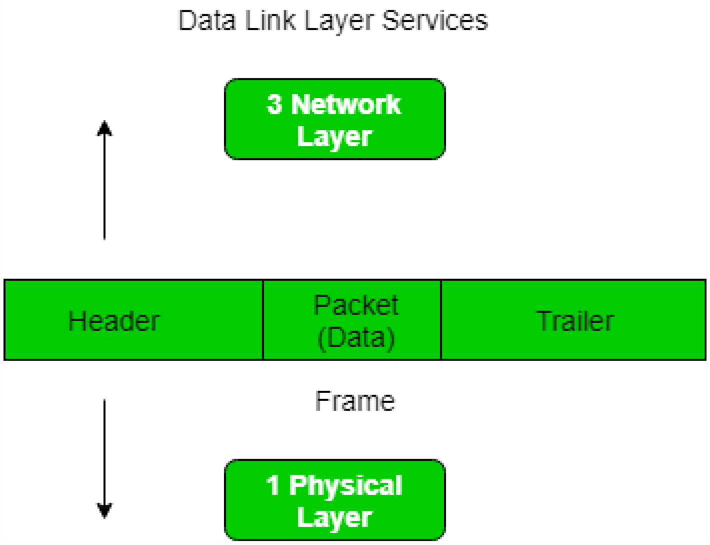
Computer Networks - Unit 2

Framing In Data Link Layer

Framing is a point-to-point connection between two computers or devices consists of a wire in which data is transmitted as a stream of bits. However, these bits must be framed into discernible blocks of information. Framing is a function of the data link layer. It provides a way for a sender to transmit a set of bits that are meaningful to the receiver. Ethernet, token ring, frame relay, and other data link layer technologies have their own frame structures. Frames have headers that contain information such as error-checking codes.



At data link layer, it extracts message from sender and provide it to receiver by providing sender's and receiver's address. The advantage of using frames is that data is broken up into recoverable chunks that can easily be checked for corruption.

Problems in Framing –

- **Detecting start of the frame:** When a frame is transmitted, every station must be able to detect it. Station detect frames by looking out for special sequence of bits that marks the beginning of the frame i.e. SFD (Starting Frame Delimeter).
- **How do station detect a frame:** Every station listen to link for SFD pattern through a sequential circuit. If SFD is detected, sequential circuit alerts station. Station checks destination address to accept or reject frame.
- **Detecting end of frame:** When to stop reading the frame.

Types of framing – There are two types of framing:

1. Fixed size – The frame is of fixed size and there is no need to provide boundaries to the frame, length of the frame itself acts as delimiter.

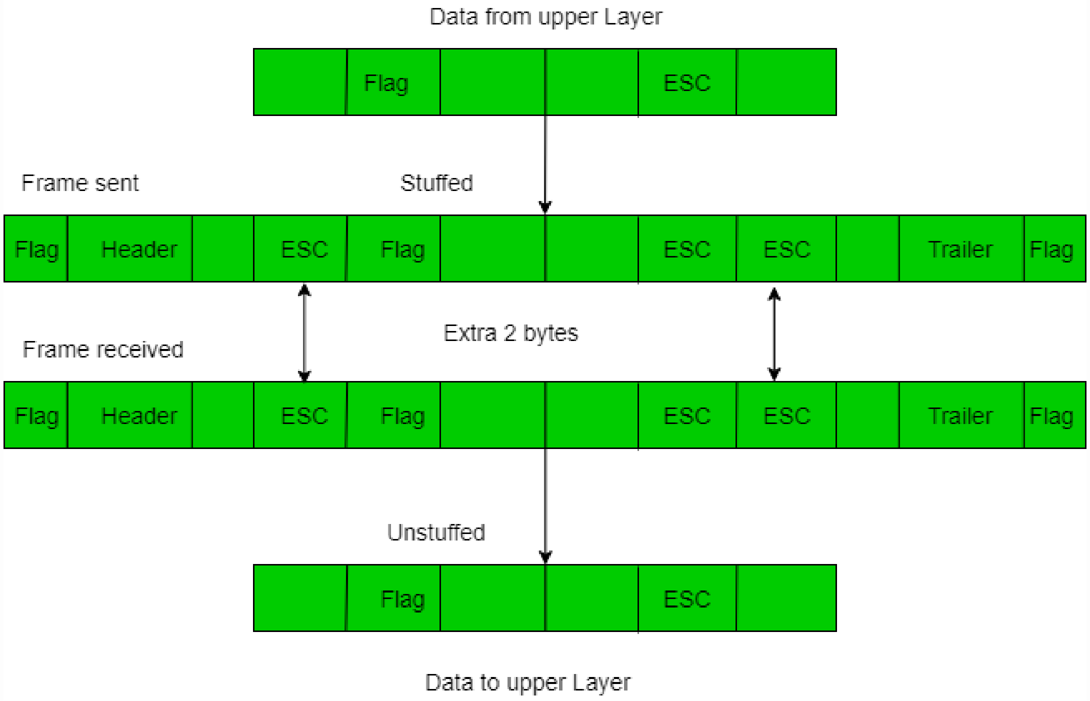
- **Drawback:** It suffers from internal fragmentation if data size is less than frame size
- **Solution:** Padding

2. Variable size – In this there is need to define end of frame as well as beginning of next frame to distinguish. This can be done in two ways:

1. **Length field –** We can introduce a length field in the frame to indicate the length of the frame. Used in **Ethernet(802.3)**. The problem with this is that sometimes the length field might get corrupted.

