UNIT II

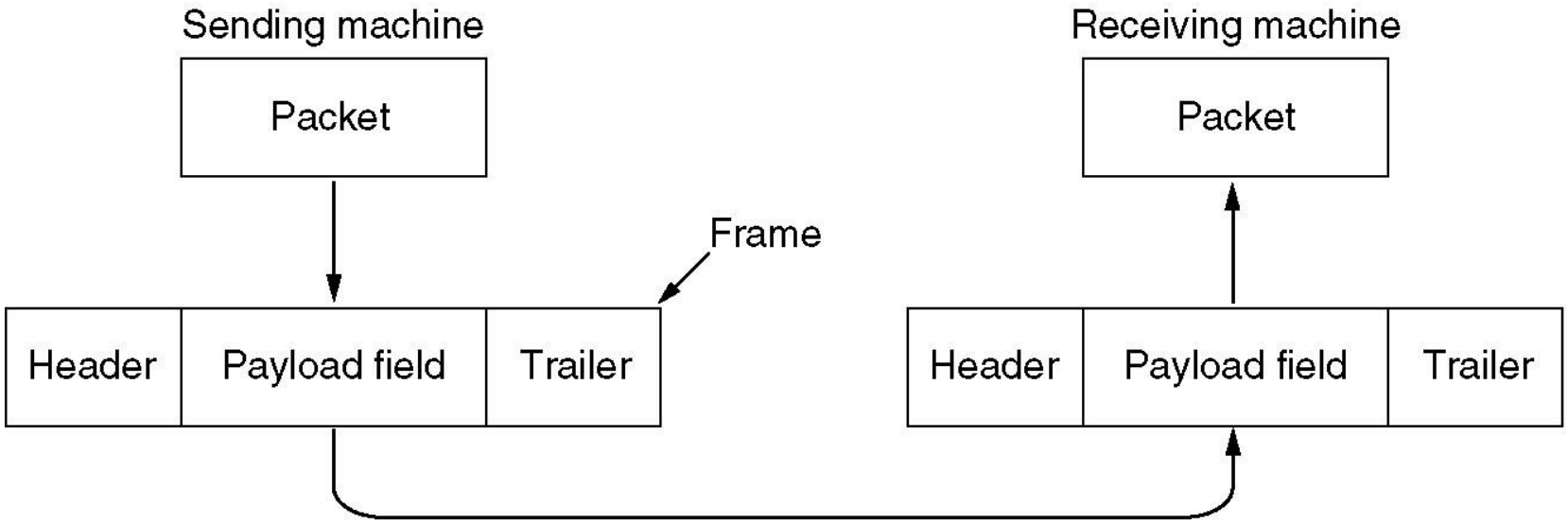
The Data Link Layer (Logical Link Control)

Prof.M. G. Devikar
Modern Education Society's
College of Engineering, Pune

Responsibilities of Data Link Layer

- Hop to Hop Delivery or Node to Node delivery
- Framing
- Error Control
- Flow Control
- Access Control (Channel)

Frame

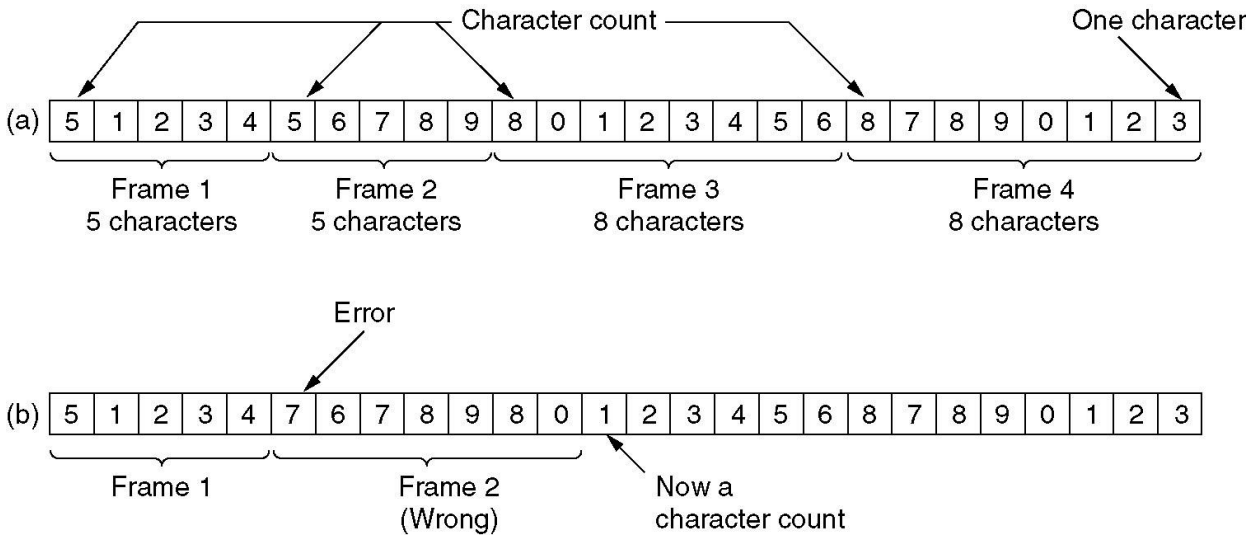


Relationship between packets and frames.

Framing Techniques

1. Character Count

In this method the field in the header is used to specify the number of characters in the frame.



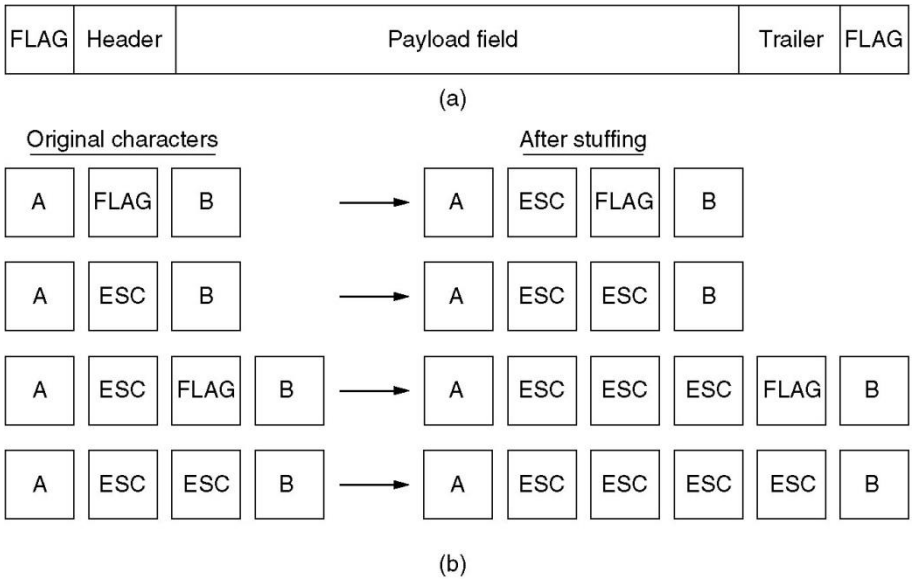
A character stream. (a) Without errors. (b) With one error.

Character Count

- The trouble with this algorithm is that the count can be garbled by a transmission error.
- For example, if the byte count of 5 in the second frame of Fig.(b) becomes a 7 due to a single bit flip, the destination will get out of synchronization. It will then be unable to locate the correct start of the next frame.
- Even if the checksum is incorrect so the destination knows that the frame is bad, it still has no way of telling where the next frame starts.
- Sending a frame back to the source asking for a retransmission does not help either, since the destination does not know how many bytes to skip over to get to the start of the retransmission. For this reason, the byte count method is rarely used by itself.

2. Flag byte with byte Stuffing

1.FLAG byte is added before and after every frame.
2. If FLAG belongs to actual data then ESC flag is used before Flag data.
3.If ESC belongs to actual data then ESC flag is used before ESC data.



(a) A frame delimited by flag bytes.
(b) Four examples of byte sequences before and after stuffing.

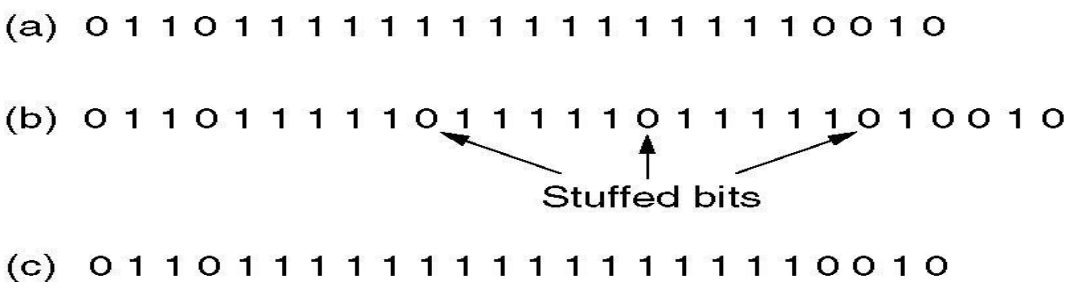
Flag byte with byte Stuffing

- Flag byte with byte Stuffing framing method gets around the problem of resynchronization after an error by having each frame start and end with special bytes. Often the same byte, called a **flag byte**, is used as both the starting and ending delimiter.
- Two consecutive flag bytes indicate the end of one frame and the start of the next. Thus, if the receiver ever loses synchronization it can just search for two flag bytes to find the end of the current frame and the start of the next frame.

3. Starting and Ending Flags with bit Stuffing

1. Each frame begins and ends with a special bit pattern 01111110,

2. Whenever sender DLL encounters 5 consecutive 1's in data, it automatically stuffs '0' bit into outgoing bit pattern,



(a) The original data.

(b) The data as they appear on the line.

(c) The data as they are stored in receiver's memory after destuffing.

Starting and Ending Flags with bit Stuffing

- With both bit and byte stuffing, a side effect is that the length of a frame now depends on the contents of the data it carries.
- For instance, if there are no flag bytes in the data, 100 bytes might be carried in a frame of roughly 100 bytes.
- If, however, the data consists solely of flag bytes, each flag byte will be escaped and the frame will become roughly 200 bytes long.
- With bit stuffing, the increase would be roughly 12.5% as 1 bit is added to every byte.

Error Coding

1.Environmental interference and physical defects in the communication medium can cause **random bit errors** during data transmission.

2.Error coding is a method of detecting and correcting these errors to ensure information is transferred intact from its source to its destination.

3. Error coding uses mathematical formulas to encode data bits at the source into longer bit words for transmission.

4.The "code word" can then be decoded at the destination to retrieve the information.

Error Detection and Correction

- **Error detecting codes** are the codes which only detect the error introduced in the data.
- This is achieved by adding just enough redundancy to allow the receiver to deduce that as error has occurred but not which error.
- Error detecting codes are also referred to as forward error correction.
- **Error correcting codes** are the codes which correct the error introduced in the data.
- This is done by adding lot of redundant data to enable the receiver to deduce what the transmitted data must have been.

Error Detection Codes

Three main error detecting codes are:

1. Parity Checking
2. Checksum error detection
3. Cyclic redundancy Code

Parity Checking

1. It is the easiest technique to detect single bit errors.
2. **Even parity** means adding a ‘0’ if number of ‘1’s’ is even and adding a ‘1’ if number of ‘1’s is odd.

Sequence 0 1 1 1 0 1 1

Codeword 0 1 1 1 0 1 1 1 Even parity bit 1 is added

Sequence 0 0 1 1 0 0 0

Codeword 0 0 1 1 0 0 0 Even Parity bit 0 is added

3.Here, if the receiver detects error it will discard byte and request for retransmission

Parity Checking

4. **Odd parity** means adding a 1 if number of '1's' is even and adding a '0' if number of 1's is odd.

Sequence 0 1 1 1 0 1 1

Codeword 0 1 1 1 0 1 1 0 Odd parity bit 0 is added

Sequence 0 0 1 1 0 0 0

Codeword 0 0 1 1 0 0 0 1 Odd Parity bit 1 is added

5. In case data is transmitted incorrectly, the parity bit value becomes incorrect; thus, indicating error has occurred during transmission.

6. Parity checking can check errors but it cannot tell at which location error has occurred.

Checksum

1. Useful when there are burst errors.
2. In checksum error detection scheme, the data is divided into ‘k’ segments each of ‘m’ bits.
3. In the sender’s end the segments are added using 1’s complement arithmetic to get the sum. The sum is complemented to get the checksum.
4. The checksum segment is sent along with the data segments.
5. At the receiver’s end, all received segments are added using 1’s complement arithmetic to get the sum. The sum is complemented.
6. If the result is zero, the received data is accepted; otherwise discarded.

