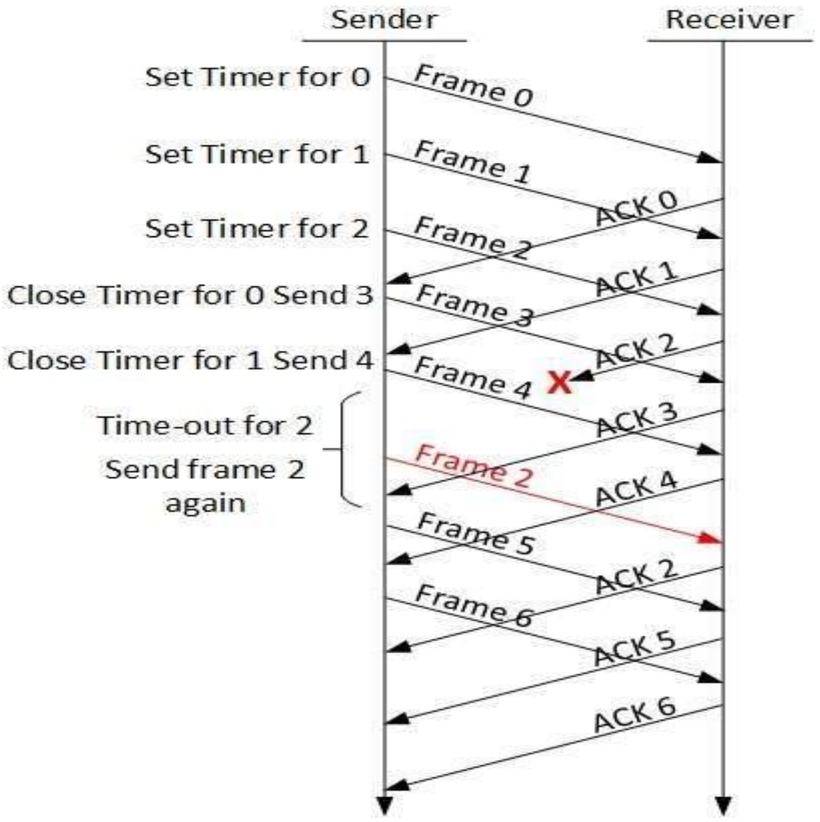


Fig 2.14 Selective-Repeat ARQ

In Selective-Repeat ARQ, the receiver while keeping track of sequence numbers, buffers the frames in memory and sends NACK for only frame which is missing or damaged.

The sender in this case, sends only packet for which NACK is received.

HDLC—High-Level Data Link Control:

These are a group of closely related protocols that are a bit old but are still heavily used. They are all derived from the data link protocol first used in the IBM mainframe world: SDLC (Synchronous Data Link Control) protocol. After developing SDLC, IBM submitted it to ANSI and ISO for acceptance as U.S. and international standards, respectively. ANSI modified it to become ADCCP (Advanced Data Communication Control Procedure), and ISO modified it to become HDLC (High-level Data Link Control). CCITT then adopted and modified HDLC for its LAP (Link Access Procedure) as part of the X.25 network interface standard but later modified it again to LAPB, to make it more compatible with a later version of HDLC. The nice thing about standards is that you have so many to choose from. Furthermore, if you do not like any of them, you can just wait for next year's model. These protocols are based on the same principles. All are bit oriented, and all use bit stuffing for data transparency. They differ only in minor, but nevertheless irritating, ways. The discussion of bit-oriented protocols that follows is intended as a general introduction. For the specific details of any one protocol, please consult the appropriate definition.

All the bit-oriented protocols use the frame structure shown in Fig.2.15. The Address field is primarily of importance on lines with multiple terminals, where it is used to identify one of the terminals. For point-to-point lines, it is sometimes used to distinguish commands from responses.

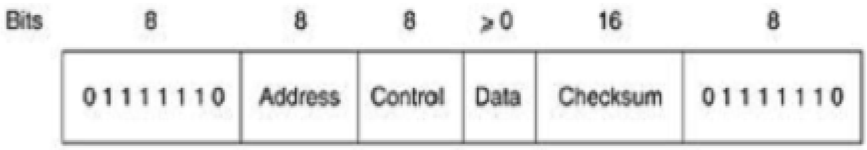


Fig 2.15 High-Level Data Link Control

Frame format for bit-oriented protocols

The Control field is used for sequence numbers, acknowledgements, and other purposes, as discussed below.

The Data field may contain any information. It may be arbitrarily long, although the efficiency of the checksum falls off with increasing frame length due to the greater probability of multiple burst errors. The Checksum field is a cyclic redundancy code. The frame is delimited with another flag sequence (01111110). On idle point-to-point lines, flag sequences are transmitted continuously.