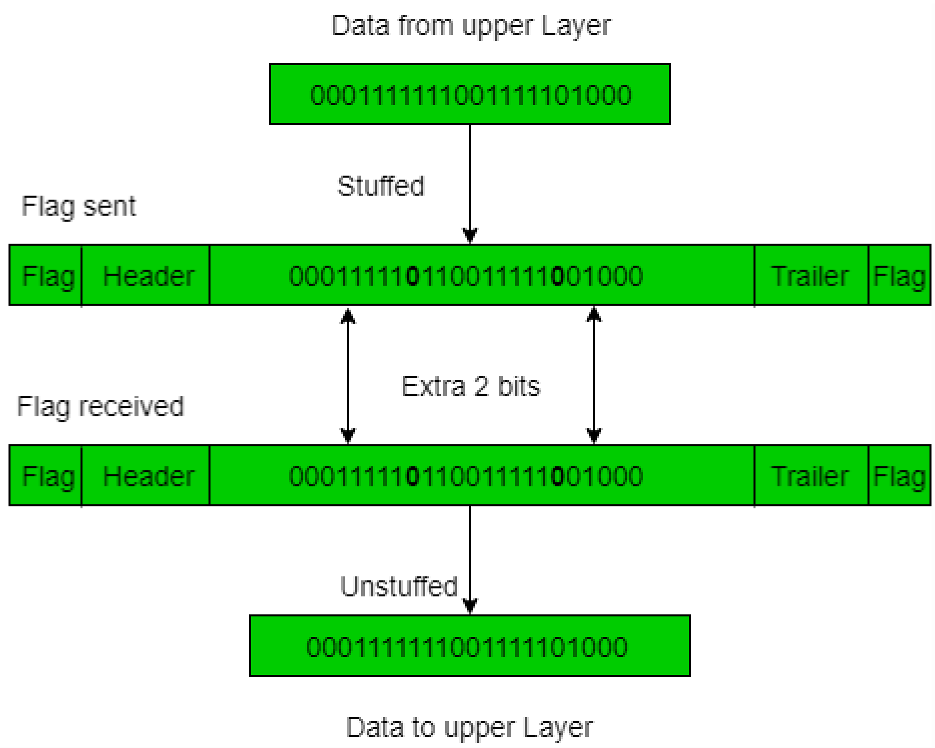
2. **End Delimeter (ED) –** We can introduce an ED(pattern) to indicate the end of the frame. Used in **Token Ring**. The problem with this is that ED can occur in the data. This can be solved by:

1. Character/Byte Stuffing: Used when frames consist of character. If data contains ED then, byte is stuffed into data to diffentiate it from ED.
Let ED = “$” –> if data contains ‘$’ anywhere, it can be escaped using ‘O’ character.
–> if data contains ‘O$’ then, use ‘OOO$'($ is escaped using O and O is escaped using O).



Disadvantage – It is very costly and obsolete method.

2. Bit Stuffing: Let ED = 01111 and if data = 01111
–> Sender stuffs a bit to break the pattern i.e. here appends a 0 in data = 0111**0**1.
–> Receiver receives the frame.
–> If data contains 011101, receiver removes the 0 and reads the data.



Examples –

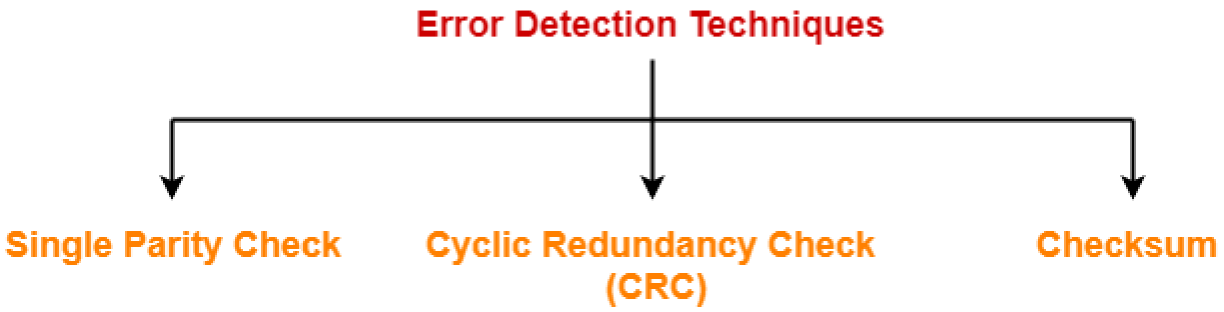
- If Data –> 011100011110 and ED –> 01111 then, find data after bit stuffing ?
  –> 0111**0**000111**0**10
- If Data –> 110001001 and ED –> 1000 then, find data after bit stuffing ?
  –> 1100**1**0100**1**1

Error Detection:

Error Detection in Computer Networks-

Error detection is a technique that is used to check if any error occurred in the data during the transmission.

Some popular error detection methods are-



1. Single Parity Check
2. Cyclic Redundancy Check (CRC)
3. Checksum

Checksum-

Checksum is an error detection method.

Error detection using checksum method involves the following steps-

Step-01:

At sender side,

- If m bit checksum is used, the data unit to be transmitted is divided into segments of m bits.
- All the m bit segments are added.
- The result of the sum is then complemented using 1’s complement arithmetic.
- The value so obtained is called as **checksum**.

Step-02:

- The data along with the checksum value is transmitted to the receiver.

Step-03:

At receiver side,

- If m bit checksum is being used, the received data unit is divided into segments of m bits.
- All the m bit segments are added along with the checksum value.
- The value so obtained is complemented and the result is checked.

Then, following two cases are possible-

Case-01: Result = 0

If the result is zero,

- Receiver assumes that no error occurred in the data during the transmission.
- Receiver accepts the data.

Case-02: Result ≠ 0

If the result is non-zero,

- Receiver assumes that error occurred in the data during the transmission.
- Receiver discards the data and asks the sender for retransmission.

Checksum Example-

Consider the data unit to be transmitted is-

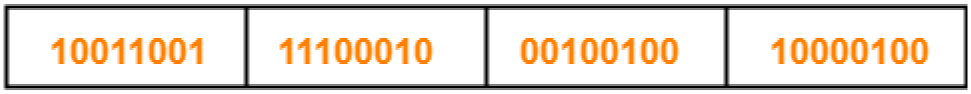
10011001111000100010010010000100

Consider 8 bit checksum is used.

Step-01:

At sender side,

The given data unit is divided into segments of 8 bits as-



10011001	11100010	00100100	10000100

Now, all the segments are added and the result is obtained as-

- 10011001 + 11100010 + 00100100 + 10000100 = 1000100011
- Since the result consists of 10 bits, so extra 2 bits are wrapped around.
- 00100011 + 10 = 00100101 (8 bits)
- Now, 1’s complement is taken which is 11011010.
- Thus, checksum value = 11011010

Step-02:

- The data along with the checksum value is transmitted to the receiver.

Step-03:

At receiver side,

- The received data unit is divided into segments of 8 bits.
- All the segments along with the checksum value are added.
- Sum of all segments + Checksum value = 00100101 + 11011010 = 11111111
- Complemented value = 00000000
- Since the result is 0, receiver assumes no error occurred in the data and therefore accepts it.

Important Notes-

Note-01:

- Consider while adding the m bit segments, the result obtained consists of more than m bits.
- Then, wrap around the extra bits and add to the result so that checksum value consists of m bits.

Note-02:

- While calculating the checksum, if checksum value is needed, then assume it to be zero.
- After calculating the checksum value, substitute the checksum value in the checksum field.
- This will be required during checksum calculation of **IP Header**, **TCP Header** and **UDP Header**.

Note-03:

- The checksum is used in the internet by several protocols although not at the data link layer.

