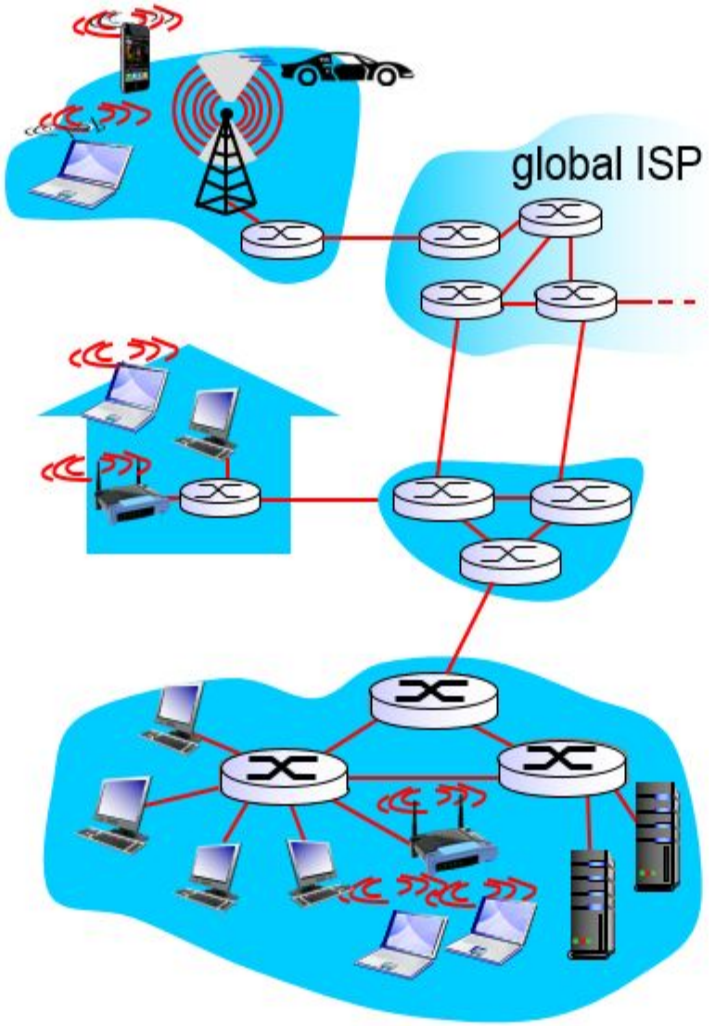
UNIT-1.2
Data Link Layer

Prepared By: Assi. Prof. Namrata Goswami, CSE, IITE

Introduction

- Host and routers are as nodes.
- Communication channels that connect adjacent nodes along communication path, its called links.
  - ✓ Wired links
  - ✓ Wireless links
  - ✓ LANs
- In this layer packet is form of frame from encapsulate datagram.
- This layer has responsibility of transferring datagram from one node to physically adjacent node over a link.



Link Layer Services

- **Framing**
  - ✓ Encapsulate datagram into frame.
  - ✓ Adding header and trailer.
- **Link Access**
  - ✓ "MAC" addresses used in frame headers to identify source and destination. Its different from IP address.
- **Reliable delivery**
  - ✓ If this layer protocol provides reliable delivery service, it guarantees to move each network-layer datagram across the link without error.
  - ✓ A link-layer reliable delivery service can be achieved with acknowledgments and retransmissions.
- **Flow Control**
  - ✓ Pacing between adjacent sending and receiving nodes.

Link Layer Services – Cont…

- Error Detection & Correction
  - ✓ Errors caused by signal attenuation, noise.
  - ✓ Receiver detects presence of errors.
  - ✓ Sender send signal for retransmission or drops frame.
  - ✓ Receiver identifies *and corrects* bit error(s) without resorting to retransmission.

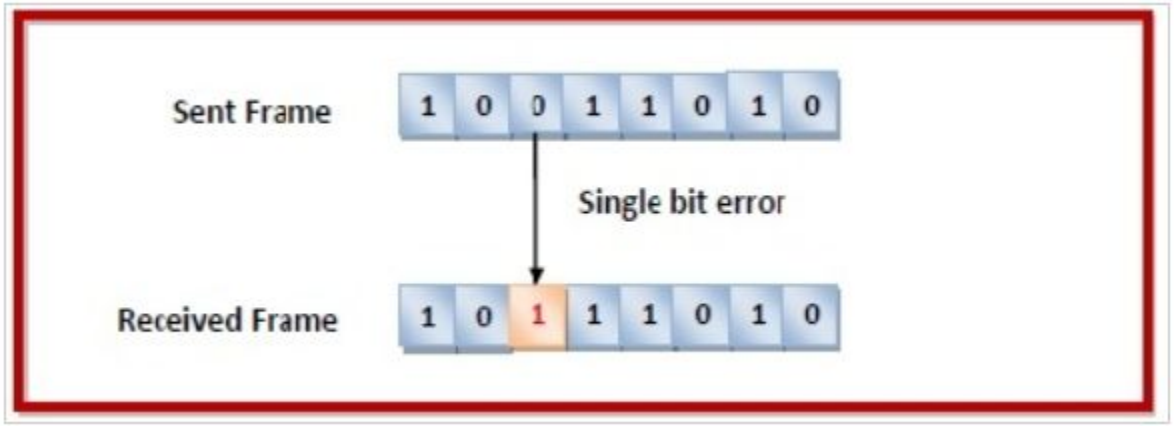
Types of Errors

Errors:

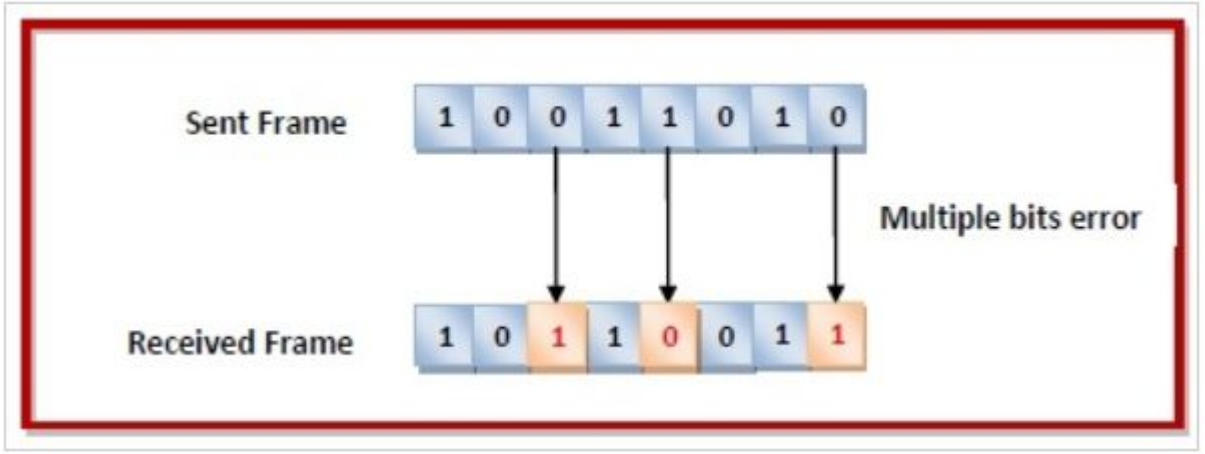
When bits are transmitted over the computer network, they are subject to get corrupted due to interference and network problems.

The corrupted bits leads to spurious data being received by the destination and are called errors.

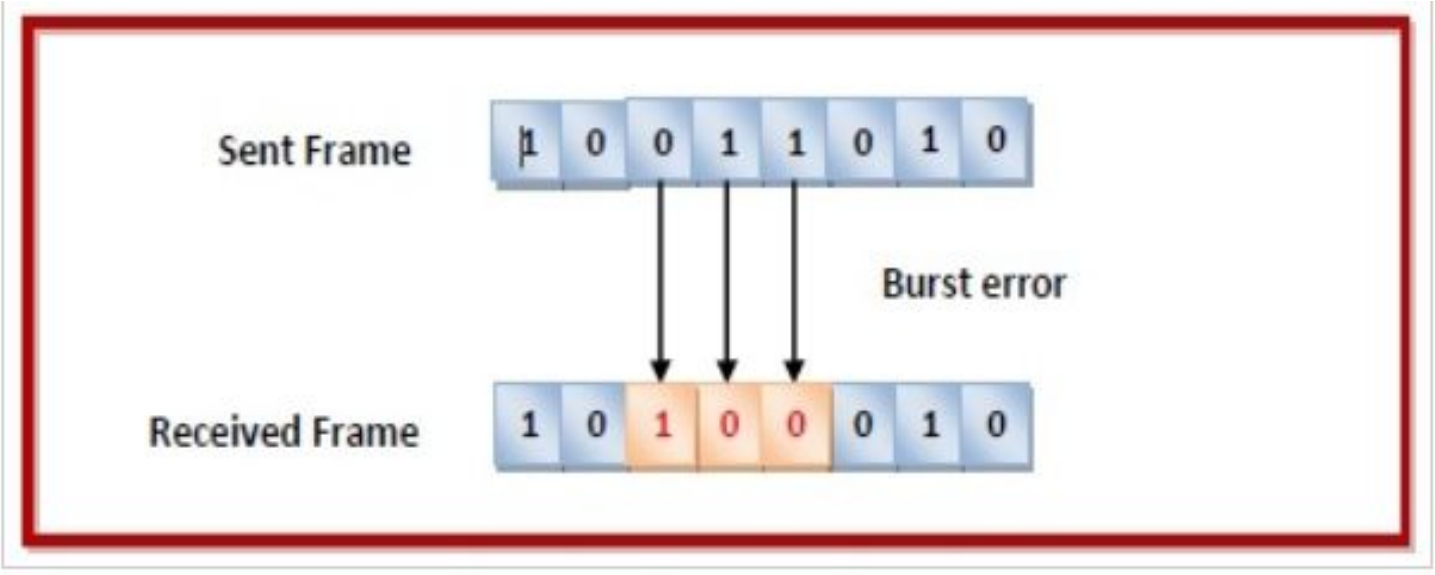
Single bit error − In the received frame, only one bit has been corrupted, i.e. either changed from 0 to 1 or from 1 to 0.



Multiple bits error − In the received frame, more than one bits are corrupted.



Burst error − In the received frame, more than one consecutive bits are corrupted.



