Unit 3 - Data Link Layer

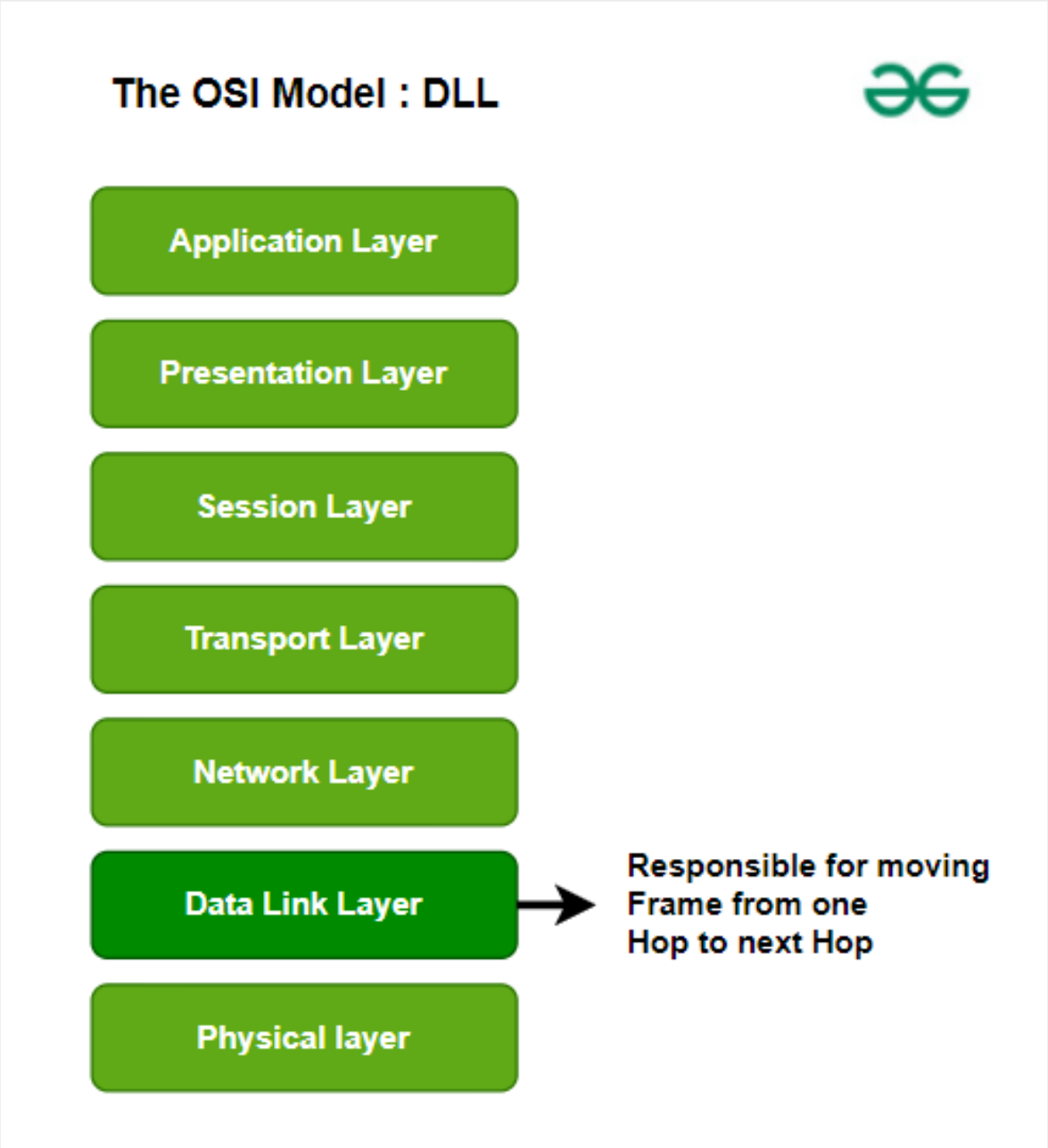
We use frames in data link layers

Transport layers (segments) - Network layer (packets) -> data Link layer (frames) -> physical layers (bits)

Last Updated : 18 Jun, 2024

The data link layer is the second layer from the bottom in the OSI (Open System Interconnection) network architecture model. It is responsible for the node-to-node delivery of data. Its major role is to ensure error-free transmission of information. DLL is also responsible for encoding, decoding, and organizing the outgoing and incoming data.

This is considered the most complex layer of the OSI model as it hides all the underlying complexities of the hardware from the other above layers. In this article, we will discuss Data Link Layer in Detail along with its functions, and sub-layers.



OSI Model: Data Link Layer

Sub-Layers of The Data Link Layer

The data link layer is further divided into two sub-layers, which are as follows:

Logical Link Control (LLC)

This sublayer of the data link layer deals with multiplexing, the flow of data among applications and other services, and LLC is responsible for providing error messages and acknowledgments as well.

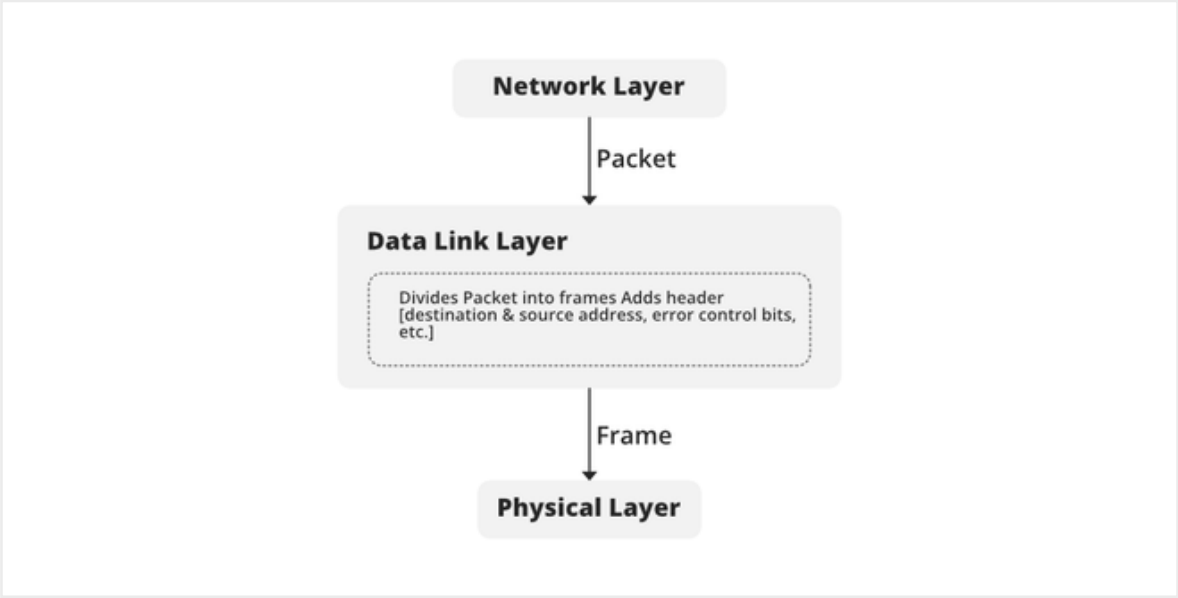
Media Access Control (MAC)

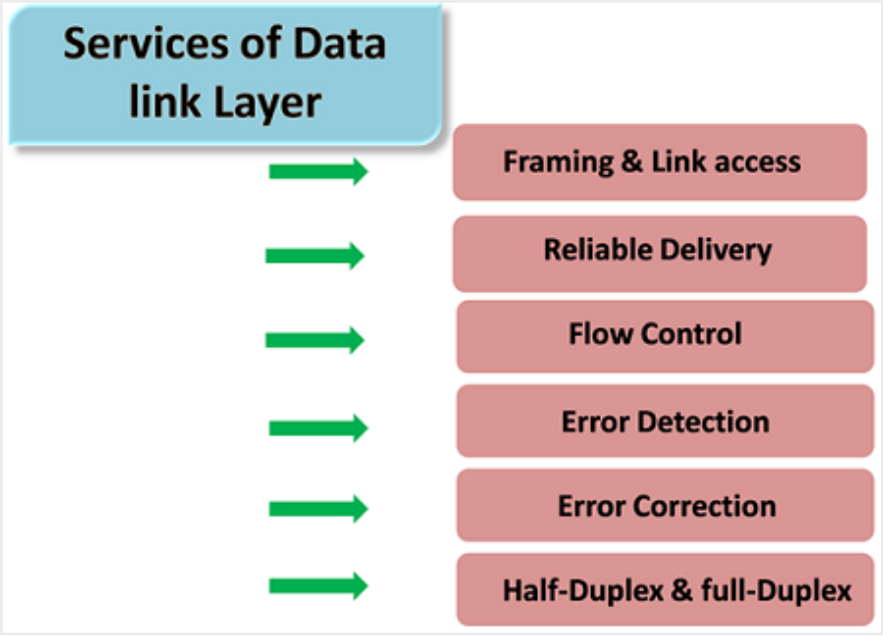
MAC sublayer manages the device's interaction, responsible for addressing frames, and also controls physical media access.

The data link layer receives the information in the form of packets from the Network layer, it divides packets into frames and sends those frames bit-by-bit to the underlying physical layer.

Functions of The Data-link Layer

There are various benefits of data link layers s let's look into it.





Framing

The packet received from the Network layer is known as a frame in the Data link layer. At the sender's side, DLL receives packets from the Network layer and divides them into small frames, then, sends each frame bit-by-bit to the physical layer. It also attaches some

special bits (for error control and addressing) at the header and end of the frame. At the receiver's end, DLL takes bits from the Physical layer organizes them into the frame, and sends them to the Network layer.

Addressing

The data link layer encapsulates the source and destination's MAC address/ physical address in the header of each frame to ensure node-to-node delivery. MAC address is the unique hardware address that is assigned to the device while manufacturing.

Error Control

Data can get corrupted due to various reasons like noise, attenuation, etc. So, it is the responsibility of the data link layer, to detect the error in the transmitted data and correct it using error detection and correction techniques respectively. DLL adds error detection bits into the frame's header, so that receiver can check received data is correct or not. It adds reliability to physical layer by adding mechanisms to detect and retransmit damaged or lost frames.

Flow Control

If the receiver's receiving speed is lower than the sender's sending speed, then this can lead to an overflow in the receiver's buffer and some frames may get lost. So, it's the responsibility of DLL to synchronize the sender's and receiver's speeds and establish flow control between them.

Media Access Control

When multiple devices share the same communication channel there is a high probability of collision, so it's the responsibility of DLL to check which device has control over the channel and CSMA/CD and CSMA/CA can be used to avoid collisions and loss of frames in the channel.

Flow Control

- It is a set of procedures that tells the sender how much data it can transmit before the data overwhelms the receiver.
- The receiving device has limited speed and limited memory to store the data. Therefore, the receiving device must be able

to inform the sending device to stop the transmission temporarily before the limits are reached.
- It requires a buffer, a block of memory for storing the information until they are processed.

Two methods have been developed to control the flow of data:
- Stop-and-wait
- Sliding window

Stop and Wait ARQ

Sender and receiver window size is 1 for each.
Stop and Wait ARQ is a Sliding Window Protocol method used for the reliable delivery of data frames. The stop-and-wait ARQ is used for noisy channels or links to handle flow and error control between sender and receiver. The Stop and Wait ARQ protocol sends a data frame and then waits for an acknowledgment (ACK) from the receiver. The Stop and Wait ARQ protocol sends a data frame and then waits for an acknowledgment (ACK) from the receiver. The ACK indicates that the receiver successfully received the data frame. After receiving the ACK from the receiver, the sender delivers the next data frame. So there is a stop before the next data frame is transferred, hence it is known as the Stop and Wait ARQ protocol.

Characteristics of Stop and Wait ARQ

- Used in Connection-oriented communication.
- It offers error and flow control.
- It is used in Data Link and Transport Layers
- Stop and Wait for ARQ mainly implements the Sliding Window Protocol concept with Window Size 1
- It uses a link between sender and receiver as a half-duplex link
- Throughput = 1 Data packet/frame per RTT
- If the bandwidth*Delay product is very high, then they stop and wait for protocol if it is not so useful. The sender has to keep waiting for acknowledgments before sending the processed next packet.
- It is an example of "**Closed Loop OR connection-oriented** " protocols
- It is a special category of SWP where its window size is 1
- Irrespective of the number of packets sender is having stop

and wait for protocol requires only 2 sequence numbers 0 and 1

Useful Terms in Stop and Wait Protocol

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

Propagation Delay = (Distance between routers) / (Velocity of propagation)

- RoundTripTime (**RTT**) = Amount of time taken by a packet to reach the receiver + Time taken by the Acknowledgement to reach the sender
- TimeOut (**TO**) = 2* RTT
- Time To Live (**TTL**) = 2* TimeOut. (Maximum TTL is 255 seconds)

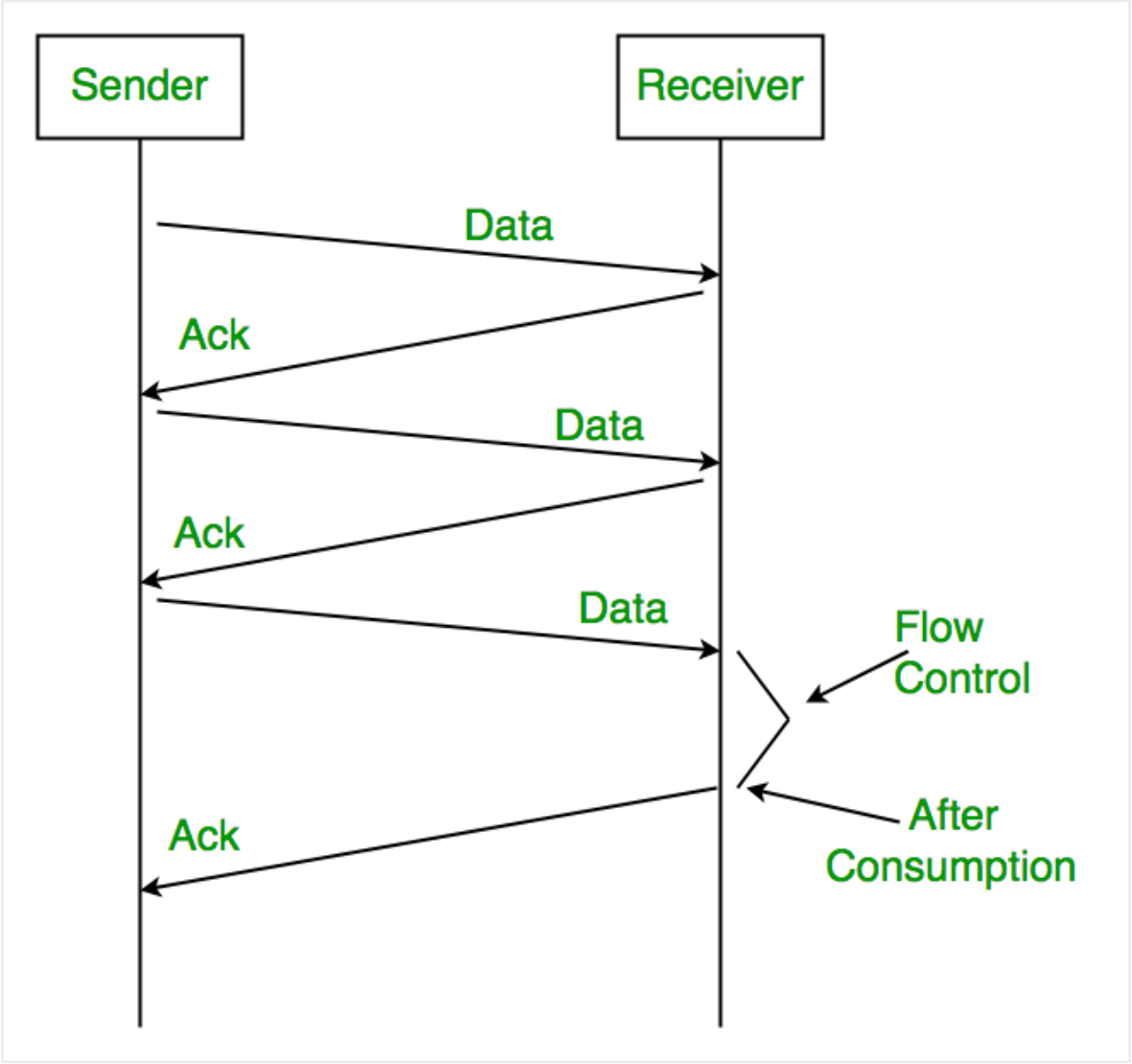
Simple Stop and Wait

At Sender

- Rule 1: Send one data packet at a time.
- Rule 2: Send the next packet only after receiving acknowledgment for the previous.

At Receiver

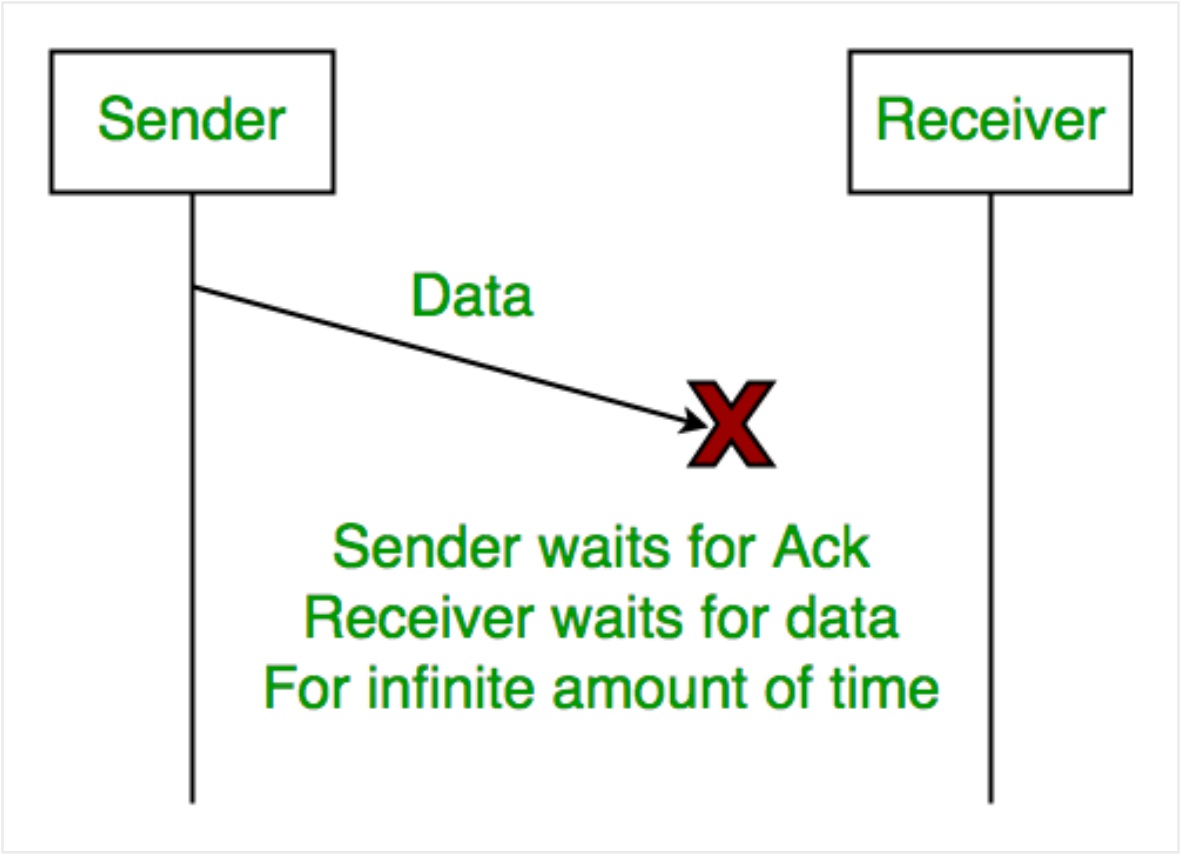
- Rule 1: Send acknowledgement after receiving and consuming a data packet.
- Rule 2: After consuming packet acknowledgement need to be sent (Flow Control)



Problems Associated with Stop and Wait

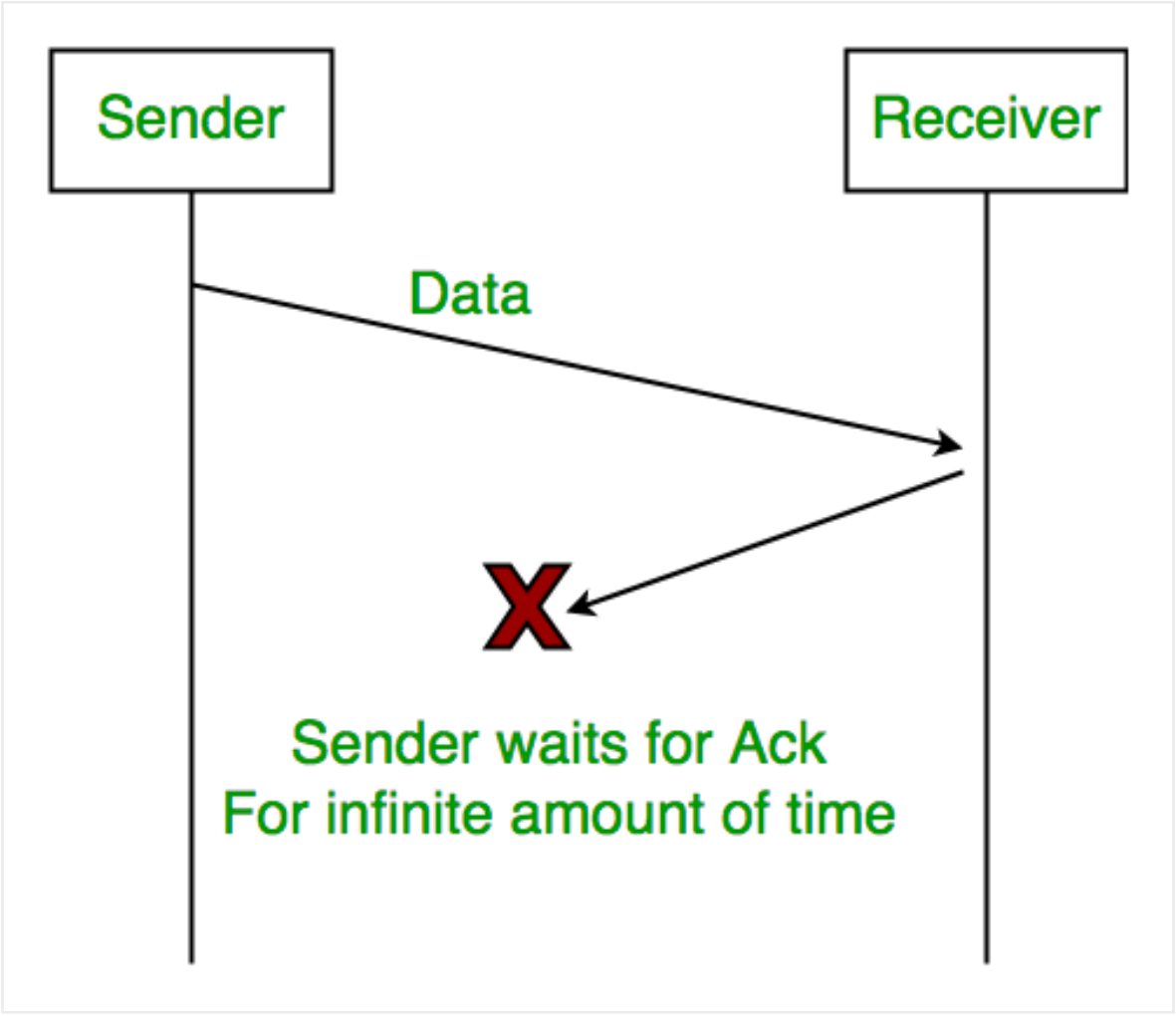
1. Lost Data

Assume the sender transmits the data packet and it is lost. The receiver has been waiting for the data for a long time. Because the data is not received by the receiver, it does not transmit an acknowledgment. The sender does not receive an acknowledgment, it will not send the next packet. This problem is caused by a loss of data.