PRACTICE PROBLEM BASED ON CHECKSUM ERROR DETECTION METHOD-

Problem-

Checksum value of 1001001110010011 and 1001100001001101 of 16 bit segment is-

1. 1010101000011111
2. 1011111000100101
3. 1101010000011110
4. 1101010000111111

Solution-

We apply the above discussed algorithm to calculate the checksum.

- 1001001110010011 + 1001100001001101 = 10010101111100000
- Since, the result consists of 17 bits, so 1 bit is wrapped around and added to the result.
- 0010101111100000 + 1 = 0010101111100001
- Now, result consists of 16 bits.
- Now, 1's complement is taken which is 1101010000011110
- Thus, checksum value = 1101010000011110

Thus, Option (C) is correct.

CRC (Cyclic Redundancy Check) :

- Cyclic Redundancy Check (CRC) is an error detection method.
- It is based on binary division.

CRC Generator-

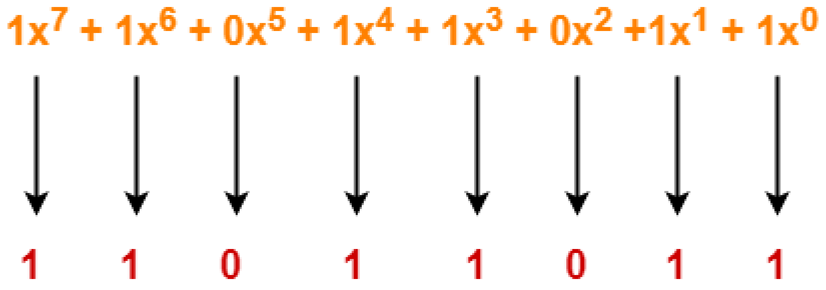
- CRC generator is an algebraic polynomial represented as a bit pattern.
- Bit pattern is obtained from the CRC generator using the following rule-

> The power of each term gives the position of the bit and the coefficient gives the value of the bit.

Example-

Consider the CRC generator is $x^7 + x^6 + x^4 + x^3 + x + 1$.

The corresponding binary pattern is obtained as-



Thus, for the given CRC generator, the corresponding binary pattern is 11011011.

Properties Of CRC Generator-

The algebraic polynomial chosen as a CRC generator should have at least the following properties-

Rule-01:

- It should not be divisible by x.
- This condition guarantees that all the burst errors of length equal to the length of polynomial are detected.

Rule-02:

- It should be divisible by x+1.
- This condition guarantees that all the burst errors affecting an odd number of bits are detected.

Important Notes-

If the CRC generator is chosen according to the above rules, then-

- CRC can detect all single-bit errors

- CRC can detect all double-bit errors provided the divisor contains at least three logic 1’s.
- CRC can detect any odd number of errors provided the divisor is a factor of x+1.
- CRC can detect all burst error of length less than the degree of the polynomial.
- CRC can detect most of the larger burst errors with a high probability.

Steps Involved-

Error detection using CRC technique involves the following steps-

Step-01: Calculation Of CRC At Sender Side-

At sender side,

- A string of n 0’s is appended to the data unit to be transmitted.
- Here, n is one less than the number of bits in CRC generator.
- Binary division is performed of the resultant string with the CRC generator.
- After division, the remainder so obtained is called as **CRC**.
- It may be noted that CRC also consists of n bits.

Step-02: Appending CRC To Data Unit-

At sender side,

- The CRC is obtained after the binary division.
- The string of n 0’s appended to the data unit earlier is replaced by the CRC remainder.

Step-03: Transmission To Receiver-

- The newly formed code word (Original data + CRC) is transmitted to the receiver.

Step-04: Checking at Receiver Side-

At receiver side,

- The transmitted code word is received.
- The received code word is divided with the same CRC generator.
- On division, the remainder so obtained is checked.

The following two cases are possible-

Case-01: Remainder = 0

If the remainder is zero,

- Receiver assumes that no error occurred in the data during the transmission.
- Receiver accepts the data.

Case-02: Remainder ≠ 0

If the remainder is non-zero,

- Receiver assumes that some error occurred in the data during the transmission.
- Receiver rejects the data and asks the sender for retransmission.

PRACTICE PROBLEMS BASED ON CYCLIC REDUNDANCY CHECK (CRC)-

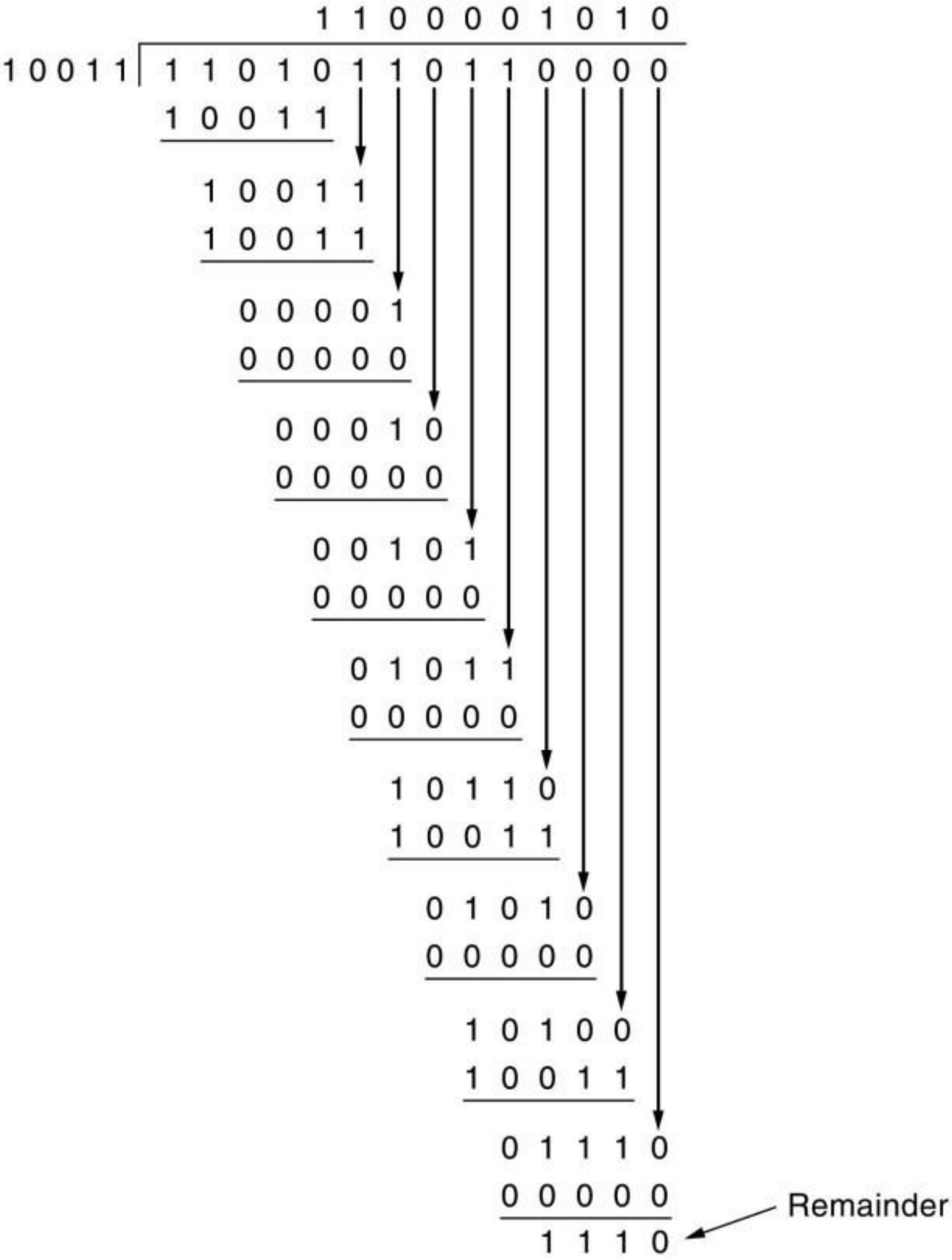
Problem-01:

A bit stream 1101011011 is transmitted using the standard CRC method. The generator polynomial is x^4+x+1. What is the actual bit string transmitted?

Solution-

- The generator polynomial $G(x) = x^4 + x + 1$ is encoded as 10011.
- Clearly, the generator polynomial consists of 5 bits.
- So, a string of 4 zeroes is appended to the bit stream to be transmitted.
- The resulting bit stream is 1101011011**0000**.

Now, the binary division is performed as-


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

From here, CRC = 1110.

Now,

- The code word to be transmitted is obtained by replacing the last 4 zeroes of 1101011011**0000** with the CRC.
- Thus, the code word transmitted to the receiver = 1101011011**1110**.

Problem-02:

A bit stream 10011101 is transmitted using the standard CRC method. The generator polynomial is x^3+1.

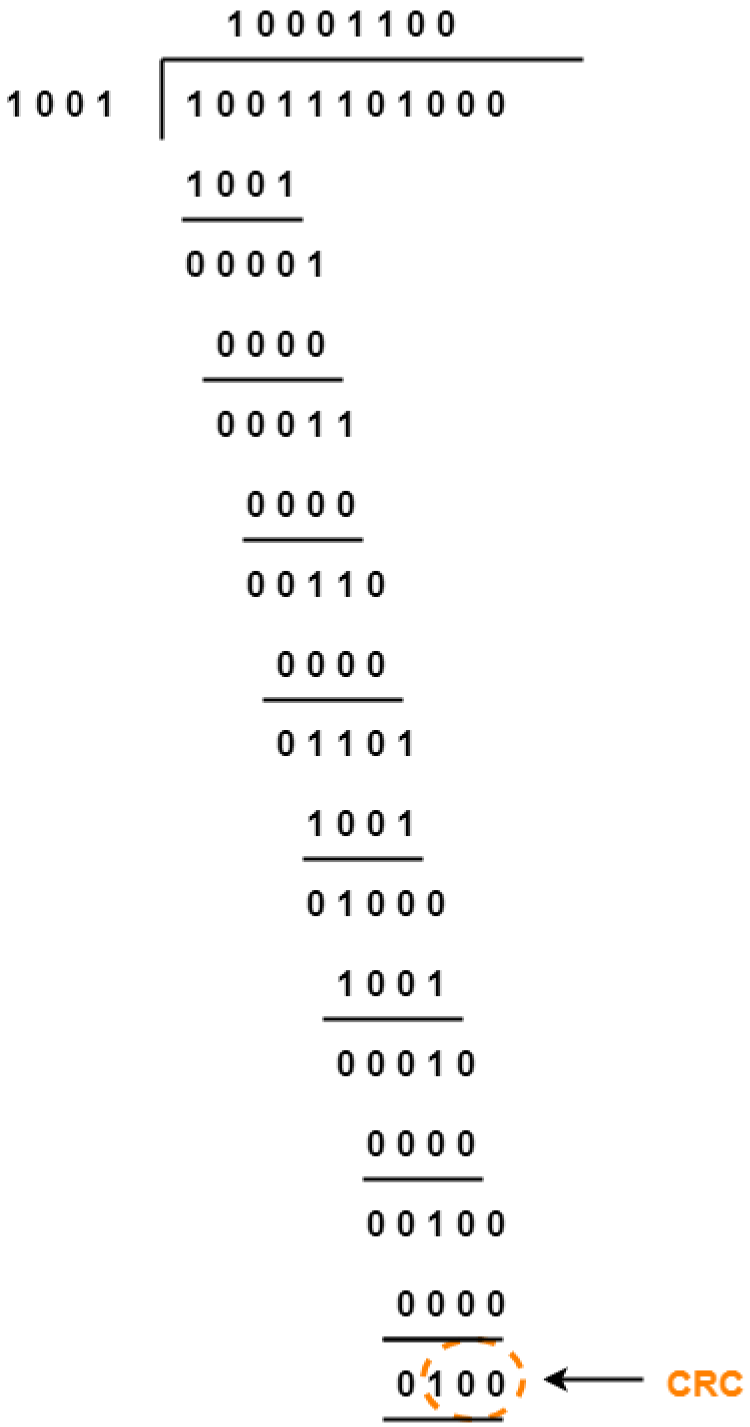
1. What is the actual bit string transmitted?
2. Suppose the third bit from the left is inverted during transmission. How will receiver detect this error?

Solution-

Part-01:

- The generator polynomial G(x) = x^3 + 1 is encoded as 1001.
- Clearly, the generator polynomial consists of 4 bits.
- So, a string of 3 zeroes is appended to the bit stream to be transmitted.
- The resulting bit stream is 10011101**000**.

Now, the binary division is performed as-



From here, CRC = 100.

Now,

- The code word to be transmitted is obtained by replacing the last 3 zeroes of 10011101**000** with the CRC.

- Thus, the code word transmitted to the receiver = 10011101**100**.