Error detection and correction

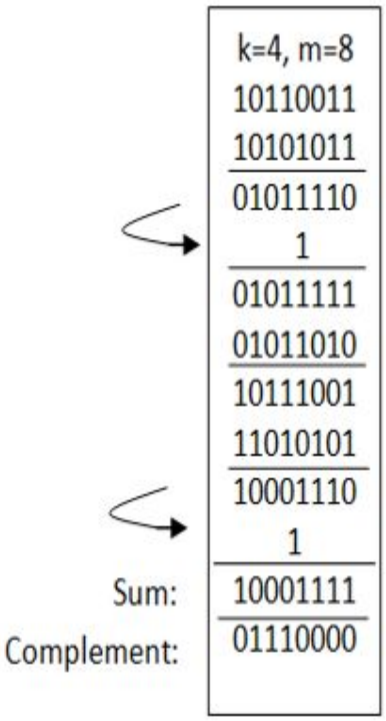
- **Errors** are introduced into the binary data transmitted from the sender to the receiver due to noise during transmission.

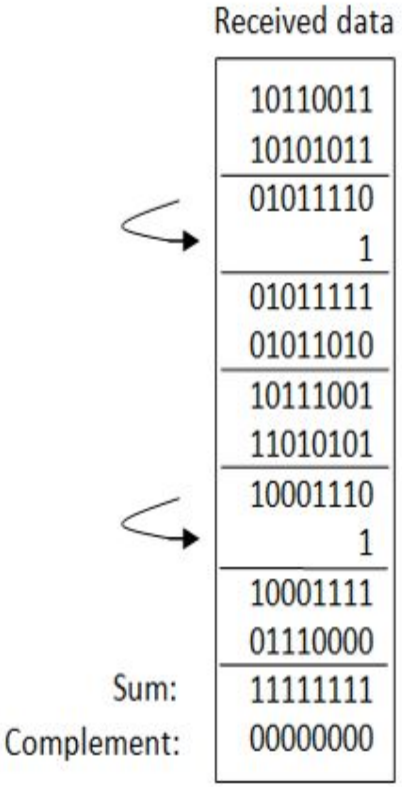
- Error detection methods are used to check whether the receiver has received correct data or corrupted data.

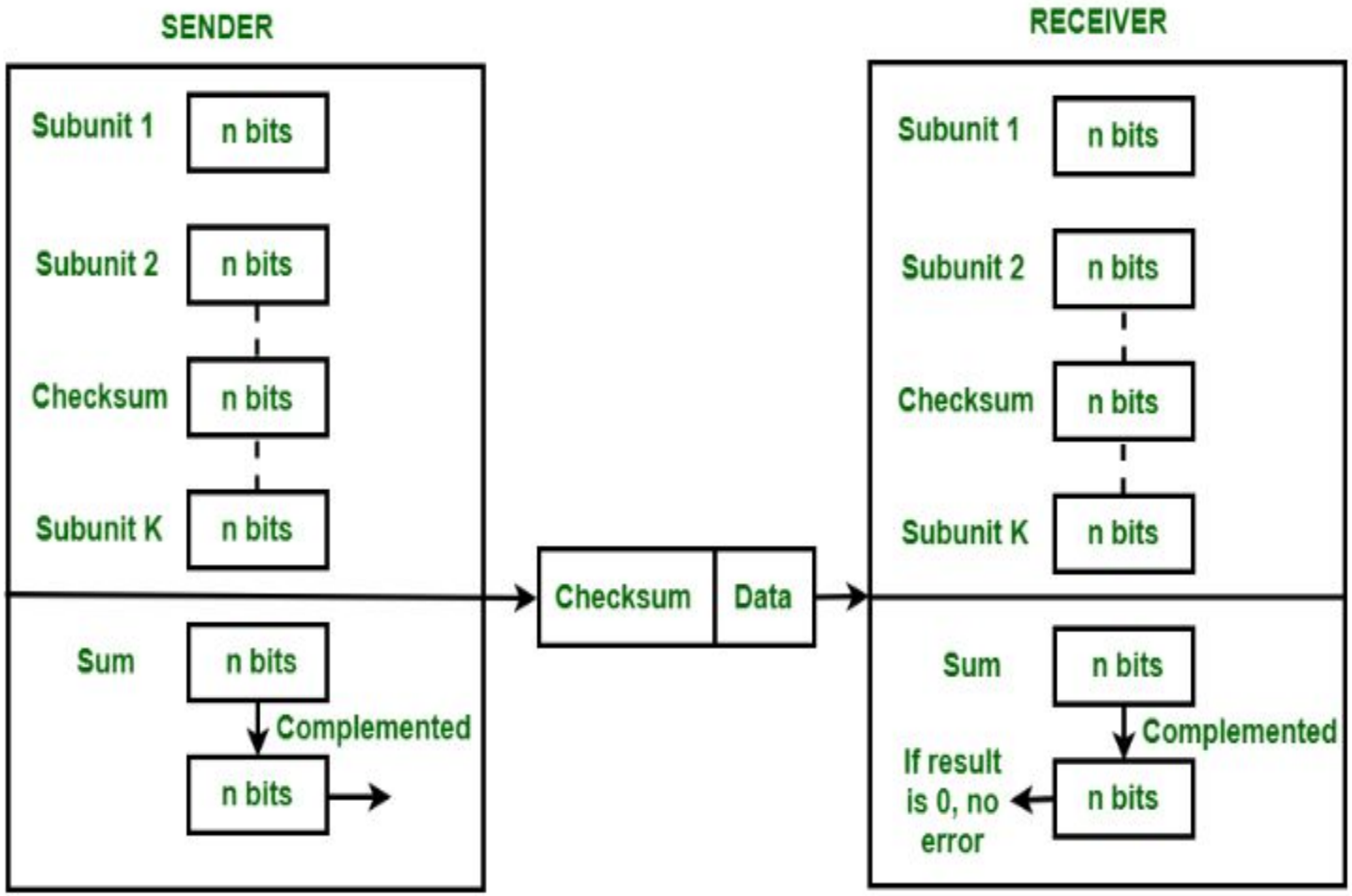
- Error correction is used to correct the detected errors during the transmission of data from sender to receiver.

Checksum

- Data is divided into k segments each of m bits.
- Sender side: Segments are added using 1's complement arithmetic to get the sum.
- sum is complemented to get the checksum.
- Checksum segment is sent along with the data segments.
- Receiver Side: All received segments are added using 1's complement arithmetic to get complemented sum.
- If the result is zero, the received data is accepted; otherwise discarded.






SENDER
RECEIVER
Subunit 1
n bits
Subunit 2
n bits
Checksum
n bits
Subunit K
n bits
Sum
n bits
Complemented
n bits
Checksum
Data
Subunit 1
n bits
Subunit 2
n bits
Checksum
n bits
Subunit K
n bits
Sum
n bits
Complemented
If result is 0, no error
n bits

Advantage :

The checksum detects all the errors involving an odd number of bits as well as the error involving an even number of bits.

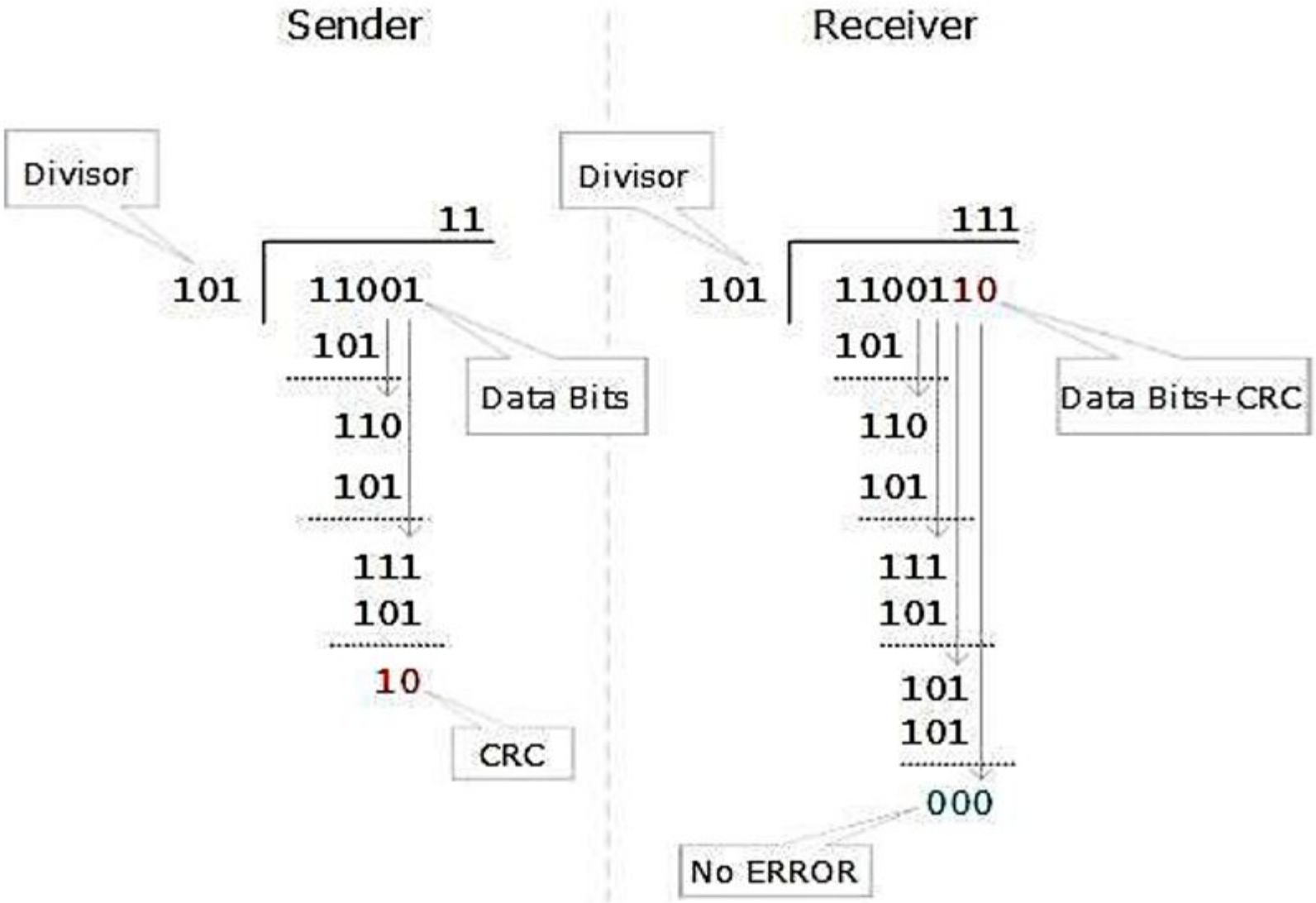
Disadvantage :

The main problem is that the error goes undetected if one or more bits of a subunit is damaged and the corresponding bit or bits of a subunit are damaged and the corresponding bit or bits of opposite value in second subunit are also damaged. This is because the sum of those columns remains unchanged.