2. Lost Acknowledgement

Assume the sender sends the data, which is also received by the receiver. The receiver sends an acknowledgment after receiving the packet. In this situation, the acknowledgment is lost in the network. The sender does not send the next data packet because it does not receive acknowledgement, under the stop and wait protocol, the next packet cannot be transmitted until the preceding packet's acknowledgment is received.

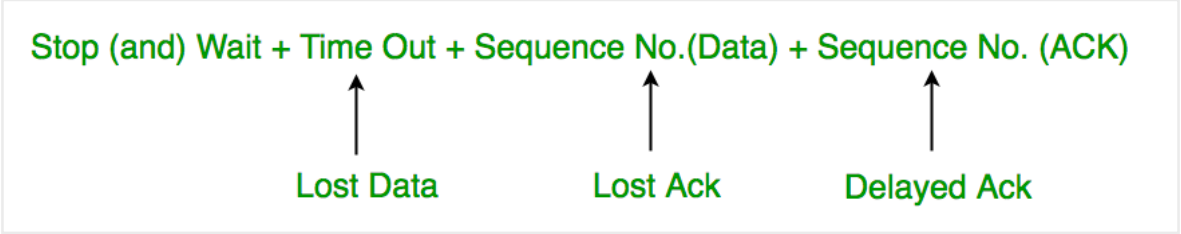


3. Delayed Acknowledgement/Data

Assume the sender sends the data, which is also received by the receiver. The receiver then transmits the acknowledgment, which is received after the sender's timeout period. After a timeout on the sender side, a long-delayed acknowledgement might be wrongly considered as acknowledgement of some other recent packet.

Stop and Wait for ARQ (Automatic Repeat Request)

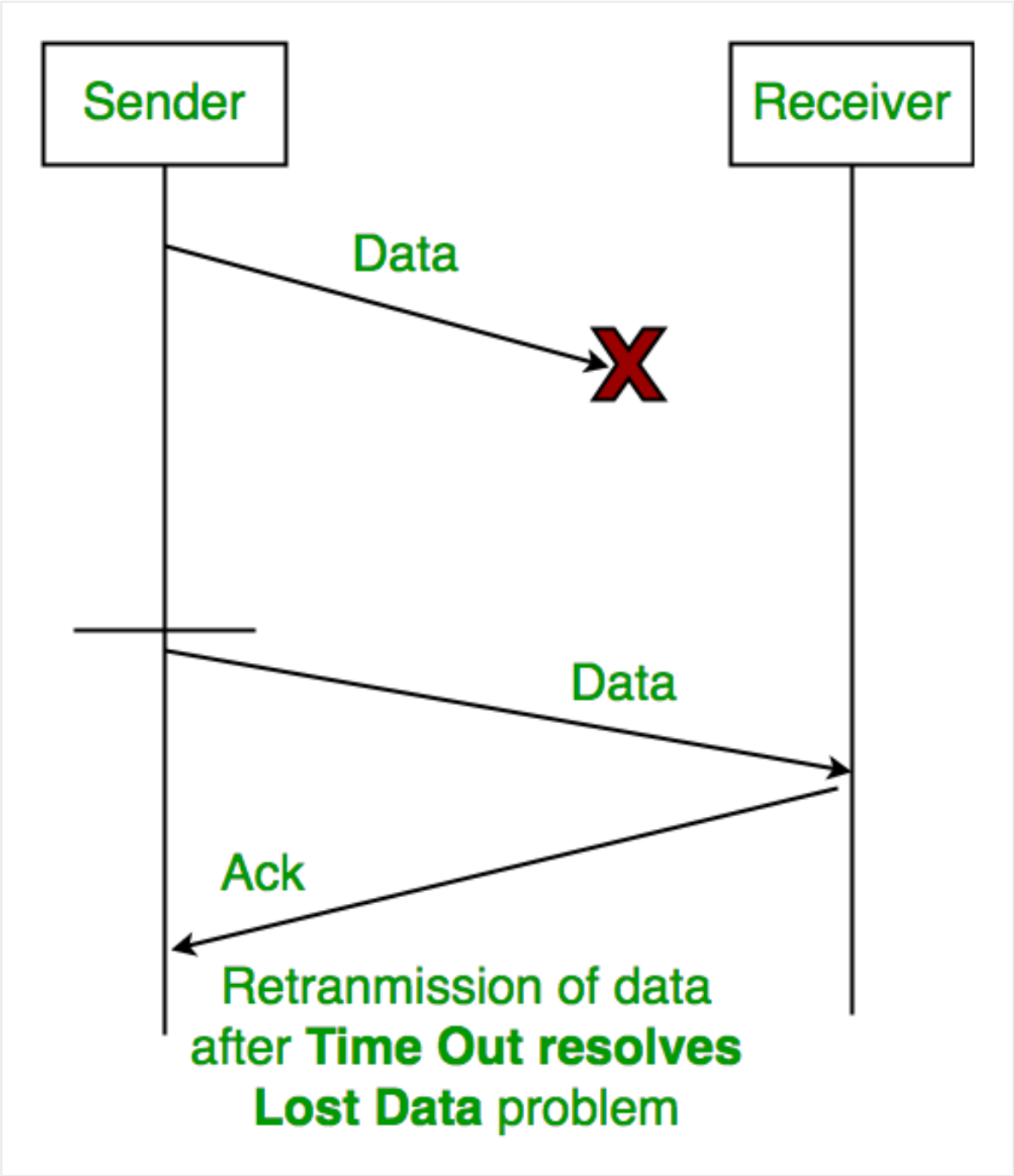
The above 3 problems are resolved by Stop and Wait for ARQ (Automatic Repeat Request) that does both error control and flow control.



1. Time Out

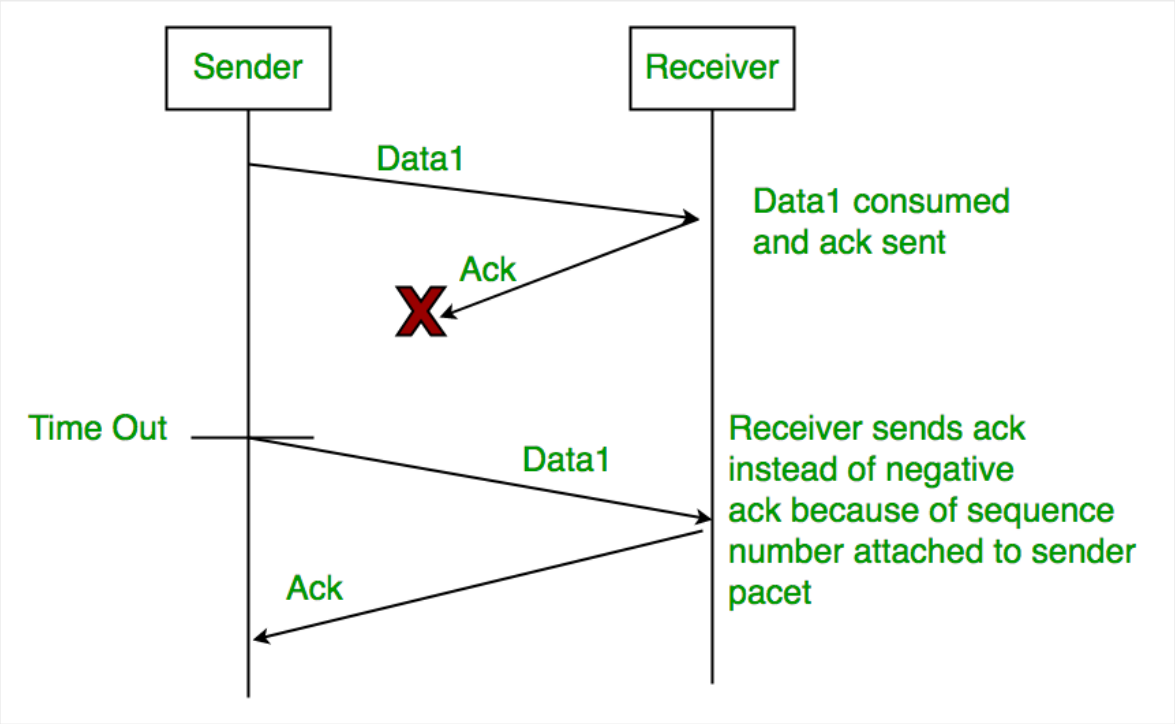
Timeout refers to the duration for which the sender waits for an

acknowledgment (ACK) from the receiver after transmitting a data packet. If the sender does not receive an ACK within this timeout period, it assumes that the frame was lost or corrupted and retransmits the frame.



2. Sequence Number (Data)

In Stop-and-Wait ARQ, the sender assigns sequence numbers to each data frame it sends. This allows the receiver to identify and acknowledge each frame individually, ensuring reliable delivery of data packets. After sending a frame, the sender waits for an acknowledgment before sending the next frame.

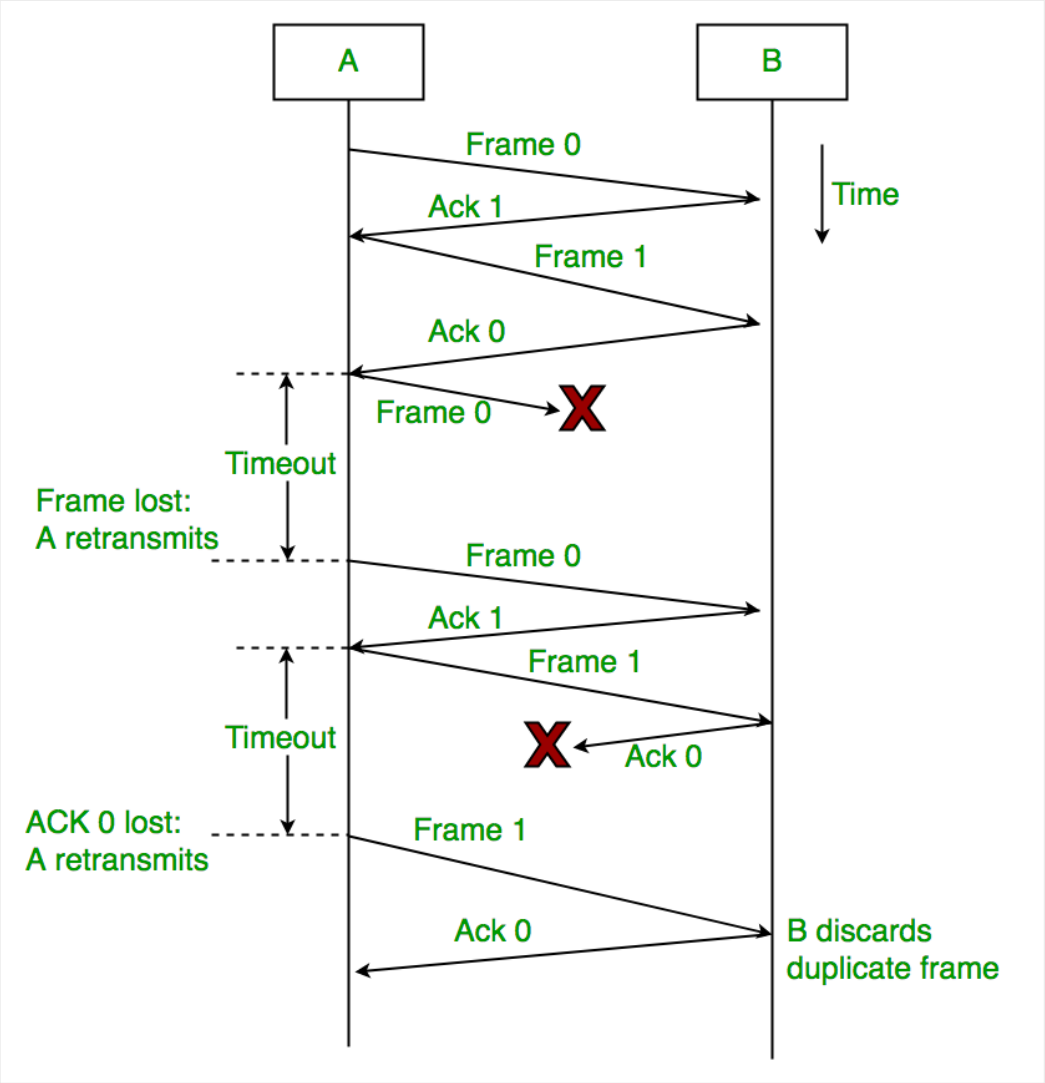


3. Sequence Number(Acknowledgement)

Similarly, sequence numbers are also used in acknowledgments (ACKs) sent by the receiver to acknowledge received data frames. When the receiver successfully receives a data frame, it sends an ACK back to the sender, indicating the sequence number of the next expected frame. The sender uses this ACK to determine whether the transmission was successful and whether it can proceed to send the next frame.

Working of Stop and Wait for ARQ

- Sender A sends a data frame or packet with sequence number 0.
- Receiver B, after receiving the data frame, sends an acknowledgement with sequence number 1 (the sequence number of the next expected data frame or packet)
  There is only a one-bit sequence number that implies that both sender and receiver have a buffer for one frame or packet only.



Constraints in Stop and Wait ARQ

Stop and Wait ARQ has very less efficiency , it can be improved by increasing the window size. Also , for better efficiency , Go back N and Selective Repeat Protocols are used. The Stop and Wait ARQ solves the main three problems but may cause big performance issues as the sender always waits for acknowledgement even if it has the next packet ready to send. 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 through a high-speed connection). To solve this problem, we can send more than one packet at a time with a larger sequence number. We will be discussing these protocols in the next articles. So Stop and Wait ARQ may work fine where propagation delay is very less for example LAN connections but performs badly for distant connections like satellite connections.

Advantages of Stop and Wait ARQ

- **Simple Implementation:** Stop and Wait ARQ is a simple

protocol that is easy to implement in both hardware and software. It does not require complex algorithms or hardware components, making it an inexpensive and efficient option.

- **Error Detection:** Stop and Wait ARQ detects errors in the transmitted data by using checksums or cyclic redundancy checks (CRC). If an error is detected, the receiver sends a negative acknowledgment (NAK) to the sender, indicating that the data needs to be retransmitted.
- **Reliable:** Stop and Wait ARQ ensures that the data is transmitted reliably and in order. The receiver cannot move on to the next data packet until it receives the current one. This ensures that the data is received in the correct order and eliminates the possibility of data corruption.
- **Flow Control:** Stop and Wait ARQ can be used for flow control, where the receiver can control the rate at which the sender transmits data. This is useful in situations where the receiver has limited buffer space or processing power.
- **Backward Compatibility:** Stop and Wait ARQ is compatible with many existing systems and protocols, making it a popular choice for communication over unreliable channels.

Disadvantages of Stop and Wait ARQ

- **Low Efficiency:** Stop and Wait ARQ has low efficiency as it requires the sender to wait for an acknowledgment from the receiver before sending the next data packet. This results in a low data transmission rate, especially for large data sets.
- **High Latency:** Stop and Wait ARQ introduces additional latency in the transmission of data, as the sender must wait for an acknowledgment before sending the next packet. This can be a problem for real-time applications such as video streaming or online gaming.
- **Limited Bandwidth Utilization:** Stop and Wait ARQ does not utilize the available bandwidth efficiently, as the sender can transmit only one data packet at a time. This results in underutilization of the channel, which can be a problem in situations where the available bandwidth is limited.
- **Limited Error Recovery:** Stop and Wait ARQ has limited error recovery capabilities. If a data packet is lost or corrupted, the sender must retransmit the entire packet, which can be time-consuming and can result in further delays.

- **Vulnerable to Channel Noise:** Stop and Wait ARQ is vulnerable to channel noise, which can cause errors in the transmitted data. This can result in frequent retransmissions and can impact the overall efficiency of the protocol.

Sliding Window Protocol

The sliding window is a technique for sending multiple frames at a time. It controls the data packets between the two devices where reliable and gradual delivery of data frames is needed. It is also used in TCP (Transmission Control Protocol).
In this technique, each frame has sent from the sequence number. The sequence numbers are used to find the missing data in the receiver end. The purpose of the sliding window technique is to avoid duplicate data, so it uses the sequence number.

Types of Sliding Window Protocol

Sliding window protocol has two types:

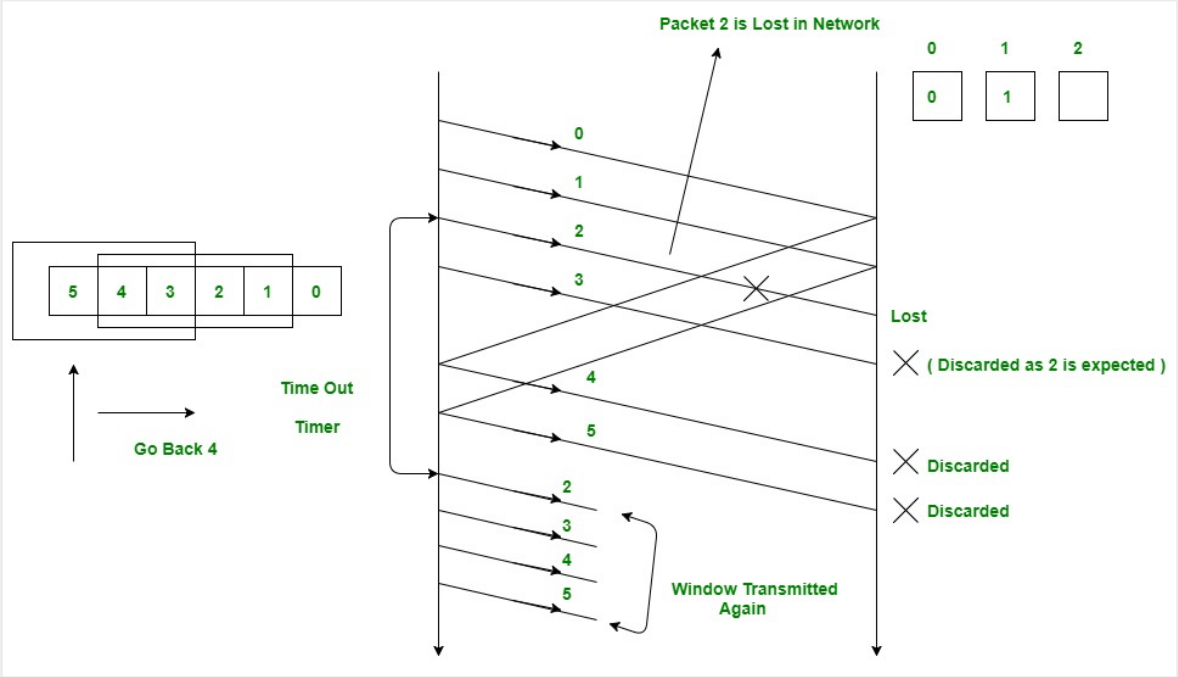
1. Go-Back-N ARQ
2. Selective Repeat ARQ

Go-Back-N ARQ

Here sender window size will be 2^m - 1;
Go-Back-N ARQ protocol is also known as Go-Back-N Automatic Repeat Request. It is a data link layer protocol that uses a sliding window method. In this, if any frame is corrupted or lost, all subsequent frames have to be sent again.
The size of the sender window is N in this protocol. For example, Go-Back-8, the size of the sender window, will be 8. The receiver window size is always 1.
If the receiver receives a corrupted frame, it cancels it. The receiver does not accept a corrupted frame. When the timer expires, the sender sends the correct frame again. The design of the Go-Back-N ARQ protocol is shown below.
For total 12 bits of data and 4 window size -
0 1 2 3 0 1 2 3 0 1 2 3 -> sequence number

Receiver window size will be always 1 here

The example of Go-Back-N ARQ is shown below in the figure.

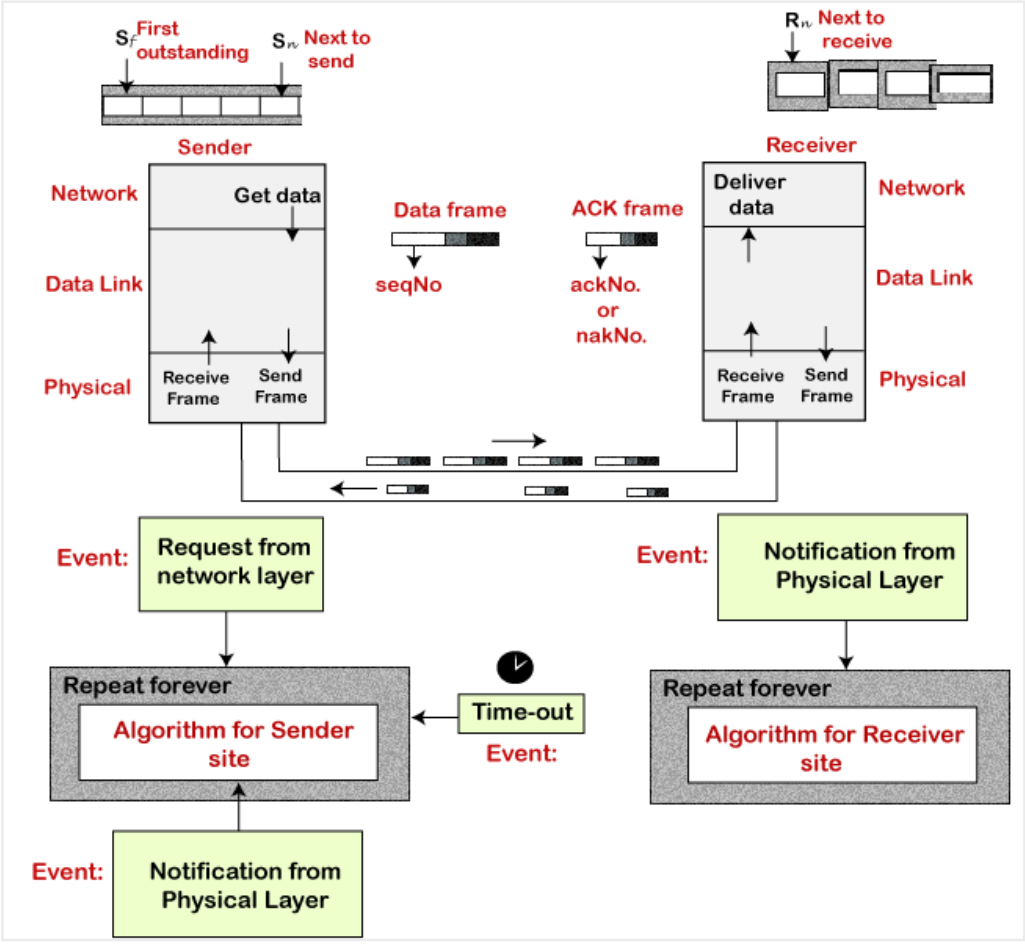


Selective Repeat ARQ

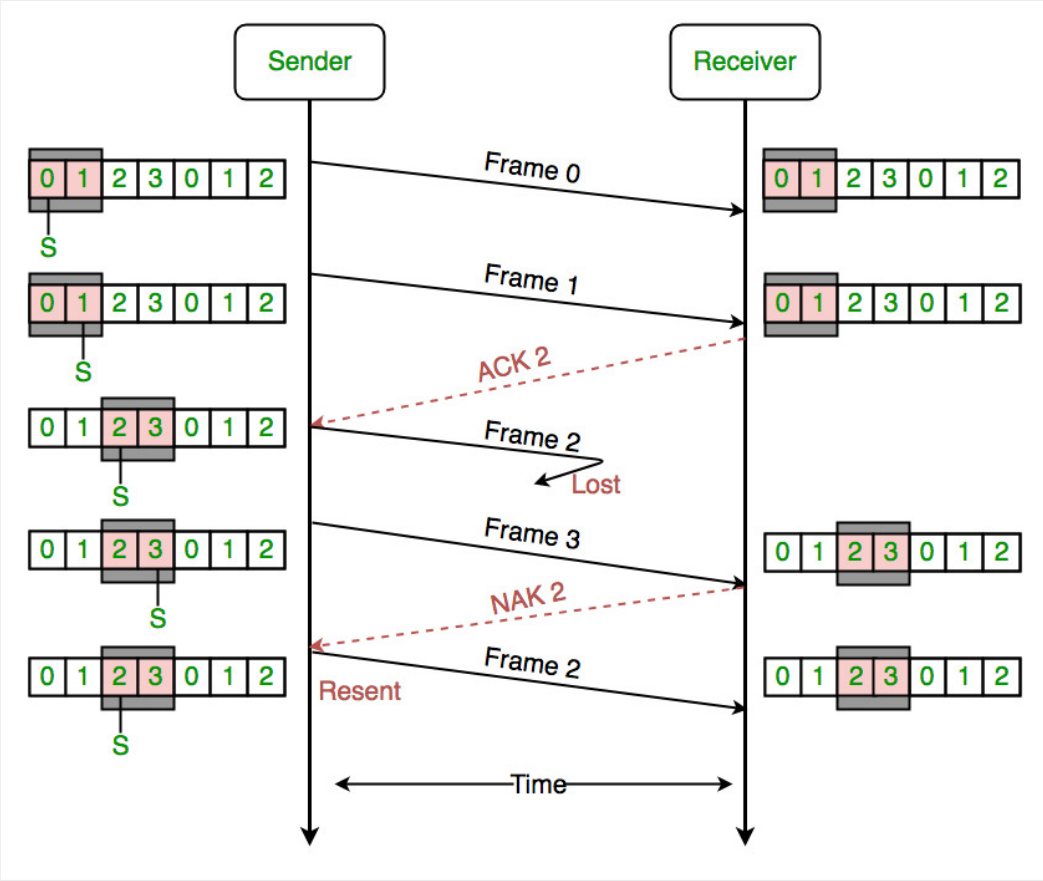
Sender window size : (2)^(m-1) = Receiver window size.