Original Data

10011001	11100010	00100100	10000100
1	2	3	4

k=4, m=8

Sender

```
1   10011001
2   11100010
    --------
   (1)01111011
          1
    --------
    01111100
3   00100100
    --------
    10100000
4   10000100
    --------
   (1)00100100
          1
    --------
Sum: 00100101
CheckSum: 11011010
```

Reciever

```
1   10011001
2   11100010
    --------
   (1)01111011
          1
    --------
    01111100
3   00100100
    --------
    10100000
4   10000100
    --------
   (1)00100100
          1
    --------
    00100101
    11011010
    --------
Sum: 11111111
Complement: 00000000
```

Conclusion: Accept Data

Cyclic Redundancy Check

1. Cyclic Redundancy Check is the most powerful and easy to implement technique.
2. Unlike checksum scheme, which is based on addition, CRC is based on binary division.
3. In CRC, a sequence of redundant bits, called cyclic redundancy check bits, are appended to the end of data unit so that the resulting data unit becomes exactly divisible by a second, predetermined binary number.
4. At the destination, the incoming data unit is divided by the same number. If at this step there is no remainder, the data unit is assumed to be correct and is therefore accepted.
5. A remainder indicates that the data unit has been damaged in transit and therefore must be rejected.

Cyclic Redundancy Check

Example :

1.Consider the original message is k= 1101 and the Generator polynomial is x3+x+1,

2. CRC generator will generate a sequence which is 1011.

3. If the CRC generator generates n bits then append (n-1) zeros at the end of original message

4. So the new sequence is 1101000 (i.e k appended by 3 zeros)

5. Now perform binary division or Ex-OR (operation), hence we have to divide 1101000 by 1011, which produces remainder as 001, so the bit frame (k+r) is actually being transmitted.

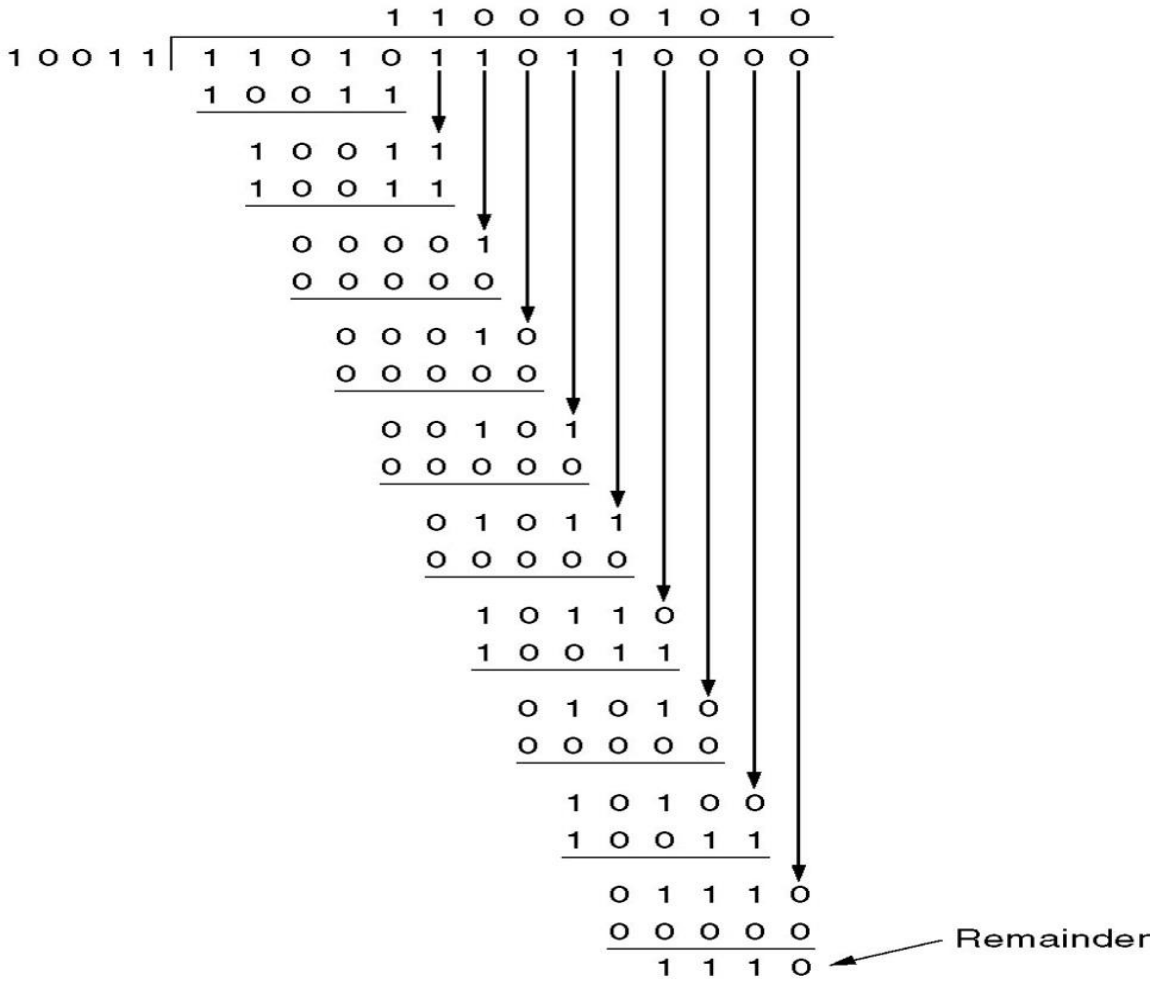
6. At the receiving end, if the received number, i.e., 1101001 is divided by the same generator polynomial 1011 to get the remainder as 000, it can be assumed that the data is free of errors.

Cyclic Redundancy Check

Frame : 1 1 0 1 0 1 1 0 1 1

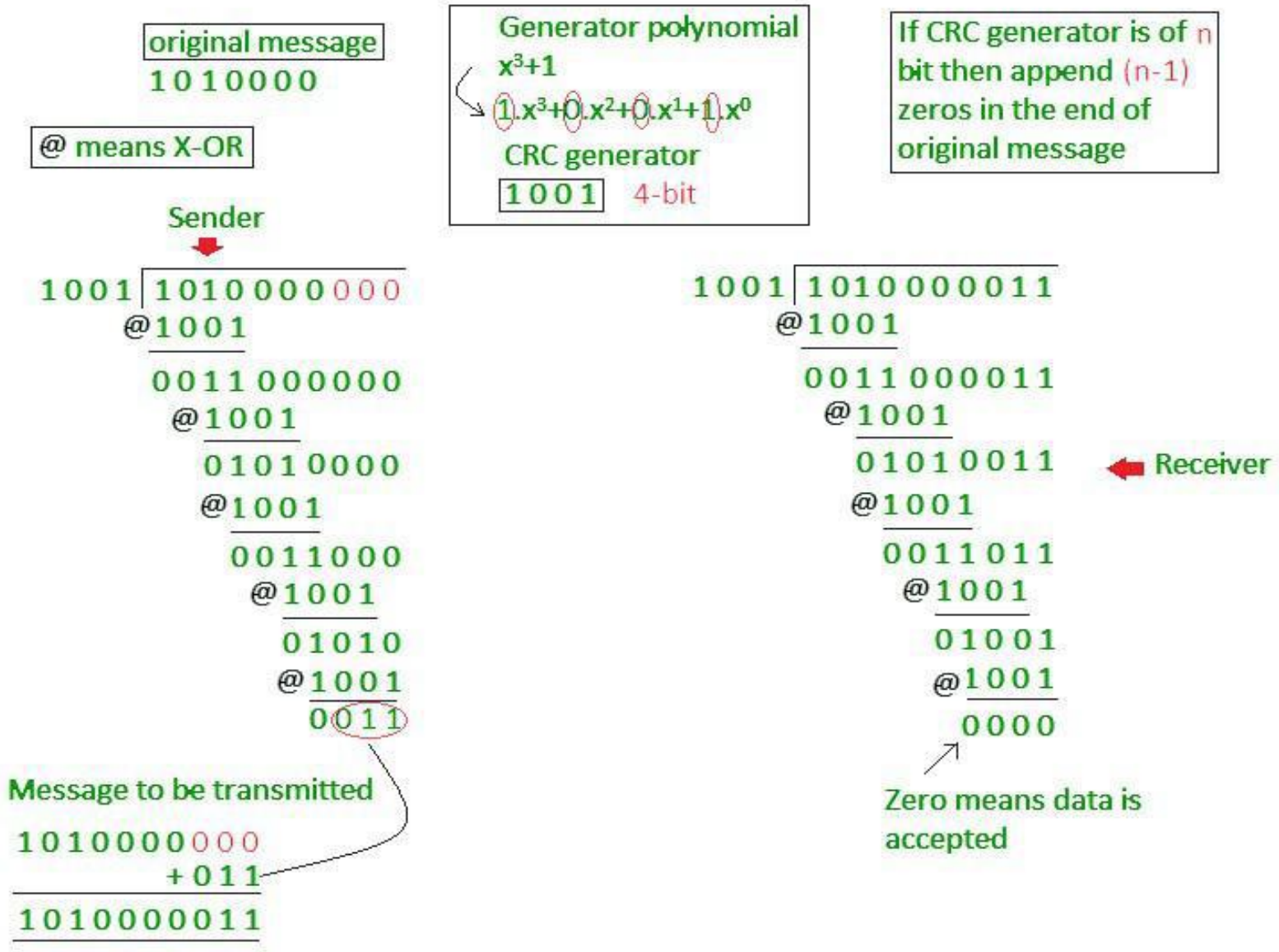
Generator: 1 0 0 1 1

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



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

Cyclic Redundancy Check



Error Correcting Codes

The techniques that we have discussed so far can detect errors, but do not correct them.

Error Correction can be handled in two ways.

- One is when an error is discovered; the receiver can have th sender retransmit the entire data unit. This is known as **backward error correction**.
- In the other, receiver can use an error-correcting code, which automatically corrects certain errors. This is known as **forward error correction**.

Hamming Code

1. Hamming code is a set of error-correction codes that can be used to **detect and correct the errors** that can occur when the data is moved or stored from the sender to the receiver. It is **technique developed by R.W. Hamming for error correction**.
2. Redundant bits are extra binary bits that are generated and added to the information-carrying bits of data transfer to ensure that no bits were lost during the data transfer.

Hamming Code

3. The number of redundant bits can be calculated using the following formula:

$2^r \geq m + r + 1$ where, r = redundant bit, m = data bit

Suppose the number of data bits is 7, then the number of redundant bits can be calculated using:

$2^4 \geq 7 + 4 + 1$

Thus, the number of redundant bits= 4

Algorithm for Hamming Code

The Hamming Code is simply the use of extra parity bits to allow the identification of an error.

1.Write the bit positions starting from 1 in binary form (1, 10, 11, 100, etc).

2.All the bit positions that are a power of 2 are marked as parity bits (1, 2, 4, 8, etc) and all the other bit positions are marked as data bits.

3.Each data bit is included in a unique set of parity bits, as determined its bit position in binary form.
a. Parity bit 1 covers all the bits positions whose binary representation includes a 1 in the least significant position (1, 3, 5, 7, 9, 11, etc).
b. Parity bit 2 covers all the bits positions whose binary representation includes a 1 in the second position from the least significant bit (2, 3, 6, 7, 10, 11, etc).

Algorithm for Hamming Code

c. Parity bit 4 covers all the bits positions whose binary representation includes a 1 in the third position from the least significant bit (4–7, 12–15, 20–23, etc).
d. Parity bit 8 covers all the bits positions whose binary representation includes a 1 in the fourth position from the least significant bit bits (8–15, 24–31, 40–47, etc).

Note: 1. Since we check for even parity set a parity bit to 1 if the total number of ones in the positions it checks is odd.

2. Set a parity bit to 0 if the total number of ones in the positions it checks is even.

Example: Hamming Code

Determining the position of redundant bits –

These redundancy bits are placed at the positions which correspond to the power of 2.