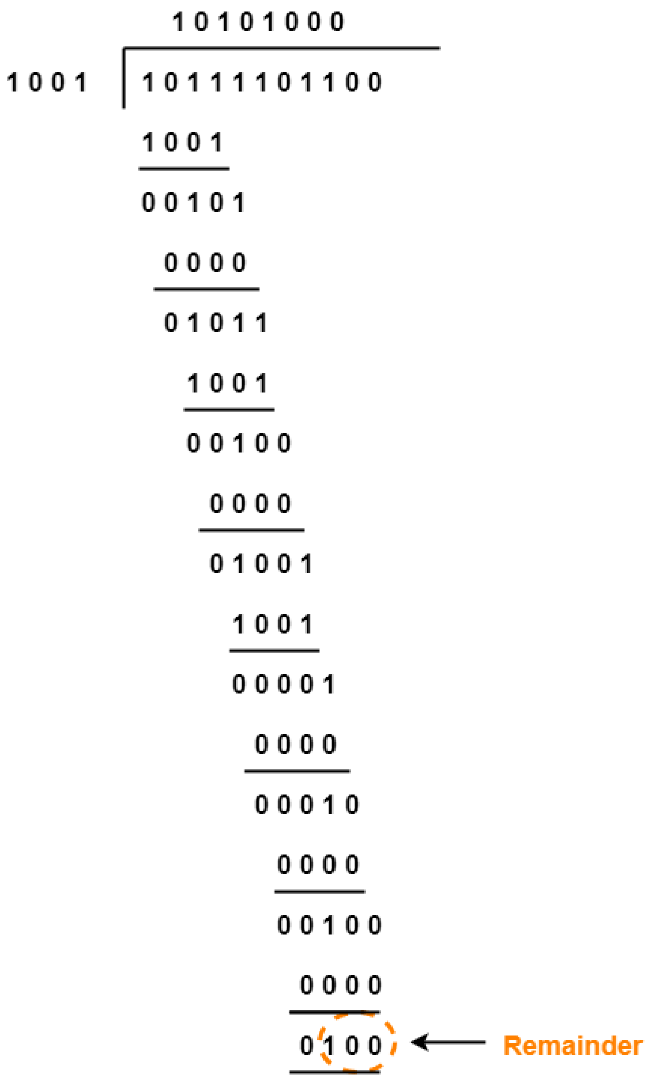
Part-02:

According to the question,

- Third bit from the left gets inverted during transmission.
- So, the bit stream received by the receiver = 10111101100.

Now,

- Receiver receives the bit stream = 10111101100.
- Receiver performs the binary division with the same generator polynomial as-



```
                 10101000
          ______________
1001  |  10111101100
         1001
         ----
         00101
          0000
          -----
          01011
           1001
           ----
           00100
            0000
            -----
            01001
             1001
             ----
             00001
              0000
              -----
              00010
               0000
               -----
               00100
                0000
                ----
                0100   <-- Remainder
                ----
```

From here,

- The remainder obtained on division is a non-zero value.
- This indicates to the receiver that an error occurred in the data during the transmission.
- Therefore, receiver rejects the data and asks the sender for retransmission.

Sliding Window Protocols:

What are elementary Data Link Layer protocols? Explain Sliding Window Protocols in detail?

Answer:- Elementary Data Link Layer Protocol:- Data Link Layer protocols are responsible for error free node to node delivery of the frames. Following are the three main Data Link Layer Protocols.

Stop & Wait Protocol:- In this method acknowledgement is expected after every frame transmission.
Acknowledgement is necessary for previous frame before starting transmission of next frame.
This is reliable protocol but the problem is of delay as acknowledgements will consume half of the bandwidth.

Sliding-Window Protocol:- Sliding Windows are the most important data link layer protocols. In this method. A trade-off is made between delay and reliability.
Acknowledgements are requested after transmission of a number of frames not after every frame.
The number of frames after which acknowledgements are needed refers to the Window size.
The Window size for any communication can be found using following formula.

e.g. If there are 8 frame in a Packet. Then Window size will be 3.

Types of Sliding Window protocols :- Sliding Window protocols may be categories into two categories:-

a) Go Back N- Sliding Window protocol:- In this Sliding window protocol, if error is occurred in nth frame, all
remaining frames are ignored and from n-frame onward re-transmission is required.
Error was occurred in 3rd Frame thus re-transmission is required from 3rd onward frame.

b) Selective Repeat-Sliding Window protocol:- In this Sliding window protocol, if error is occurred in nth frame, remaining frames are continued to transmit and only nth-frame re-transmission is required at last of that frame series.

Data link layer and channel allocation

What are two sub-layers of Data Link Layer? What are various channel allocation schemes?

Answer: Data Link Layer: Data link layer is the second layer i.e. next to Physical Layer. Data link layer is responsible to accept packets from Network layer and to divide into frames, ensure error control during the transmission of frames flow control and contention resolution in case multiple stations are competing for single channel.

Tasks of Data Link Layer are divided into categories - first category called channel allocation is the responsibility of the first sub layer named as MAC (Medium Access Control)-sub layer of data link layer. LAN topologies are basically bus topology configured differently. LANs usually have multipoint line configuration. When packets are deployed on a LAN, the entire capacity of the line is usually used for that transmission. Because of this, some form of traffic regulation must take place for the nodes attached to a LAN. This sub-layer is responsible to ensure effective channel allocation to the stations if multiple stations are requesting same channel.

Second sub layer called LLC(Logical Link Control)-sub layer is responsible for all other duties of the Data Link Layer viz. Framing, accepting packets from upper layer, error control etc.

Various Channel Allocation Schemes: Channel Allocation schemes can be categories into two categories:-

a) **Static Channel Allocation Schemes** : In this category a fixed dedicated portion of the bandwidth or time or wavelength etc is assigned to a particular station.

Following are the examples of the Static Channel Allocation schemes:-

1) **Time Division Multiplexing** - A multiplexing technique by which multiple data signals can be transmitted over a common communication channel in different time slots is known as **Time Division Multiplexing (TDM)**.

It allows the division of the overall time domain into various fixed length time slots. A single frame is said to be transmitted when it's all signal components gets transmitted over the channel.

2) **Frequency Division Multiplexing. -** In frequency-division multiplexing (FDM), multiple signals are combined for transmission on a single communications line or channel, with each signal assigned to a different frequency (subchannel) within the main channel.

To accommodate the successful transmission of multiple signals over a single line, FDM separates assigned bands by strips of unused frequencies called *guard bands*. This prevents overlapping between signal frequencies over a shared medium.

A signal is generated and modulated by a sending device and is carried over the separated bands. The modulated signals are combined using a multiplexer (mux) and transmitted over the communication channel. At the receiving end, the combined signal goes through a demultiplexer (demux) to extract the individual signals.

b) **Dynamic Channel Allocation Schemes** :- Static channel allocation schemes can work only for a small number of
the stations, In case number of stations is large, dedicated allocation of the resources is not effective. Thus
dynamic channel allocation schemes are the only option in cases when a large number of stations are
competing for the single channel.
Following are the examples of Dynamic Channel Allocation Schemes:-
1) ALOHA 2) Slotted ALOHA 3) CSMA 4)CSMA/CD

What is random access protocol?