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.

If the receiver receives a corrupt frame, it does not directly discard it. It sends a negative acknowledgment to the sender. 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. The design of the Selective Repeat ARQ protocol is shown below.



The example of the Selective Repeat ARQ protocol is shown below in the figure.



Difference between the Go-Back-N ARQ and Selective Repeat ARQ?

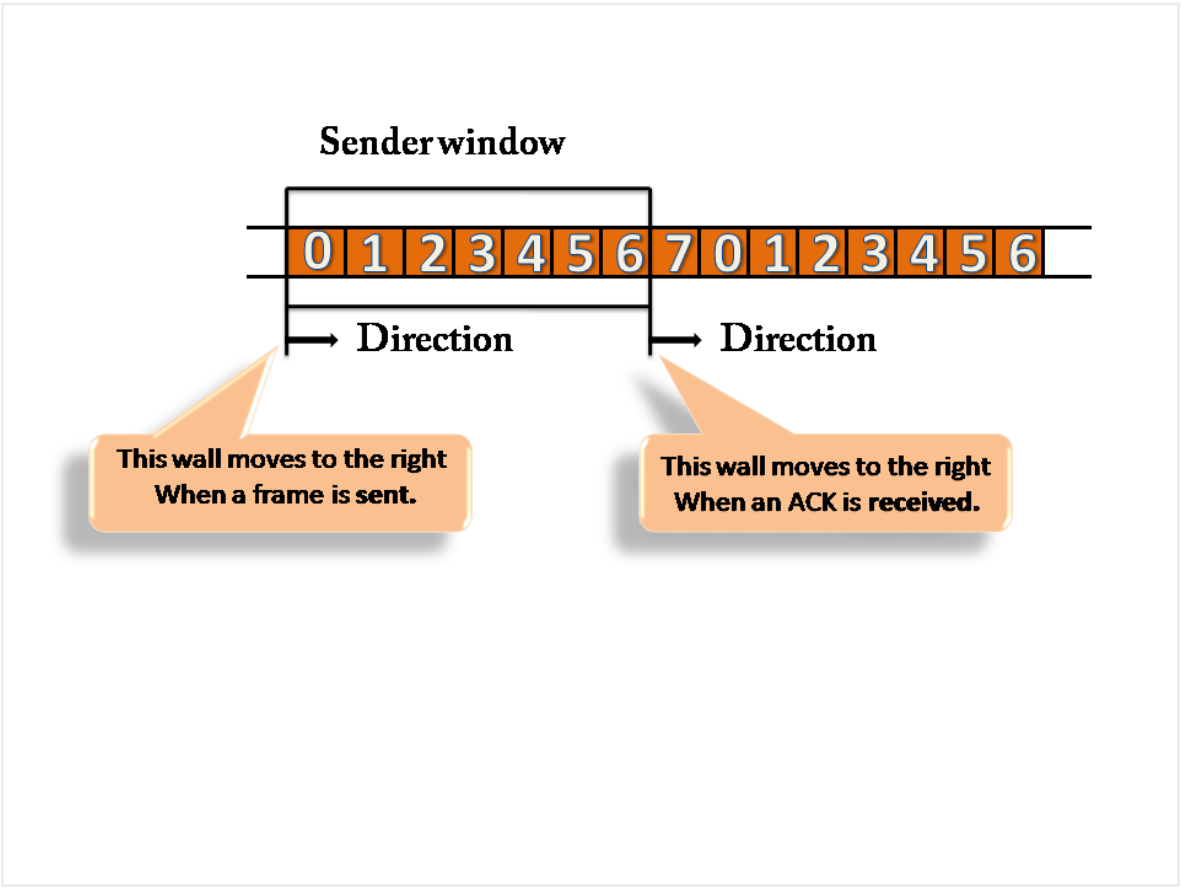
Go-Back-N ARQ	Selective Repeat ARQ
If a frame is corrupted or lost in it,all subsequent frames have to be sent again.	In this, only the frame is sent again, which is corrupted or lost.
If it has a high error rate,it wastes a lot of bandwidth.	There is a loss of low bandwidth.
It is less complex.	It is more complex because it has to do sorting and searching as well. And it also requires more storage.
It does not require sorting.	In this, sorting is done to get the frames in the correct order.
It does not require searching.	The search operation is performed in it.

It is used more.	It is used less because it is more complex.
Go-Back-N Protocol	Selective Repeat Protocol
In Go-Back-N Protocol, if the sent frame are find suspected then all the frames are re-transmitted from the lost packet to the last packet transmitted.	In selective Repeat protocol, only those frames are re-transmitted which are found suspected.
Sender window size of Go-Back-N Protocol is N where in binary bits -> 2^m-1.	Sender window size of selective Repeat protocol is also N **(2)^(m-1)**.
Receiver window size of Go-Back-N Protocol is 1.	Receiver window size of selective Repeat protocol is N. **(2)^(m-1)**.
Go-Back-N Protocol is less complex.	Selective Repeat protocol is more complex.
In Go-Back-N Protocol, neither sender nor at receiver need sorting.	In selective Repeat protocol, receiver side needs sorting to sort the frames.
In Go-Back-N Protocol, type of Acknowledgement is cumulative. If a acknowledgement failed we need to send the entire window again as it sends akg of next expected byte.	In selective Repeat protocol, type of Acknowledgement is individual. If a acknowledgement failed only that frame will be sent later on as it send ackg for each byte.
In Go-Back-N Protocol, Out-of-Order packets are NOT Accepted (discarded) and the entire window is re-transmitted.	In selective Repeat protocol, Out-of-Order packets are Accepted.

In Go-Back-N Protocol, if Receives a corrupt packet, then also, the entire window is re-transmitted.	In selective Repeat protocol, if Receives a corrupt packet, it immediately sends a negative acknowledgement and hence only the selective packet is retransmitted.
Efficiency of Go-Back-N Protocol is N/(1+2*a) & N = 2^m-1 & a = pt/tt	Efficiency of selective Repeat protocol is also N/(1+2*a). & N = **(2)^(m-1)**. & a = pt/tt
Implementation is easy	Implementation in coding part is hard because of searching and sorting algorithms for missing bits.

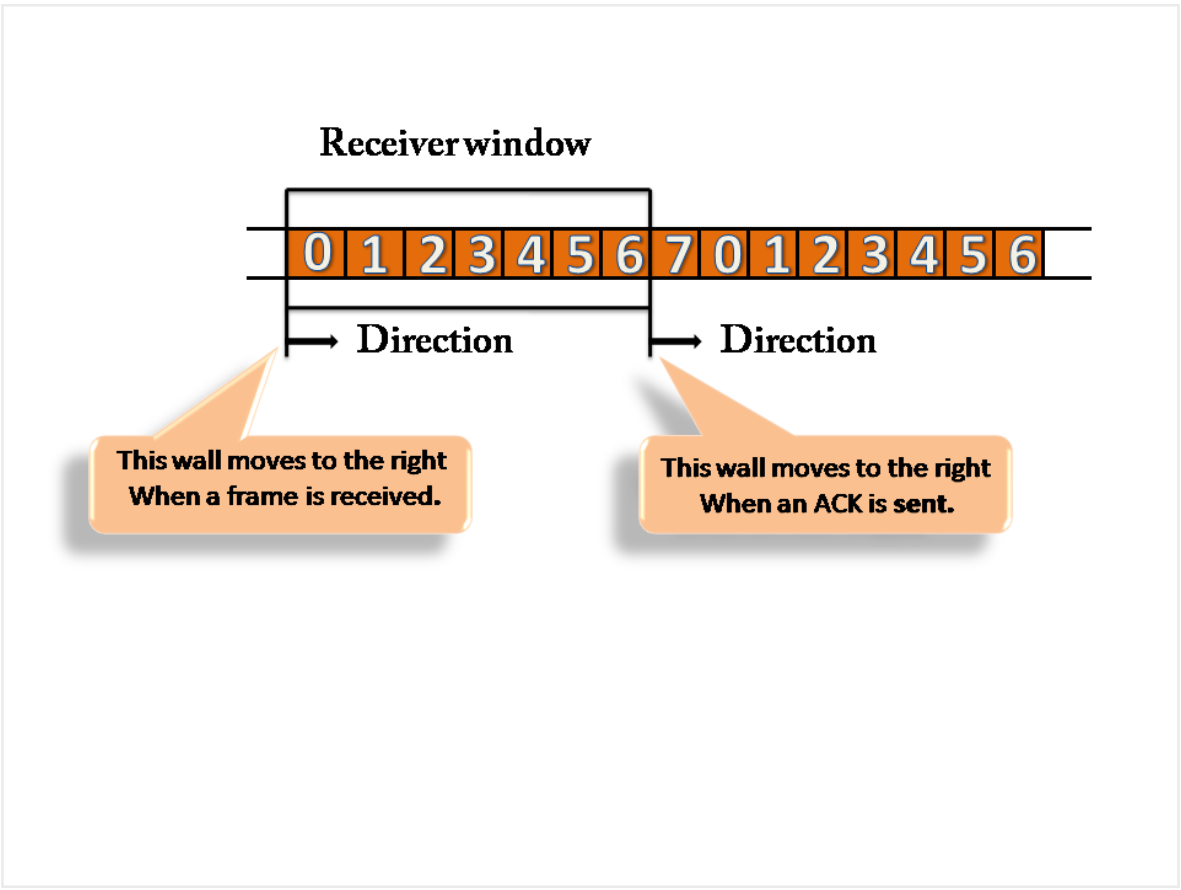
Sender Window

- At the beginning of a transmission, the sender window contains n-1 frames, and when they are sent out, the left boundary moves inward shrinking the size of the window. For example, if the size of the window is w if three frames are sent out, then the number of frames left out in the sender window is w-3.
- Once the ACK has arrived, then the sender window expands to the number which will be equal to the number of frames acknowledged by ACK.
- For example, the size of the window is 7, and if frames 0 through 4 have been sent out and no acknowledgement has arrived, then the sender window contains only two frames, i.e., 5 and 6. Now, if ACK has arrived with a number 4 which means that 0 through 3 frames have arrived undamaged and the sender window is expanded to include the next four frames. Therefore, the sender window contains six frames (5,6,7,0,1,2).



Receiver Window

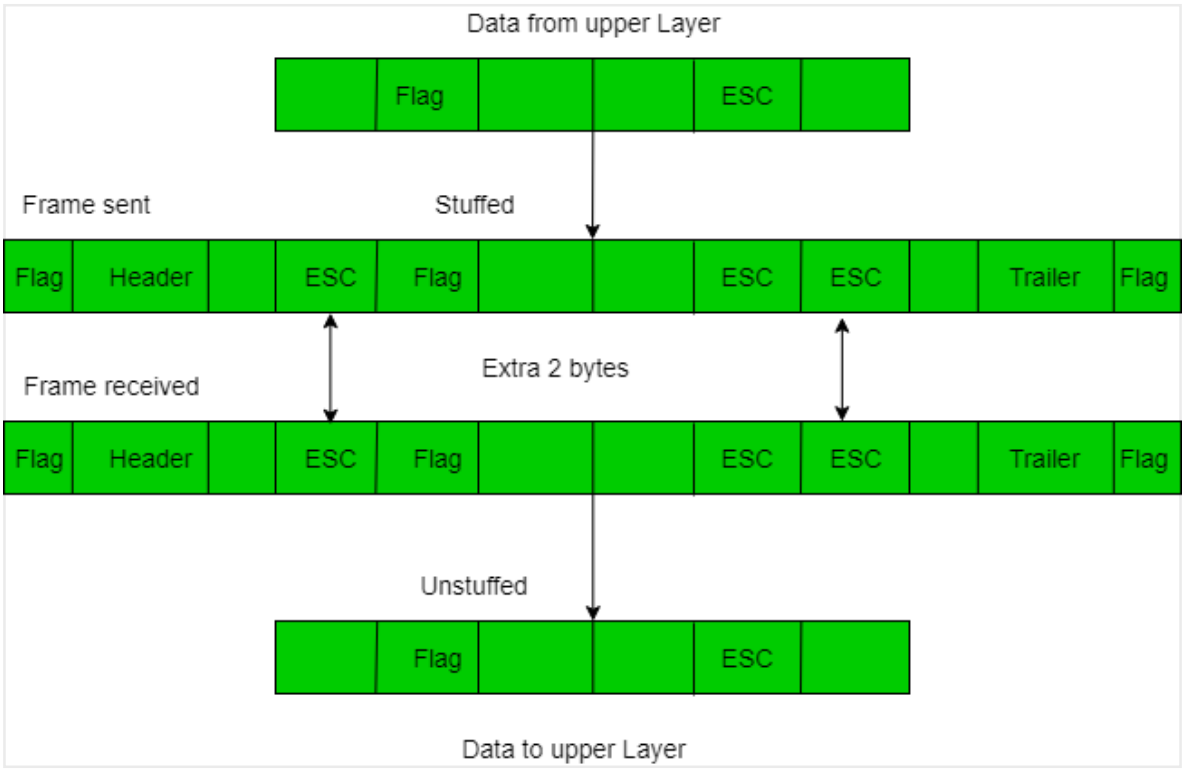
- At the beginning of transmission, the receiver window does not contain n frames, but it contains n-1 spaces for frames.
- When the new frame arrives, the size of the window shrinks.
- The receiver window does not represent the number of frames received, but it represents the number of frames that can be received before an ACK is sent. For example, the size of the window is w, if three frames are received then the number of spaces available in the window is (w-3).
- Once the acknowledgement is sent, the receiver window expands by the number equal to the number of frames acknowledged.
- Suppose the size of the window is 7 means that the receiver window contains seven spaces for seven frames. If the one frame is received, then the receiver window shrinks and moving the boundary from 0 to 1. In this way, window shrinks one by one, so window now contains the six spaces. If frames from 0 through 4 have sent, then the window contains two spaces before an acknowledgement is sent.



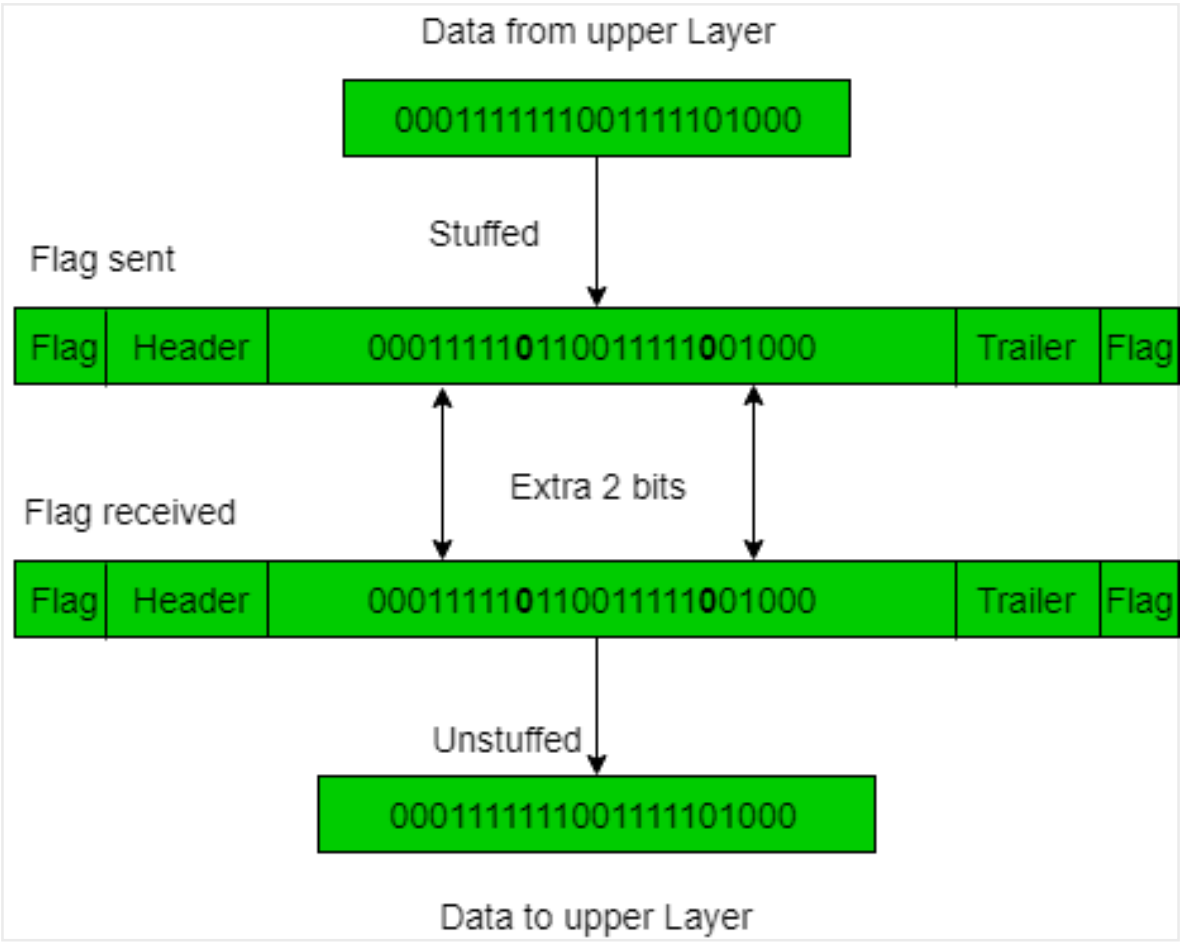
Framing in Data Link Layer

-> We use starting and ending flag in data part, We use esc before flag or another esc if we want to skip that flag or that particular flag is part of data then we use esc in character byte stuffing.

-> In bit stuffing we use a 0 after 5 continuos ones to separate digits to detect on receiver end that its a part of data not the header



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 01101, receiver removes the 0 and reads the data.



Examples:

- If Data –> 011100011110 and ED –> 0111 then, find data after bit stuffing.

--> 011**0**10001**0**11**0**0

- If Data –> 110001001 and ED –> 1000 then, find data after bit stuffing?

--> 1100**1**010**0**1**1**

Error Detection in Computer Networks

-> Here the error detection techniques we use are by using redundant bits means we are adding extra bits to our data to ensure that we be able to detect the error properly and to correct error we use hamming code error detection and correction method

-> We use check sum & cyclic redundancy test (arc) to detect the error in bits.

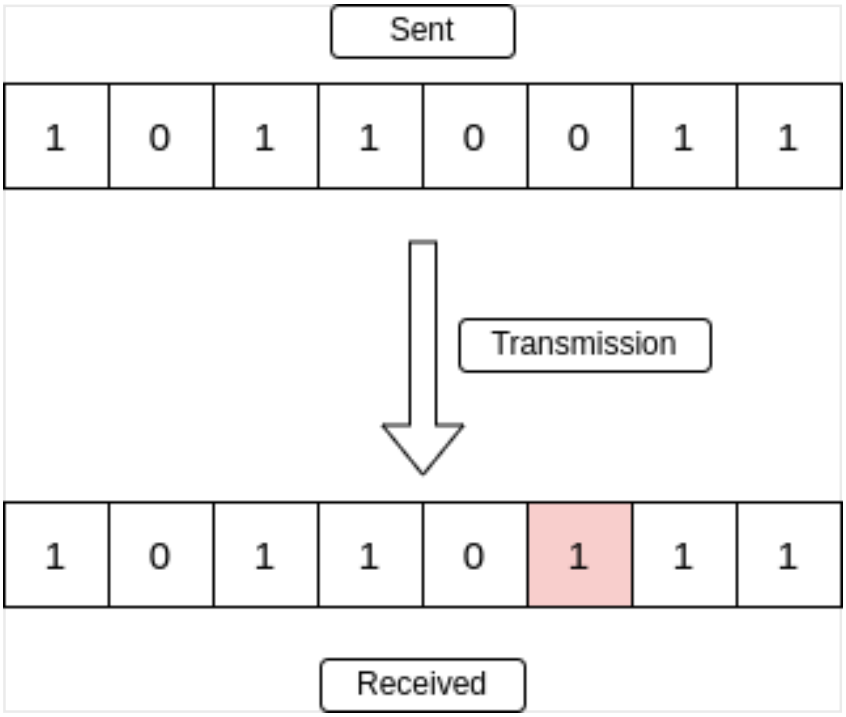
-> For a 1gb/sec bandwidth we have 1 bit sent in 1/10^9 ns which means 1 bit can have error in this time, although mostly we have burst error in most of the cases.

Error is a condition when the receiver's information does not match the sender's. Digital signals suffer from noise during transmission that can introduce errors in the binary bits traveling from sender to receiver. That means a 0 bit may change to 1 or a 1 bit may change to 0.
Data (Implemented either at the Data link layer or Transport Layer of the OSI Model) may get scrambled by noise or get corrupted whenever a message is transmitted. To prevent such errors, error-detection codes are added as extra data to digital messages. This helps in detecting any errors that may have occurred during message transmission.

Types of Errors

Single-Bit Error

A single-bit error refers to a type of data transmission error that occurs when one bit (i.e., a single binary digit) of a transmitted data unit is altered during transmission, resulting in an incorrect or corrupted data unit.