Consider The number of data bits = 7

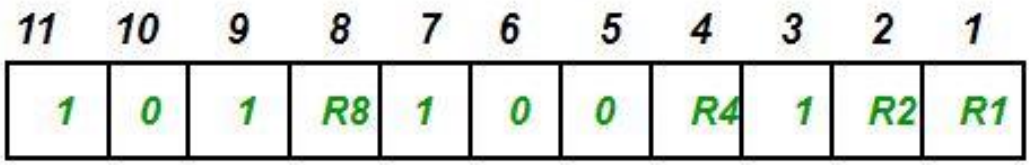
The number of redundant bits = 4

The total number of bits = 11

The redundant bits are placed at positions corresponding to power of 2 i.e 1, 2, 4, and 8

Example: Hamming Code

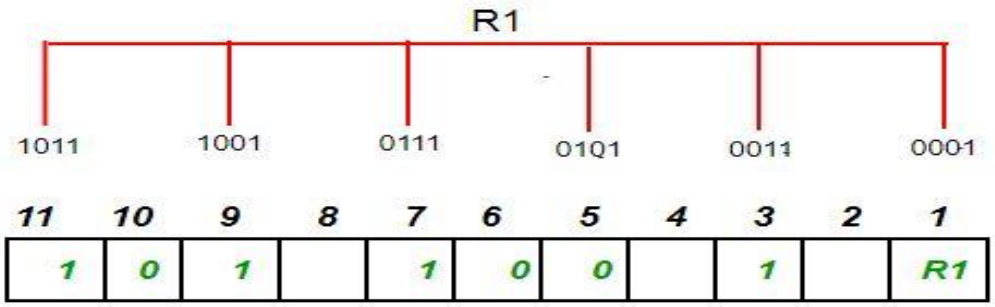
Suppose the data to be transmitted is 1011001, the bits will be placed as follows:



Example: Hamming Code

Determining the Parity bits –

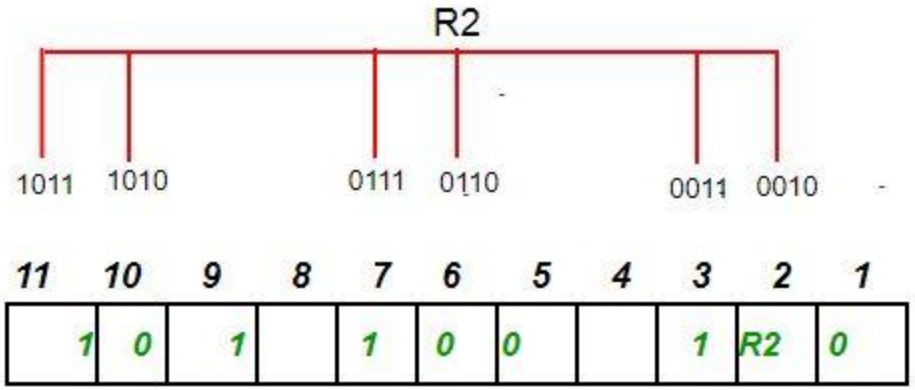
1. R1 bit is calculated using parity check at all the bits positions whose binary representation includes a 1 in the least significant position. R1: bits 1, 3, 5, 7, 9, 11



To find the redundant bit R1, we check for even parity. Since the total number of 1's in all the bit positions corresponding to R1 is an even number the value of R1 (parity bit's value) = 0

Example: Hamming Code

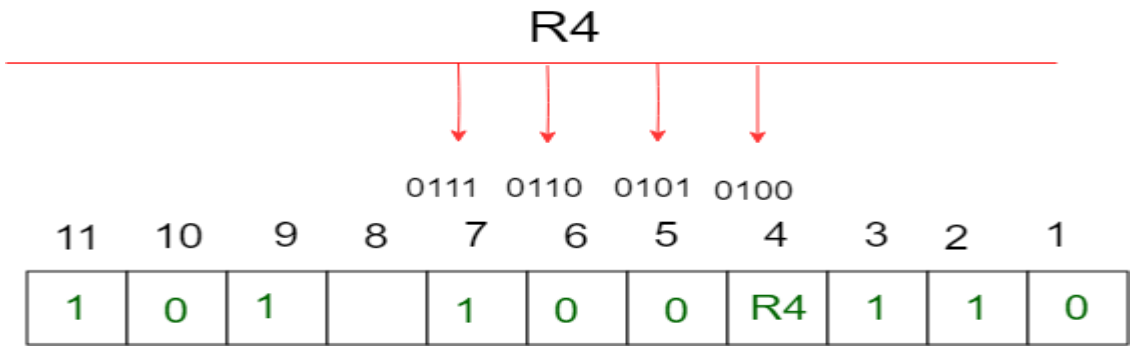
2. R2 bit is calculated using parity check at all the bits positions whose binary representation includes a 1 in the second position from the least significant bit.R2: bits 2,3,6,7,10,11



To find the redundant bit R2, we check for even parity. Since the total number of 1's in all the bit positions corresponding to R2 is an odd number the value of R2(parity bit's value)=1

Example: Hamming Code

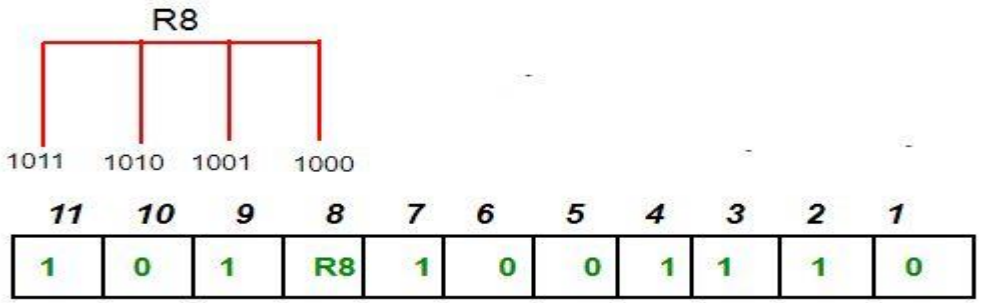
3. R4 bit is calculated using parity check at all the bits positions whose binary representation includes a 1 in the third position from the least significant bit.R4: bits 4, 5, 6, 7



To find the redundant bit R4, we check for even parity. Since the total number of 1's in all the bit positions corresponding to R4 is an odd number the value of R4(parity bit's value) = 1

Example: Hamming Code

4. R8 bit is calculated using parity check at all the bits positions whose binary representation includes a 1 in the fourth position from the least significant bit.R8: bit 8,9,10,11



To find the redundant bit R8, we check for even parity. Since the total number of 1's in all the bit positions corresponding to R8 is an even number the value of R8(parity bit's value)=0.

Example: Hamming Code

Hence, R1=0, R2=1,R4= 1, R8=0

Thus, the data transferred is:

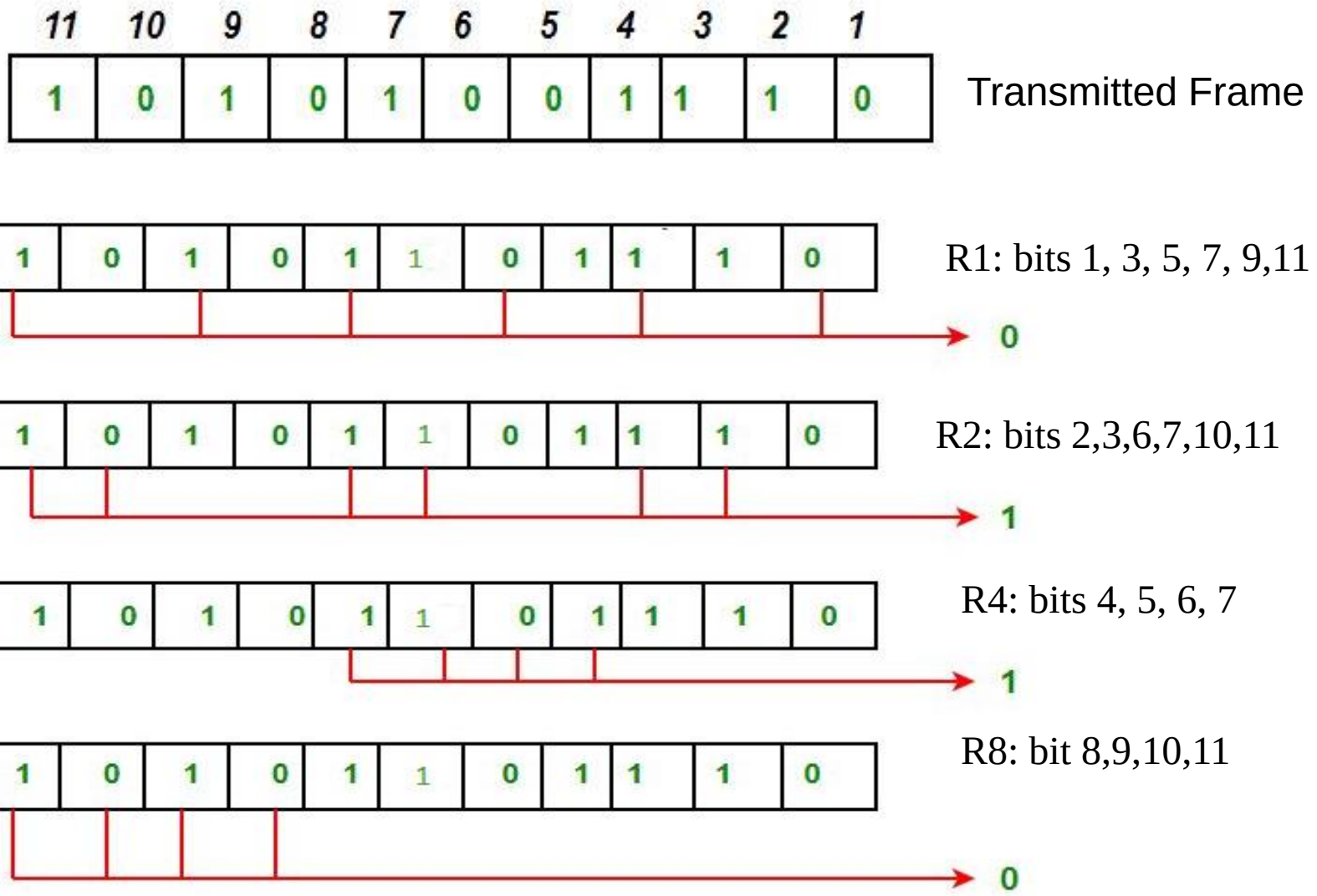
11	10	9	8	7	6	5	4	3	2	1
1	0	1	0	1	0	0	1	1	1	0

Example: Hamming Code

How to make Error detection and correction – ?????

Suppose in the above example the 6th bit is changed from 0 to 1 during data transmission, then it gives new parity values in the binary number:

Example: Hamming Code



Example: Hamming Code

- The bits give the binary number as 0110 whose decimal representation is 6. Thus, the bit 6 contains an error. To correct the error the 6th bit is changed from 1 to 0.

Flow Control

- Data-link layer is responsible for implementation of point-to-point flow and error control mechanism.
- When a data frame is sent from one host to another over a single medium, it is required that the sender and receiver should work at the same speed. That is, sender sends at a speed on which the receiver can process and accept the data. What if the speed (hardware/software) of the sender or receiver differs? If sender is sending too fast the receiver may be overloaded, (swamped) and data may be lost.

Flow Control

- The usual way to ensure reliable delivery is to provide the sender with some feedback about what is happening at the other end of the line. Typically, the protocol calls for the receiver to send back special control frames bearing positive or negative acknowledgements about the incoming frames.
- If the sender receives a positive acknowledgement about a frame, it knows the frame has arrived safely.
- On the other hand, a negative acknowledgement means that something has gone wrong and the frame must be transmitted again.

Data Link Protocols

- A Simplex Stop-and-Wait Protocol
- Stop and Wait ARQ (Automatic Repeat Request)
- Sliding Window Protocol

1. Go Back N
2. Selective Repeat

Simple Stop and Wait

Sender:

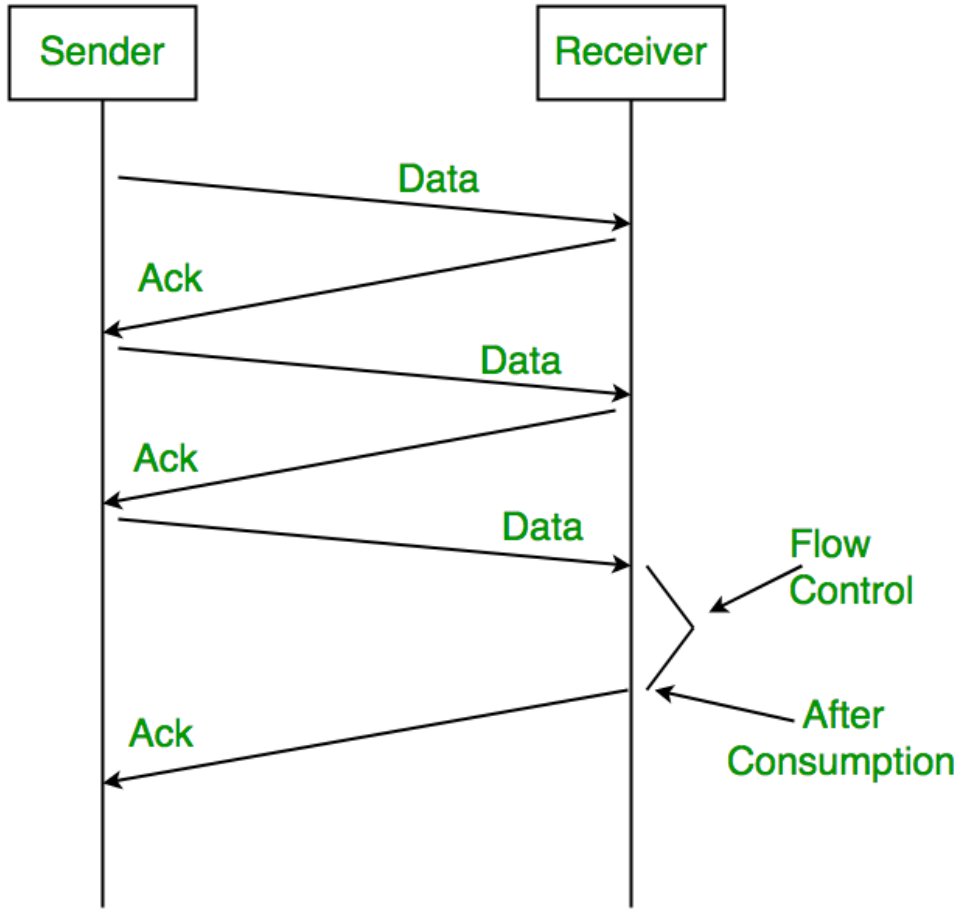
Rule 1) Send one data packet at a time.