Random access protocol is one of the three types of multiple access protocol. The multiple access protocol offers channel utilization to various users. It operates in the MAC layer of the OSI model.

The word **random** corresponds to anything which is not time or sequence-specific. From random access protocol, we mean that anytime multiple users can access the channel according to their requirement as each individual is equally prioritized. However, its simplicity has some drawbacks which we will discuss later.

Introduction to ALOHA

ALOHA protocol offers a simple communication scheme where each individual station of the network may transmit the data whenever it wishes to do so. It is one of the important assets of satellite communication as basically satellite communication allows the transmission of the information signal from earth station towards the satellite in space i.e., uplink transmission.

Also, the satellite retransmits the signal back to the earth-based station or stations i.e., downlink.

To understand this, consider the figure below where a central computer forms a connection with 3 individual remote stations:

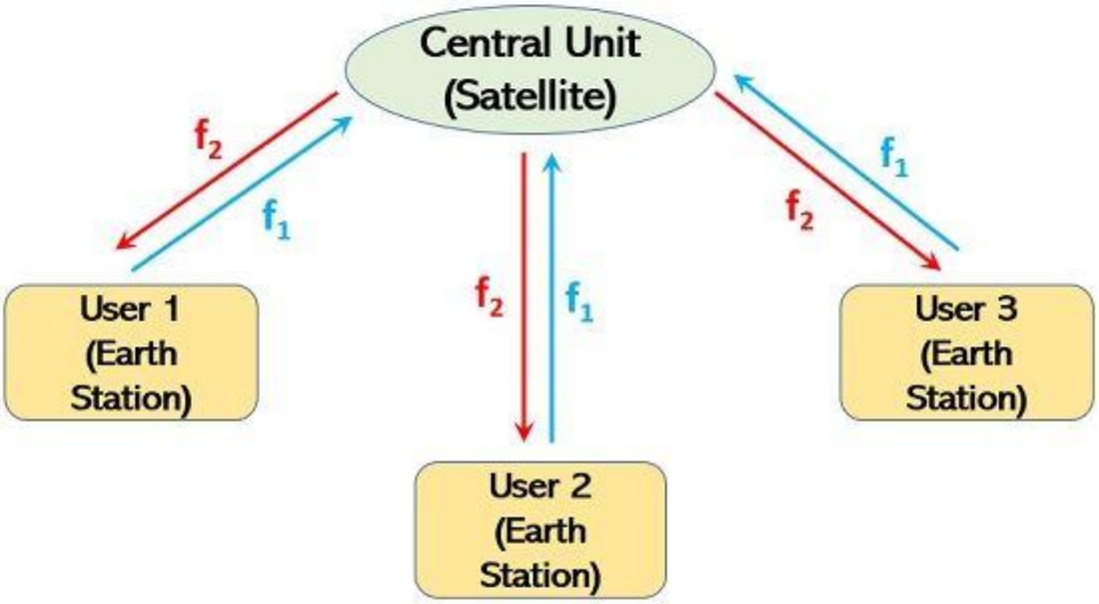


Illustration of simplified ALOHAnet

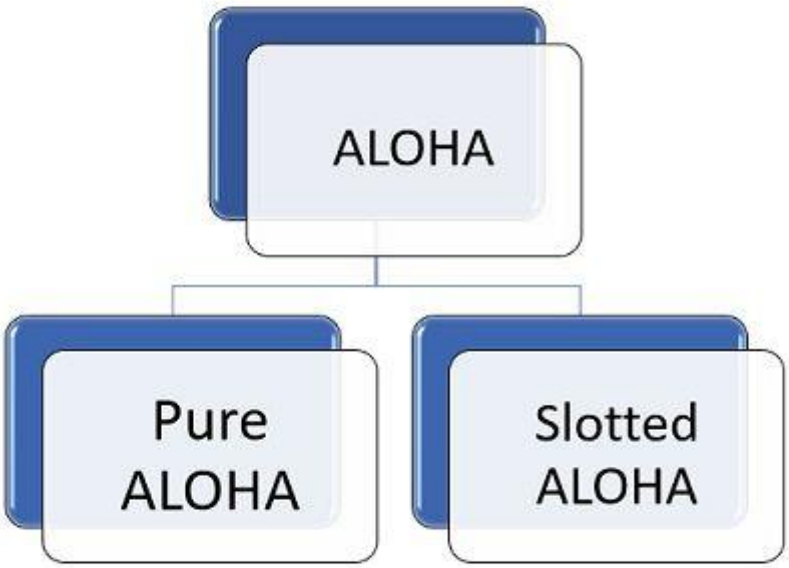
Here all 3 users operate in an uncoordinated manner. This means that at the time of transmission no station is aware of the transmission of data from other stations within the network. So, various uncoordinated users try to share the available resources in order to send the data to the central unit.

Thus, is called a **contention scheme** because a form of disagreement occurs between the stations whose data will be transmitted through the channel among all of them.

ALOHA is one of the protocols or methods associated with the utilization of the medium for the transmission of data frames from earth stations to satellite or from satellite to earth stations. It is a contention scheme where no centralized control exists and all the stations (nodes) are equally prioritized. This scheme shows suitability towards bursty traffic.