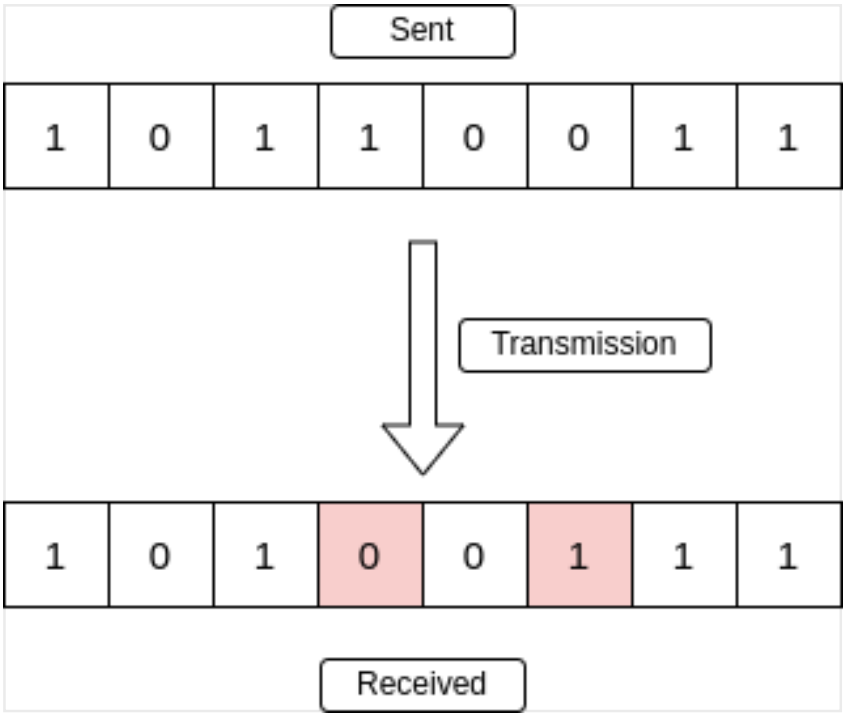


Single-Bit Error

Multiple-Bit Error

A multiple-bit error is an error type that arises when more than one bit in a data transmission is affected. Although multiple-bit errors are

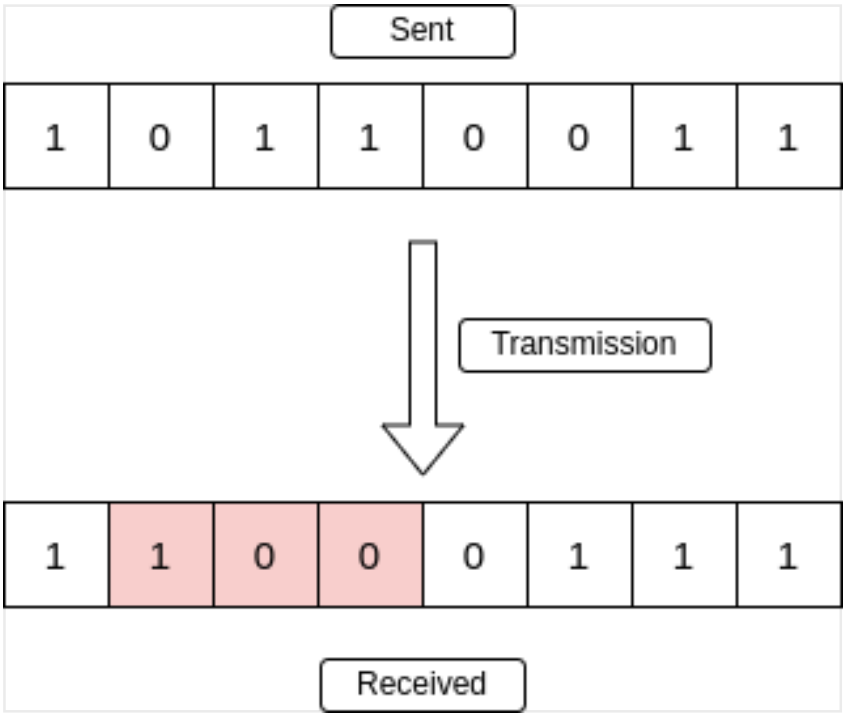
relatively rare when compared to single-bit errors, they can still occur, particularly in high-noise or high-interference digital environments.



Multiple-Bit Error

Burst Error

When several consecutive bits are flipped mistakenly in digital transmission, it creates a burst error. This error causes a sequence of consecutive incorrect values.



Burst Error

Error Detection Methods

To detect errors, a common technique is to introduce redundancy bits that provide additional information. Various techniques for error detection include:

- Simple Parity Check
- Two-Dimensional Parity Check
- Checksum
- Cyclic Redundancy Check (CRC)

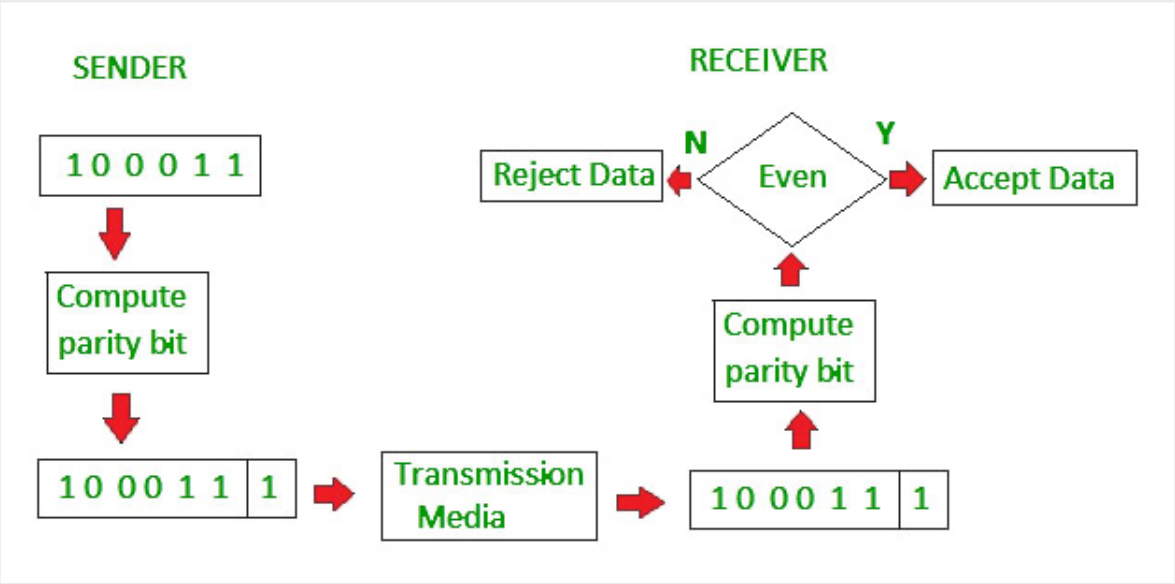
Simple Parity Check

-> most of the time we use even parity
-> here we add 1 if data bits are odd else 0 if data bits are even
eg. 100101 1 -> 1 added as parity to make it even
-> it can detect odd parity error or only single bit error but fails to detect multiple bits error
-> It can't detect which bit has the error
-> It is used in large data transfer where some error could be neglected, like voice calls etc.
-> Hamming distance - The number of position at which 2 windows bits has different values, can be calculated after xor operation of both.
Hamming distance d must be d - 1 to detect the error in simple parity check
0101 xor 1000 -> 1101

Simple-bit parity is a simple error detection method that involves adding an extra bit to a data transmission. It works as:

- 1 is added to the block if it contains an odd number of 1's, and
- 0 is added if it contains an even number of 1's

This scheme makes the total number of 1's even, that is why it is called even parity checking.



Advantages of Simple Parity Check

- Simple parity check can detect all single bit error.
- Simple parity check can detect an odd number of errors.
- **Implementation**: Simple Parity Check is easy to implement in both hardware and software.
- **Minimal Extra Data**: Only one additional bit (the parity bit) is added per data unit (e.g., per byte).
- **Fast Error Detection**: The process of calculating and checking the parity bit is quick, which allows for rapid error detection without significant delay in data processing or communication.
- **Single-Bit Error Detection**: It can effectively detect single-bit errors within a data unit, providing a basic level of error detection for relatively low-error environments.

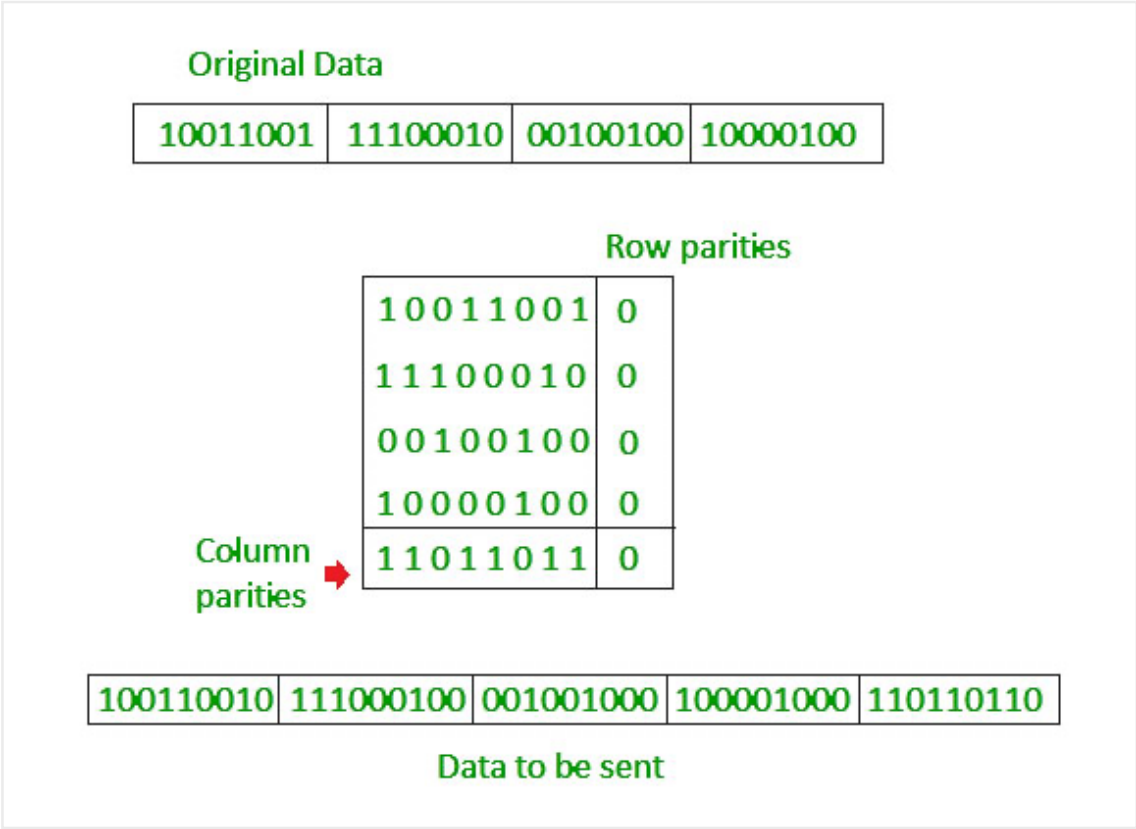
Disadvantages of Simple Parity Check

- Single Parity check is not able to detect even no. of bit error.
- **For example,** the Data to be transmitted is **101010**.
  Codeword transmitted to the receiver is 1010101 (we have used even parity).
  Let's assume that during transmission, two of the bits of code word flipped to 1111101.
  On receiving the code word, the receiver finds the no. of ones to be even and hence **no error,** *which is a wrong assumption.*
-

Two-Dimensional Parity Check

Two-dimensional Parity check bits are calculated for each row, which is equivalent to a simple parity check bit. Parity check bits are

also calculated for all columns, then both are sent along with the data. At the receiving end, these are compared with the parity bits calculated on the received data.



Advantages of Two-Dimensional Parity Check

- Two-Dimensional Parity Check can detect and correct all single bit error.
- Two-Dimensional Parity Check can detect two or three bit error that occur any where in the matrix.

Disadvantages of Two-Dimensional Parity Check

- Two-Dimensional Parity Check can not correct two or three bit error. It can only detect two or three bit error.
- If we have a error in the parity bit then this scheme will not work.

Checksum

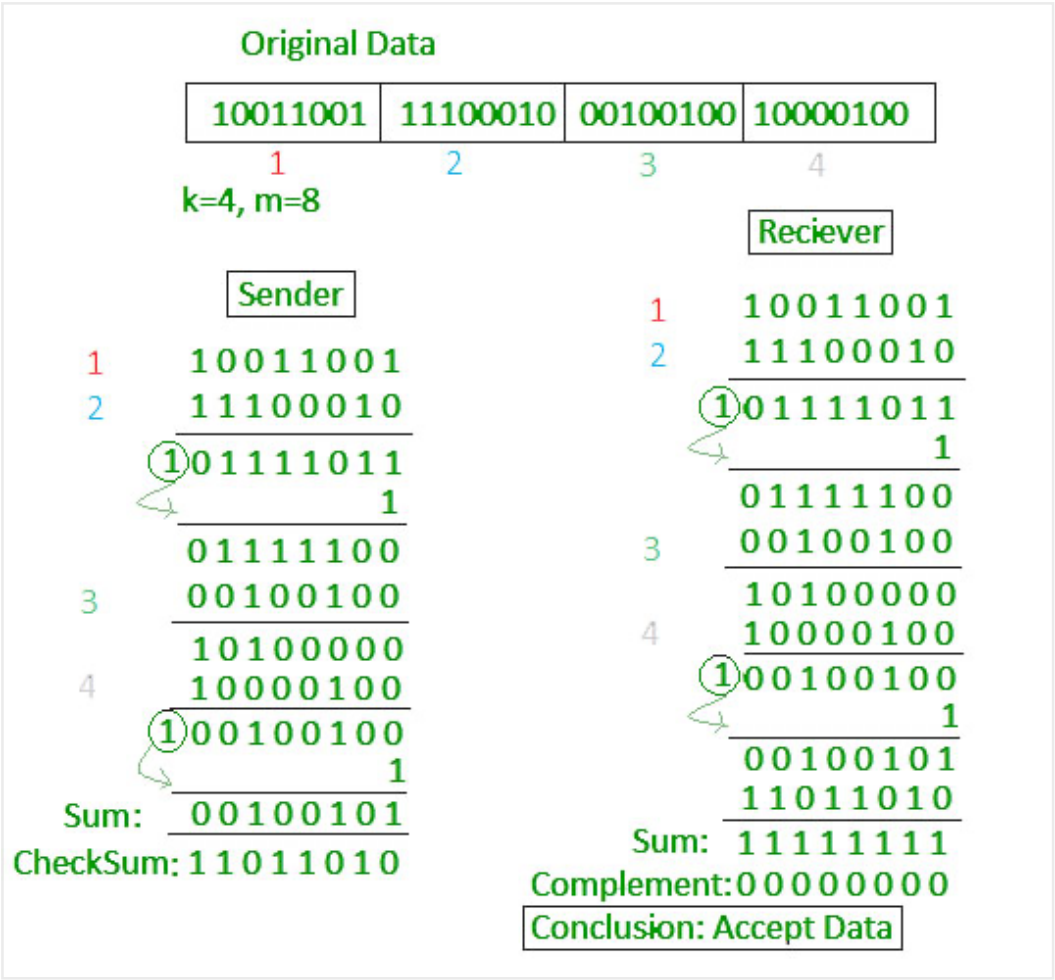
Checksum error detection is a method used to identify errors in transmitted data. The process involves dividing the data into equally sized segments and using a 1's complement to calculate the sum of these segments. The calculated sum is then sent along with the data to the receiver. At the receiver's end, the same process is repeated and if all zeroes are obtained in the sum, it means that the data is correct.

Checksum – Operation at Sender's Side

- Firstly, the data is divided into k segments each of m bits.
- On 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.
- The checksum segment is sent along with the data segments.

Checksum – Operation at Receiver's Side

- At the receiver's end, all received segments are added using 1's complement arithmetic to get the sum. The sum is complemented.
- If the result is zero, the received data is accepted; otherwise discarded.



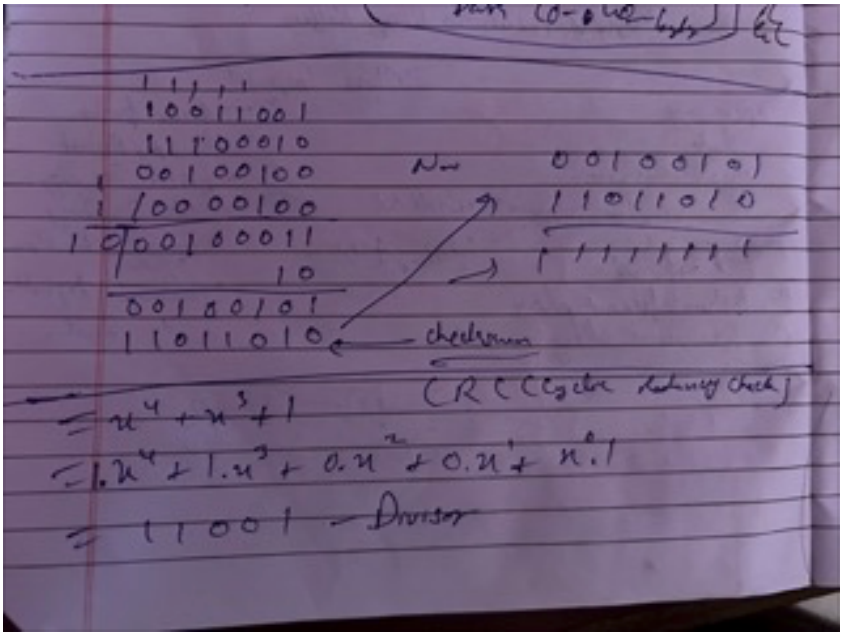
Cyclic Redundancy Check (CRC)

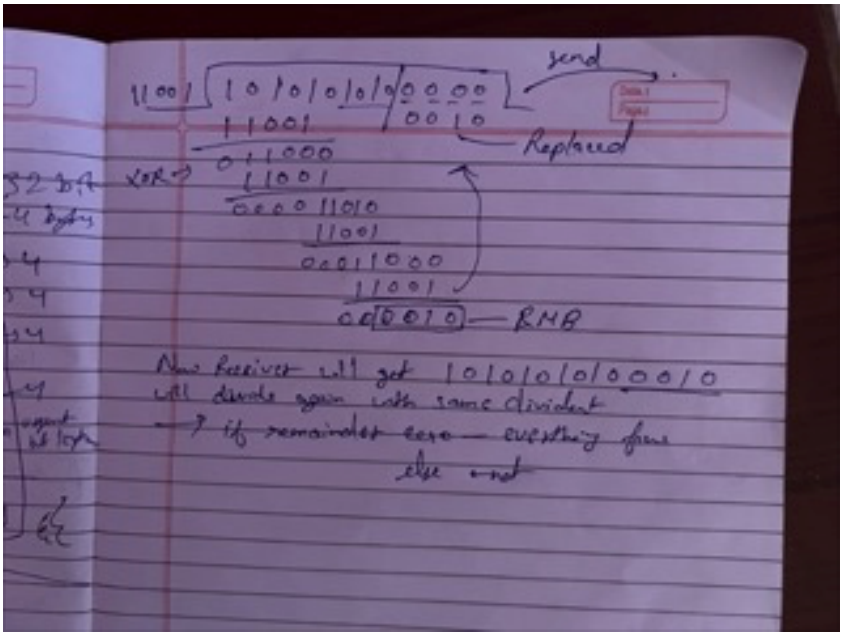
-> used in real life application currently,

-> can detect odd bit error, single bit, burst error of length equal to the polynomial degree can also be detected by this shit. .. means

We have polynomial equation here like x^4 + x^2 + 1, here the

degree of x is 4 that’s what we talking about .. .
-> total bits we send here are - > m+ r
M = number of bits in message
R = number of redundant bits
-> we must have given divisor in polynomial form
x^4+x^3+1
= 1.x^4 + 1.x^3 + 0.X^2 + 0.x^1 + 1.x^0
Now take the coefficients of x
= 11001. In bits as divisor or maybe divident in same way sometimes
-> we will add highest power of divisor to dividend here 4 -> 0000
Else if dividend given in binary form add n - 1 zero where n is number of bits.
-> xor operation while dividing
-> in xor zero for same and 1 for different bits.
-> lets say we get reminder 00010 will append in last 4 bits of reminder in place of 4 zero bits and send the whole to receiver
-> receiver will do the same division with same divident and
If reminder is zero everything if fine else error.



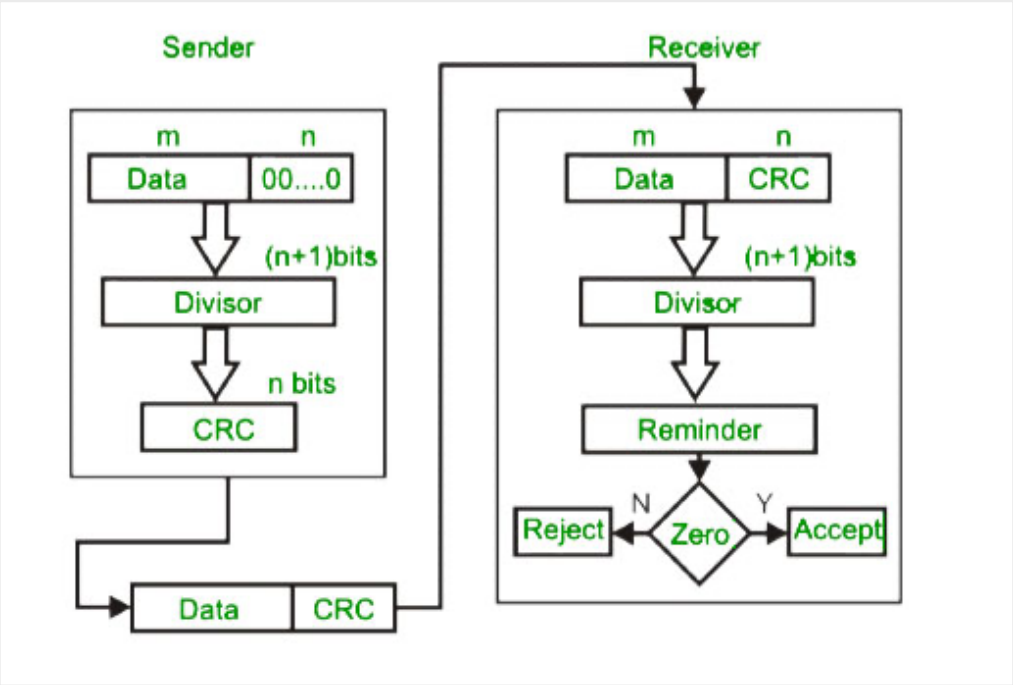


Efficient = m/N * 100

M = number of bits in message

N = number of redundant bits + data bits = total bits

- Unlike the checksum scheme, which is based on addition, CRC is based on binary division.
- In CRC, a sequence of redundant bits, called cyclic redundancy check bits, are appended to the end of the 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.



CRC Working

We have given dataword of length n and divisor of length k.