Rule 2) Send next packet only after receiving acknowledgement for previous.

Receiver:

Rule 1) Send acknowledgement after receiving and consuming of data packet.

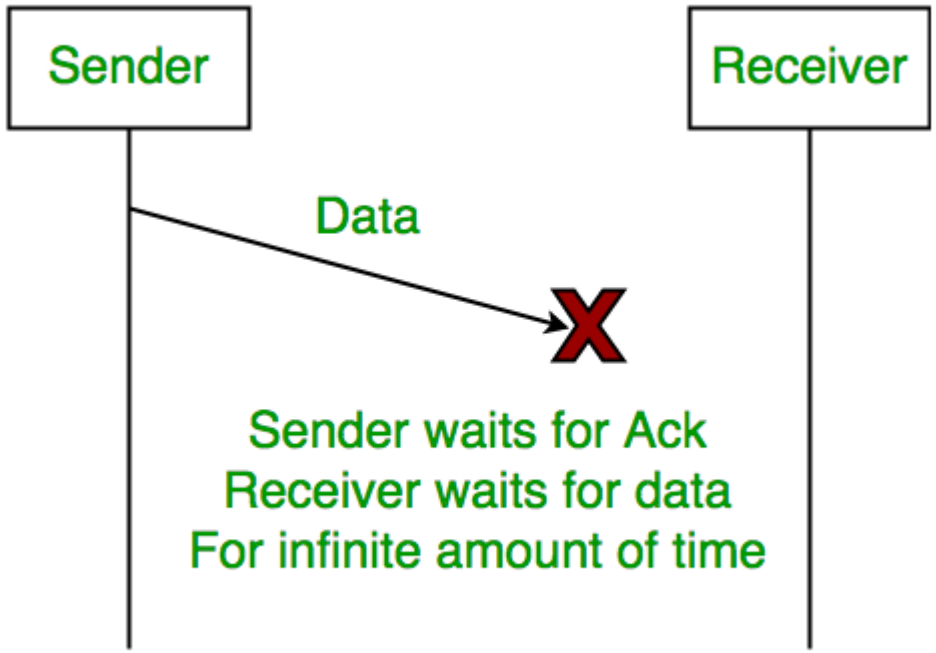
Rule 2) After consuming packet acknowledgement need to be sent (Flow Control)



Simple Stop and Wait Protocol

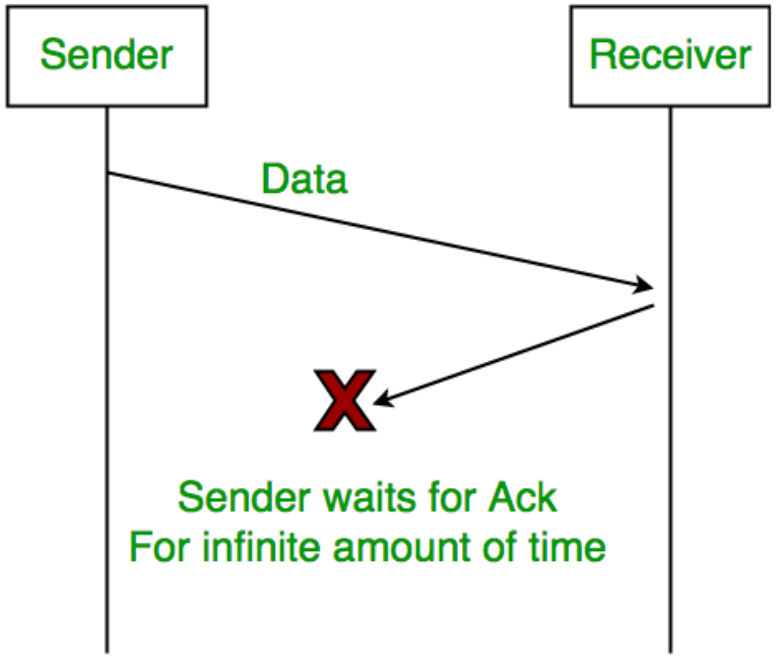
Problems

1. Lost Data



Problems

2. Lost Acknowledgement:

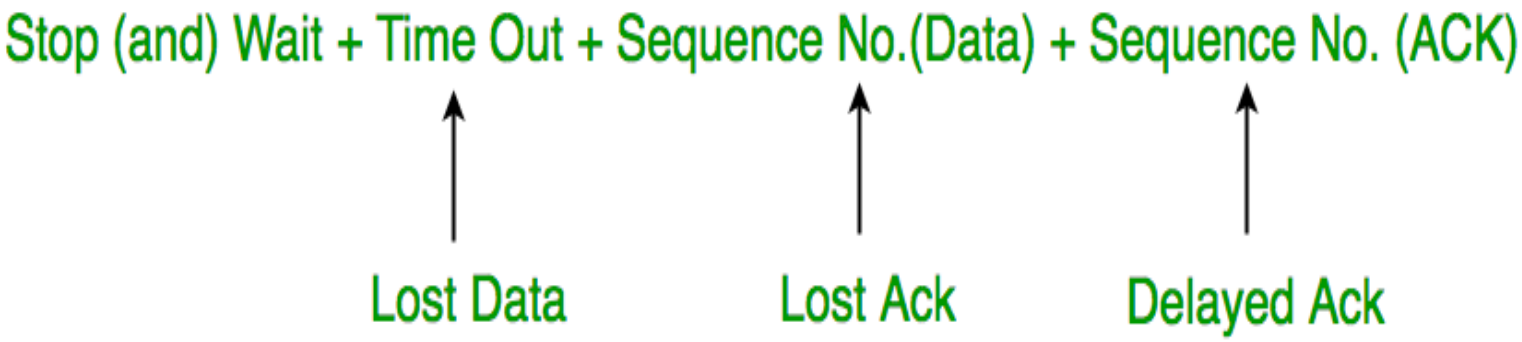


Problems

3. Delayed Acknowledgement/Data:

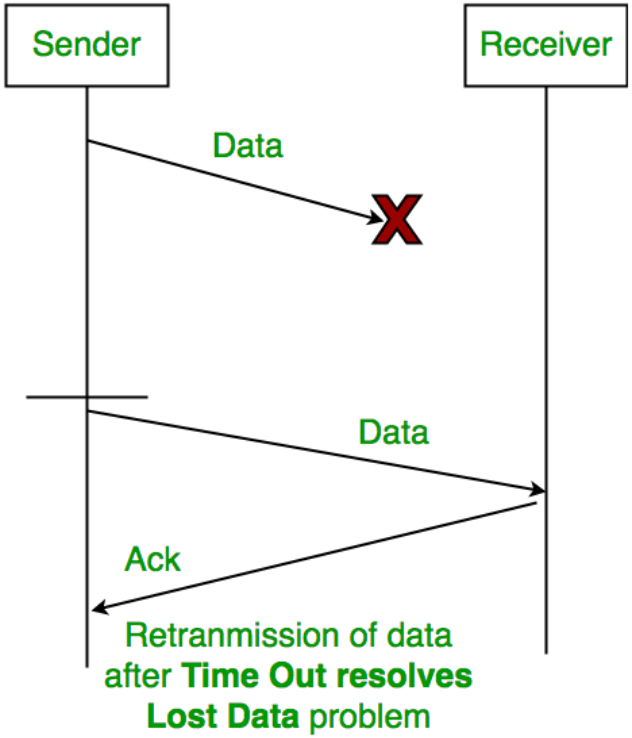
After timeout on sender side, a long delayed acknowledgement might be wrongly considered as acknowledgement of some other recent packet.

Stop and Wait ARQ
(Automatic Repeat Request)



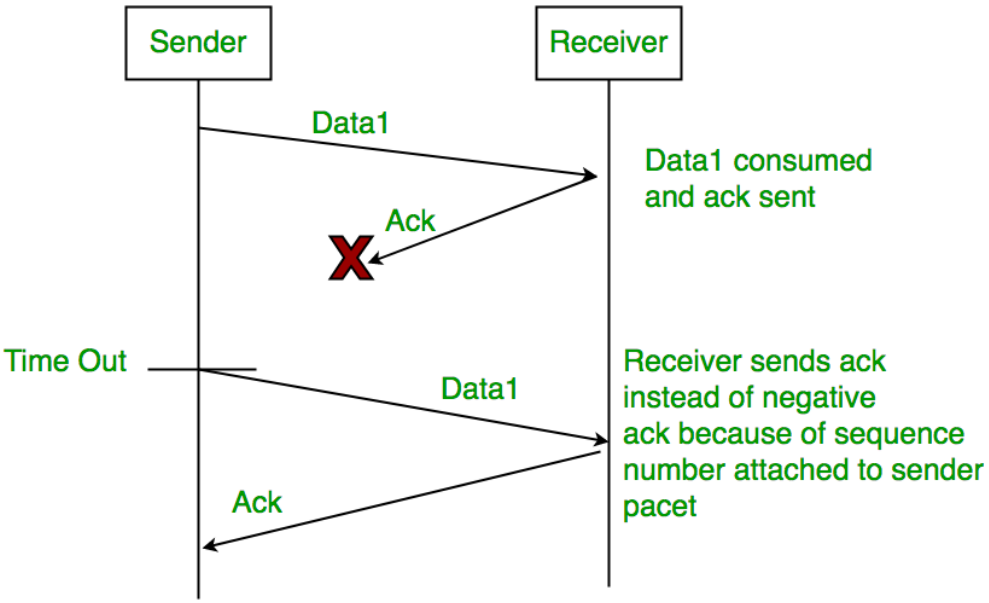
Stop and Wait ARQ
(Automtic Repeat Request)

1. Time Out:



Stop and Wait ARQ
(Automtic Repeat Request)

2. Sequence Number (Data)



Stop and Wait ARQ
(Automtic Repeat Request)

3. Delayed Acknowledgement:

This is resolved by introducing sequence number for acknowledgement also.

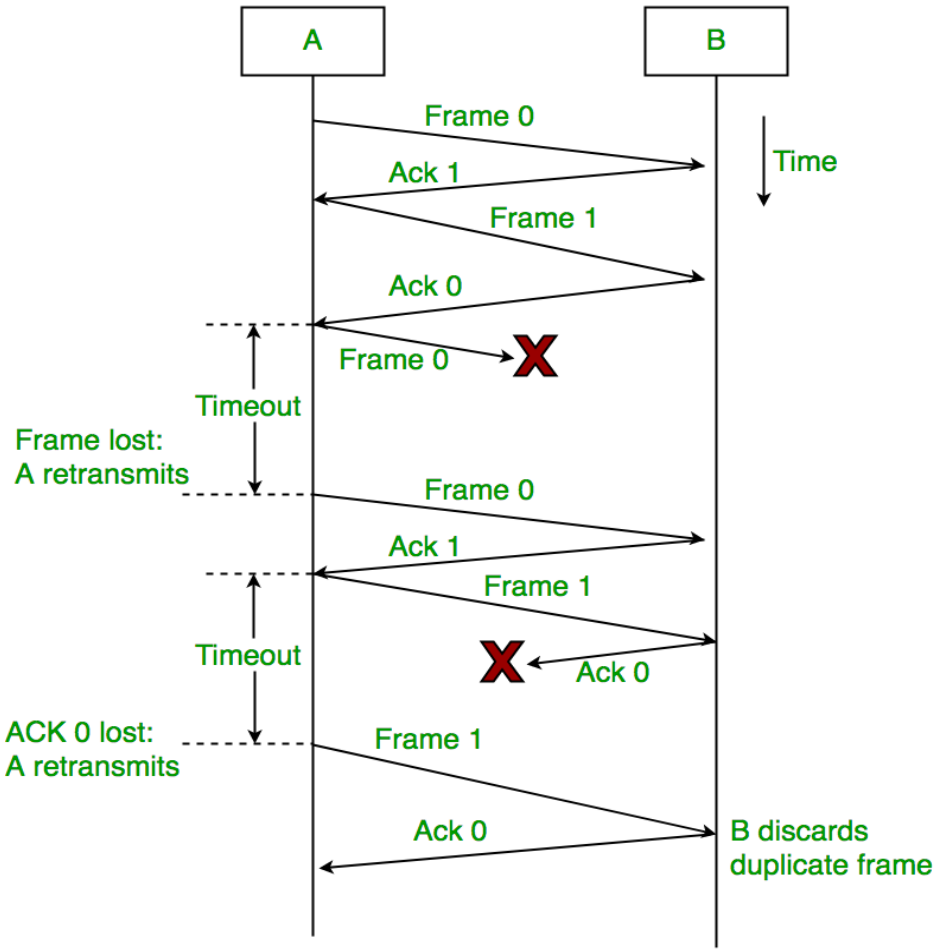
Working of Stop and Wait ARQ:

1) Sender A sends a data frame or packet with sequence number 0.