Types of ALOHA Protocol

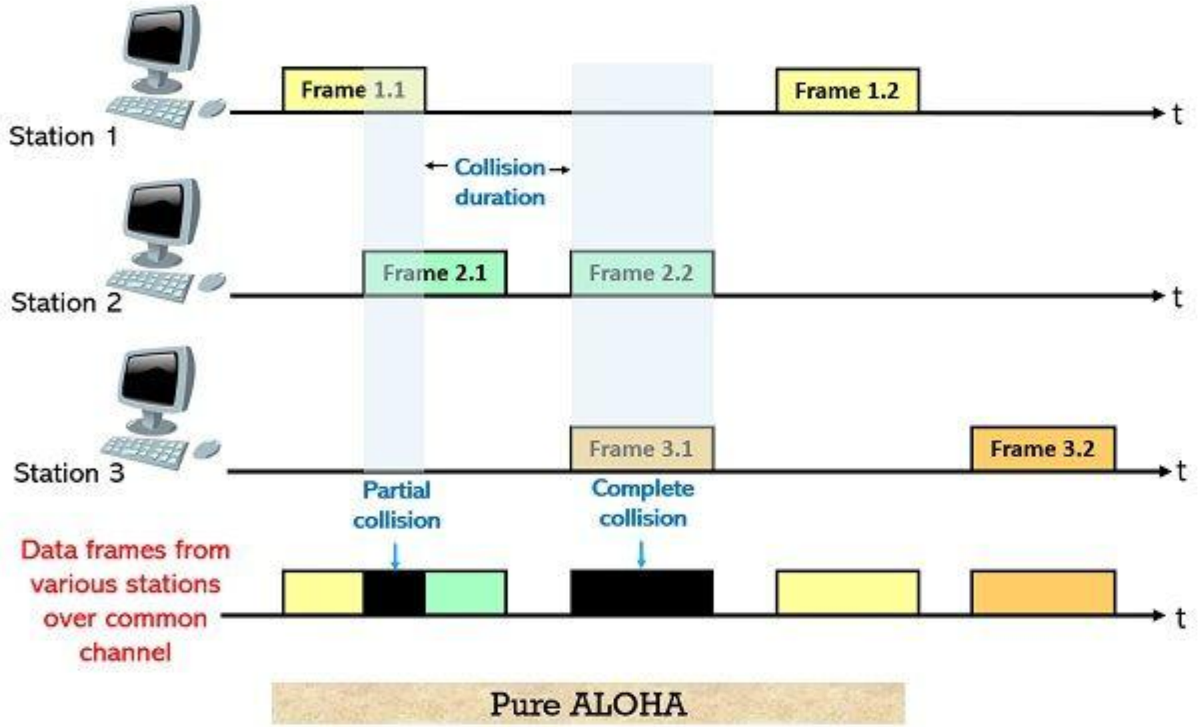
ALOHA protocol is a type of random access protocol that itself is a subclassification of medium access protocol. According to the implementation of the protocol, ALOHA is classified into 2 categories.



Let us understand each type in detail:

Pure ALOHA

Pure ALOHA is the basic type of ALOHA contention scheme in which the data frames from multiple VSATs are transmitted towards the satellite via a common channel according to the demand. Here due to the bursty nature of traffic within a network, following pure ALOHA protocol, chances of collision of data frames are huge.
This is so because none of the stations is concerned about whether any other station is transmitting at that particular instant or not. So, multiple data packets when transmitted through a common channel undergo collision. The packet collision during transmission over a common channel is shown below:

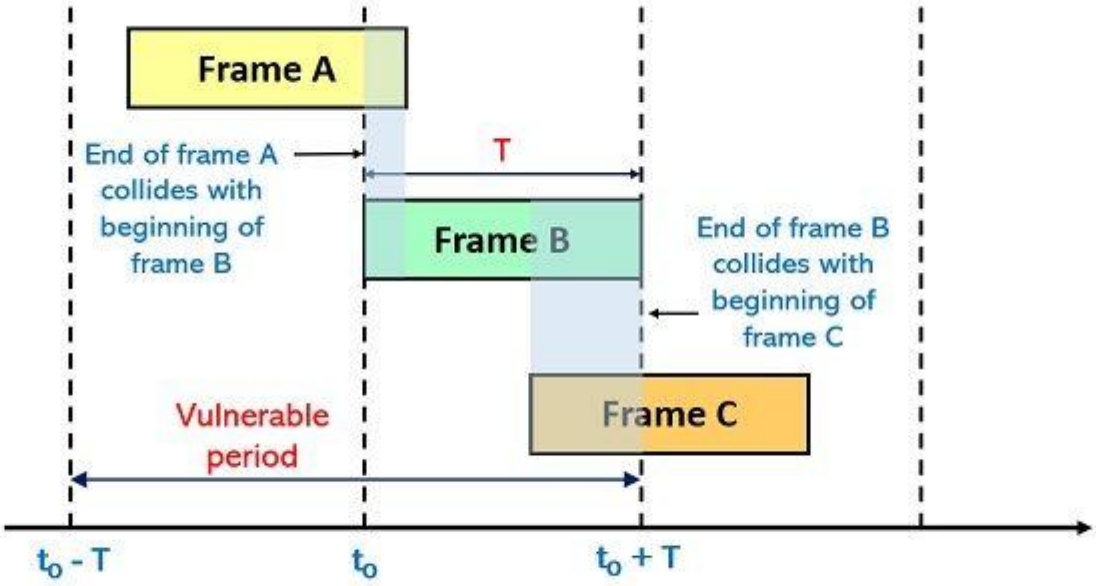


Here it is shown that data packets from three separate stations are sent over a common channel randomly. So, those data frames that are present at the same time instant over the channel will collide and be lost in the path instead of reaching the destination. It does not matter whether the collision is complete or partial because both cases will result in loss of data.

In this type, after transmitting the data, the stations wait for the acknowledgment from the satellite. However, if acknowledgment is not received within a fixed time period, then stations wait for a random amount of time. This random time duration is called **back-off time**. After completion of back-off time, the stations resend data.
The random nature of back-off time helps to reduce chances of further collision because obviously if multiple stations (who have not received acknowledgment) will retransmit the data after a fixed time period, then this will lead to data collision again.

<u>Vulnerable period in pure ALOHA</u>

Suppose in the figure shown below three individual data frames are sent over the channel resulting in a partial collision.


Frame A
End of frame A collides with beginning of frame B
T
Frame B
End of frame B collides with beginning of frame C
Frame C
Vulnerable period
$t_0 - T$
t_0
$t_0 + T$

Vulnerable time for Pure ALOHA

Here each packet is of uniform length with fixed time duration T. During transmission of a data frame A with duration T, if another data frame B is sent then even overlapping of first and last bits of the two frames will result in a collision. Hence both the frames will be completely destroyed.

Thus, once a packet is sent, the next packet will be sent by taking a margin of duration T along with the time duration T of the transmitted packet. Meaning, T before the frame to T once the frame begins. Hence, the vulnerable period will be 2T. This helps in determining channel utilization, given as throughput S

$$\mathbf{S = Ge^{-2G}}$$

: G is the total number of stations transmitting at the same instant.