Step 1: Append (k-1) zero's to the original message

Step 2: Perform modulo 2 division

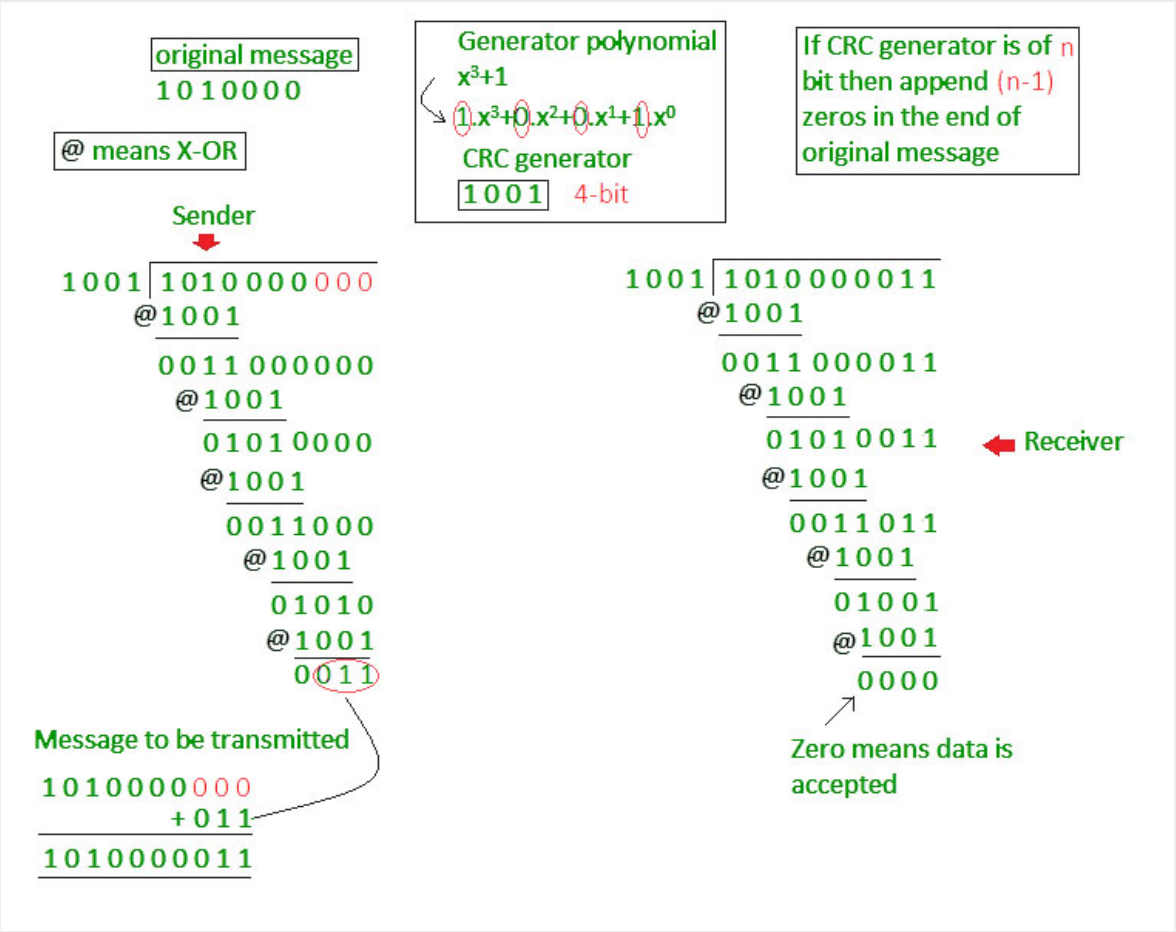
Step 3: Remainder of division = CRC

Step 4: Code word = Data with append k-1 zero's + CRC

Note:

- CRC must be k-1 bits
- Length of Code word = n+k-1 bits

Example: Let's data to be send is 1010000 and divisor in the form of polynomial is x^3+1. CRC method discussed below.



Example: Previous year GATE questions based on error detection: GATE CS 2009 Question 48 GATE CS 2007 Question 68.

Advantages of Error Detection

- **Increased Data Reliability:** Error detection ensures that the data transmitted over the network is reliable, accurate, and free from errors. This ensures that the recipient receives the same data that was transmitted by the sender.
- **Improved Network Performance:** Error detection mechanisms can help to identify and isolate network issues that are causing errors. This can help to improve the overall performance of the network and reduce downtime.
- **Enhanced Data Security:** Error detection can also help to ensure that the data transmitted over the network is secure and has not been tampered with.

Disadvantages of Error Detection

- **Overhead**: Error detection requires additional resources and processing power, which can lead to increased overhead on the network. This can result in slower network performance and increased latency.
- **False Positives:** Error detection mechanisms can sometimes

generate false positives, which can result in unnecessary retransmission of data. This can further increase the overhead on the network.
- **Limited Error Correction:** Error detection can only identify errors but cannot correct them. This means that the recipient must rely on the sender to retransmit the data, which can lead to further delays and increased network overhead.

Conclusion

In conclusion, error detection in computer networks is crucial for ensuring data is transmitted accurately and reliably. By using various techniques, like checksums, parity bits, and cyclic redundancy checks, networks can identify and correct errors that occur during data transmission. These methods help maintain the integrity of the data, prevent communication problems, and ensure that information is delivered correctly from one device to another. This reliability is essential for the smooth operation of all digital communications and applications that rely on network connectivity.

Hamming Code - Error Detection & Error Correction

-> 1011 -> first we will caluclate the parity p1, p2, p3 on sender side by using even or odd parity.
-> p1 - > d1, d3, d5, 6, 7
-> p2 -> 2, 3, 6, 7
-> p4 -> 4,5,6,7,12,13,14,15 ....

The hamming code technique, which is an error-detection and error-correction technique, was proposed by R.W. Hamming. Whenever a d a t a packet is transmitted over a network, there are possibilities that the data bits may get lost or damaged during transmission.

Let’s understand the Hamming code concept with an example.

Let’s say you have received a **7-bit Hamming code** which is 1011011.

First , let us talk about the redundant bits.

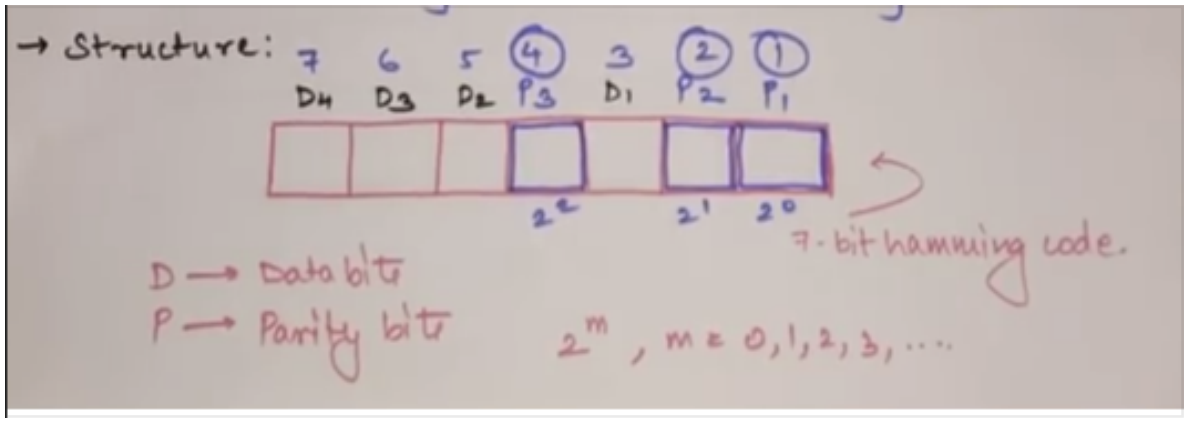
Redundant Bits :
These are some extra binary bits that are not part of the origin al data, but they are generated & added to the original data bit. All this is done to ensure that the data bits don't get damaged and if they do, we can recover them.
Now the question arises, how do we determine the number of redundant bits to be added?

Let there are 4 Data bits

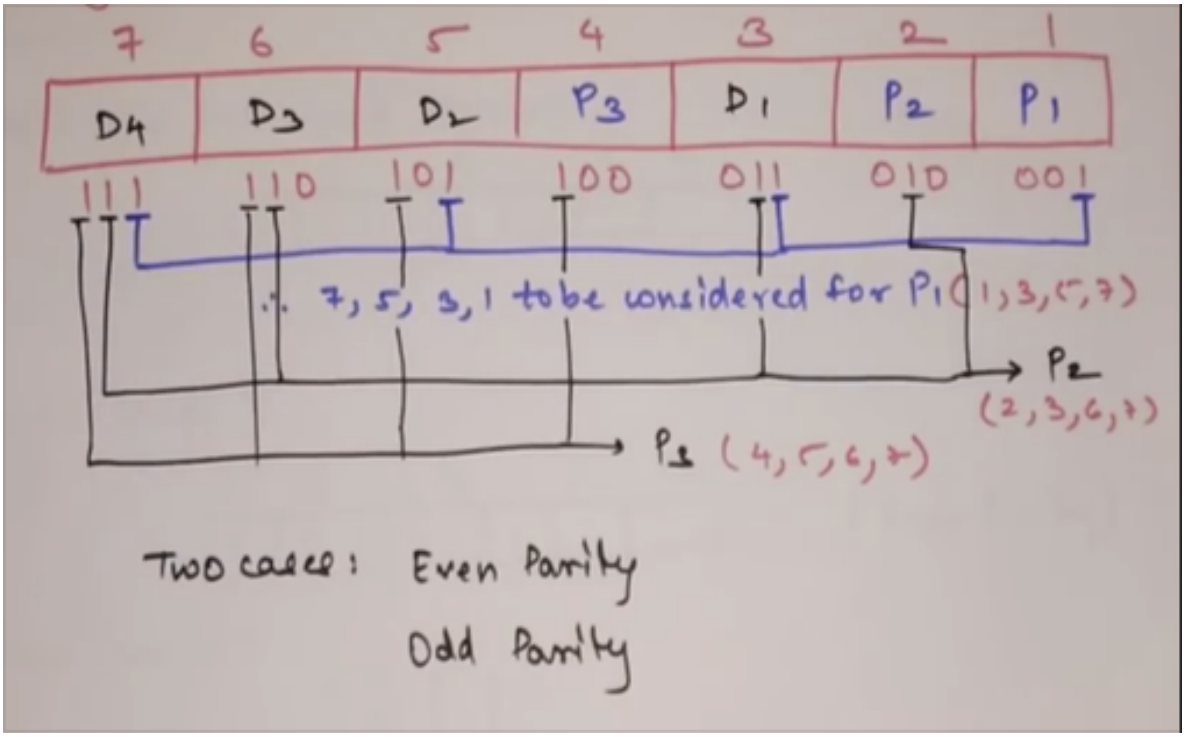
The seven bits are bit 7, bit 6, 5,4,3,2,1.
In this, the redundant bits are placed at the positions that are numbered corresponding to the power of 2, i.e., 1,2,4,8, ... Thus the locations of data bit and redundant bit are D4, D3, D2, P3, D1, P2, P1.

Hamming Codes
-> Hamming codes are linear block codes.
-> It is an error correcting code.
-> Parity bits are used here.
-> They are inserted in between the data bits.
-> The most commonly used is 7-bit hamming code.

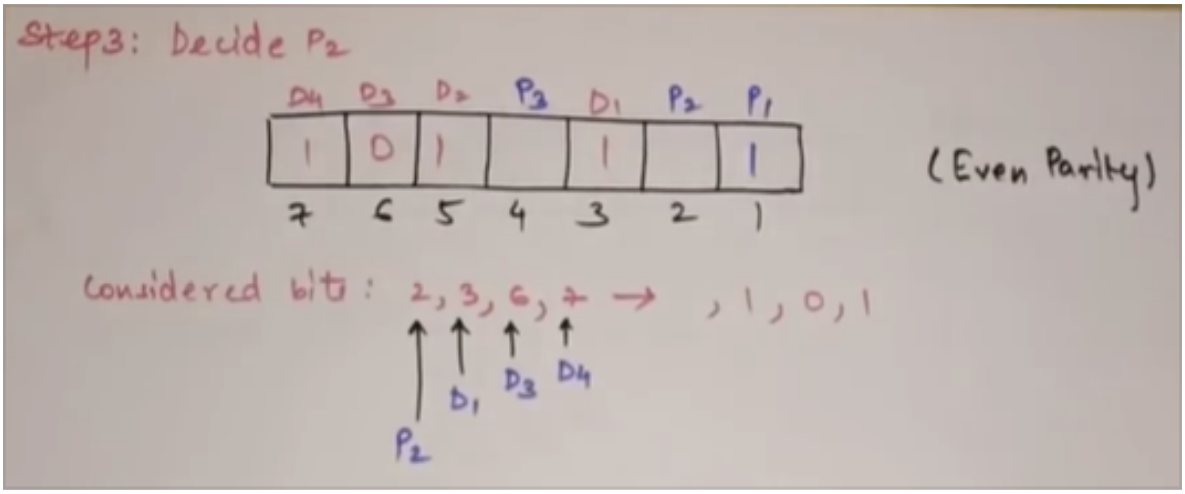
Step 1 : Structure :



Step 2 : Deciding the values for parity bits :



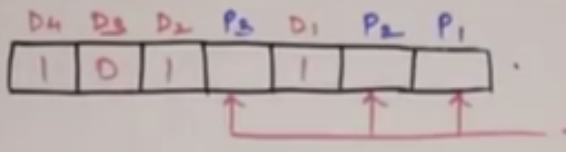
Step 3: Decide P2 :



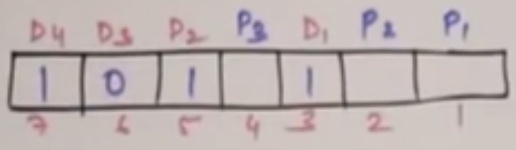
A bit word 1011 is to be transmitted construct the even parity 7-bit hamming code for this data.

→

Step1: The codeword format.

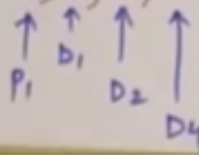


Step 2: Decide P_1

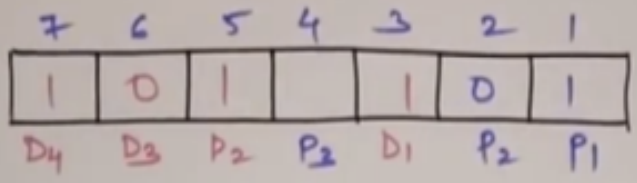


(Even Parity)

considered bits : 1, 3, 5, 7 = 1, 1, 1, 1

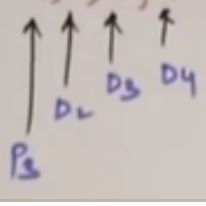


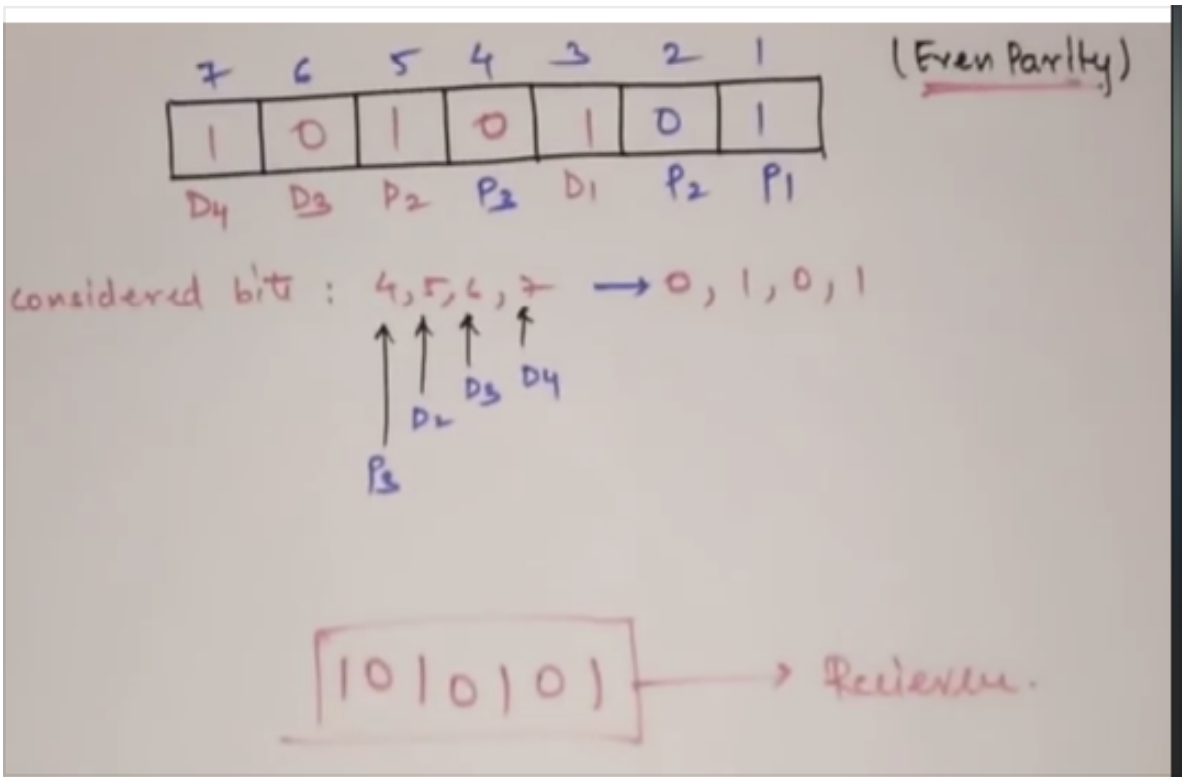
Step4: Decide P3



(Even Parity)

considered bits : 4, 5, 6, 7 → , 1, 0, 1





Example problem 1
Encode a binary word 1101 int o the even parity hamming code.
Thus the encoded 9-bit hamming code is 111001101.

How to detect and correct the error in the hamming code?

After receiving the encoded message, each parity bit along with its corresponding group of bits are checked for proper parity. While checking, the correct result of individual parity is marked as 0 and the wrong result is marked as 1.

After checking all the parity bits, a binary word is formed taking the result bits for P1 as LSB. So formed binary word gives the bit location, where there is an error.

Extra points :
Hamming Distance : The number of bits in which two codewords vary is called hamming distance.

1011001
1101101
———-
3 bits vary

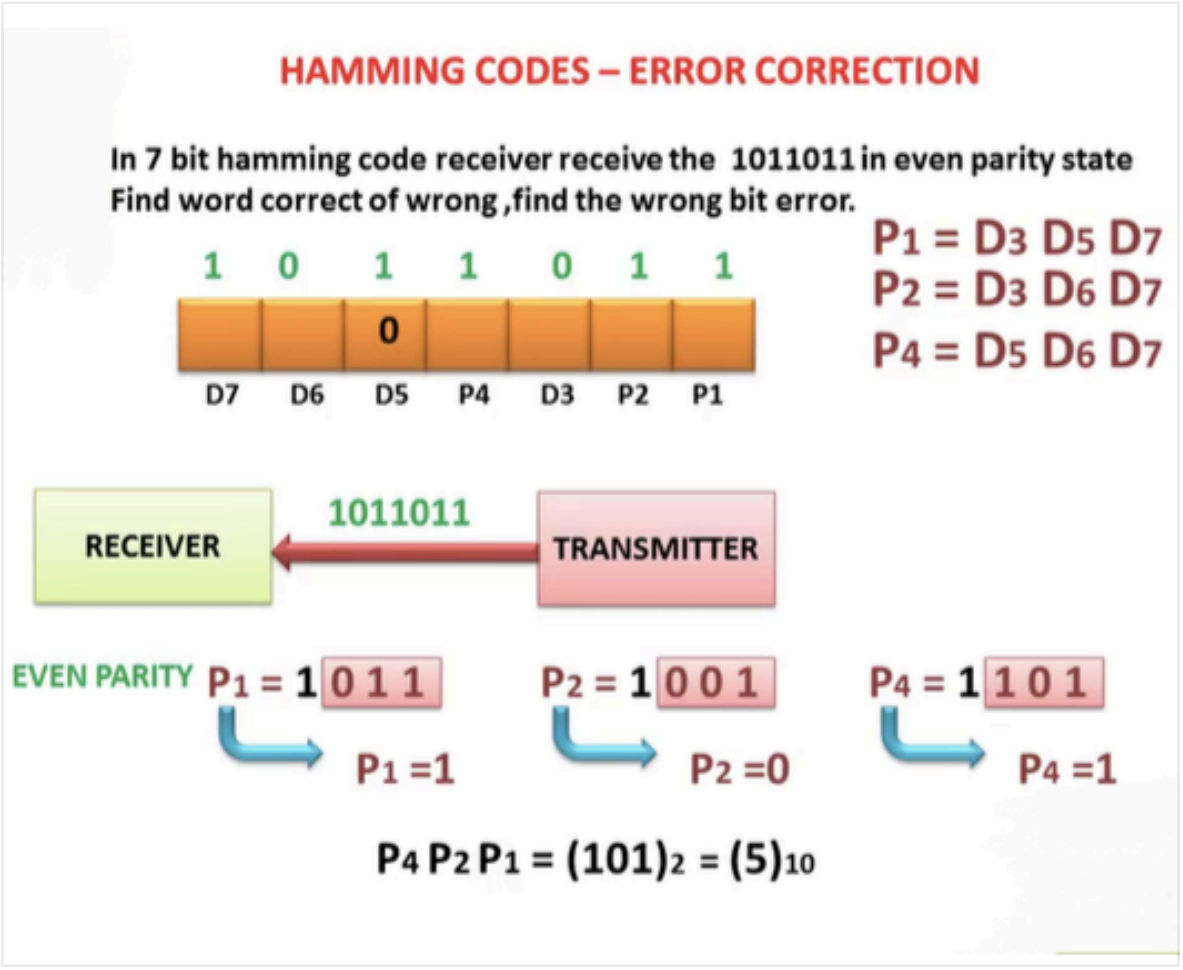
I.e. 3 Hamming distance

Example 1 Find the error in code 1011011 which has even parity decoding and correct it using hamming code error detection.

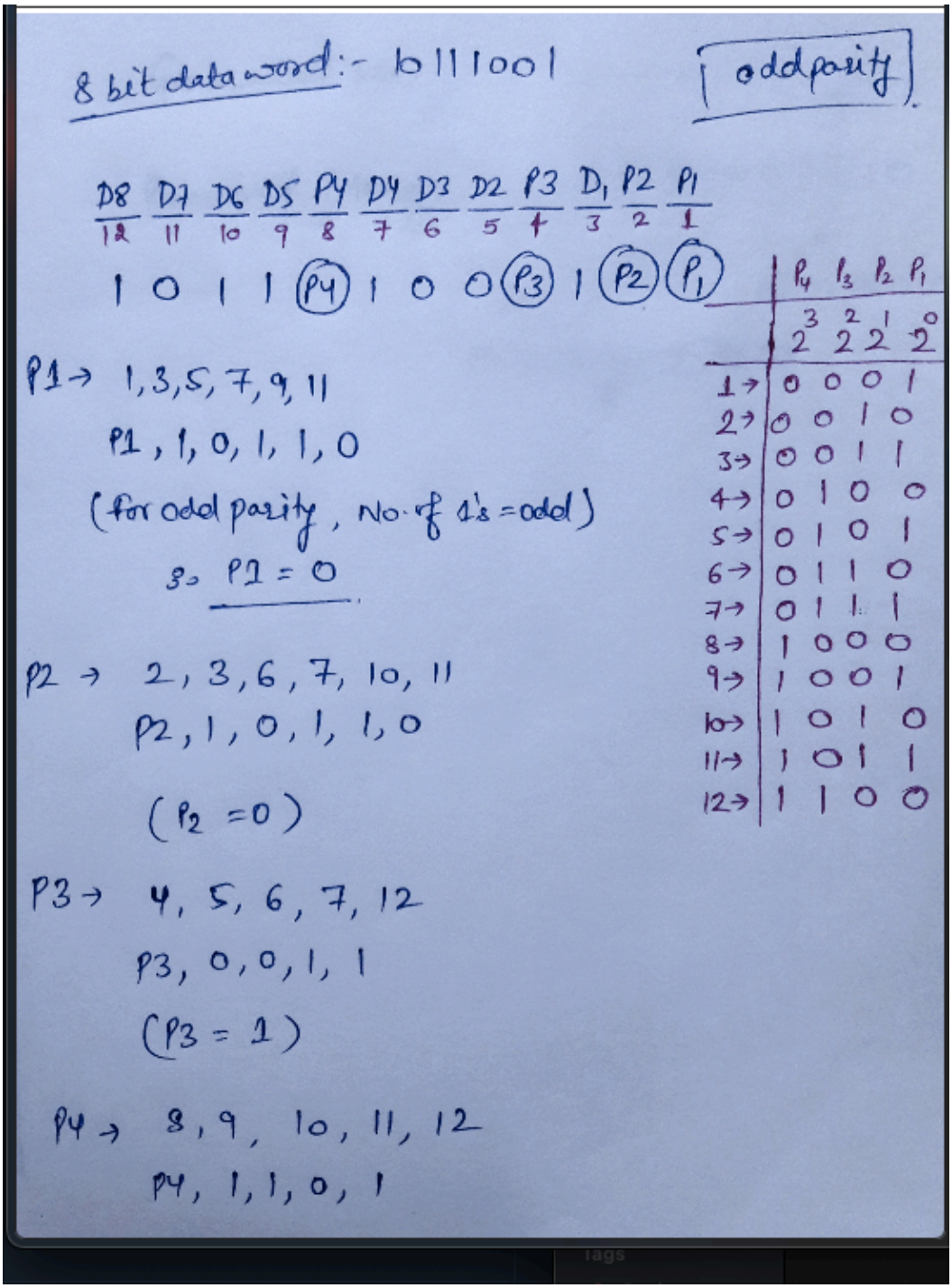
Sol : d7 d6 d5 p4 d3 p2 p1
1. 0. 1. 1. 0 1. 1
7 6. 5. 4. 3. 2. 1

Checking Errors :
For P1 -> Take one skip one. 1, 3 5 7
Code - 1 0 1 1 -> Odd Parity which is incorrect
P1 value is 1.
P2 -> Take two skip two - 2,3,6,7
Code - 1 0 0 1 ->. Even parity which is correct
P2 value is 0.
P4 -> Take four skip 4 - 4,5,6,7
Code - 1 1 0 1 -> Odd Parity which is incorrect
P4 value is 1.