2) Receiver B, after receiving data frame, sends and acknowledgement with sequence number 1 (sequence number of next expected data frame or packet).

There is only one bit sequence number that implies that both sender and receiver have buffer for one frame or packet only.

Working of Stop and Wait ARQ:



Characteristics of Stop and Wait ARQ:

1. It uses link between sender and receiver as half duplex link

2. If Bandwidth*Delay product is very high, then stop and wait protocol is not so useful. The sender has to keep waiting for acknowledgements before sending the processed next packet.

3. Bandwidth delay product is a measurement of how many bits can fill up a network link. It gives the maximum amount of data that can be transmitted by the sender at a given time before waiting for acknowledgment. Thus it is the maximum amount of unacknowledged data.

4.It is an special category of SWP where its window size is 1.

5. Irrespective of number of packets sender is having stop and wait protocol requires only 2 sequence numbers 0 and 1.

Useful Terms:

1. **Propagation Delay:** Amount of time taken by a packet to make a physical journey from one router to another router.

2. **Round Trip Time** (RTT) = 2* Propagation Delay

3. **Throughput** defines how much useful data can be transmitted per unit time.

Throughput = 1 Data packet/frame per RTT (Stop N Wait ARQ)

Drawback of Stop and Wait ARQ:

1. It causes big performance issues as sender always waits for acknowledgement even if it has next packet ready to send.

Example 1 Bandwidth delay product

Consider that the link capacity of a channel is 512 Kbps and round – trip delay time is 1000ms.

- The bandwidth delay product = BW*RTT

$$= 512 \times 10^3 \text{ bits/sec} \times 1000 \times 10^{-3} \text{ sec}$$

$$= 512{,}000 \text{ bits}$$

$$= 64{,}000 \text{ bytes}$$

$$= 62.5 \text{ KB}$$

Example 2

In a Stop-and-Wait ARQ system, the bandwidth of the line is 1 Mbps, and 1 bit takes 20 ms to make a round trip. What is the bandwidth-delay product? If the system data frames are 1000 bits in length, what is the utilization percentage of the link?

- Bandwidth delay Product:

$$(1 \times 10^{6}) \times (20 \times 10^{-3}) = 20{,}000 \text{ bits}$$

- The link utilization is only 1000/20,000, or 5 percent. For this reason, for a link with a high bandwidth or long delay, the use of Stop-and-Wait ARQ wastes the capacity of the link.

Sliding Window Protocol

Consider a situation where you have a high bandwidth connection and propagation delay is also high (you are connected to some server in some other country though a high speed connection), you can’t use this full speed due to limitations of stop and wait.

Sliding Window protocol handles this efficiency issue by sending more than one packet at a time with a larger sequence numbers.

Sliding Window Protocol

1.Sliding window protocols are data link layer protocols for reliable and sequential delivery of data frames.

2.The sliding window is also used in Transmission Control Protocol.

3. In this protocol, multiple frames can be sent by a sender at a time before receiving an acknowledgment from the receiver.

4. The term sliding window refers to the imaginary boxes to hold frames. Sliding window method is also known as windowing.

Sliding Window Protocol

1. In these protocols, the sender has a buffer called the sending window and the receiver has buffer called the receiving window.

2.The size of the sending window determines the sequence number of the outbound frames.

3. If the sequence number of the frames is an n-bit field, then the range of sequence numbers that can be assigned is 0 to 2 power $n-1$.

4. Consequently, the size of the sending window is 2 power $n-1$. Thus in order to accommodate a sending window size of 2 power $n-1$, a n-bit sequence number is chosen.

Sliding Window Protocols

- A Protocol Using Go Back N
- A Protocol Using Selective Repeat

A Protocol Using Go Back N

- Go-Back-N protocol, also called Go-Back-N Automatic Repeat request, is a data link layer protocol that uses a sliding window method for reliable and sequential delivery of data frames.
- It is a case of sliding window protocol having to send window size of N and receiving window size of 1
- In this protocol we can send several frames before receiving acknowledgments; we keep a copy of these frames until the acknowledgments arrive.
- In the Go-Back-N Protocol, the sequence numbers are modulo 2*m*, where m is the size of the sequence number field in bits**.**
- The sequence numbers range from 0 to *2 power m*- 1. For example, if *m* is 4, the only sequence numbers are 0 through 15 inclusive.

Why we need GBN?

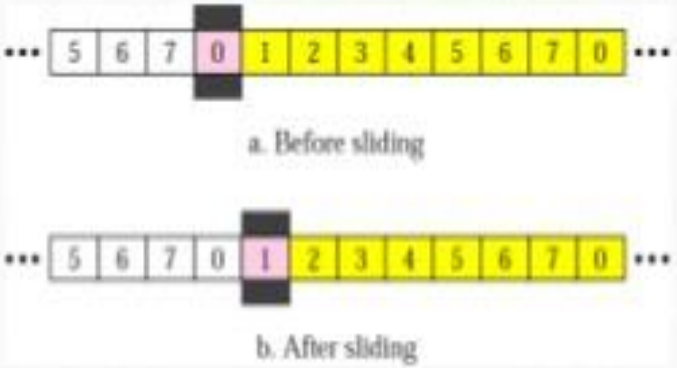
- To improve efficiency of transmission.
- To send more than one frames at a time.
- To reduce waiting time of the sender (i.e.)Sender is waiting for an acknowledgement from the receiver.

Sender sliding Window

- Sender can transmit N frames before receiving ACK.
- Its window size will be N.
- Sender will maintain a copy of sent packets in sent buffer until for the particular sent packets are acknowledged.
- Sender will resend all packets if the timeout timer runs out.
- Once data get acknowledged by the receiver, that particular data will be removed from the buffer.

Receiver sliding window

- Size of the window at the receiving site is always 1 in this protocol.
- Receiver is always looking for a specific frame to arrive in a specific order.
- Any frame arriving out of order is discarded and needs to be resent.
- Receiver window slides as shown in fig. Receiver is waiting for frame 0 in part a.



Receiver sliding window

- Receiver sends positive ACK if a frame arrived safe and in order.
- The silence of the receiver causes the timer of the unacknowledged frame to expire.
- Then the sender resends all frames, beginning with the one with the expired timer.
- For example, suppose the sender has sent frame 6, but the timer for frame 3 expires (i.e. frame 3 has not been acknowledged), then the sender goes back and sends frames 3, 4, 5, 6 again. Thus it is called Go-Back-N-ARQ
- The receiver does not have to acknowledge each frame received, it can send one cumulative ACK for several frames.

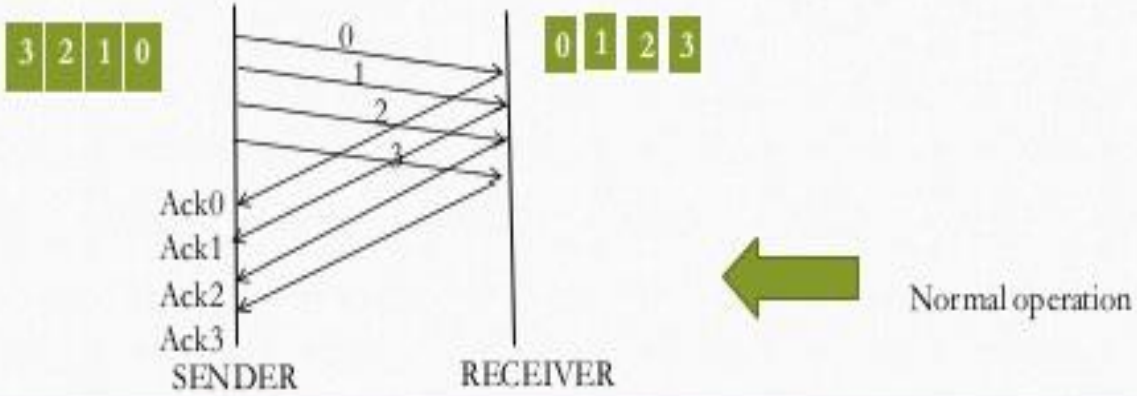
Sender window size.

- Sender window size in GBN is N itself.
- N value must be greater than 1
- If N value is equal to 1,then it is stop & wait protocol.
  Representation:
- GBN=GB10=>N=10=>sender window size.

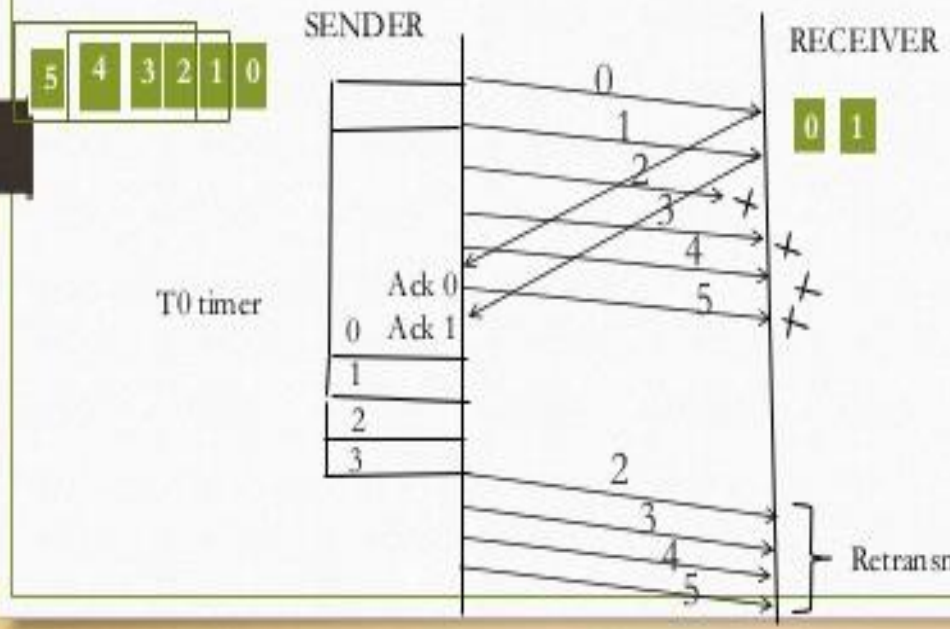
Receiver window size

- Receiver window size is always 1.

Let us consider an example,GB4=>N=4=>sender window size.