The minimum frame contains three fields and totals 32 bits, excluding the flags on either end.

There are three kinds of frames: Information, Supervisory, and Unnumbered.

The contents of the Control field for these three kinds are shown in Fig.2.16 The protocol uses a sliding window, with a 3-bit sequence number. Up to seven unacknowledged frames may be outstanding at any instant. The Seq field in Fig 2.16 (a) is the frame sequence number. The Next field is a piggybacked acknowledgement. However, all the protocols adhere to the convention that instead of piggybacking the number of the last frame received correctly, they use the number of the first frame not yet received (i.e., the next frame expected). The choice of using the last frame received or the next frame expected is arbitrary; it does not matter which convention is used, provided that it is used consistently

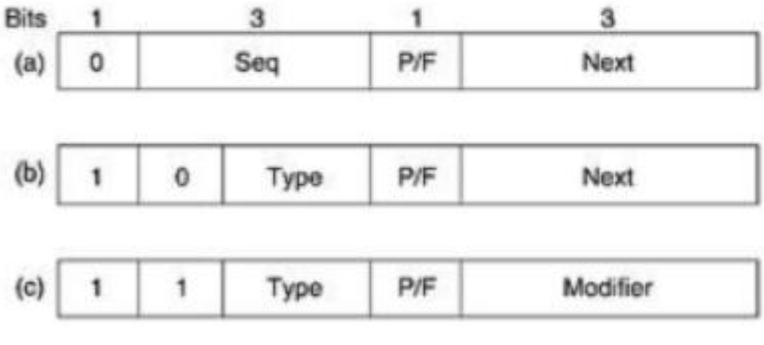


Fig 2.16 Control field of (a) an information frame, (b) a supervisory frame, (c) an unnumbered frame

The P/F bit stands for Poll/Final. It is used when a computer (or concentrator) is polling a group of terminals. When used as P, the computer is inviting the terminal to send data. All the frames sent by the terminal, except the final one, have the P/F bit set to P. The final one is set to F. In some of the protocols, the P/F bit is used to force the other machine to send a Supervisory frame immediately rather than waiting for reverse traffic onto which to piggyback the window information. The bit also has some minor uses in connection with the unnumbered frames.

The various kinds of Supervisory frames are distinguished by the Type field. Type 0 is an acknowledgement frame (officially called RECEIVE READY) used to indicate the next frame expected. This frame is used when there is no reverse traffic to use for piggybacking. **Type 1** is a negative acknowledgement frame (officially called REJECT). It is used to indicate that a transmission error has been detected. The Next field indicates the first frame in sequence not received correctly (i.e., the frame to be retransmitted). The sender is required to retransmit all

Outstanding frames starting at Next. This strategy is similar to our protocol 5 rather than our protocol 6.

Type 2 is RECEIVE NOT READY. It acknowledges all frames up to but not including next, just as RECEIVE READY does, but it tells the sender to stop sending. RECEIVE NOT READY is intended to signal certain temporary problems with the receiver, such as a shortage of buffers, and not as an alternative to the sliding window flow control. When the condition has been repaired, the receiver sends a RECEIVE READY, REJECT, or certain control frames.

Type 3 is the SELECTIVE REJECT. It calls for retransmission of only the frame specified. In this sense it is like our protocol 6 rather than 5 and is therefore most useful when the sender's window size is half the sequence space size, or less. Thus, if a receiver wishes to buffer out- of-sequence frames for potential future use, it can force the retransmission of any specific frame using Selective Reject. HDLC and ADCCP allow this frame type, but SDLC and LAPB do not allow it (i.e., there is no Selective Reject), and type 3 frames are undefined.

The third class of frame is the unnumbered frame. It is sometimes used for control purposes but can also carry data when unreliable connectionless service is called for. The various bit-oriented protocols differ considerably here, in contrast with the other two kinds, where they are nearly identical. Five bits are available to indicate the frame type, but not all 32 possibilities are used.

PPP-The Point-to-Point Protocol:

The Internet needs a point-to-point protocol for a variety of purposes, including router-to-router traffic and home user-to-ISP traffic. This protocol is PPP (Point-to-Point Protocol), which is defined in RFC 1661 and further elaborated on in several other RFCs (e.g., RFCs 1662 and 1663). PPP handles error detection, supports multiple protocols, allows IP addresses to be negotiated at connection time, permits authentication, and has many other features.

PPP provides three features:

1. A framing method that unambiguously delineates the end of one frame and the start of the next one. The frame format also handles error detection.