Cyclic Redundancy Check

- CRC is the most powerful and easy to implement technique.
- CRC is based on binary division.
- In CRC, a sequence of redundant bits, are appended to the end of data unit so that the resulting data unit becomes exactly divisible by a second, predetermined binary number.
- 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.
- A remainder indicates that the data unit has been damaged in transit and therefore must be rejected.
- The binary number, which is (r+1) bit in length, can also be considered as the coefficients of a polynomial, called Generator Polynomial.

CRC – Example:1



Hamming code

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 a **technique developed by R.W. Hamming for error correction**.

Redundant bits – 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.

- 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 ≥ 7 + 4 + 1 Thus, the number of redundant bits= 4 **Parity bits.**
- A parity bit is a bit appended to a data of binary bits to ensure that the total number of 1's in the data is even or odd.
- There are two types of parity bits:

Even parity bit: In the case of even parity, for a given set of bits, the number of 1's are counted. If that count is odd, the parity bit value is set to 1, making the total count of occurrences of 1's an even number. If the total number of 1's in a given set of bits is already even, the parity bit's value is 0.

- **Odd Parity bit –** In the case of odd parity, for a given set of bits, the number of 1's are counted. If that count is even, the parity bit value is set to 1, making the total count of occurrences of 1's an odd number. If the total number of 1's in a given set of bits is already odd, the parity bit's value is 0.

- **General Algorithm of Hamming code:** 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).
3. All the other bit positions are marked as data bits.
4. Each data bit is included in a unique set of parity bits, as determined its bit position in binary form.

5. 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.

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

Determining the position of redundant bits – These redundancy bits are placed at positions that correspond to the power of 2.

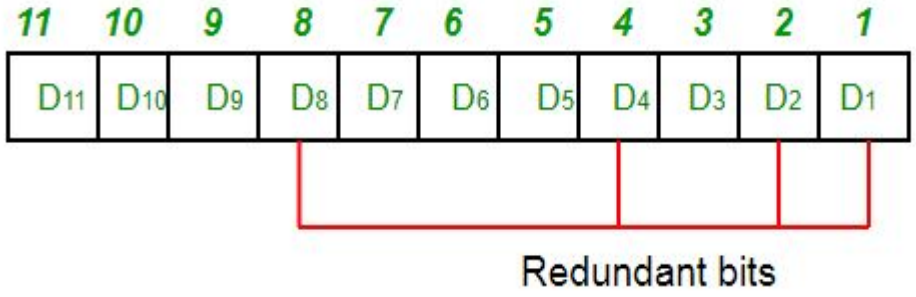
As in the above example:

The number of data bits = 7

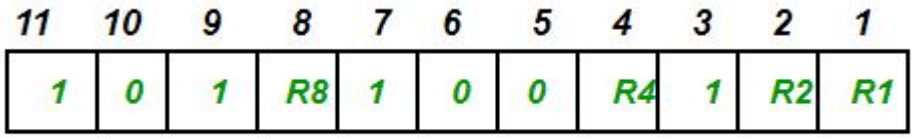
The number of redundant bits = 4

The total number of bits = 11

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

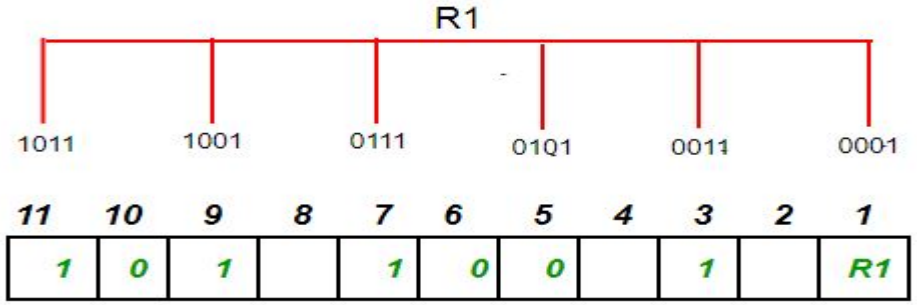


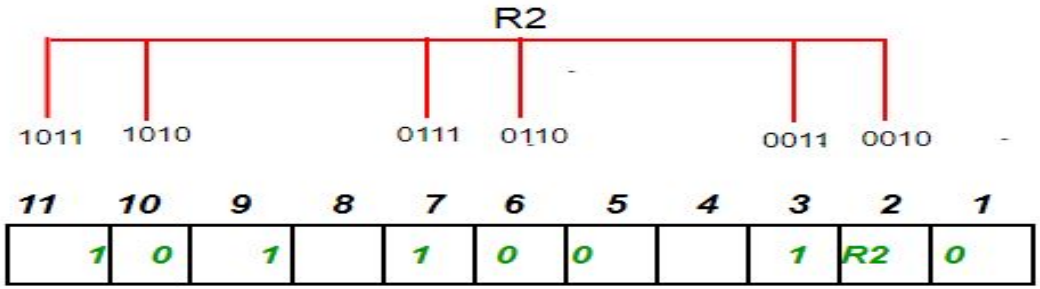
Suppose the data to be transmitted is 1011001, the bits will be placed as follows:



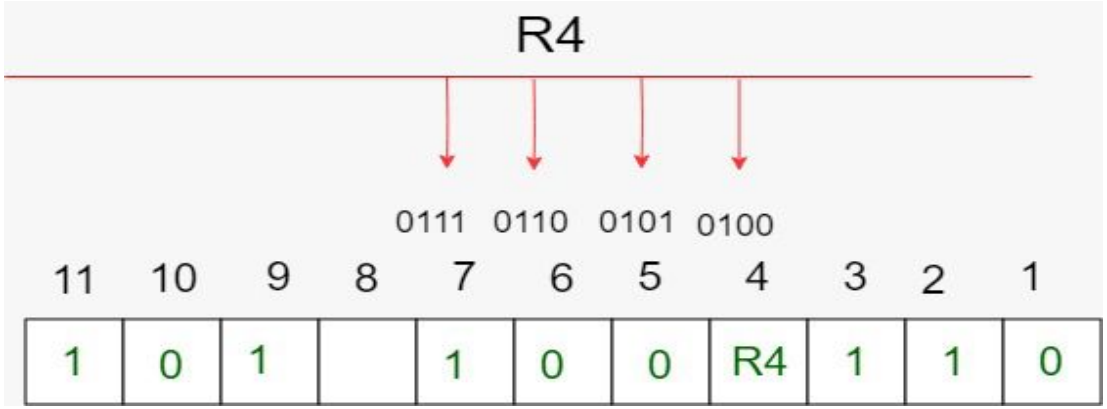
Determining the Parity bits:

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

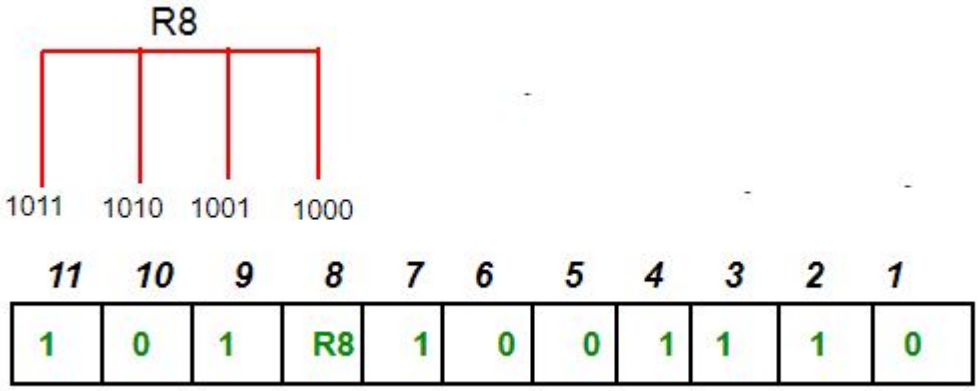




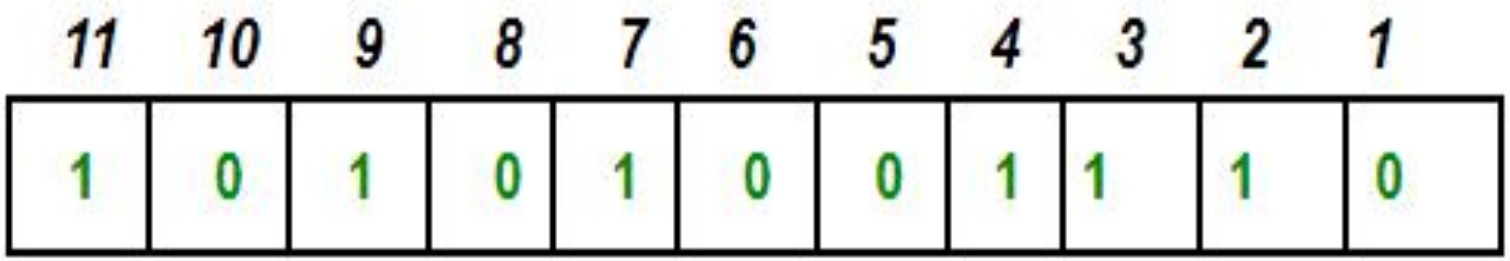
R4 bit is calculated using parity check at all the bits positions whose binary representation includes a 1 in the third position from the least significant bit. R4: bits 4, 5, 6, 7