For maximum throughput, G must be ½. Therefore,

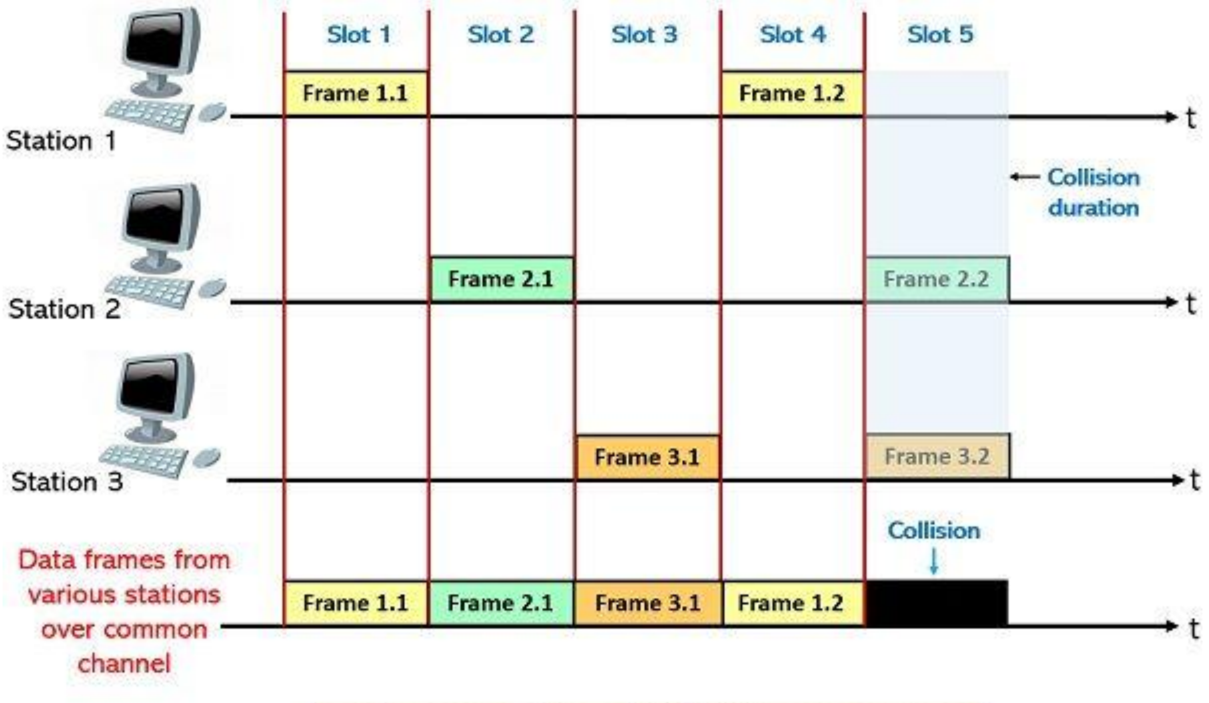
$$\mathbf{S = 0.184}$$

So, maximum efficiency will be **18.4%**.

Slotted ALOHA

It is the advanced version of pure aloha and came into existence to increase the efficiency of the former. The reason is that pure aloha has more chances to undergo collision. We have seen that in pure aloha simultaneous transmission of multiple data frames over the channel cause collision and loss of data frames.

In the slotted ALOHA scheme, multiple **slots** of discrete-time intervals are formed within the complete common channel. Thus, slotted aloha is a combinational approach where pure aloha is implemented with the slotted channel.



Slotted ALOHA

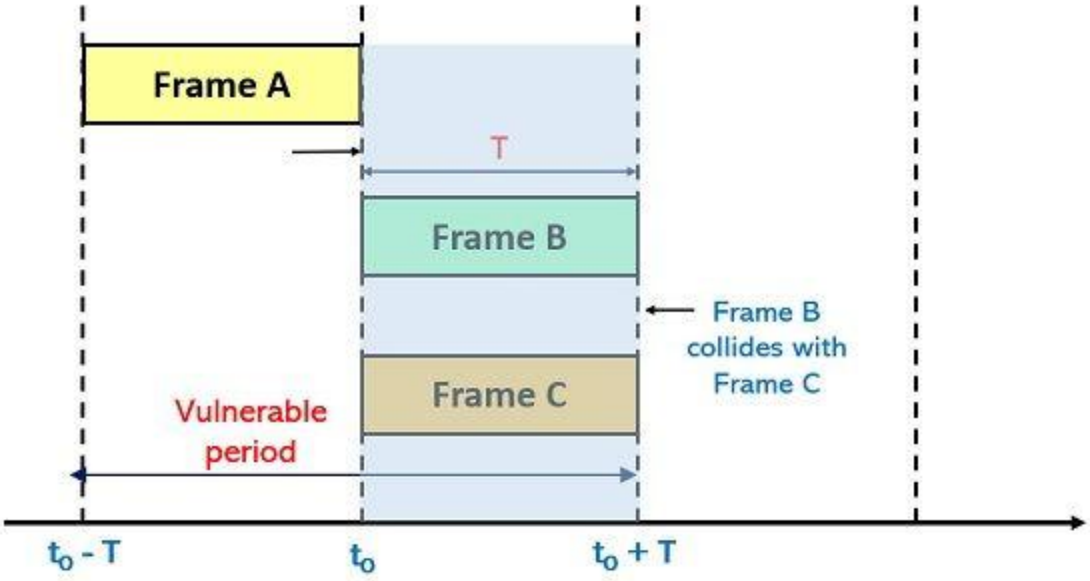
Here each slot has a duration T and data frames are allowed to get transmitted only at the beginning of each time slot. This means that no data frame can be transmitted at any arbitrary time instant. In case, any of the stations fail to transmit in its allotted slot then it is needed to wait for the next slot.

Unlike pure aloha, in slotted aloha, if collision occurs then it is of complete nature (i.e., not partial collision) and data will not be received. This occurs if more than one station is simultaneously transmitting in the same slot.

In slotted aloha, successful data transmission takes place only when each slot individually transmits only a single data frame. By doing this collision probability reduces to a large extent. If no data packet is transmitted in any of the slots, then it will remain idle. It is to be noted here if acknowledgment for any packet is not received then it is considered that it is lost after the collision and is further retransmitted in another slot after considering back-off time. The lost data packets during transmission are known as **backlogged packets**.

Vulnerable period for Slotted ALOHA

As in slotted aloha, the complete channel time is divided into slots thus, here the vulnerable time gets reduced to half than the pure aloha.



Vulnerable time for Slotted ALOHA

This is so because in slotted aloha it is clearly shown that slots are divided according to the uniform packet duration i.e., T. Also, we have discussed that each packet can only be sent at the beginning of each slot. Thus, here chances of only complete collision occur and that too just within the slot. As here each slot exhibit duration T, thus the vulnerable period will be T.

Hence, channel utilization given by throughput S will be:

$$\mathbf{S = Ge^{-G}}$$

For maximum efficiency, G should be 1. Therefore

$$\mathbf{S = 0.368}$$

Thereby offering maximum efficiency, **36.8%**.

ALOHAnet offers easy implementation but shows suitability towards light and moderate load. However, in high load conditions, it somewhat fails. This is so because even slotted aloha does not remove collision completely.