2. A link control protocol for bringing lines up, testing them, negotiating options, and bringing them down again gracefully when they are no longer needed. This protocol is called LCP (Link Control Protocol). It supports synchronous and asynchronous circuits and byte-oriented and bit-oriented encodings.

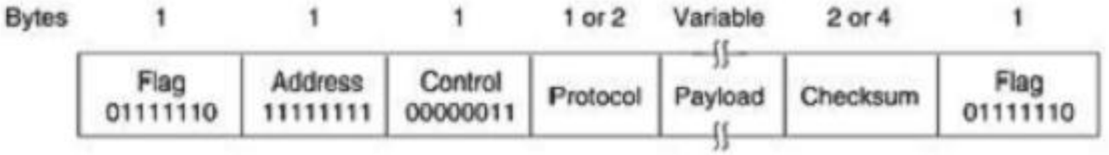
3. A way to negotiate network-layer options in a way that is independent of the network layer protocol to be used. The method chosen is to have a different NCP (Network Control Protocol) for each network layer supported.

To see how these pieces fit together, let us consider the typical scenario of a home user calling up an Internet service provider to make a home PC a temporary Internet host. The PC first calls the provider's router via a modem. After the router's modem has answered the phone and established a physical connection, the PC sends the router a series of LCP packets in the payload field of one or more PPP frames. These packets and their responses select the PPP parameters to be used.

Once the parameters have been agreed upon, a series of NCP packets are sent to configure the network layer. Typically, the PC wants to run a TCP/IP protocol stack, so it needs an IP address.

There are not enough IP addresses to go around, so normally each Internet provider gets a block of them and then dynamically assigns one to each newly attached PC for the duration of its login session. If a provider owns n IP addresses, it can have up to n machines logged in simultaneously, but its total customer base may be many times that. The NCP for IP assigns the IP address. At this point, the PC is now an Internet host and can send and receive IP packets, just as hardwired hosts can. When the user is finished, NCP tears down the network layer connection and frees up the IP address. Then LCP shuts down the data link layer connection. Finally, the computer tells the modem to hang up the phone, releasing the physical layer connection.

1. The PPP frame format was chosen to closely resemble the HDLC frame format, since there was no reason to reinvent the wheel. The major difference between PPP and HDLC is that PPP is character oriented rather than bit oriented. In particular, PPP uses byte stuffing on dial-up modem lines, so all frames are an integral number of bytes. It is not possible to send a frame consisting of 30.25 bytes, as it is with HDLC. Not only can PPP frames be sent over dialup telephone lines, but they can also be sent over SONET or true bit-oriented HDLC lines (e.g., for router-router connections). The PPP frame format is shown in Fig.2.17.



Bytes	1	1	1	1 or 2	Variable	2 or 4	1
	Flag 01111110	Address 11111111	Control 00000011	Protocol	Payload	Checksum	Flag 01111110

Fig 2.17 The PPP full frame format for unnumbered mode operation

All PPP frames begin with the standard HDLC flag byte (01111110), which is byte stuffed if it occurs within the payload field. Next comes the Address field, which is always set to the binary value 11111111 to indicate that all stations are to accept the frame. Using this value avoids the issue of having to assign data link addresses.

The Address field is followed by the Control field, the default value of which is 00000011. This value indicates an unnumbered frame. In other words, PPP does not provide reliable transmission using sequence numbers and acknowledgements as the default. In noisy environments, such as wireless networks, reliable transmission using numbered mode can be used. The exact details are defined in RFC 1663, but in practice it is rarely used. Since the Address and Control fields are always constant in the default configuration, LCP provides the necessary mechanism for the two parties to negotiate an option to just omit them altogether and save 2 bytes per frame.

The fourth PPP field is the Protocol field. Its job is to tell what kind of packet is in the Payload field. Codes are defined for LCP, NCP, IP, IPX, AppleTalk, and other protocols. Protocols starting with a 0 bit are network layer protocols such as IP, IPX, OSI CLNP, XNS. Those starting with a 1 bit are used to negotiate other protocols. These include LCP and a different NCP for each network layer protocol supported. The default size of the Protocol field is 2 bytes, but it can be negotiated down to 1 byte using LCP. The Payload field is variable length, up to some negotiated maximum. If the length is not negotiated using LCP during line setup, a default length of 1500 bytes is used. Padding may follow the payload if need be. After the Payload field comes the Checksum field, which is normally 2 bytes, but a 4-byte checksum can be negotiated.