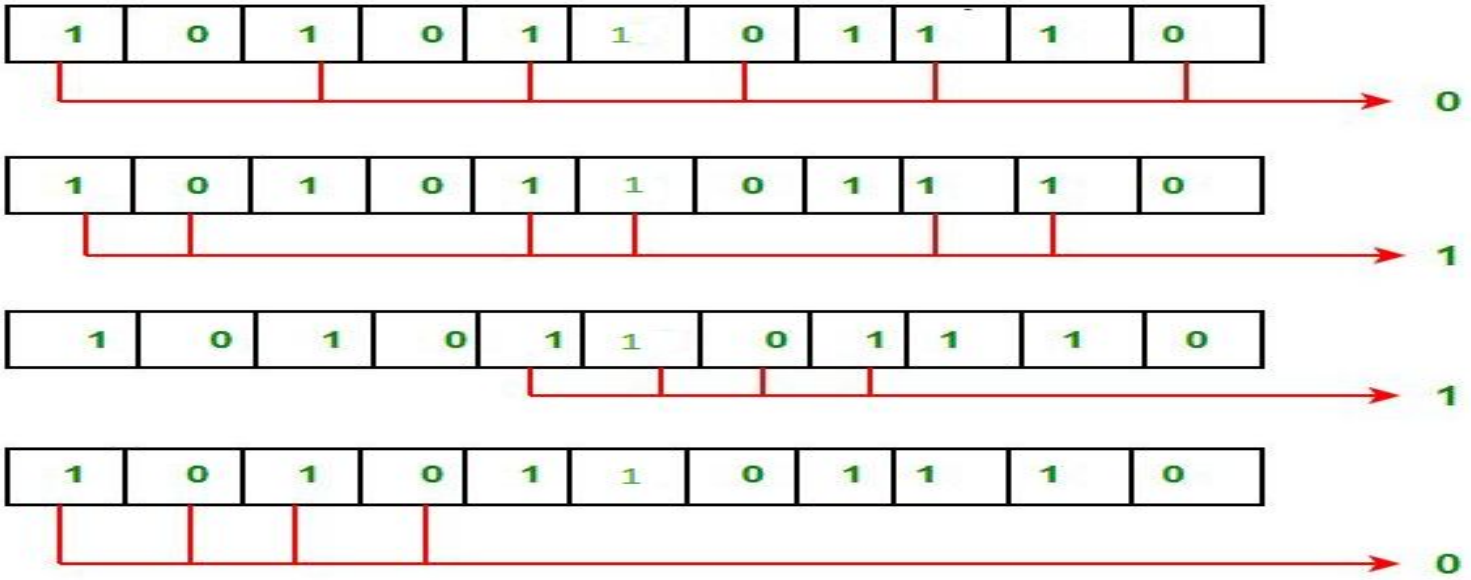
R8 bit is calculated using parity check at all the bits positions whose binary representation includes a 1 in the fourth position from the least significant bit.
R8: bit 8,9,10,11



number the value of R8(parity bit's value)=0.



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:



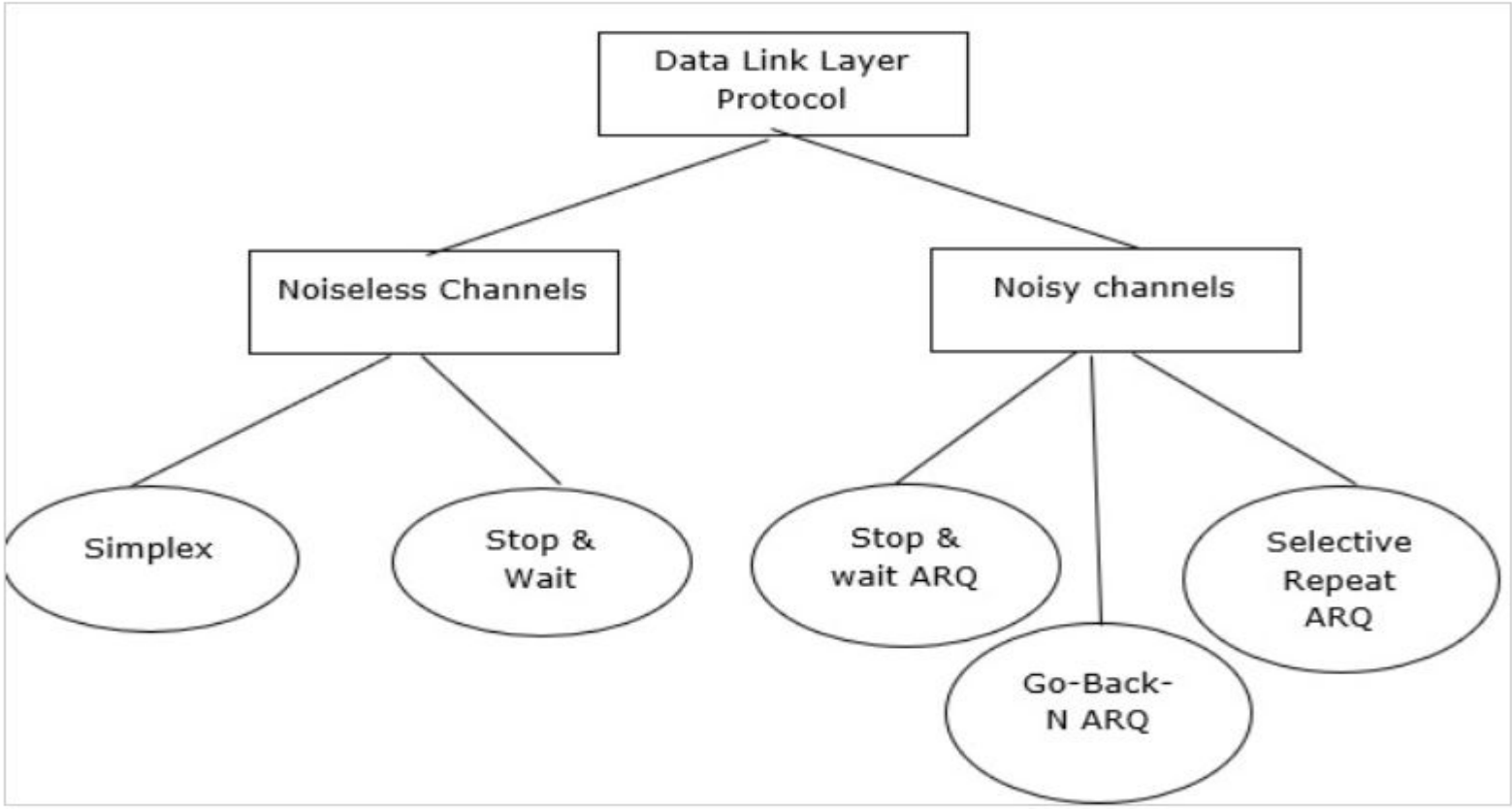
- The bits give the binary number 0110 whose decimal representation is 6. Thus, bit 6 contains an error. To correct the error the 6th bit is changed from 1 to 0.

Here are some of the features of Hamming code:

- Error Detection and Correction
- Redundancy
- Efficiency
- Widely Used
- Single Error Correction
- Limited Multiple Error Correction

Data link protocols for noisy and noiseless channels

- Data link layer protocols are divided into two categories based on whether the transmission channel is noiseless or noisy.



Noiseless Channels

There are two noiseless channels which are as follows −

- Simplex channel
- Stop & wait channel

Let us consider an ideal channel where no frames are lost, duplicated, or corrupted. We introduce two protocols for this type of channel. These two protocols are as follows −

- Protocol that does not use flow control.
- Protocol that uses the flow control.

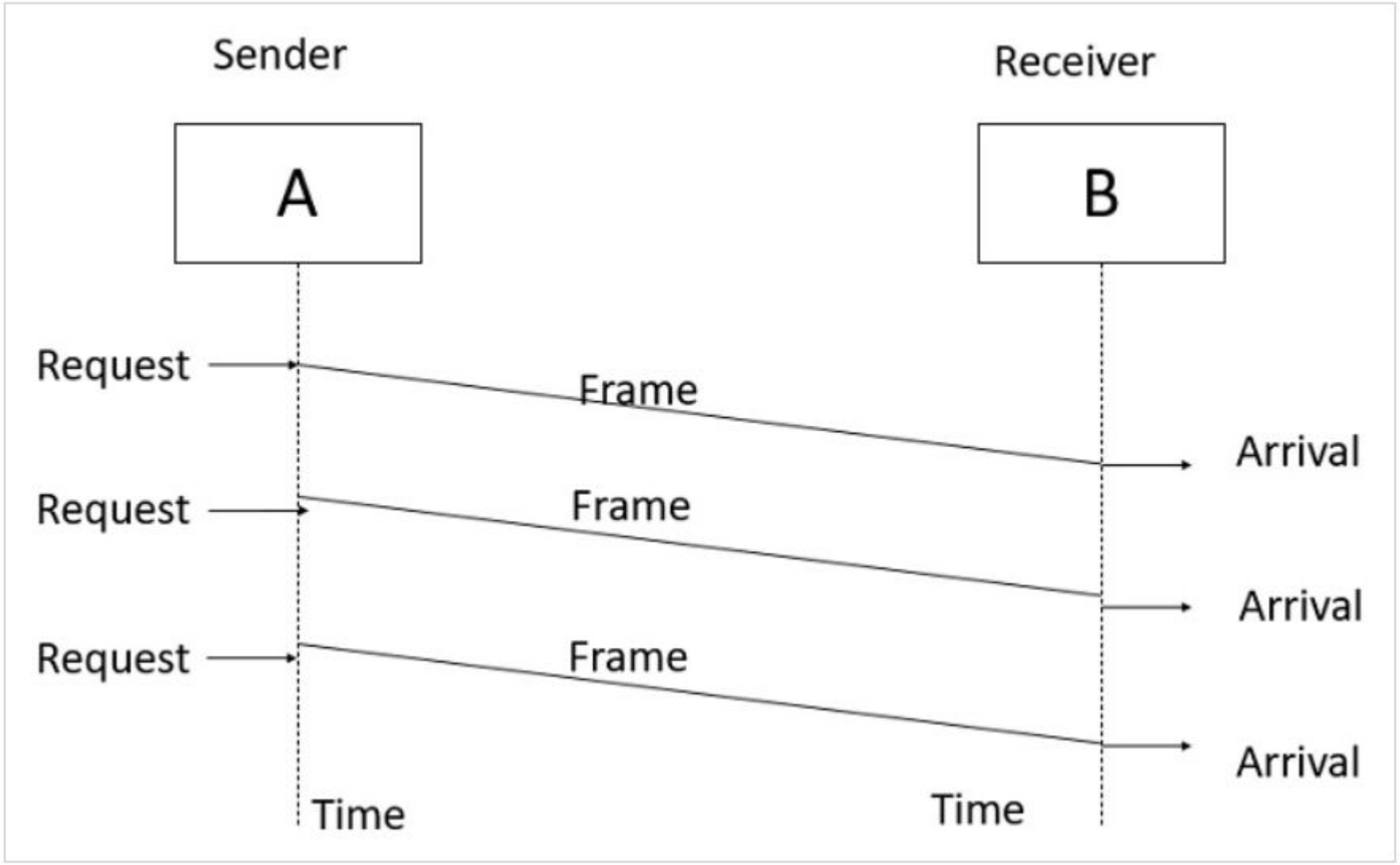
Simplest Protocol

Step 1 − Simplest protocol that does not have flow or error control.