Data packet is lost



Acknowledgement

There are two types of acknowledgement used in GBN

- Cumulative Acknowledgement

• Traffic is less

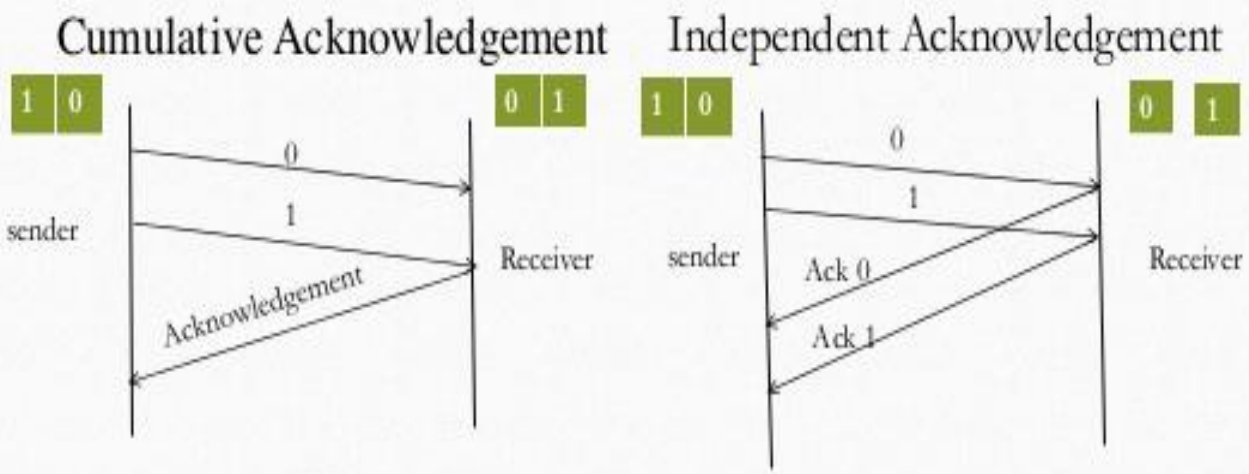
• Reliability is less

- Independent Acknowledgement

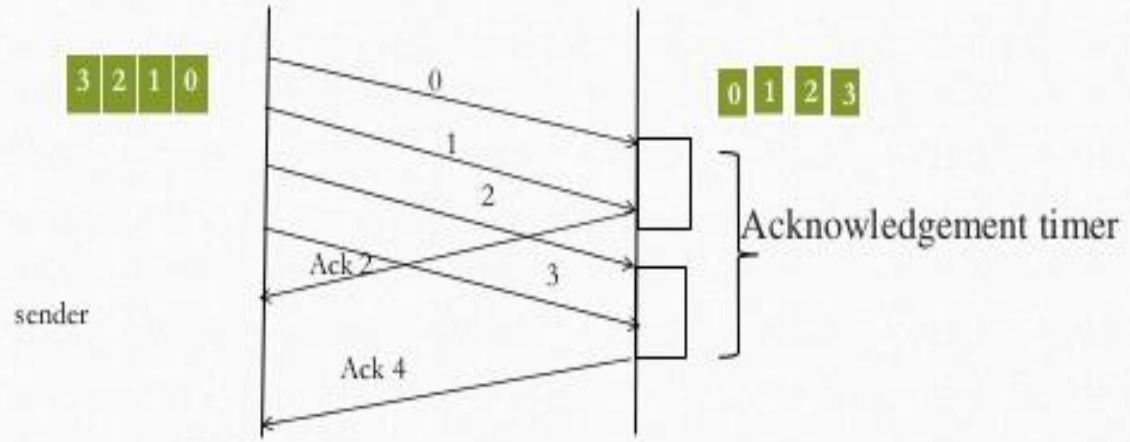
• Traffic is more

• Reliability is more

Graphical Representations



Acknowledgement Timer



Cumulative Acknowledgement

- **Case: 1** Acknowledgement Timer is too big:
- ADVANTAGE : More packets can be acknowledged at a time at the receiver side.
- DISADVANTAGE: Timeout(To) timer runs out at the sender side.
- **Case : 2** Acknowledgement Timer is too small:
  •DISADVANTAGE: Acknowledgement will be send by the receiver for every packets. • So , Acknowledgement timer should not be too small or too big. It should be a timer with fixed size. • Timeout timer must be greater than the Acknowledgement timer.

Go Back N issues

- There is no need to buffer out-of-order packets; they are simply discarded (As window size is 1).However, this protocol is inefficient if the underlying network protocol loses a lot of packets.
- Each time a single packet is lost or corrupted, the sender resends all outstanding packets although some of these packets may have been received safe and sound, but out of order.
- If the network layer is losing many packets because of congestion in the network, the resending of all of these outstanding packets makes the congestion worse

Selective Repeat

- Selective Repeat ARQ is also known as the Selective Repeat Automatic Repeat Request. It is a data link layer protocol that uses a sliding window method.
- The Go-back-N ARQ protocol works well if it has fewer errors. But if there is a lot of error in the frame, lots of bandwidth loss in sending the frames again. So, we use the Selective Repeat ARQ protocol.
- In this protocol, the size of the sender window is always equal to the size of the receiver window. The size of the sliding window is always greater than 1.

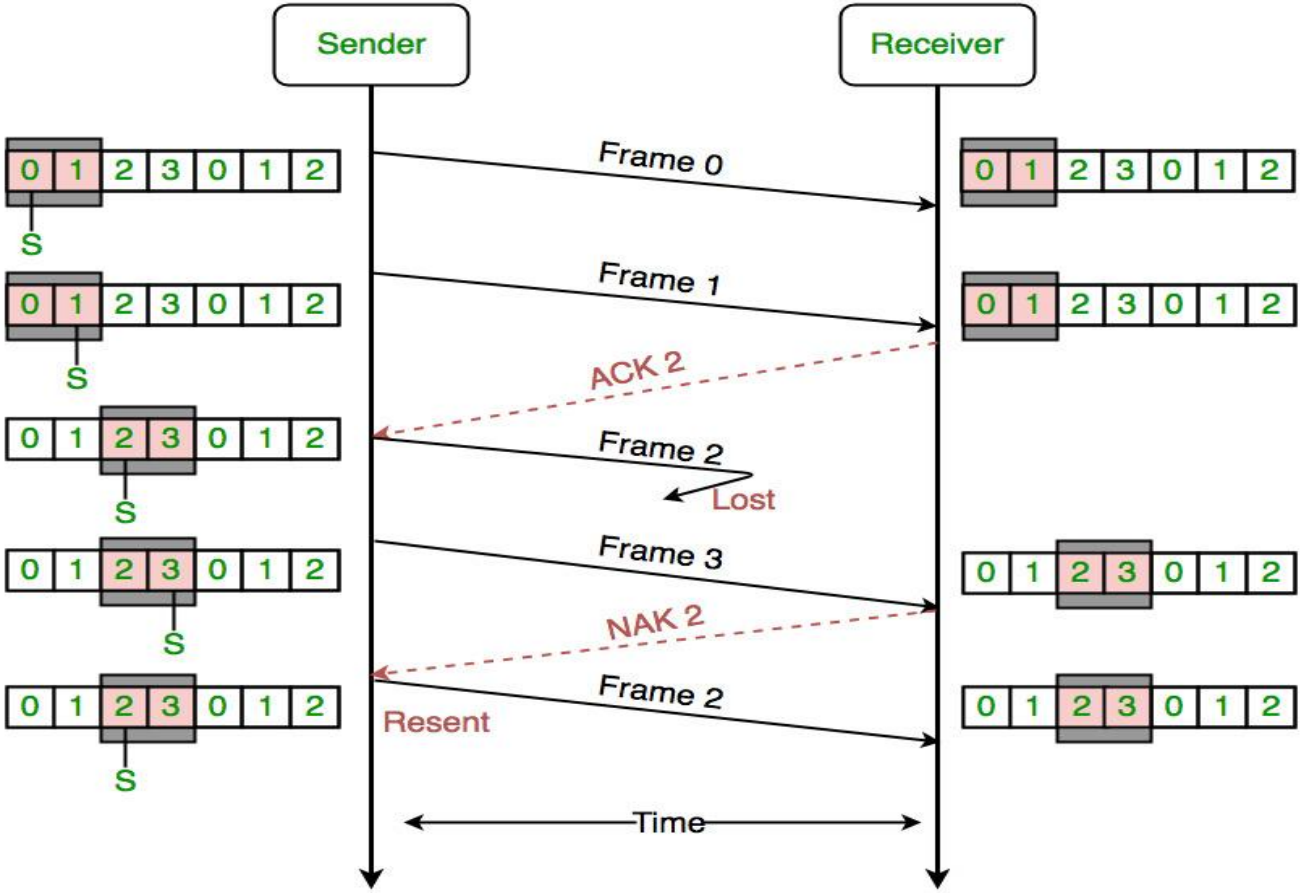
Selective Repeat

- If the receiver receives a corrupt frame, it does not directly discard it. It sends a negative acknowledgment to the sender NAK. The sender sends that frame again as soon as on the receiving negative acknowledgment. There is no waiting for any time-out to send that frame.
- It uses two windows of equal size: a sending window that stores the frames to be sent and a receiving window that stores the frames receive by the receiver. The size is half the maximum sequence number of the frame. For example, if the sequence number is from 0 – 15, the window size will be 8.

Selective Repeat

- Sender's Windows (Ws) = Receiver's Windows (Wr).
- Window size should be less than or equal to half the sequence number in SR protocol. This is to avoid packets being recognized incorrectly. If the windows size is greater than half the sequence number space, then if an ACK is lost, the sender may send new packets that the receiver believes are retransmissions.
- Sender retransmit un-ACKed packets after a timeout – Or upon a NAK if NAK is employed.
- Receiver ACKs all correct packets.Receiver stores correct packets until they can be delivered in order to the higher layer.

Selective Repeat



Difference between Stop and Wait, Go back N and Selective Repeat

PROPERTIES	STOP AND WAIT	GO BACK N	SELECTIVE REPEAT
Sender window size	1	N	N
Receiver Window size	1	1	N
Minimum Sequence number	2	N+1	2N
Efficiency	1/(1+2*a)	N/(1+2*a)	N/(1+2*a)
Type of Acknowledgement	Individual	Cumulative	Individual
Supported order at Receiving end	-	In-order delivery only	Out-of-order delivery as well
Number of retransmissions in case of packet drop	1	N	1

Where, a = Ratio of Propagation delay and Transmission delay

High-Level Data Link Control (HDLC)

- High-level Data Link Control (HDLC) is a group of communication protocols of the data link layer for transmitting data between network points or nodes. Since it is a data link protocol, data is organized into frames. A frame is transmitted via the network to the destination that verifies its successful arrival.
- HDLC protocol embeds information in a **data frame** that allows devices to **control data flow and correct errors**.
- It is a bit - oriented protocol that is applicable for both point - to - point and multipoint communications.
- It has been widely implemented because it supports both half-duplex and full-duplex communication lines, point-to-point (peer to peer) and multi-point networks

HDLC Overview

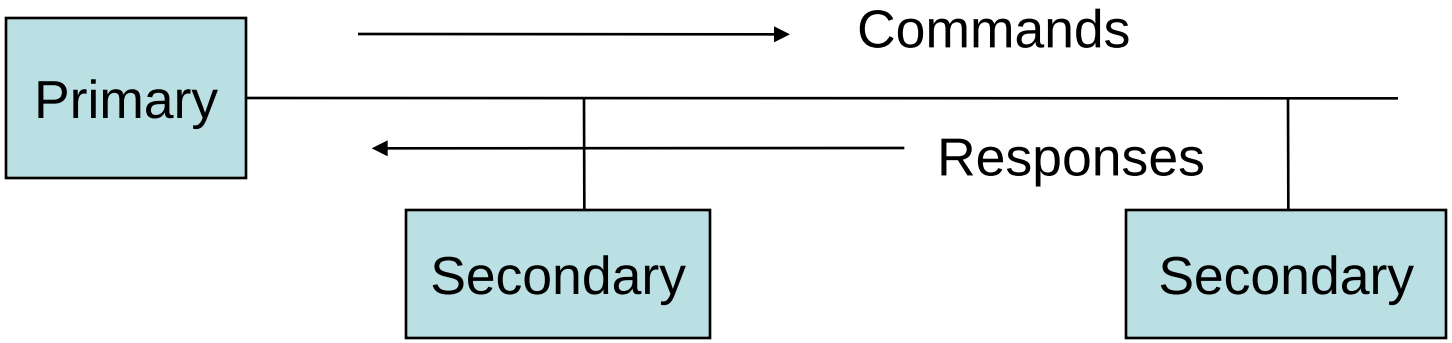
- Defines three types of stations
  - Primary
  - Secondary
  - Combined
- Defines three types of data transfer mode
  - Normal Response mode
  - Asynchronous Response mode
  - Asynchronous Balanced mode
- Three types of frames
  - Unnumbered
  - information
  - Supervisory

HDLC

- The three stations are :
  - Primary station
    - Has the responsibility of controlling the operation of data flow the link.
    - Handles error recovery
    - Frames issued by the primary station are called *commands.*
  - Secondary station,
    - Operates under the control of the primary station.
    - Frames issued by a secondary station are called *responses*.
    - The primary station maintains a separate logical link with each secondary station.
  - Combined station,
    - Acts as both as primary and secondary station.
    - Does not rely on other for sending data

HDLC

Unbalanced Mode



Balanced mode

HDLC

- The three modes of data transfer operations are
  - Normal Response Mode (NRM)

1. This is the mode in which the primary station initiates transfers to the secondary station.

2. The secondary station can only transmit a response when, and only when, it is instructed to do so by the primary station.

3. After receiving permission from the primary station, the secondary station initiates its transmission.

4. NRM is used most frequently in Unbalanced configuration, where the primary station controls the link. It is good for multi-point links

HDLC

– Asynchronous Response Mode (ARM)

1. Same as NRM except that the secondary station does not have to wait to receive explicit permission from the primary station to transfer any frames.
2. Reduces overhead as no frames need to be sent to allow secondary nodes to transmit
3. Transmission proceeds when channel is detected idle, used mostly in point-to-point-links.
4. This is when the ARM link is operating at half-duplex.
5. If the ARM link is operating at full duplex, the secondary station can transmit at any time.