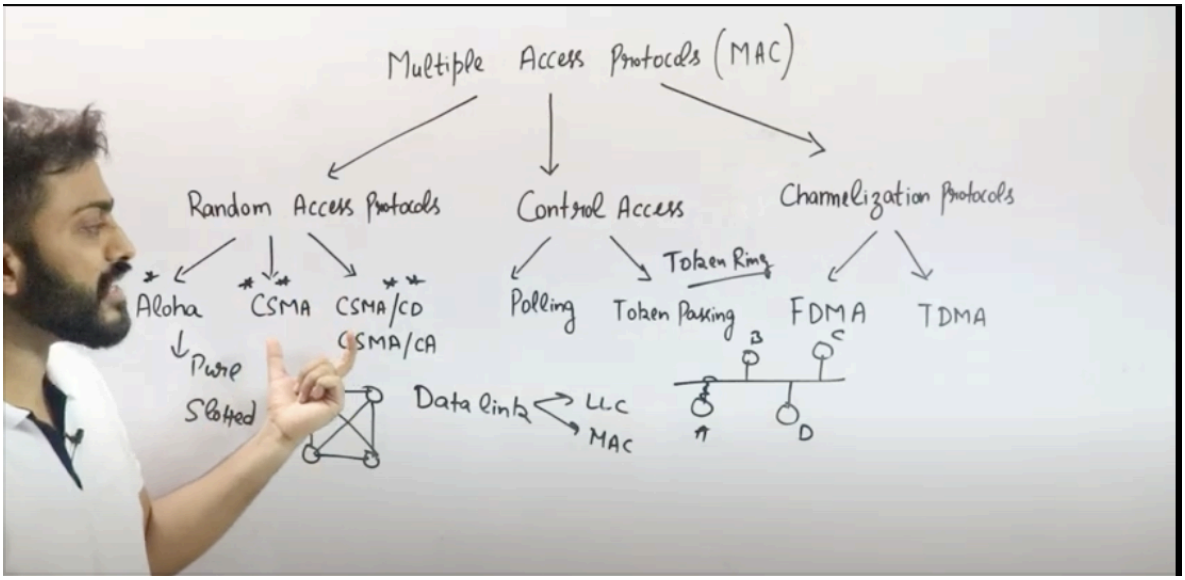
Now to correct the code , because P4 & P1 is not equal to zero receiver code is wrong
Correcting Error - Error - 1 0 1. = 5 decimal value
So, It says the error is at 5th position.
1011011 -> invert the 5th position binary val
1001011 is the answer



Ex - Given the 8-bit data word 10111001, generate the 12-bit composite word for the hamming code that detects and corrects single bit error using odd parity at the receiving end, it can be received as 101101001110 find out the error position



Media Access Control Protocol

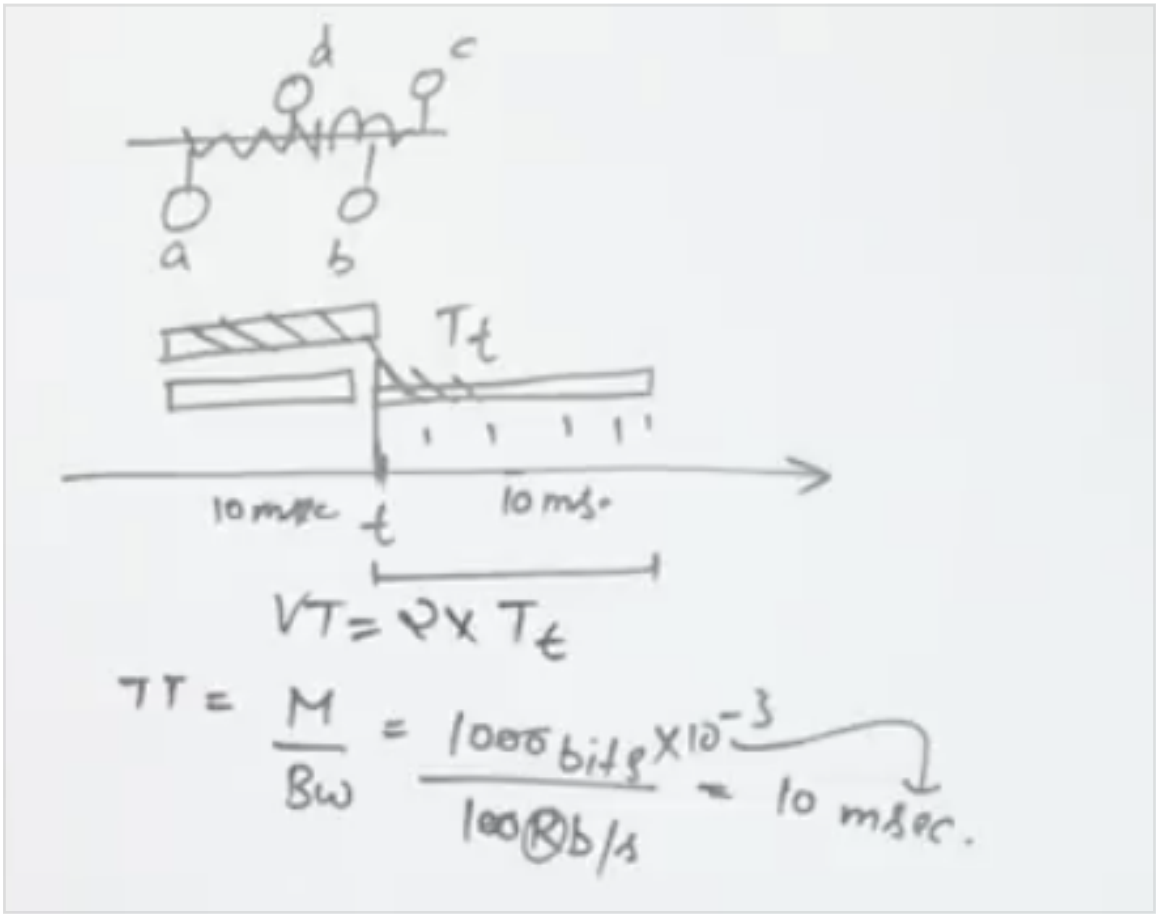


What is aloha?

Aloha is designed for wireless LAN (Local Area Network) but can also be used in a shared medium to transmit data. In aloha, any station can transmit data to a channel at any time. It does not require any carrier sensing.

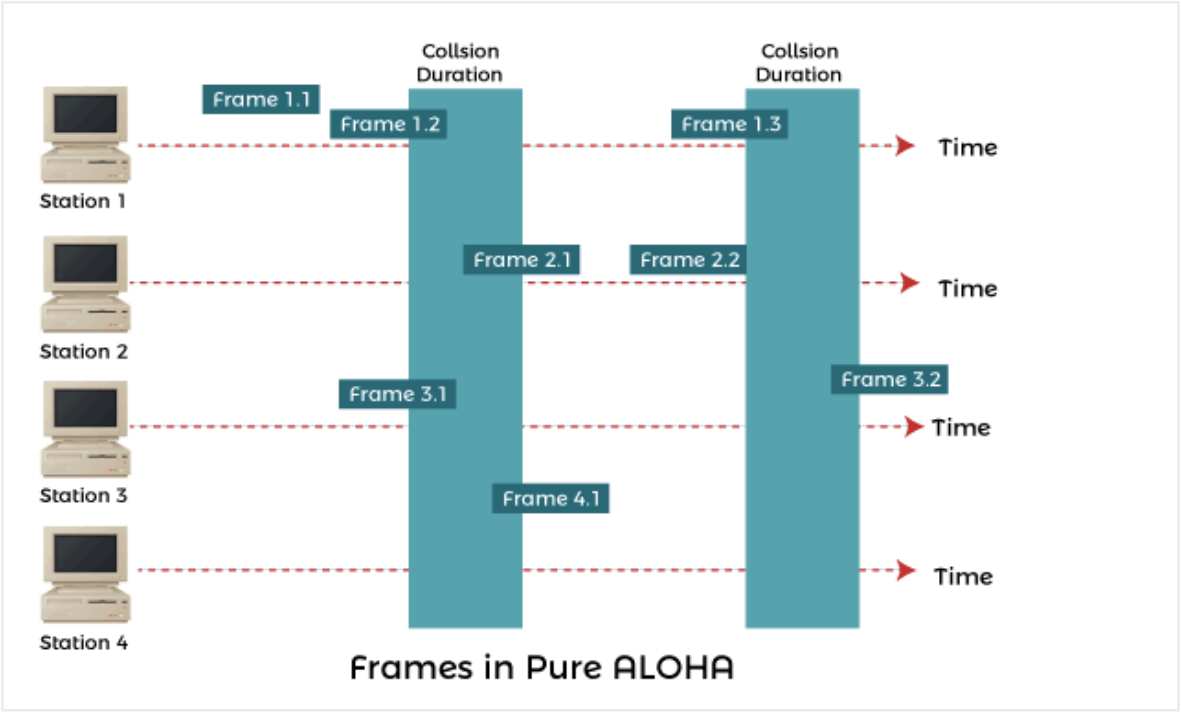
Pure Aloha

-> Random Access Protocol - any use can send data to anyone at anytime hence high changes of collision
-> We won't sense the carrier or medium that either someone already sending or receiving data.
-> Acknowledgement will be sent after receiving the data if ack or data lost retransmission will take place
-> Lan based won't use now a days ...
-> Only transmission time, no propagation time
-> Transmission time = Message/Bandwidth = 1000bits msg/ 100kb/s = 10m/sec
-> Venerable time = 2 × TT (transmission time)
During this transmission time if others will send data even in the last time, collision will occur



-> Efficiency N = G * e^-2G
here -2 is due to venerable time,
G = no. of station wants to send data in that transmission time.
DN/DG = G * e^-2G
= -2 + e^-2G (1) = 0
= G = 1/2 which means the efficiency will be higher if 1/2 no. of stations send data in tt time
Putting G = 1/2
= 1/2 * 2^-2*1/2
= 18.4%

Pure aloha is used when data is available for sending over a channel at stations. In pure Aloha, when each station transmits data to a channel without checking whether the channel is idle or not, the chances of collision may occur, and the data frame can be lost.



Frames in Pure ALOHA

When a station transmits the data frame to a channel without checking whether the channel is free or not, there will be a possibility of the collision of data frames. Station expects the acknowledgement from the receiver, and if the acknowledgement of the frame is received at the specified time, then it will be OK; otherwise, the station assumes that the frame is destroyed. Then station waits for a random amount of time, and after that, it retransmits the frame until all the data are successfully transmitted to the receiver.

Slotted Aloha

-> Timeline is divided in slots and each of them can only send transmission at the begging of slop not when in between of end.
-> vulnerable time vt = Tt -> A device can start sharing only at the beginning of the time slot
-> efficiency will be maximum if on a single slot only single user will transmit otherwise collision will occur
-> less changes of collision
-> Efficiency N = G * e^-G

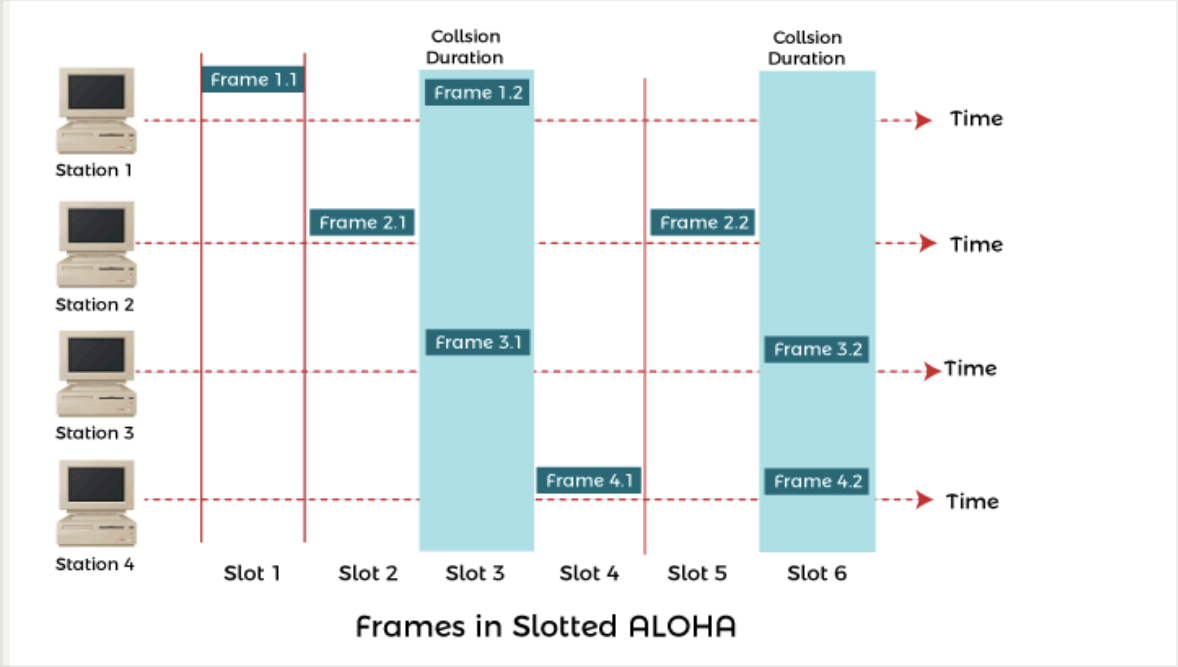
$$Dn/DG = -1 + e^{-G} = 0$$

$$= G = 1$$

Putting value of G

$$N = 1 \times e^{-1} = 1/e = 2.71 = 36.8\%$$

There is a high possibility of frame hitting in pure aloha, so slotted aloha is designed to overcome it. Unlike pure aloha, slotted aloha does not allow the transmission of data whenever the station wants to send it.



In slotted Aloha, the shared channel is divided into a fixed time interval called slots. So that, if a station wants to send a frame to a shared channel, the frame can only be sent at the beginning of the slot, and only one frame is allowed to be sent to each slot. If the station is failed to send the data, it has to wait until the next slot. However, there is still a possibility of a collision because suppose if two stations try to send a frame at the beginning of the time slot.

Pure aloha v/s slotted aloha

Now, let's see the comparison chart between pure aloha and slotted aloha. We are comparing both terms on the basis of characteristics to make the topic more clear and understandable.

S.no.	On the basis of	Pure Aloha	Slotted Aloha

1.	**Basic**	In pure aloha, data can be transmitted at any time by any station.	In slotted aloha, data can be transmitted at the beginning of the time slot.
2.	**Introduced by**	It was introduced under the leadership of Norman Abramson in 1970 at the University of Hawaii.	It was introduced by Robert in 1972 to improve pure aloha's capacity.
3.	**Time**	Time is not synchronized in pure aloha. Time is continuous in it.	Time is globally synchronized in slotted aloha. Time is discrete in it.
4.	**Number of collisions**	It does not decrease the number of collisions to half.	On the other hand, slotted aloha enhances the efficiency of pure aloha. It decreases the number of collisions to half.
5.	**Vulnerable time**	In pure aloha, the vulnerable time is = 2 x Tt	Whereas, in slotted aloha, the vulnerable time is = Tt

6.	**Successful transmission**	In pure aloha, the probability of the successful transmission of the frame is - S = G * e-2G	In slotted aloha, the probability of the successful transmission of the frame is - S = G * e-G
7.	**Throughput**	The maximum throughput in pure aloha is about 18%.	The maximum throughput in slotted aloha is about 37%.

Conclusion

From the above discussion, it can be said that slotted aloha is somewhat better than pure aloha. It is because there is less possibility of collision in slotted aloha.

So, that's all about the article. Hope it will be helpful and informative to you.

CSMA (Carrier Sense Multiple Access) Difference between 1-persistent, p-persistent and Non-persistent CSMA

Carrier Sense Multiple Access (CSMA) is like a set of rules for how devices share the same communication channel to avoid collisions. Imagine you're in a room with several people trying to talk at once. In 1-persistent CSMA everyone keeps checking to see if the room is free and speaks up immediately when it is which often leads to people talking over each other. In p-persistent CSMA the room is divided into time slots and people decide to speak based on a set chance reducing the chances of talking over each other. Non-persistent CSMA is like waiting and then checking again randomly so people aren't constantly trying to talk at once which cuts down on collisions but might delay the conversation. Each method has its trade-offs with 1-persistent being quick but collision-prone p-persistent balancing speed and collisions and non-persistent minimizing collisions at the cost of potential delays.

1. 1-Persistent CSMA

In 1-persistent CSMA, the station continuously senses the channel to check its state i.e. idle or busy so that it can transfer data or not. In case when the channel is busy, the station will wait for the channel to become idle. When the station finds an idle channel, it transmits the frame to the channel without any delay. It transmits the frame with probability 1. Due to probability 1, it is called 1-persistent CSMA. The problem with this method is that there are a large number of chances for a collision it is because there is a chance w bhi hen two or more stations find the channel in an idle state and the transmit frames at the same time. On the time when a collision occurs, the station has to wait for a random time for the channel to be idle and to start again. Another chance of collision is that suppose two stations are in connections and data frames are being transferred, then all the intermediate stations will sense that and won't transmit the data, if the data frame transmission is ended then all the intermediate stations sense that and sends the data frames at a time, which makes collision rate even worse.

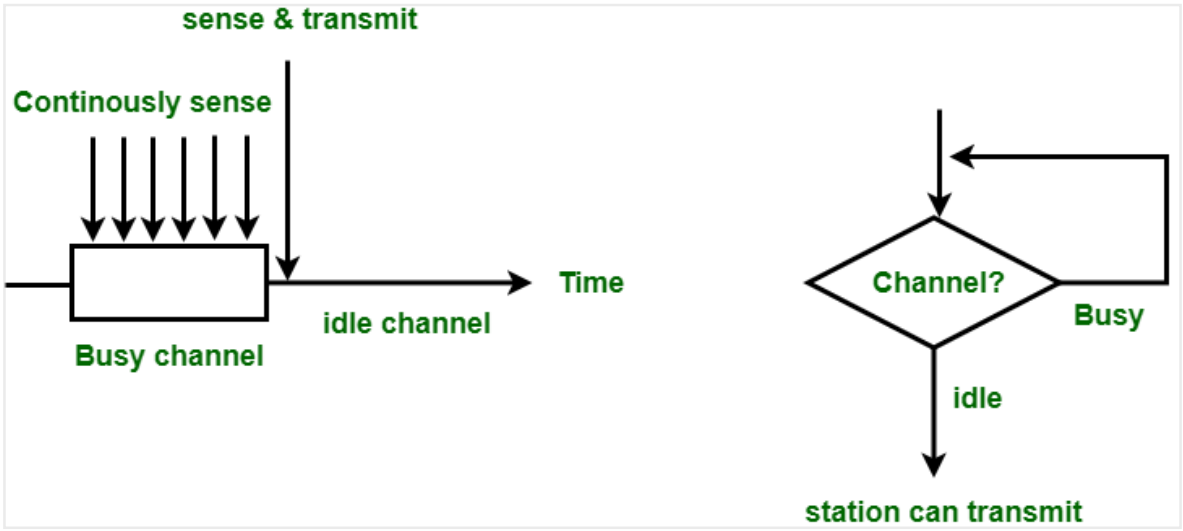


Figure – 1-persistent CSMA

2. P-Persistent CSMA

This is the method that is used when channel has time-slots and that time-slot duration is equal to or greater than the maximum propagation delay time. When the station is ready to send the frames, it will sense the channel. If the channel found to be busy, the channel will wait for the next slot. If the channel found to be idle, it transmits the frame with probability p, thus for the left probability i.e. q which is equal to 1-p the station will wait for the beginning of the next time

slot. In case, when the next slot is also found idle it will transmit or wait again with the probabilities p and q. This process is repeated until either the frame gets transmitted or another station has started transmitting.

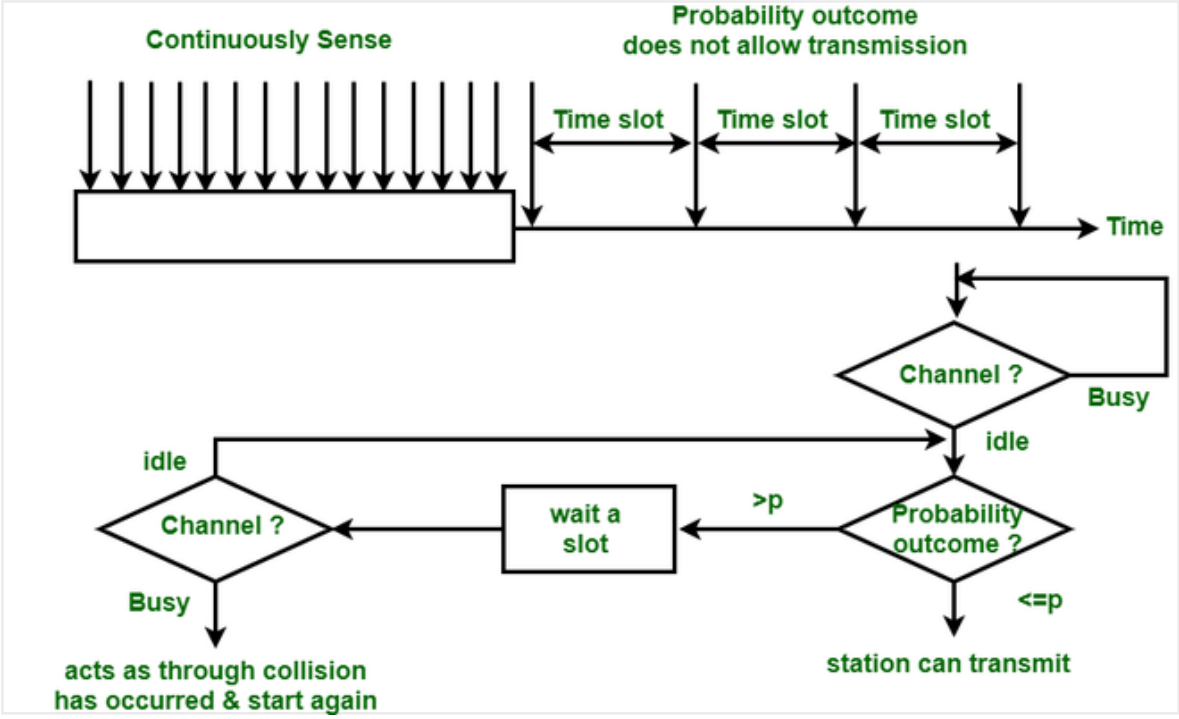


Figure – p-persistent CSMA

3. Non-Persistent CSMA

In this method, the station that has frames to send, only that station senses for the channel. In case of an idle channel, it will send frame immediately to that channel. In case when the channel is found busy, it will wait for the random time and again sense for the state of the station whether idle or busy. In this method, the station does not immediately sense for the channel for only the purpose of capturing it when it detects the end of the previous transmission. The main advantage of using this method is that it reduces the chances of collision. The problem with this is that it reduces the efficiency of the network.