Step 2 − It is a unidirectional protocol where data frames are traveling in one direction that is from the sender to receiver.

Step 3 − Let us assume that the receiver can handle any frame it receives with a processing time that is small enough to be negligible, the data link layer of the receiver immediately removes the header from the frame and hands the data packet to its network layer, which can also accept the packet immediately.

Simplest Protocol



Stop-and-Wait Protocol

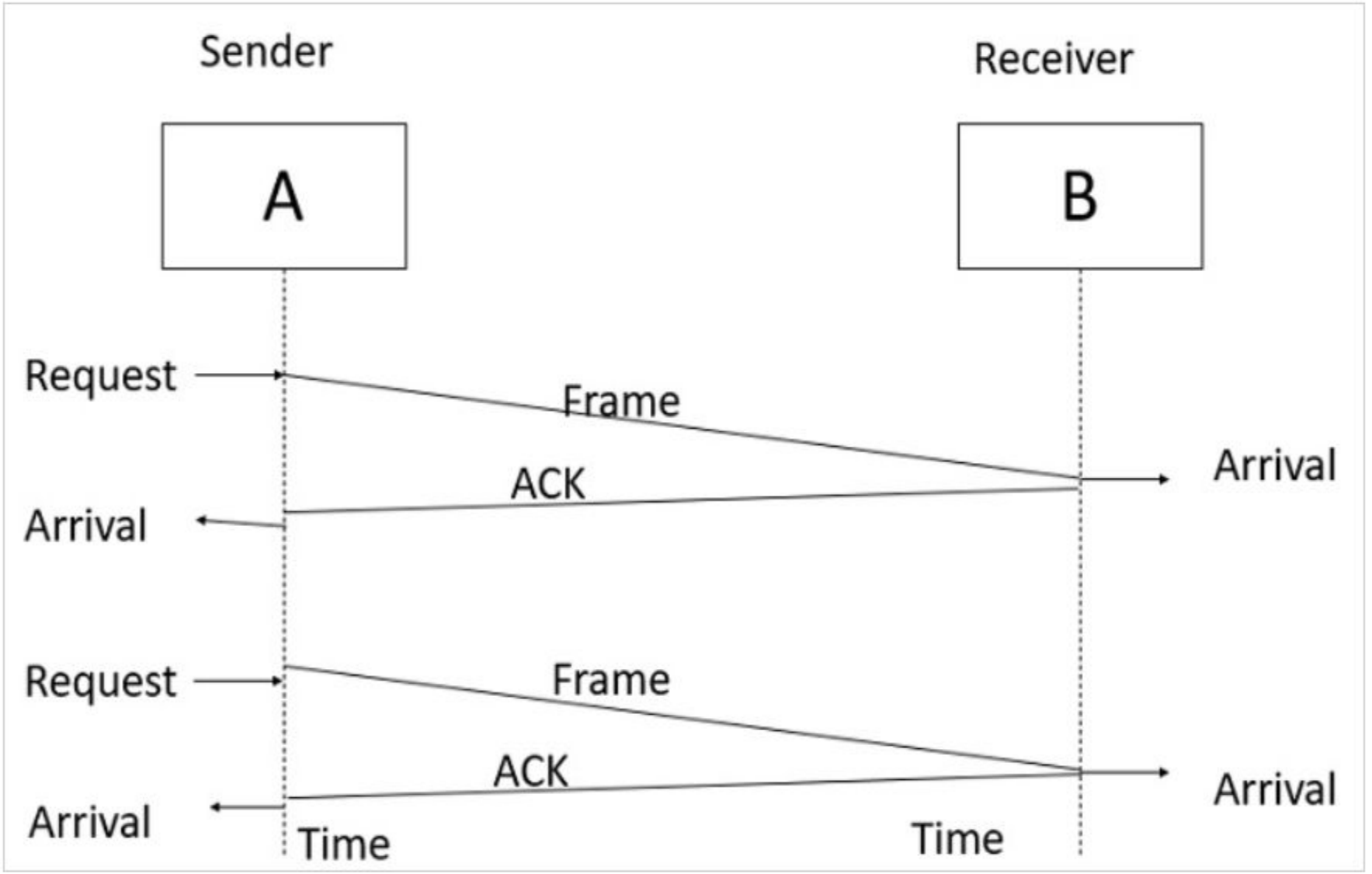
Step 1 − If the data frames that arrive at the receiver side are faster than they can be processed, the frames must be stored until their use.

Step 2 − Generally, the receiver does not have enough storage space, especially if it is receiving data from many sources. This may result in either discarding of frames or denial of service.

Step 3 − To prevent the receiver from becoming overwhelmed with frames, the sender must slow down. There must be ACK from the receiver to the sender.

Step 4 − In this protocol the sender sends one frame, stops until it receives confirmation from the receiver, and then sends the next frame.

Step 5 − We still have unidirectional communication for data frames, but auxiliary ACK frames travel from the other direction. We add flow control to the previous protocol.


Sender
Receiver
A
B
Request
Frame
Arrival
ACK
Arrival
Request
Frame
ACK
Arrival
Arrival
Time
Time

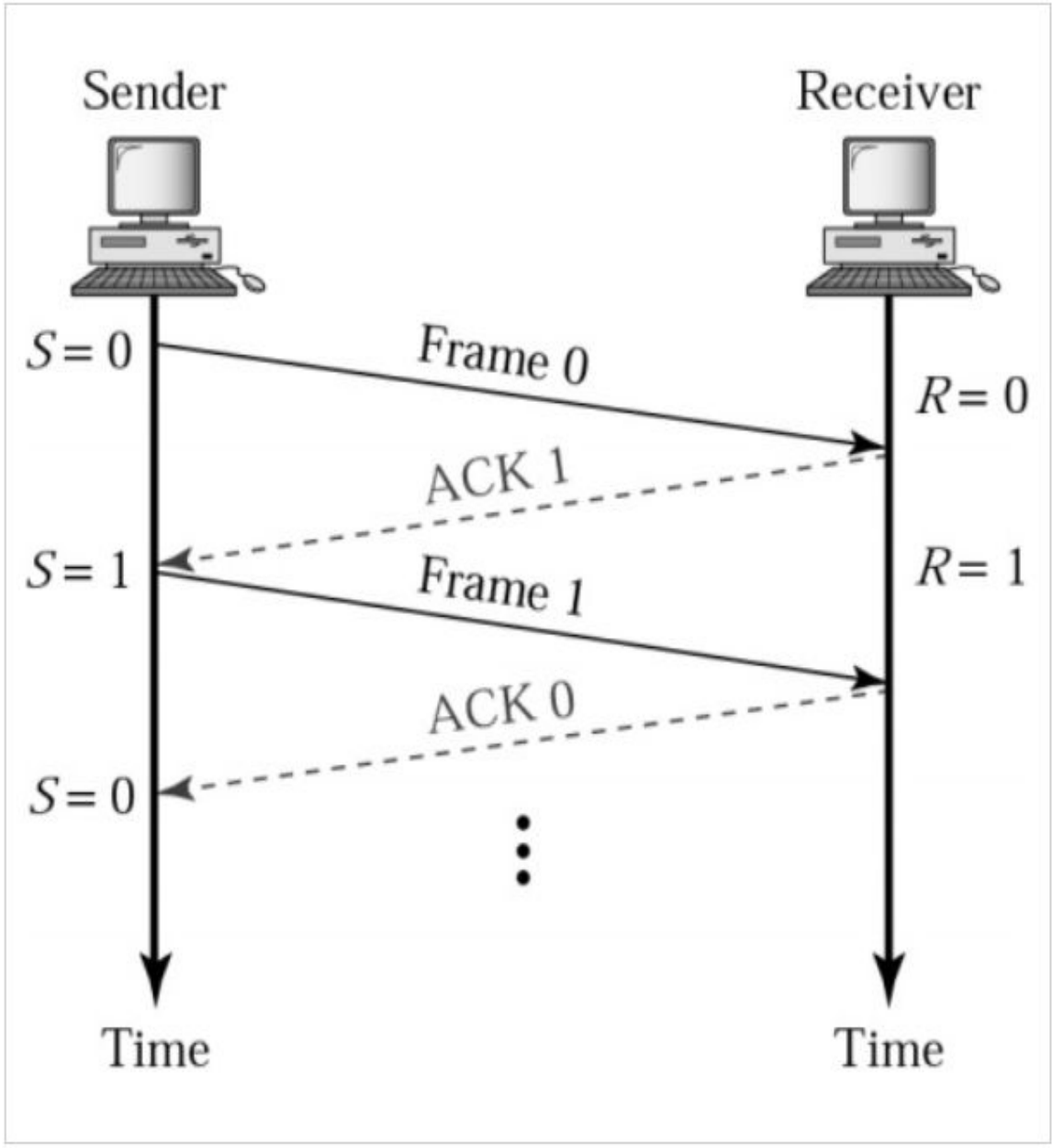
Stop & wait Automatic Repeat Request (ARQ)

Step 1 − In a noisy channel, if a frame is damaged during transmission, the receiver will detect with the help of the checksum.

Step 2 − If a damaged frame is received, it will be discarded, and the transmitter will retransmit the same frame after receiving a proper acknowledgement.

Step 3 − If the acknowledgement frame gets lost and the data link layer on 'A' eventually times out. Not having received an ACK, it assumes that its data frame was lost or damaged and sends the frame containing packet 1 again. This duplicate frame also arrives at the data link layer on 'B', thus part of the file will be duplicated and protocol is said to be failed.

Step 4 − To solve this problem, assign a sequence number in the header of the message.