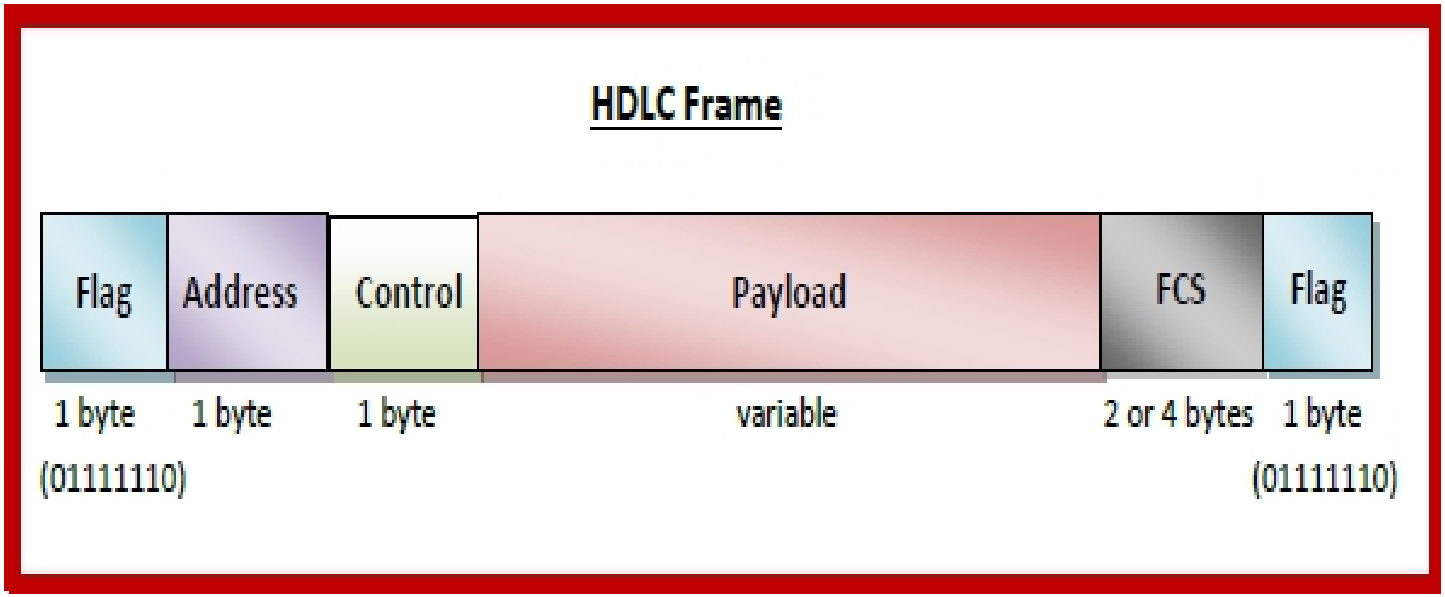
HDLC

Asynchronous Balanced Mode (ABM)

1. This mode is used in case of combined stations. There is no need for permission on the part of any station in this mode. This is because combined stations do not require any sort of instructions to perform any task on the link

2. Mainly used in point-to-point links, for communication between combined stations

Data Link Control HDLC frame structure



The fields of a HDLC frame

1. **Flag** − It is an 8-bit sequence that marks the beginning and the end of the frame. The bit pattern of the flag is 01111110.

2. **Address** − It contains the address of the receiver. If the frame is sent by the primary station, it contains the address(es) of the secondary station(s). If it is sent by the secondary station, it contains the address of the primary station. The address field may be from 1 byte to several bytes.

The fields of a HDLC frame

3. Control − It is 1 or 2 bytes containing flow and error control information.

HDLC uses the control field (C) to determine how to control the communications process. This field contains the commands, responses and sequences numbers used to maintain the data flow accountability of the link, defines the functions of the frame and initiates the logic to control the movement of traffic between sending and receiving

stations.

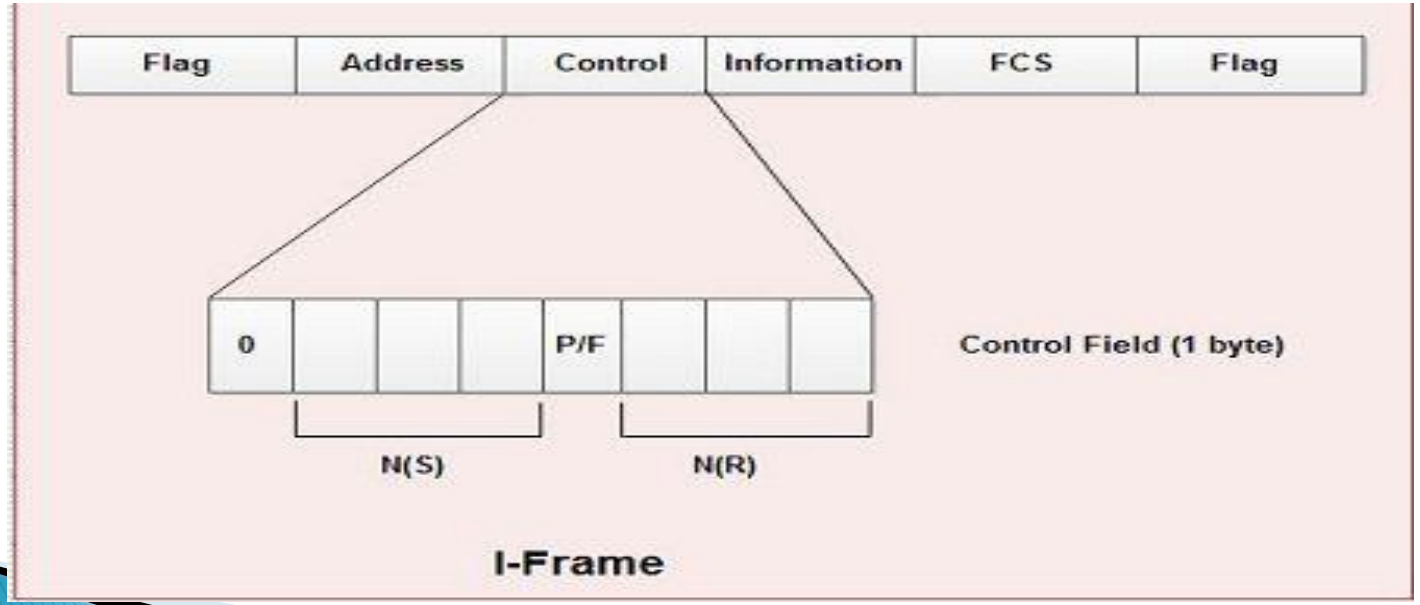
4. Payload − This carries the data from the network layer. Its length may vary from one network to another.

5. FCS − It is a 2 byte or 4 bytes Frame Check Sequence for error detection. The standard code used is CRC (cyclic redundancy code)

Types of HDLC Frames

- I-frame – **Information frames**

- I-frames carry user's data and control information about user's data.
- I-frame carries user data in the information field



I-frame – Information frames

- The first bit of control field is always zero, *i.e.* the presence of zero at this place indicates that it is I-frame.
- • Bit number 2, 3 & 4 in control field is called N(S) that specifies the sequence number of the frame. Thus it specifies the number of the frame that is currently being sent. Since it is a 3.bit field, only eight sequence numbers are possible 0, 1,2,3,4,5,6, 7 (000 to 111).

I-frame – Information frames

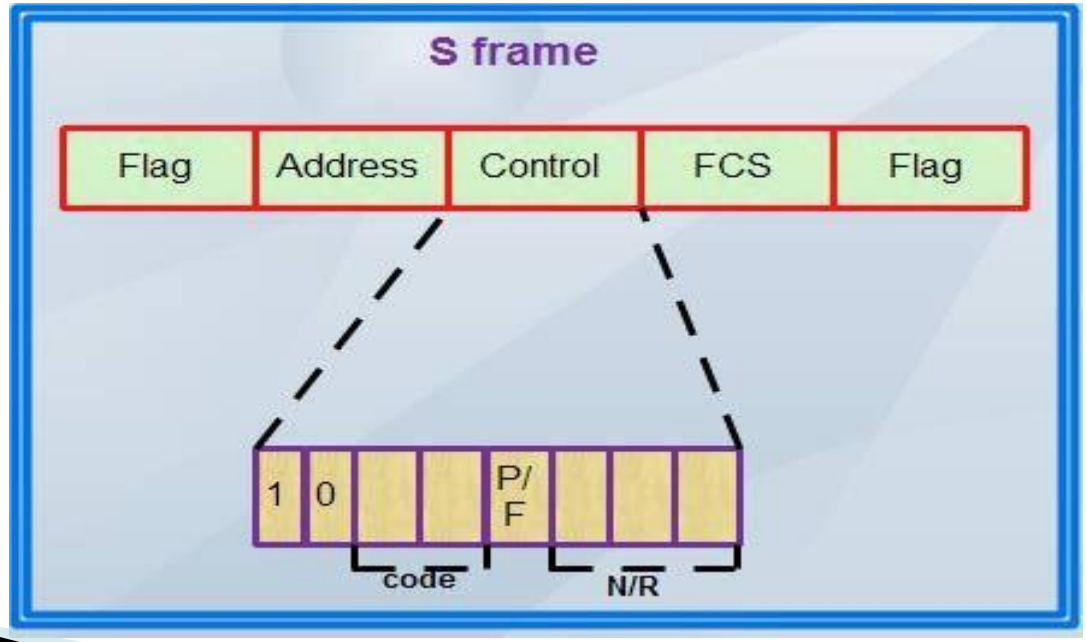
- Bit number 5 in control field is P/F i.e. Poll/Final and is used for these two purposes. It has, meaning only when it is set i.e. when P/F=1. It can represent the following two cases.
- (i) It means poll when frame is sent by a primary station to secondary (when address field contains the address of receiver).
- (ii) It means final when frame is sent by secondary to a primary (when the address field contains the address of the sender).
- • Bit number 6, 7, and 8 in control field specifies N(R) i.e. the sequence number of the frame expected in return in two-way communication.

I-frame – Information frames

- If last frame received was error-free then N(R) number will be that of the next frame is sequence. If the last frame was not received correctly, the N(R) number will be the number of the damaged frame, asking for its retransmission.

Supervisory frame

- S-frame carries control information, primarily data link layer flow and error controls.
- It does not contain information field.
- The format of S-frame is shown in diagram.



Supervisory frame

- The first two bits in the control field of S-frame are always 10.
- Then there is a bit code field that specifies four types of S-frame with combination 00,01, 10, 11 as shown in table :-

Table: Types of S-frame

Code	Command
00	RR Receive Ready
01	REJ Reject
10	RNR Receive Not Ready
11	SREJ Selective Reject

Supervisory frame

1. RR, Receive Ready-used to acknowledge frames when no I-frames are availab1e to piggyback the acknowledgement.
2. REJ Reject-used by the receiver to send a NAK when error has occurred.
3. RNR Receive Not Ready-used for flow control.
4. SREJ Selective Reject-indicates to the transmitter that it should retransmit the frame indicated in the N(R) subfield.