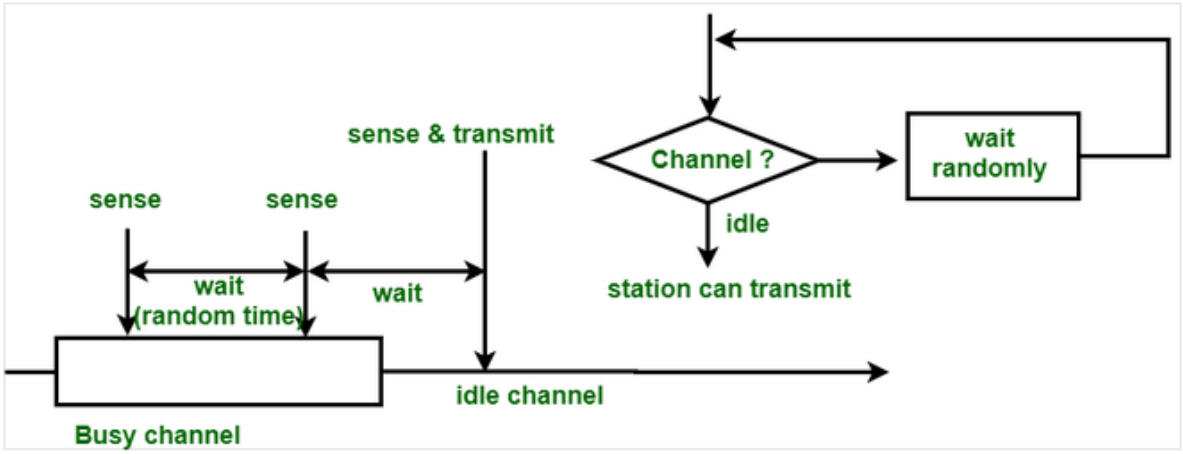


Figure – Non-persistent CSMA

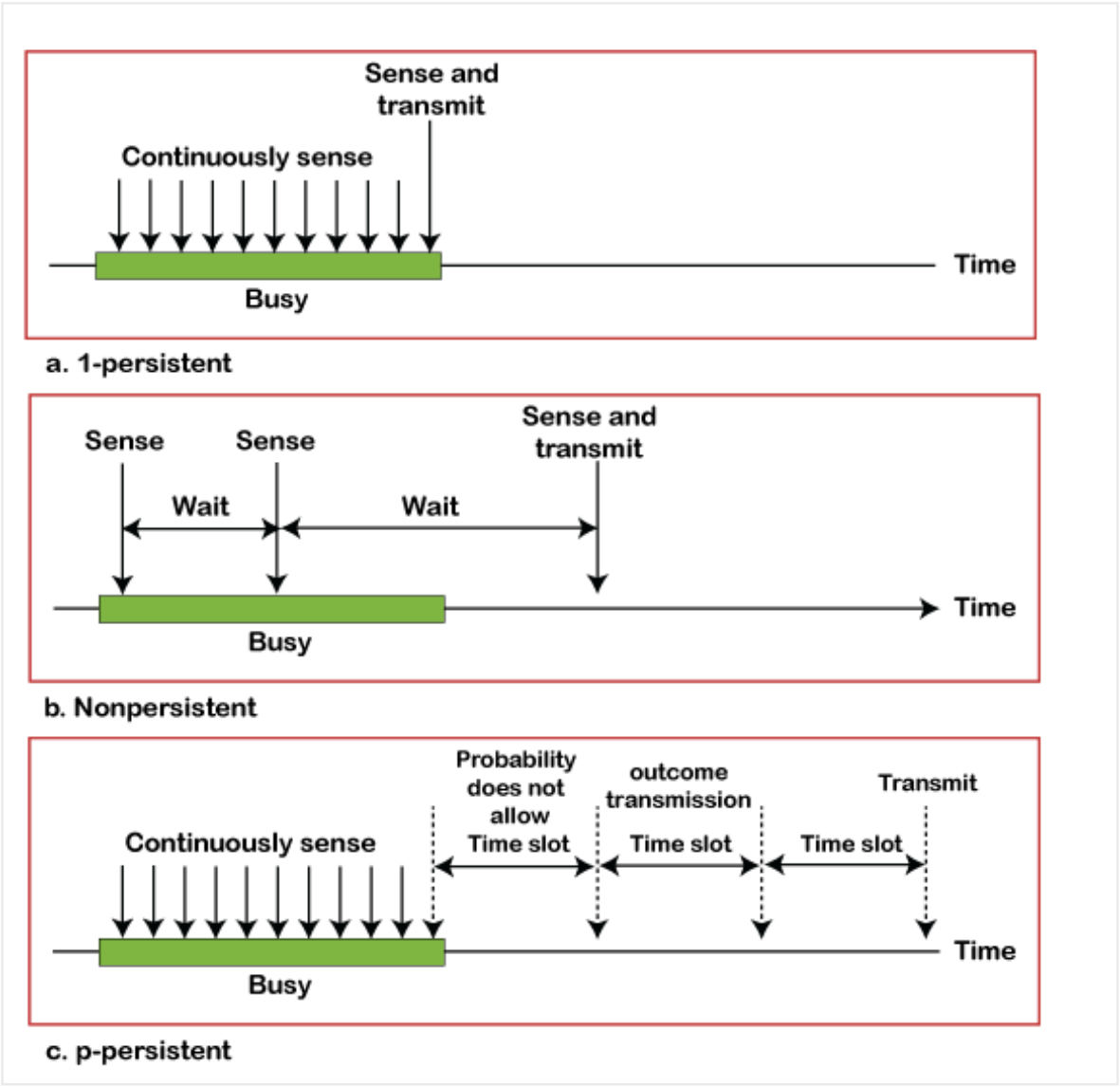
Difference Between 1-persistent, p-persistent and Non-persistent CSMA

Parameter	1-persistent CSMA	p-persistent CSMA	Non-persistent CSMA
Carrier Sense	It sends with the probability of 1 when channel is idle.	It sends with the probability of p when channel is idle.	It send when channel is idle.
Waiting	It continuously senses the channel or carrier.	It waits for the next time slot.	It will wait for the random amount of time to check the carrier.
Chances of Collision	There is highest chances of collision in this.	Less chances as compared to 1-persistence and non-persistence.	Less chances as compared to 1-persistence but more than the p-persistence.

Utilization	It's utilization is above ALOHA as frames are only sent when the channel is idle.	It's utilization is depend upon the probability p.	It's utilization is above 1-persistent as not all the stations constantly check the channel at the same time.
Delay Low Load	It is low as frames are sent when the channel become idle.	It is large when p is small as station will not always send when channel is idle.	It is small as station will send whenever channel is found idle but longer than 1-persistent since it checks for the random time when busy.
Delay High Load	It is high due to collision.	It is large when the probability p of sending is small when channel is idle and channel is rarely idle.	It is longer than 1-persistent as channel is checked randomly when busy.

Conclusion

1-persistent CSMA aggressively transmits data as soon as the channel is idle leading to high collision rates but quick data transmission under low load. p-persistent CSMA balances between transmission and waiting reducing collisions by introducing a probabilistic approach making it suitable for networks with time-slots. Non-persistent CSMA minimizes collisions by waiting for random periods before rechecking the channel though this can lead to longer delays. Understanding these differences helps in selecting the appropriate method for specific network environments ensuring optimal performance and efficiency.

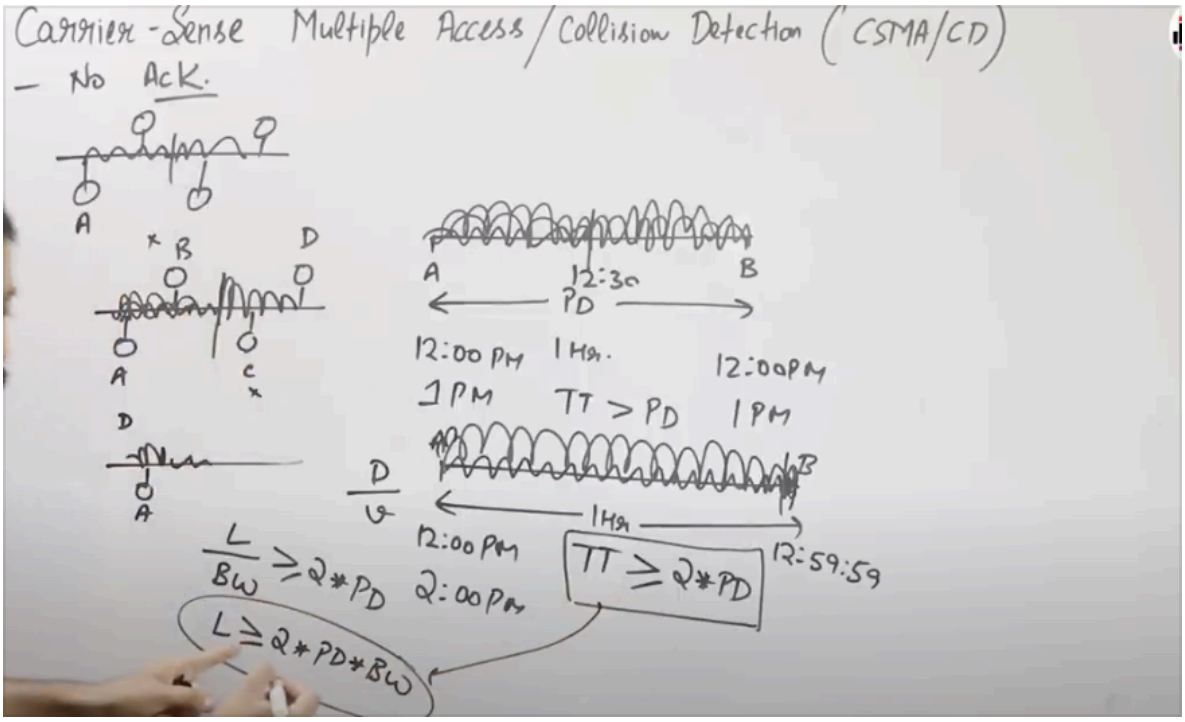


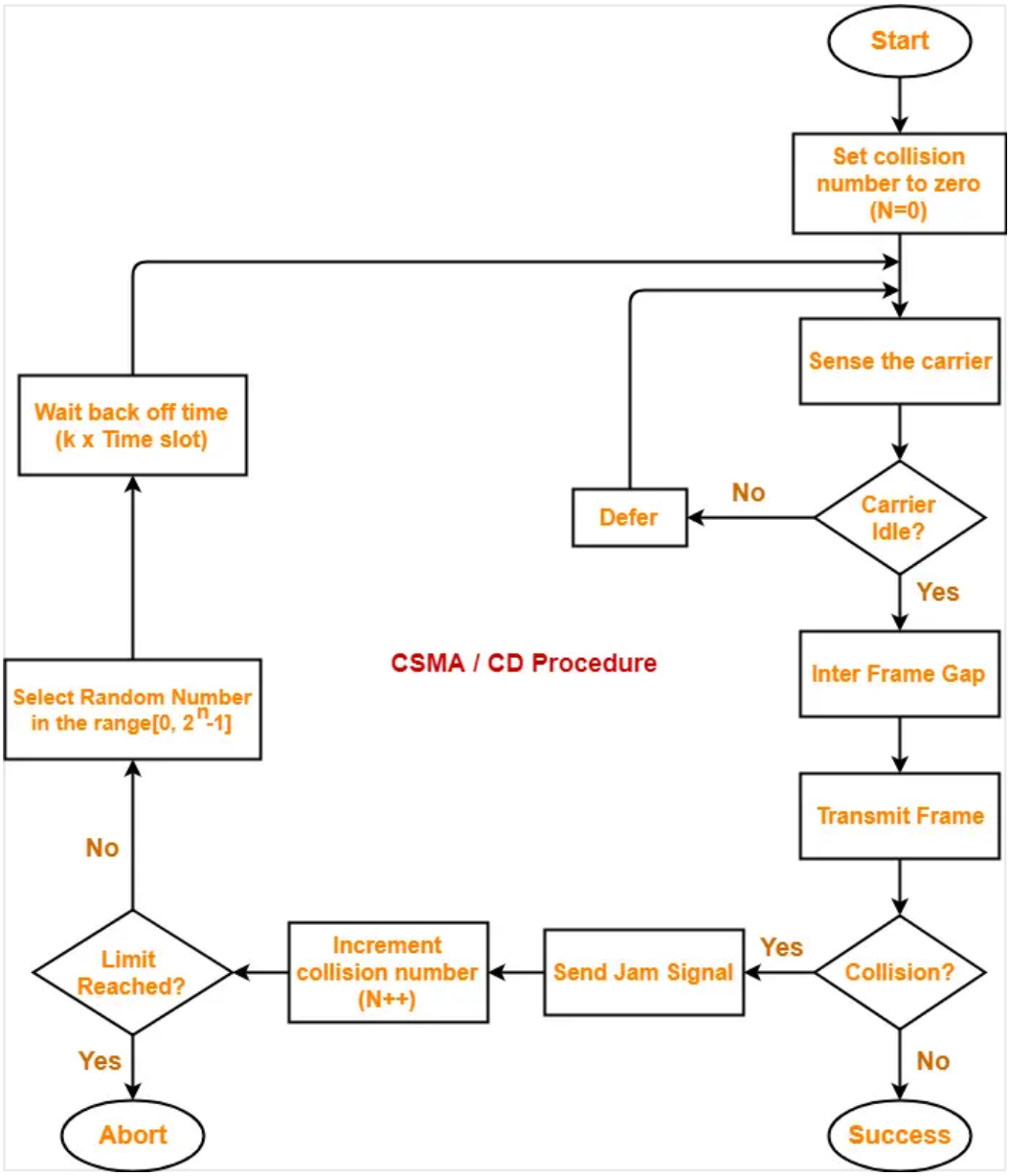
Type In Carrier Sense Multiple Access (CSMA)

1. Carrier Sense Multiple Access with Collision Detection (CSMA/CD):

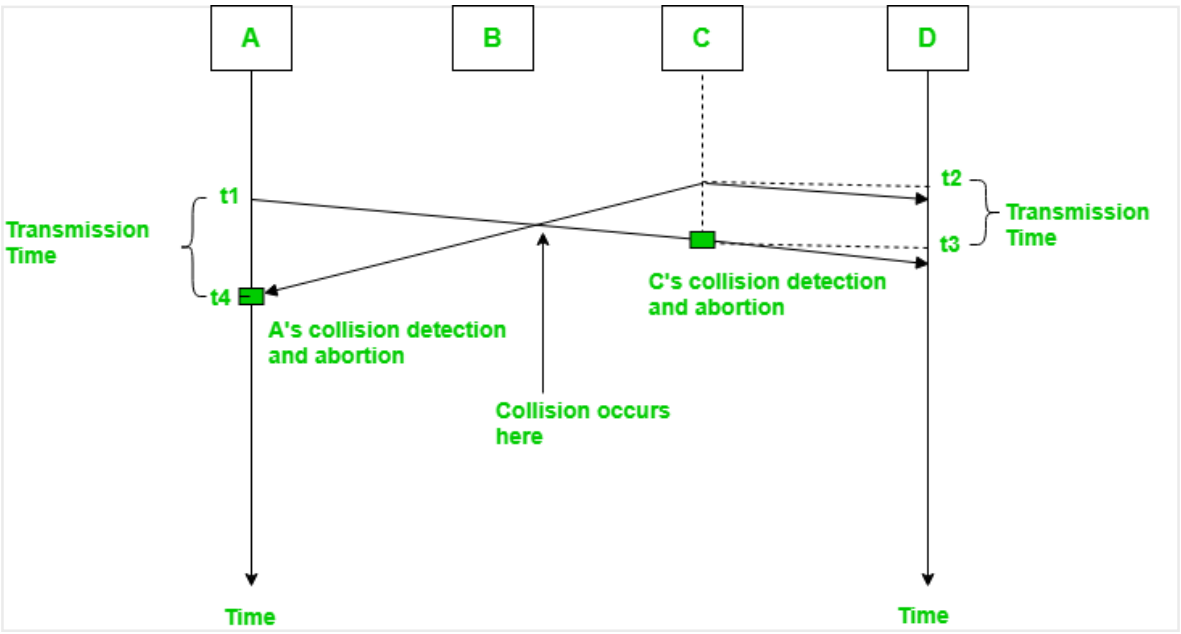
-> There is no acknowledgment system as we talking about ethernet and stuff ... already have data in the channel and adding acknowledgement will further increase the number of collisions.
-> Here the user will sense if data is collided when the collision data comes near it while he is still transferring the data
-> If user stops transferring data before collision data comes back in that situation user can’t decide either the collision data is of his own or some other users

-> so the transmission time should be TT >= 2 * Propagation Delay.
Tt = Length of msg / Bandwidth.
Propogation Delay = Distance / velocity
-> Efficiency N = 1/1 + 6.44a
a= PD/TT





In this method, a station monitors the medium after it sends a frame to see if the transmission was successful. If successful, the transmission is finished, if not, the frame is sent again.



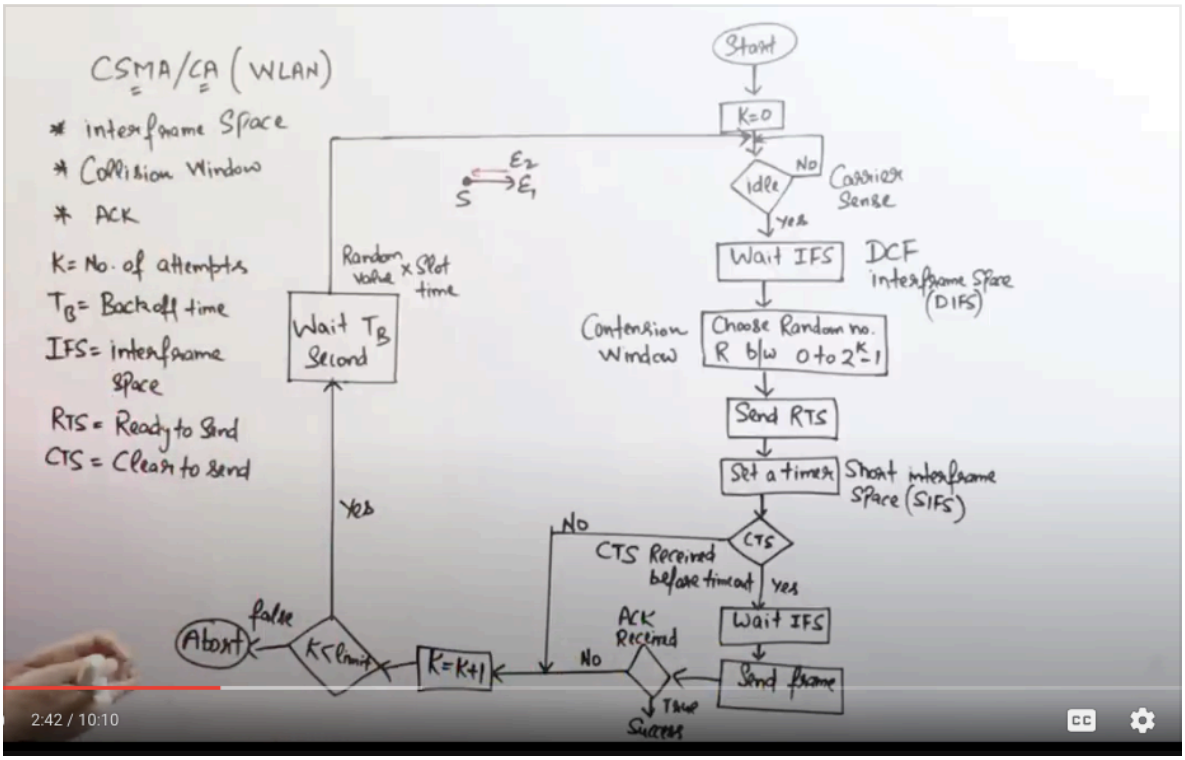
In the diagram, *starts* sending the first bit of its frame at t1 and since C sees the channel idle at t2, starts sending its frame at t2. C detects A's frame at t3 and aborts transmission. A detects C's frame at t4 and aborts its transmission. Transmission time for C's frame is, therefore, t3-t2 and for A's frame is t4-t1
So, the **frame transmission time (Tfr) should be at least twice the maximum propagation time (Tp)**. This can be deduced when the two stations involved in a collision are a maximum distance apart.
Process: The entire process of collision detection can be explained as follows:

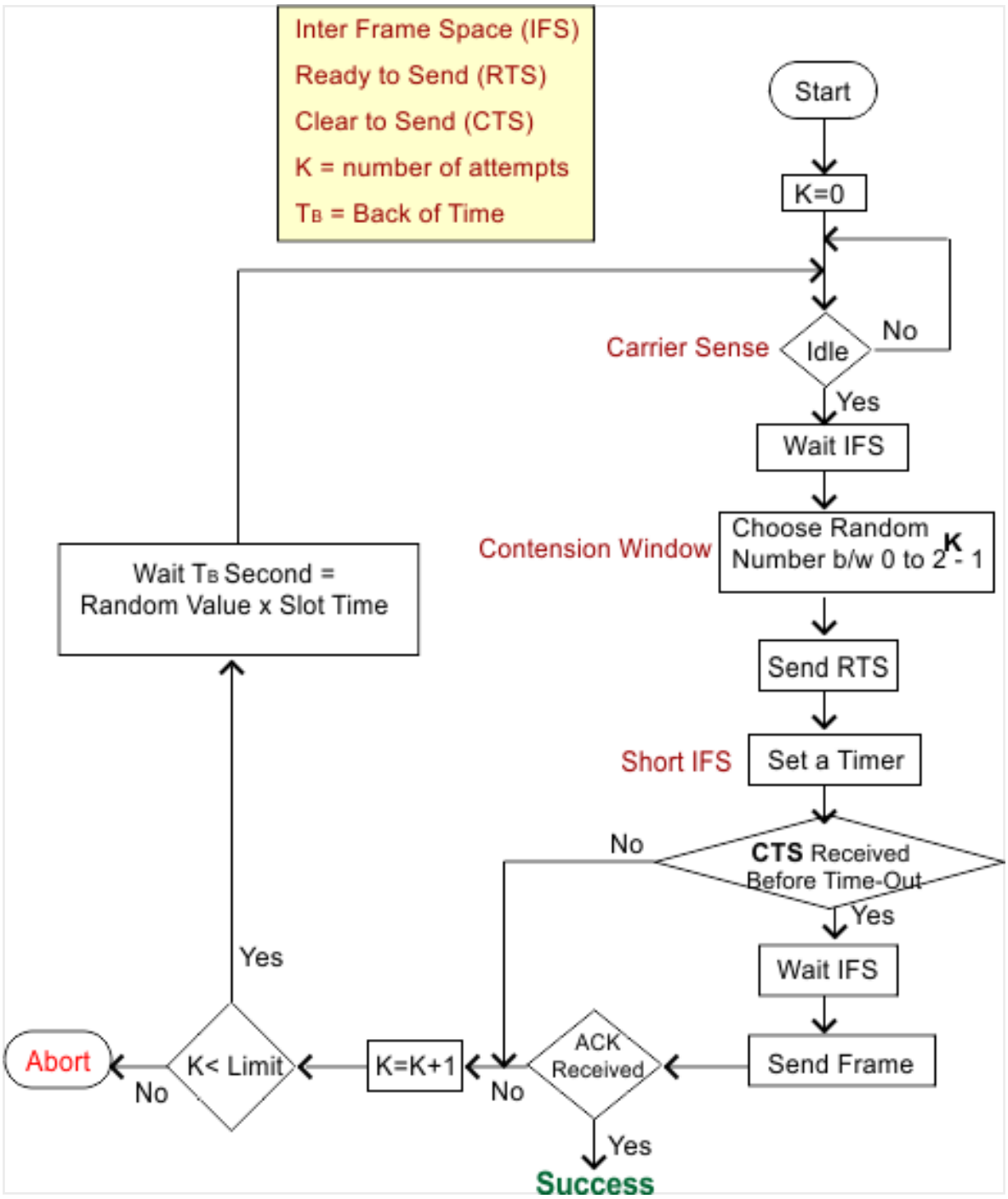
Throughput and Efficiency: The throughput of CSMA/CD is much greater than pure or slotted ALOHA.