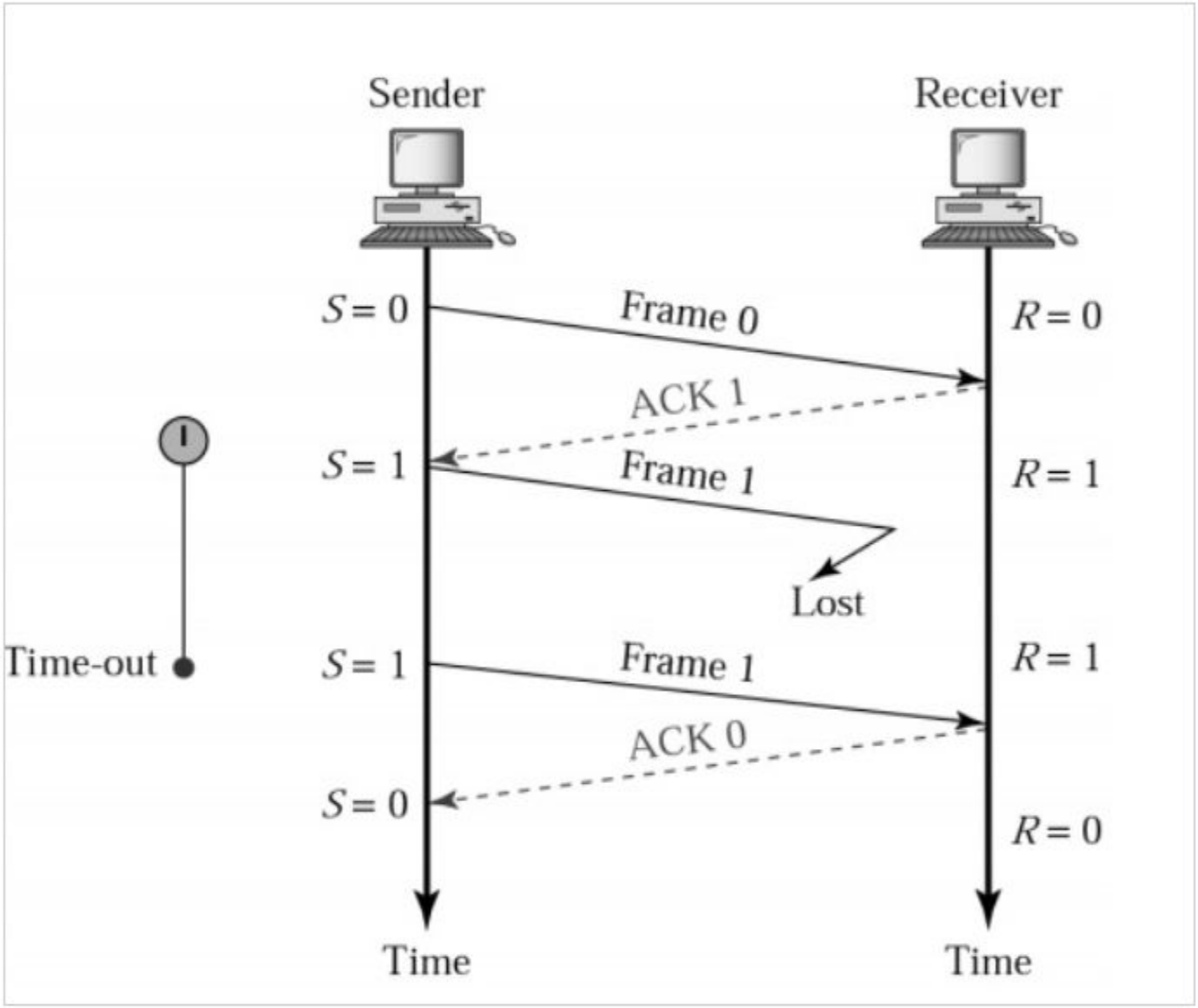

Sender
Receiver
S = 0
Frame 0
R = 0
ACK 1
S = 1
Frame 1
R = 1
ACK 0
S = 0
Time
Time

Step 5 − The receiver checks the sequence number to determine if the message is a duplicate since only the message is transmitted at any time.

Step 6 − The sending and receiving station needs only a 1-bit alternating sequence of '0' or '1' to maintain the relationship of the transmitted message and its ACK/ NAK.

Step 7 − A modulo-2 numbering scheme is used where the frames are alternatively labelled with '0' or '1' and positive acknowledgements are of the form ACK 0 and ACK 1.

Stop & Wait ARQ with Lost frame is as follows −



Go-Back-N ARQ

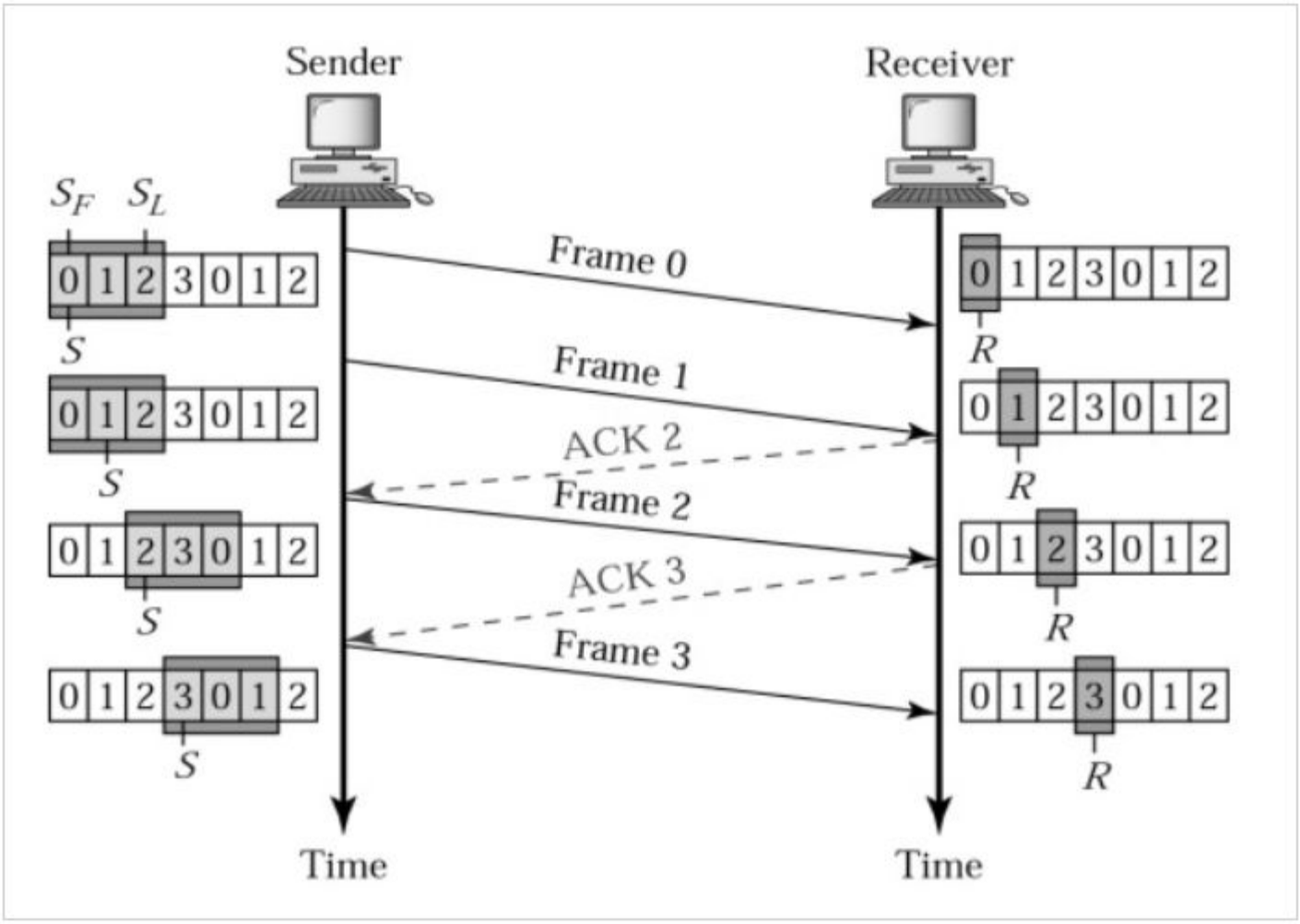
Step 1 − In this protocol we can send several frames before receiving acknowledgements.

Step 2 − we keep a copy of these frames until the acknowledgment arrives.

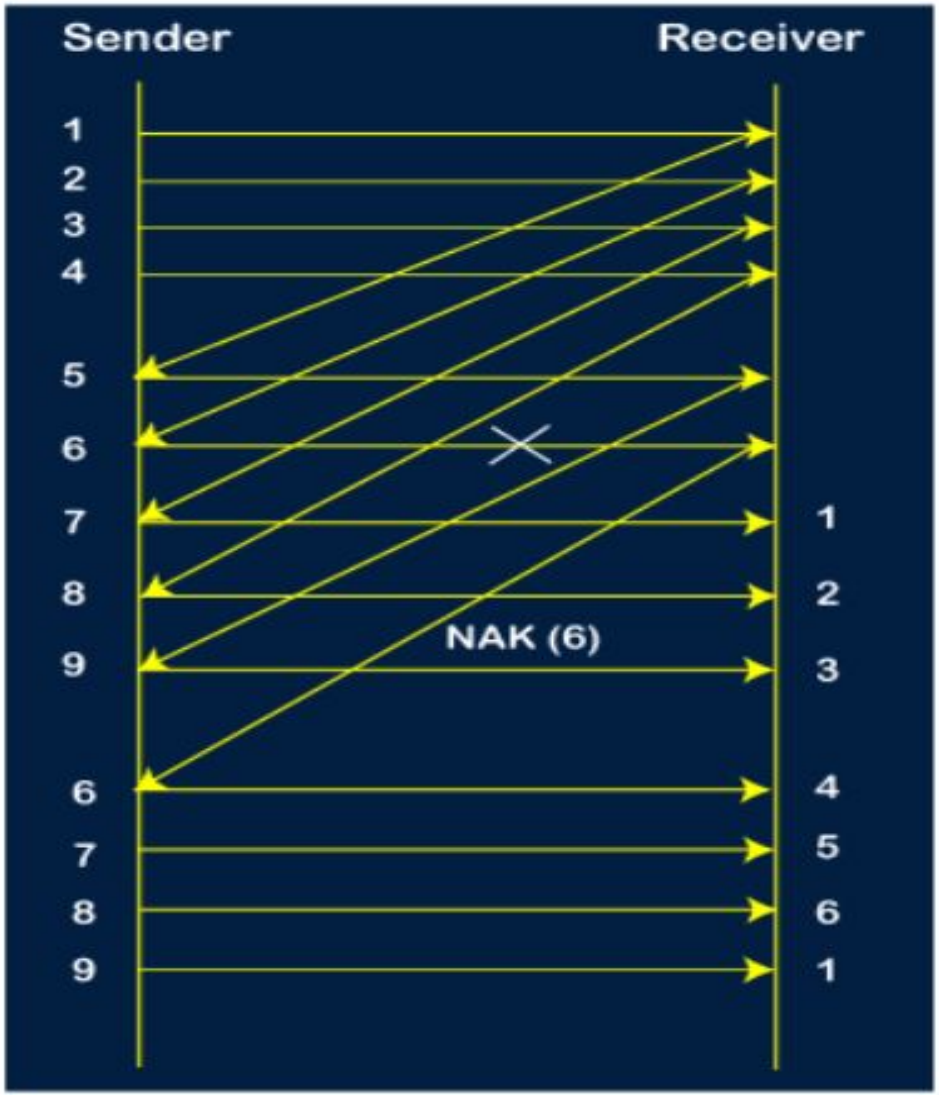
Step 3 − Frames from a sending station are numbered sequentially. However, we need to include the sequence number of each frame in the header; we need to set a limit.