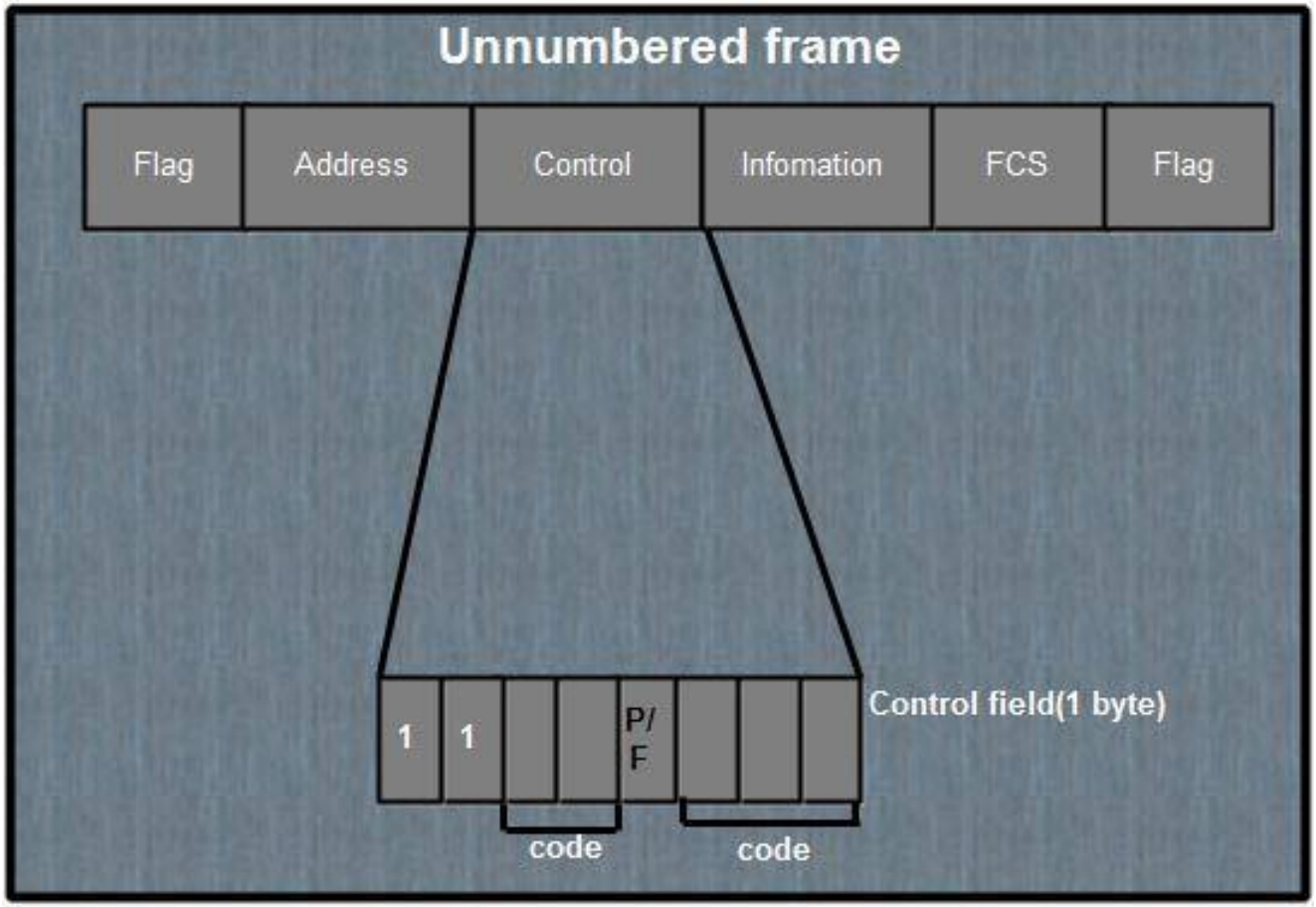
Supervisory frame

- There is no N(S) field in control field of S-frame as S-frames do not transmit data.
- *P/F* bit is the fifth bit and serves the same purpose as discussed earlier.
- Last three bits in control field indicates N(R) *i.e.* they correspond to the ACK or NAK value.

Unnumbered frame

- U-frames are reserved for system management and information carried by them is used for managing the link
- • U-frames are used to exchange session management and control information between the two connected devices.
- • Information field in U-frame does not carry user information rather, it carries system management information.
- • The frame format of U-frame is shown in diagram.
- • U-frame is identified by the presence of 11 in the first and second bit position in control field.
- • These frames do not contain N(S) or N(R) in control field.

Unnumbered frame



Unnumbered frame

- U-frame contains two code fields, one two hit and other three bit.
- • These five bits can create upto 32 different U-frames.
- • .*P/F* bit in control field has same purpose in V-frame as discussed earlier.

Point to Point Data Link Control

- One sender, one receiver, one link: easier than broadcast link:
  - No Media Access Control
  - No need for explicit MAC addressing
  - E.g., dialup link, ISDN line
- PPP is most commonly used data link protocol. It is used to connect the Home PC to the server of ISP via a modem.

Services Provided by PPP

. PPP defines the format of the frame to be exchanged between the devices.

2. It defines link control protocol (LCP) for:-

(a) Establishing the link between two devices.
(b) Maintaining this established link.
(c) Configuring this link.
(d) Terminating this link after the transfer.

3. It defines how network layer data are encapsulated in data link frame.

Services Provided by PPP

4. PPP provides error detection. Not Correction

5. PPP supports multiple protocols.

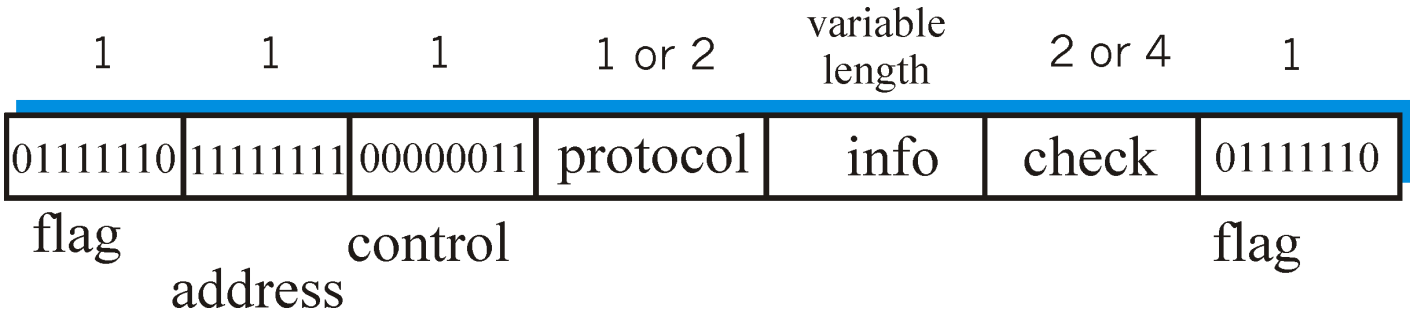
6. PPP allows the IP address to be assigned at the connection time

   i.e. dynamically. Thus a temporary IP address can be assigned to each host.

7. PPP provides multiple network layer services supporting a variety of network layer protocol. For this PPP uses a protocol called NCP (Network Control Protocol).

8. It also defines how two devices can authenticate each other.

PPP Data Frame



PPP Data Frame

1. Flag field: Flag field marks the beginning and end of the PPP frame. Flag byte is 01111110. (1 byte)

2. Address field: This field is of 1 byte and is always 11111111. This address is the broadcast address i.e. all the stations accept this frame.

3. Control field: This field is also of 1 byte. This field uses the format of the U-frame (unnumbered) in HDLC. The value is always 00000011 to show that the frame does not contain any sequence numbers and there is no flow control or error control.

4. Protocol field: This field specifies the kind of packet in the data field i.e. what is being carried in data field.

PPP Data Frame

. Protocol field: This field specifies the kind of packet in the data field i.e. what is being carried in data field.

5. Data field: Its length is variable. If the length is not negotiated using LCP during line set up, a default length of 1500 bytes is used. It carries user data or other information.

6. FCS field: The frame checks sequence. It is either of 2 bytes or 4 bytes. It contains the checksum.

PPP Stack

- Although PPP is a data link layer protocol, PPP uses another set of other protocols to establish the link, authenticate the parties involved, and carry the network layer data.

- Three sets of protocols are defined to make PPP powerful:

1. The Link Control Protocol (LCP),
2. two Authentication Protocols (APs),
3. and several Network Control Protocols (NCPs).

- At any moment, a PPP packet can carry data from one of these protocols in its data field.

The Link Control Protocol (LCP)

- LCP is responsible for establishing, maintaining, configuring, and terminating links.
- It also provides negotiation mechanisms to set options between the two endpoints.

Both endpoints of the link must reach an agreement about the options before the link can be established.

All LCP packets are carried in the payload field of the PPP frame with the protocol

Authentication Protocols

Authentication plays a very important role in PPP because PPP is designed for use over dial-up links where verification of user identity is necessary.

- Authentication means validating the identity of a user who needs to access a set of resources.

- PPP has created two protocols for authentication:
- Password Authentication Protocol
- and Challenge Handshake Authentication Protocol.

PAP (Password Authentication Protocol)

- This protocol provides two step authentication procedures:

Step 1: User name and password is provided by the user who wants to access a system.

Step 2: The system checks the validity of user name and password and either accepts or denies the connection.

- PAP packets are also carried in the data field of PPP frames.

There are three PAP packets.

1. **Authenticate-request**: used to send user name & password.

2. **Authenticate-ack**: used by system to allow the access.

3. **Authenticate-nak**: used by system to deny the access.

CHAP (Challenge Handshake Authentication Protocol)

- It provides more security than PAP.
- In this method, password is kept secret, it is never sent on-line.
- It is a three-way handshaking authentication protocol:

1. System sends. a challenge packet to the user. This packet contains a value, usually a few bytes.
2. Using a predefined function, a user combines this challenge value with the user password and sends the resultant packet back to the system.
3. System then applies the same function to the password of the user and challenge value and creates a result. If result is same as the result sent in the response packet, access is granted, otherwise, it is denied.

CHAP (Challenge Handshake Authentication Protocol)

There are 4 types of CHAP packets:

1. **Challenge**-used by system to send challenge value.

2. **Response**-used by the user to return the result of the calculation.

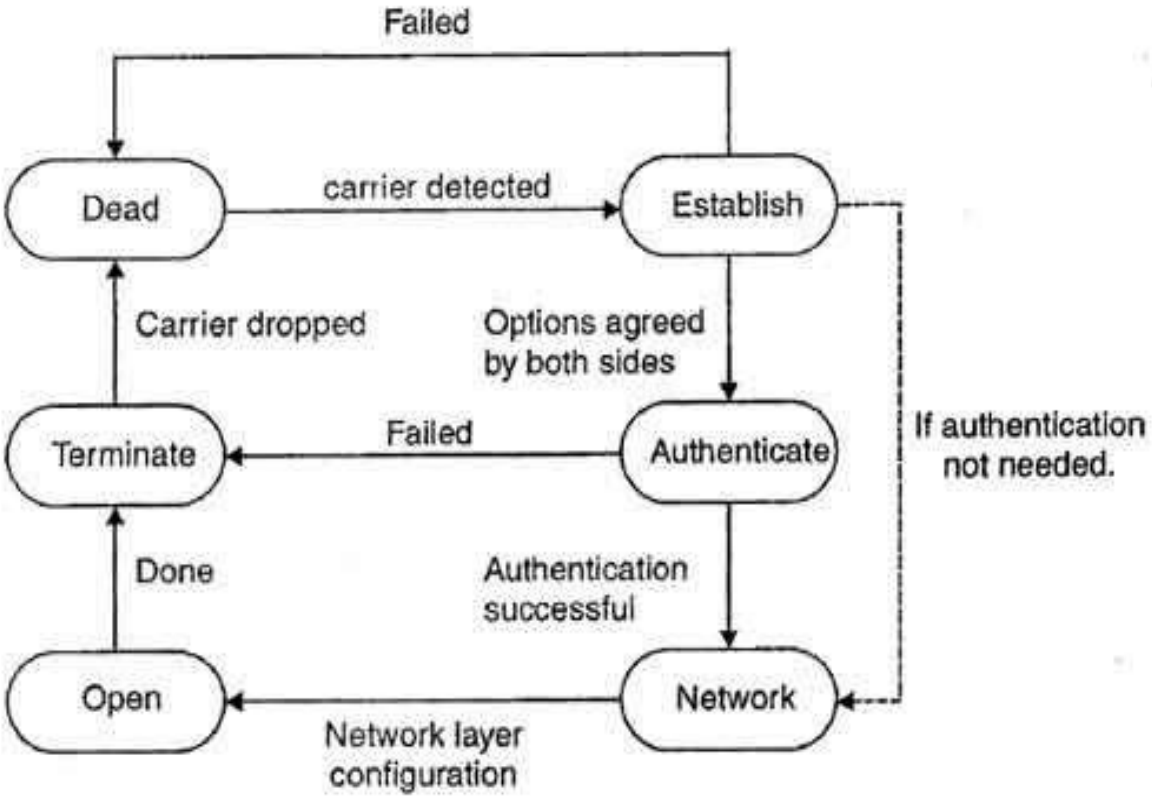
3. **Success**-used by system to allow access to the system.

4. **Failure**-used by the system to deny access to the system.

Network Control Protocol (NCP)

- After establishing the link and authenticating the user, PPP connects to the network layer. This connection is established by NCP.
- Therefore NCP is a set of control protocols that allow the encapsulation of the data coming from network layer.
- After the network layer configuration is done by one of the NCP protocols, the users can exchange data from the network layer.
- None of the NCP packets carry networks layer data. They just configure the link at the network layer for the incoming data.

Transition Phases in PPP



Transition phases

Transition Phases in PPP

1. **Dead:** In dead phase the link is not used. There is no active carrier and the line is quiet.

2. **Establish:** Connection goes into this phase when one of the nodes start communication. In this phase, two parties negotiate the options. If negotiation is successful, the system goes into authentication phase or directly to networking phase. LCP packets are used for this purpose.

3. **Authenticate:** This phase is optional. The two nodes may decide during the establishment phase, not to skip this phase. However if they decide to proceed with authentication, they send several authentication packets. If the result is successful, the connection goes to the networking phase; otherwise, it goes to the termination phase.

Transition Phases in PPP

4. Network: In network phase, negotiation for the network layer protocols takes place. PPP specifies that two nodes establish a network layer agreement before data at the network layer can be exchanged. This is because PPP supports several protocols at network layer. If a node is running multiple protocols simultaneously at the network layer, the receiving node needs to know which protocol will receive the data.

5. Open: In this phase, data transfer takes place. The connection remains in this phase until one of the endpoints wants to end the connection.

6. Terminate: In this phase connection is terminated.