1. In summary, PPP is a multiprotocol framing mechanism suitable for use over modems, HDLC bit-serial lines, SONET, and other physical layers. It supports error detection, option negotiation, header compression, and, optionally, reliable transmission using an HDLC type frame format.

Framing: To provide service to the network layer, the data link layer must use the service provided to it by the physical layer. What the physical layer does is accept a raw bit stream and attempt to deliver it to the destination. This bit stream is not guaranteed to be error free. The number of bits received may be less than, equal to, or more than the number of bits transmitted, and they may have different values. It is up to the data link layer to detect and, if necessary, correct errors. The usual approach is for the data link layer to break the bit stream up into discrete frames and compute the checksum for each frame. When a frame arrives at the destination, the checksum is recomputed. If the newly computed checksum is different from the one contained in the frame, the data link layer knows that an error has occurred and takes steps to deal with it (e.g., discarding the bad frame and possibly also sending back an error report). Breaking the bit stream up into frames is more difficult than it at first appears. One way to achieve this framing is to insert time gaps between frames, much like the spaces between words in ordinary text. However, networks rarely make any guarantees about timing, so it is possible these gaps might be squeezed out or other gaps might be inserted during transmission. Since it is too risky to count on timing to mark the start and end of each frame, other methods have been devised. We will look at four methods: 1. Character count. 2. Flag bytes with byte stuffing. 3. Starting and ending flags, with bit stuffing. 4. Physical layer coding violations. The first framing method uses a field in the header to specify the number of characters in the frame. When the data link layer at the destination sees the character count, it knows how many characters follow and hence where the end of the frame is. This technique is shown in Fig.2.18(a) for four frames of sizes 5, 5, 8, and 8 characters, respectively.

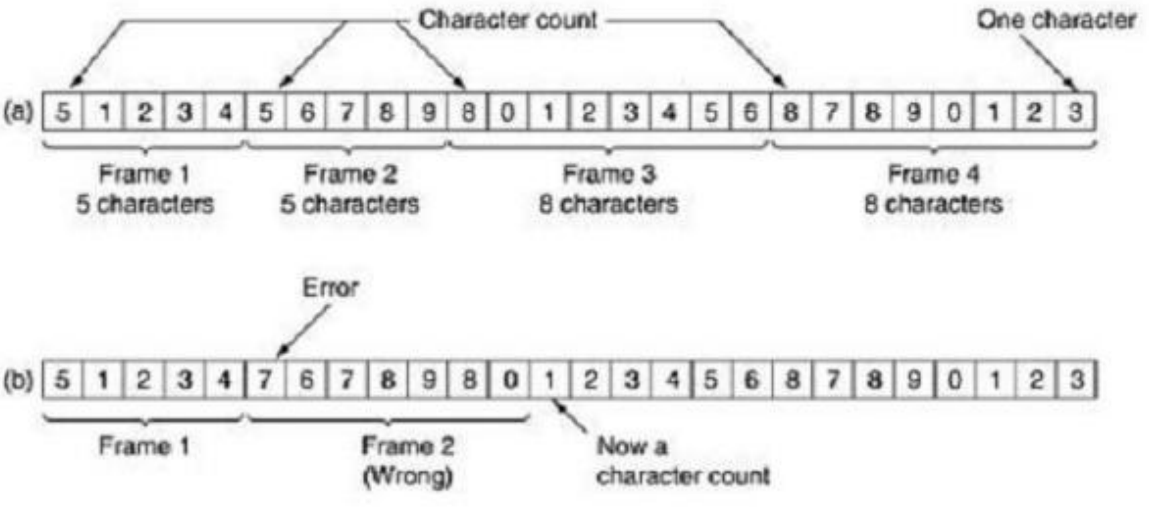


Fig 2.18 Framing

Part – A

1. List the layers of OSI model?
2. List the different network topologies.
3. Explain the function of data link layer.
4. Which layer of OSI provides synchronization?
5. Which layer of OSI provides routing functions?
6. Give any two physical layer interface standards?
7. What is parity check?
8. What is CRC?
9. What are the different ARQ Schemes?
10. What is ARQ?
11. What is the advantage of selective repeat ARQ over GO-back-N ARQ?
12. What is piggybacking?
13. What is bit stuffing?
14. Give the frame format for HDLC?
15. What is flow control mechanism?
16. What are the protocols for flow control in the data link layer?

Part – B

1. Explain Stop-N-Wait flow control and sliding window flow control protocols.
2. Given message M=1110101100 and pattern p=010100. Compute FCS, Explain how error detection in carried out in CRC.
3. Explain Go-back-N ARQ and selective Repeat ARQ with timing diagrams.
4. Explain High level Data Link (HDLC) protocol.