Step 4 − If the header of the frame allows m bits for the sequence number, the sequence numbers range from 0 to 2^m-1. We can also repeat the sequence numbers.

For m = 2, the range of sequence numbers is: 0 to 3, i.e.
0,1,2,3, 0,1,2,3,…

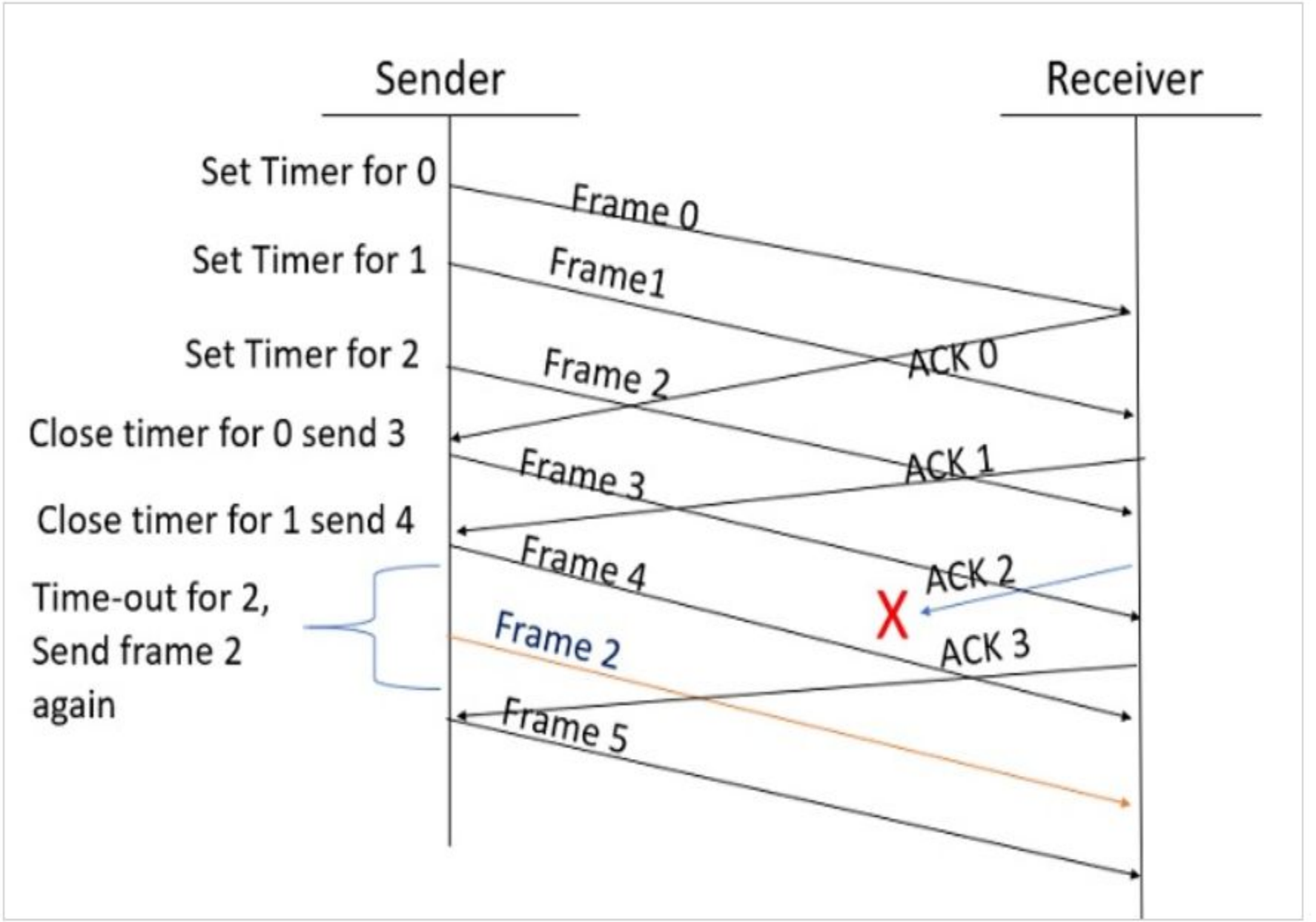


As it is Go-BACK, so it retransmits all the packets of the current window. It will resend 6, 7, 8, 9. The counter values of 6, 7, 8, 9 are 4, 5, 6, 1, respectively. In this case, the 8th packet is lost as it has a 6-counter value, so the counter variable will again be restarted from 1.



Selective Repeat ARQ

- It is also known as Sliding Window Protocol and used for error detection and control in the data link layer.
- In the selective repeat, the sender sends several frames specified by a window size even without the need to wait for individual acknowledgement from the receiver as in Go-Back-N ARQ. In selective repeat protocol, the retransmitted frame is received out of sequence.
- In Selective Repeat ARQ only the lost or error frames are retransmitted, whereas correct frames are received and buffered.
- The receiver while keeping track of sequence numbers buffers the frames in memory and sends NACK for only frames which are missing or damaged. The sender will send/retransmit a packet for which NACK is received.


Sender
Receiver
Set Timer for 0
Frame 0
Set Timer for 1
Frame1
Set Timer for 2
Frame 2
ACK 0
Close timer for 0 send 3
Frame 3
ACK 1
Close timer for 1 send 4
Frame 4
ACK 2
X
Time-out for 2,
Send frame 2
again
Frame 2
ACK 3
Frame 5

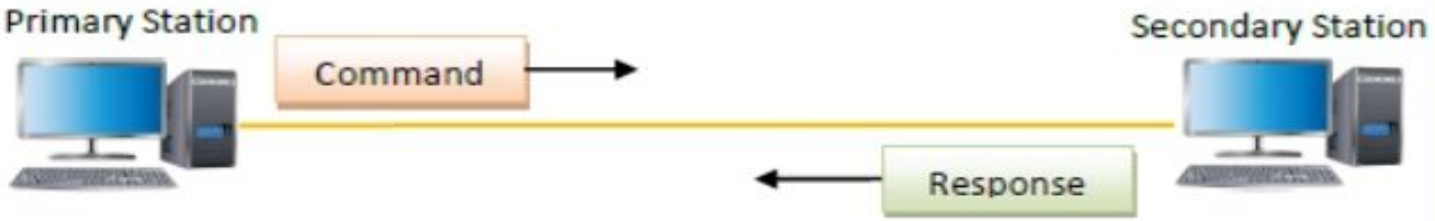
High-level Data Link Control (HDLC) Protocol

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.

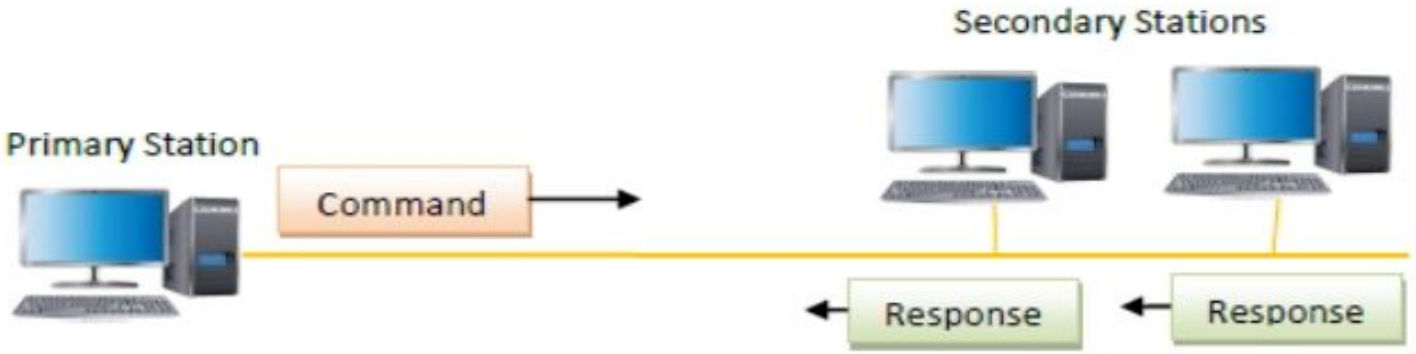
A frame is transmitted via the network to the destination that verifies its successful arrival.

It is a bit - oriented protocol that is applicable for both point - to - point and multipoint communications.

Normal Response Mode



Point – to – point communication

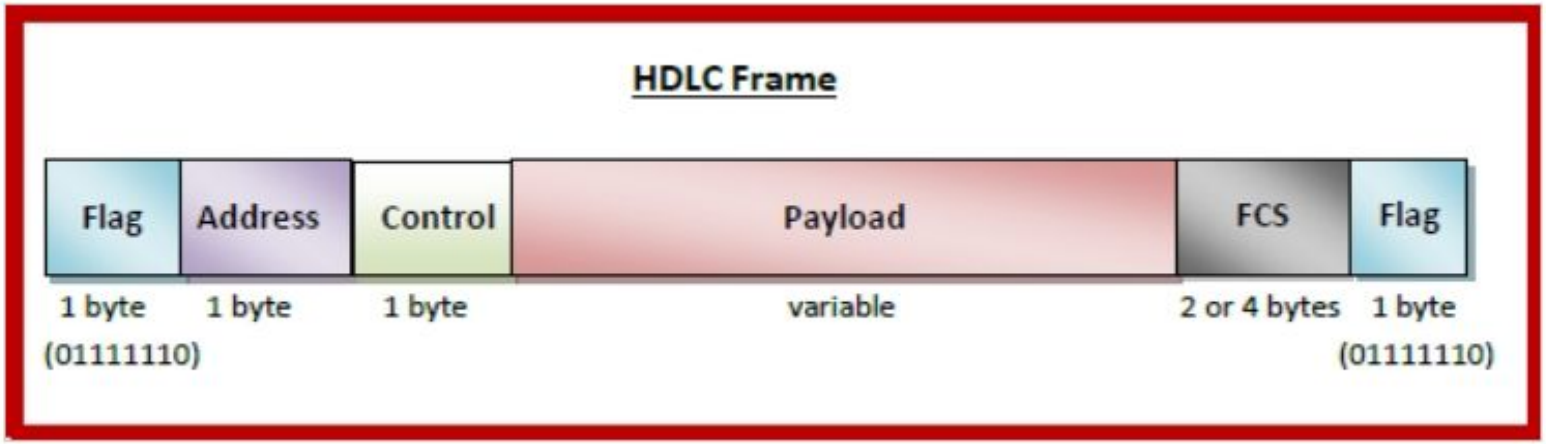


Multipoint communication

HDLC Frame

HDLC is a bit - oriented protocol where each frame contains up to six fields. The structure varies according to the type of frame. The fields of a HDLC frame are −

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.



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.

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

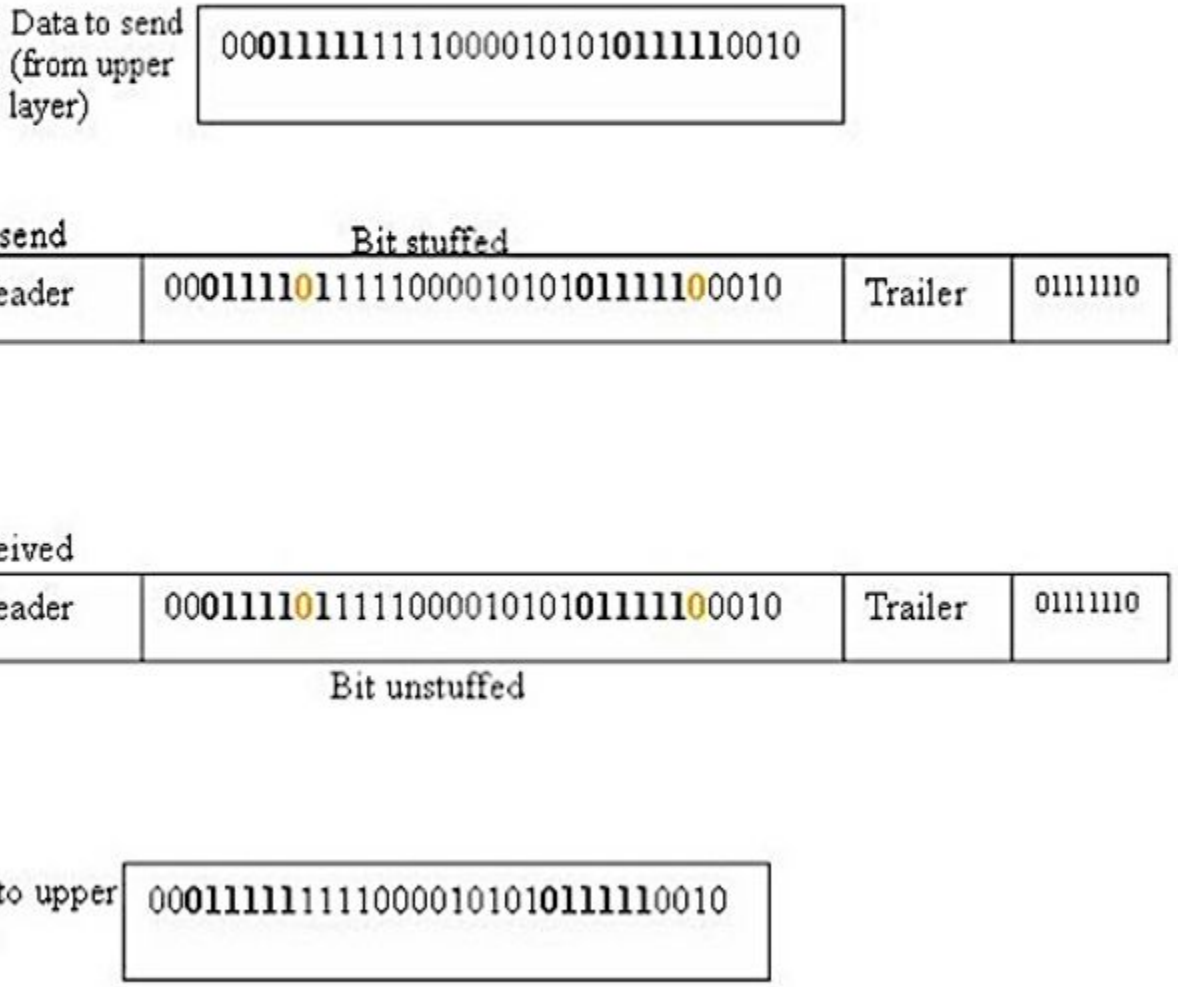
Payload − This carries the data from the network layer. Its length may vary from one network to another.

FCS − It is a 2 byte or 4 bytes frame check sequence for error detection. The standard code used is CRC (cyclic redundancy code).

Bit Stuffing

- In a bit-oriented protocol, the data to send is a series of bits.
- In order to distinguish frames, most protocols use a bit pattern of 8-bit length (01111110) as flag at the beginning and end of each frame.
- Here also cause the problem of appearance of flag in the data part to deal with this an extra bit added.
- This method is called bit stuffing.
- If a 0 and five successive 1 bits are encountered, an extra 0 is added.
- The receiver node removes the extra-added zero.

Bit Stuffing - Example



Point-to-Point Protocol (PPP)

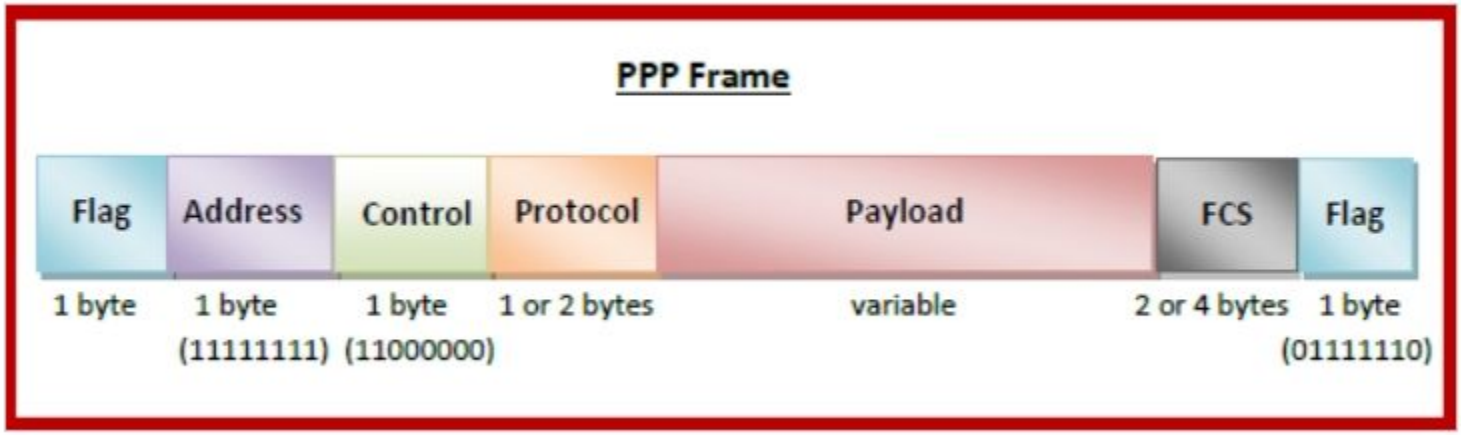
- Point - to - Point Protocol (PPP) is a communication protocol of the data link layer that is used to transmit multiprotocol data between two directly connected (point-to-point) computers.
- It is a **byte - oriented protocol** that is widely used in broadband communications having heavy loads and high speeds.

Services Provided by PPP

- Defining the frame format of the data to be transmitted.
- Defining the procedure of establishing link between two points and exchange of data.
- Stating authentication rules of the communicating devices.
- Providing address for network communication.
- Providing connections over multiple links.

PPP Frame

- **Flag** − 1 byte that marks the beginning and the end of the frame. The bit pattern of the flag is 01111110.
- **Address** − 1 byte which is set to 11111111 in case of broadcast.
- **Control** − 1 byte set to a constant value of 11000000.
- **Protocol** − 1 or 2 bytes that define the type of data contained in the payload field.



- **Payload** − This carries the data from the network layer. The maximum length of the payload field is 1500 bytes. However, this may be negotiated between the endpoints of communication.

- **FCS** − It is a 2 byte or 4 bytes frame check sequence for error detection. The standard code used is CRC (cyclic redundancy code).

Byte Stuffing

- Problem of resynchronization by having each frame start and end with special bytes.
- A flag byte is used to separate the frame as both the starting and ending delimiter.
- This technique is called *byte stuffing* or *character stuffing*.
- In this way, if the receiver ever loses synchronization, it can just search for the flag byte to find the end of the current frame.
- Two consecutive flag bytes indicate the end of one frame and start of the next one.
- To solve this problem is to have the sender's data link layer insert a special escape byte (ESC) just before each "accidental" flag byte in the data.
- The data link layer on the receiving end removes the escape byte before the data are given to the network layer.

Byte Stuffing - Example