- For the 1-persistent method, throughput is 50% when G=1.
- For the non-persistent method, throughput can go up to 90%.

2. Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) –

-> In Carrier Sense Multiple Access / Collision avoidance - > This one is used in WLAN or say wifi etc wireless tech because we can't detect the collision here as we do in csma/cd.
-> DCF -> Distributed coordination function
IFS -> Interframe Space
RTS -> Ready To Send
CTS -> Clear to send


CSMA/CA (WLAN)
* Interframe Space
* Collision Window
* ACK
K= No. of attempts
T_B= Backoff time
IFS= Interframe Space
RTS = Ready to Send
CTS = Clear to send
Start
K=0
Idle
No
Carrier Sense
yes
Wait IFS
DCF Interframe Space (DIFS)
Contension Window
Choose Random no. R b/w 0 to 2^K-1
Send RTS
Set a timer
Short interframe Space (SIFS)
CTS
CTS Received before timeout
No
yes
Wait IFS
Send frame
ACK Received
No
True
Success
K=K+1
K<limit
false
Abort
yes
Random value x Slot time
Wait T_B Second
E_2
E_1
S
2:42 / 10:10



The basic idea behind CSMA/CA is that the station should be able to receive while transmitting to detect a collision from different stations. In wired networks, if a collision has occurred then the energy of the received signal almost doubles, and the station can sense the possibility of collision. In the case of wireless networks, most of the energy is used for transmission, and the energy of the received signal increases by only 5-10% if a collision occurs. It can't be used by the station to sense collision. Therefore **CSMA/CA has been specially designed for wireless networks**.

These are three types of strategies:

1. **InterFrame Space (IFS):** When a station finds the channel

busy it senses the channel again, when the station finds a channel to be idle it waits for a period of time called **IFS time**. IFS can also be used to define the priority of a station or a frame. Higher the IFS lower is the priority.
2. **Contention Window:** It is the amount of time divided into slots. A station that is ready to send frames chooses a random number of slots as **wait time**.
3. **Acknowledgments:** The positive acknowledgments and time-out timer can help guarantee a successful transmission of the frame.

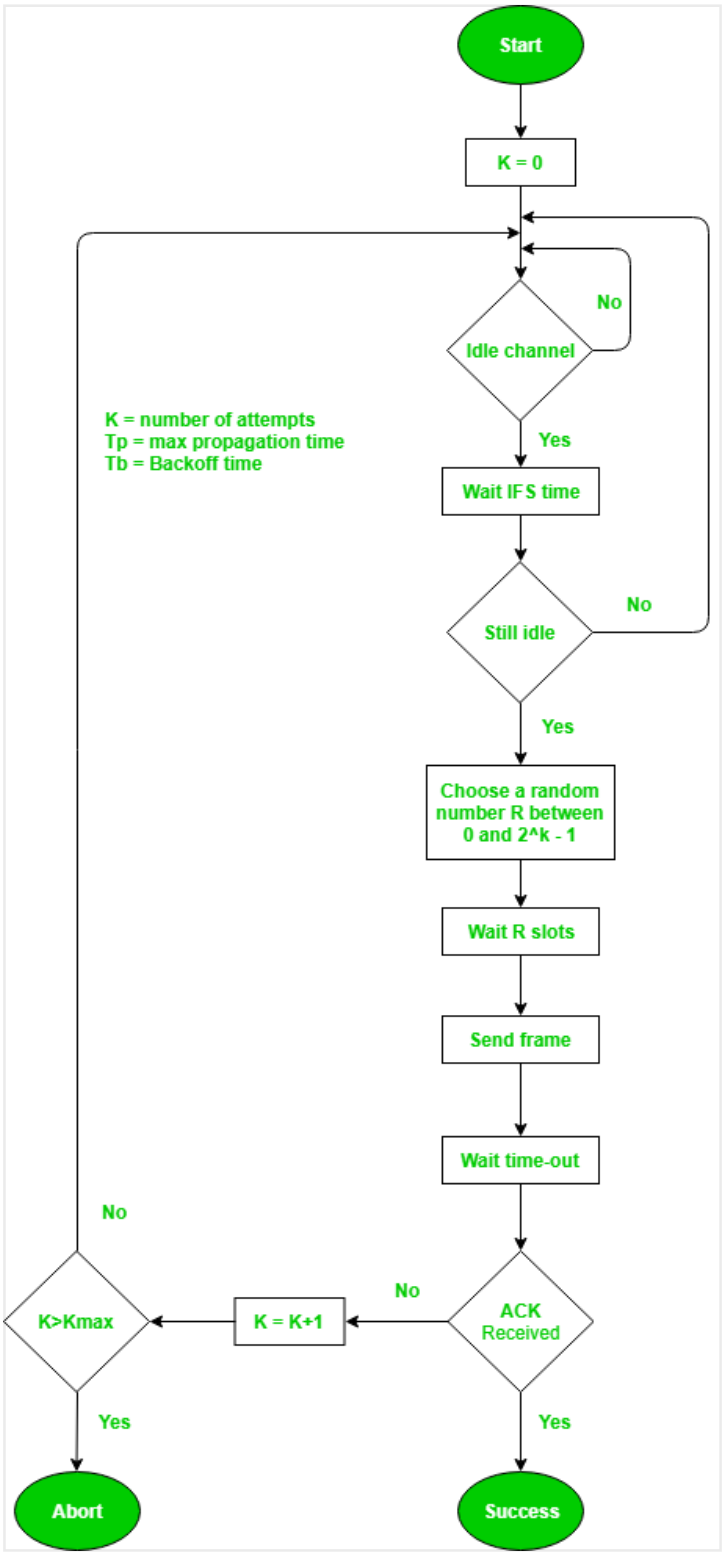
Characteristics of CSMA/CA :

1. **Carrier Sense**: The device listens to the channel before transmitting, to ensure that it is not currently in use by another device.
2. **Multiple Access**: Multiple devices share the same channel and can transmit simultaneously.
3. **Collision Avoidance**: If two or more devices attempt to transmit at the same time, a collision occurs. CSMA/CA uses random backoff time intervals to avoid collisions.
4. **Acknowledgment (ACK)**: After successful transmission, the receiving device sends an ACK to confirm receipt.
5. **Fairness**: The protocol ensures that all devices have equal access to the channel and no single device monopolizes it.
6. **Binary Exponential Backoff**: If a collision occurs, the device waits for a random period of time before attempting to retransmit. The backoff time increases exponentially with each retransmission attempt.
7. **Interframe Spacing**: The protocol requires a minimum amount of time between transmissions to allow the channel to be clear and reduce the likelihood of collisions.
8. **RTS/CTS Handshake**: In some implementations, a Request-To-Send (RTS) and Clear-To-Send (CTS) handshake is used to reserve the channel before transmission. This reduces the chance of collisions and increases efficiency.
9. **Wireless Network Quality**: The performance of CSMA/CA is greatly influenced by the quality of the wireless network, such as the strength of the signal, interference, and network congestion.
10. **Adaptive Behavior**: CSMA/CA can dynamically adjust its

behavior in response to changes in network conditions, ensuring the efficient use of the channel and avoiding congestion.

Overall, CSMA/CA balances the need for efficient use of the shared channel with the need to avoid collisions, leading to reliable and fair communication in a wireless network.

Process: The entire process of collision avoidance can be explained as follows:



Comparisonof various protocols:

Protocol	Transmission behavior	Collision detection method	Efficiency	Use cases

Pure ALOHA	Sends frames immediately	No collision detection	Low	Low-traffic networks
Slotted ALOHA	Sends frames at specific time slots	No collision detection	Better than pure ALOHA	Low-traffic networks
CSMA/CD	Monitors medium after sending a frame, retransmits if necessary	Collision detection by monitoring transmissions	High	Wired networks with moderate to high traffic
CSMA/CA	Monitors medium while transmitting, adjusts behavior to avoid collisions	Collision avoidance through random backoff time intervals	High	Wireless networks with moderate to high traffic and high error rates

Difference Between CSMA/CA and CSMA/CD

CSMA/CD	CSMA/CA
CSMA / CD is effective after a collision.	Whereas CSMA / CA is effective before a collision.
CSMA / CD is used in wired networks.	Whereas CSMA / CA is commonly used in wireless networks.
It only reduces the recovery time.	Whereas CSMA/ CA minimizes the possibility of collision.
CSMA / CD resends the data frame whenever a conflict occurs.	Whereas CSMA / CA will first transmit the intent to send for data transmission.

CSMA / CD is used in 802.3 standard.	While CSMA / CA is used in 802.11 standard.
It is more efficient than simple CSMA(Carrier Sense Multiple Access).	While it is similar to simple CSMA(Carrier Sense Multiple Access).
It is the type of CSMA to detect the collision on a shared channel.	It is the type of CSMA to avoid collision on a shared channel.
It is work in MAC layer.	It is also work in MAC layer.

Ethernet Frame Format

-> uses csma/cd technology
-> introduced in 1983
-> uses in lan
-> IEEE 802.3

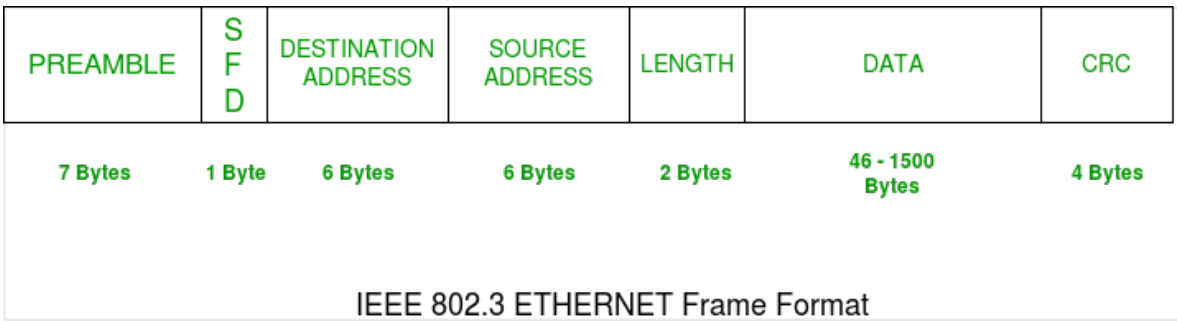
-> Types of ethernet :
1. **10 baseband 2 -> 10mbs and 200 m -> thick ethernet**
2. **10 base 5 -> 10 -> coaxial ethernet**
3. **100mbs**

-> topology used are bus and star
-> 1mbps/ 400gb/sec
-> MAC address - 48 bit - 6bytes
-> first 8 bits preamble and sfd added by physical layer in constant 101010101 format just to alert the receiver to became ready to recive
-> here the source address is sender's MAC address while the destination address is the next hob or next node address not the receiver MAC address which is still unknown to sender because data link layer works in node to node delivery.
-> we use arp packets to gets the MAC address of next node although we already know next node ip address.
-> length would be 2 bytes -> 16bits -> 0 - 65535
-> data length should be atleast 46 bytes because coma/cs needs to have TT >= 2 * PD, Pd = D/velocity
-> CRC is the error detection with cyclic redundancy check ... bits 4bytes

The basic frame format which is required for all MAC implementation

is defined in **IEEE 802.3 standard**. Though several optional formats are being used to extend the protocol's basic capability. Ethernet frame starts with the Preamble and SFD, both work at the physical layer. The ethernet header contains both the Source and Destination MAC address, after which the payload of the frame is present. The last field is CRC which is used to detect the error. Now, let's study each field of basic frame format.

Ethernet (IEEE 802.3) Frame Format



IEEE 802.3 ETHERNET Frame Format

1. **PREAMBLE:** Ethernet frame starts with a 7-Bytes Preamble. This is a pattern of alternative 0's and 1's which indicates starting of the frame and allow sender and receiver to establish bit synchronization. Initially, PRE (Preamble) was introduced to allow for the loss of a few bits due to signal delays. But today's high-speed Ethernet doesn't need a Preamble to protect the frame bits. PRE (Preamble) indicates the receiver that frame is coming and allow the receiver to lock onto the data stream before the actual frame begins.
2. **Start of frame delimiter (SFD):** This is a 1-Byte field that is always set to 10101011. SFD indicates that upcoming bits are starting the frame, which is the destination address. Sometimes SFD is considered part of PRE, this is the reason Preamble is described as 8 Bytes in many places. The SFD warns station or stations that this is the last chance for synchronization.
3. **Destination Address:** This is a 6-Byte field that contains the MAC address of the machine for which data is destined.
4. **Source Address:** This is a 6-Byte field that contains the MAC address of the source machine. As Source Address is always an individual address (Unicast), the least significant bit of the first byte is always 0.
5. **Length:** Length is a 2-Byte field, which indicates the length of

the entire Ethernet frame. This 16-bit field can hold a length value between 0 to 65535, but length cannot be larger than 1500 Bytes because of some own limitations of Ethernet.

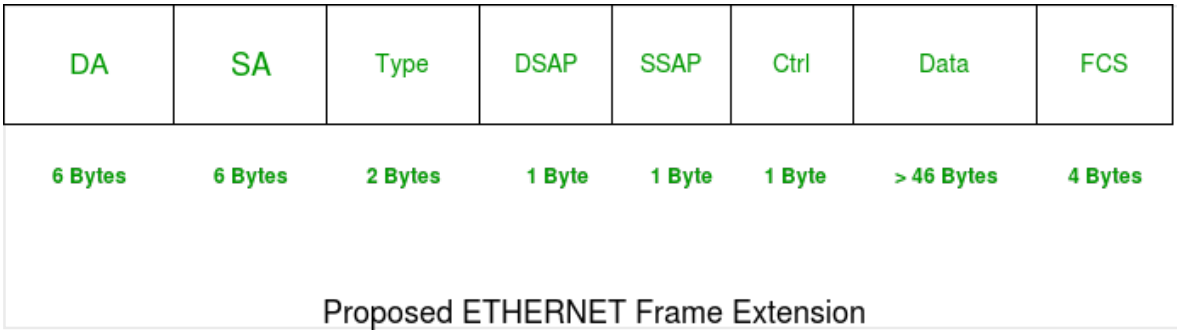
6. **Data:** This is the place where actual data is inserted, also known as **Payload** . Both IP header and data will be inserted here if Internet Protocol is used over Ethernet. The maximum data present may be as long as 1500 Bytes. In case data length is less than minimum length i.e. 46 bytes, then padding 0's is added to meet the minimum possible length.
7. **Cyclic Redundancy Check (CRC):** CRC is 4 Byte field. This field contains a 32-bits hash code of data, which is generated over the Destination Address, Source Address, Length, and Data field. If the checksum computed by destination is not the same as sent checksum value, data received is corrupted.
8. **VLAN Tagging:** The Ethernet frame can also include a VLAN (Virtual Local Area Network) tag, which is a 4-byte field inserted after the source address and before the EtherType field. This tag allows network administrators to logically separate a physical network into multiple virtual networks, each with its own VLAN ID.
9. **Jumbo Frames:** In addition to the standard Ethernet frame size of 1518 bytes, some network devices support Jumbo Frames, which are frames with a payload larger than 1500 bytes. Jumbo Frames can increase network throughput by reducing the overhead associated with transmitting a large number of small frames.
10. **Ether Type Field:** The EtherType field in the Ethernet frame header identifies the protocol carried in the payload of the frame. For example, a value of 0x0800 indicates that the payload is an IP packet, while a value of 0x0806 indicates that the payload is an ARP (Address Resolution Protocol) packet.
11. **Multicast and Broadcast Frames:** In addition to Unicast frames (which are sent to a specific destination MAC address), Ethernet also supports Multicast and Broadcast frames. Multicast frames are sent to a specific group of devices that have joined a multicast group, while Broadcast frames are sent to all devices on the network.
12. **Collision Detection:** In half-duplex Ethernet networks, collisions can occur when two devices attempt to transmit

data at the same time. To detect collisions, Ethernet uses a Carrier Sense Multiple Access with Collision Detection (CSMA/CD) protocol, which listens for activity on the network before transmitting data and backs off if a collision is detected.

Note: *Size of frame of Ethernet IEEE 802.3 varies 64 bytes to 1518 bytes including data length (46 to 1500 bytes).*

Brief Overview on Extended Ethernet Frame (Ethernet II Frame)

Standard IEEE 802.3 basic frame format is discussed above in detail. Now let's see the extended Ethernet frame header, using which we can get a Payload even larger than 1500 Bytes.



Proposed ETHERNET Frame Extension

DA [Destination MAC Address]: ***6 bytes*SA** [Source MAC Address]: ***6 bytes*Type** [0x8870 (Ethertype)]: ***2 bytes*DSAP** [802.2 Destination Service Access Point] : ***1 byte*SSAP** [802.2 Source Service Access Point] : ***1 byte*Ctrl** [802.2 Control Field] : ***1-byte*Data** [Protocol Data] : ***> 46 bytes*FCS** [Frame Checksum]: ***4 bytes*** Although length field is missing in Ethernet II frame, the frame length is known by virtue of the frame being accepted by the network interface.

Advantages

- **Simple Format:** The Ethernet frame format is simple and easy to understand, making it easy to implement and troubleshoot Ethernet networks.
- **Flexibility:** The Ethernet frame format is flexible and can accommodate different data sizes and network topologies, making it suitable for a wide range of network applications.
- **Widely Adopted:** The Ethernet frame format is widely adopted and supported by a large number of vendors and network devices, ensuring compatibility and interoperability.

- **Error Detection:** The Ethernet frame format includes a cyclic redundancy check (CRC) field for error detection, which helps to ensure data integrity during transmission.
- **Support for VLANs:** The Ethernet frame format supports virtual local area networks (VLANs), which allows network administrators to logically partition a physical LAN into multiple smaller virtual LANs for improved network management and security.

Disadvantages

- **Limited Frame Size:** The Ethernet frame format has a maximum frame size of 1500 bytes, which can limit the amount of data that can be transmitted in a single frame and can result in increased overhead due to fragmentation and reassembly of larger packets.
- **Broadcast Storms:** Ethernet networks use broadcast transmissions to send frames to all devices on the network, which can lead to broadcast storms if too many devices send broadcast frames simultaneously, resulting in network congestion and performance issues.
- **Security Vulnerabilities:** The Ethernet frame format does not include built-in security features, making Ethernet networks vulnerable to security threats such as eavesdropping and spoofing .
- **Limited Speed:** Ethernet networks have a limited maximum speed, which may not be sufficient for high-speed applications or large-scale networks.
- **Limited Distance:** The maximum distance between two devices on an Ethernet network is limited, which can restrict the physical coverage of the network.

Conclusion

A fundamental aspect associated with network communication is Ethernet frame format specified under IEEE 802.3 standard. It enables one reliable and flexible way of sending data through the network. Even though it’s rather basic and widely utilized, it has its drawbacks such as maximum frame size and likelihood of security attacks. Thus, there is need for Ethernet professionals to comprehend this format in order to efficiently design, implement or troubleshoot any Ethernet-based local area network (LAN).

Difference between Token Bus and Token Ring

Token Bus Network	Token Ring Network
In the token bus network, the token is passed along a virtual ring.	While in the token ring network the token is passed over a physical ring.
The token bus network is simply designed for large factories.	While the token ring network is designed for the offices.
The token bus network is defined by the IEEE 802.4 standard.	While the token ring network is defined by the IEEE 802.5 standard.
Token bus network provides better bandwidth.	While the token ring network does not provide better bandwidth as compared to the token bus.
In a token bus network, Bus topology is used.	While in token ring network, Star or Ring topology is used.
The maximum time it takes to reach the last station in a token bus network cannot be calculated.	While the maximum time to reach the last station in the token ring network can be calculated.
In a token bus network, coaxial cable is used	In token ring network, twisted pair and fiber optic is used.
In a token bus network, the cable length is 200m to 500m.	In a token ring network, the cable length is 50m to 1000m.
In token bus network, distributed algorithm provide maintenance.	In a token ring network, a designated monitor station performs station maintenance.
The priority handling mechanism is not associated with the transmission of data through workstations with this network.	The priority handling mechanism is associated with the transmission of data through workstations with this network.

These networks are not much reliable.	These networks are reliable.
It does not keep routing details.	It keeps the information of routing.
The network is less expensive compared to the Token Ring network.	It is expensive.

Conclusion

Token Bus and Token Ring are two example of token passing networks, however, their configuration and structure are different. Token Bus is a bus topological network that has the flexibility of efficiency but suffers from problems of flexible maintenance and scalability. Token Ring is the other the topology that we have that is a ring topology providing collision-free reliable performance at the expenses of the cost and complexity. Although both are no longer used at present, knowing the distinction between the two is relevant to the learning the progression of network protocols.

IEEE Standards :

The IEEE 802 family of standards is a set of protocols and standards for Local Area Networks (LANs) and Metropolitan Area Networks (MANs).

Here are some of the most common IEEE standards for LANs and WANs:

LAN Standards:

- **IEEE 802.3:** Defines the physical and data link layer specifications for Ethernet networks, including Ethernet, Fast Ethernet, Gigabit Ethernet, and 10 Gigabit Ethernet, CSMA-CD.
- **IEEE 802.4:** Defines the Token Bus network access method.
- **IEEE 802.5:** Defines the Token Ring network access method.
- **IEEE 802.11:** Defines the standards for Wireless LAN (Wi-Fi) networks, including 802.11a, 802.11b, 802.11g, 802.11n, 802.11ac, and 802.11ax.

WAN Standards:

- **IEEE 802.17:** Defines the standard for Resilient Packet Ring (RPR), a high-speed WAN technology that provides fault tolerance and high availability.

Other relevant standards:

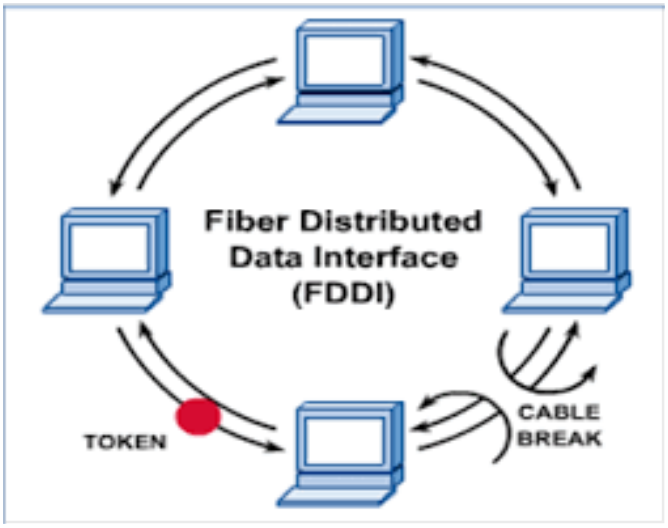
- **IEEE 802.1:** Defines the architecture and services for LANs

and MANs.
- **IEEE 802.2:** Defines the Logical Link Control (LLC) layer protocol.

Fiber Distributed Data Interface (FDDI)

is a network standard that uses fiber optic cables to transmit data in a local area network (LAN):

- Speed: FDDI can transmit data at 100 megabits per second (Mbps).
- Range: FDDI networks can extend up to 200 kilometers (124 miles).
- Protocol: FDDI uses the token ring protocol.
- Suitability: FDDI is suitable for LANs, metropolitan area networks (MANs), campus area networks (CANs), and connecting servers in a room.
- Obolescence: FDDI has been largely superseded by other networking technologies, such as Fast Ethernet and Gigabit Ethernet.
- Historical significance: FDDI was an important technological development that helped pave the way for more modern networking technologies.



FDDI networks typically have two token rings, one primary and one secondary. The primary ring can hold up to 100 Mbps, while the secondary ring can also store data, increasing the capacity to 200 Mbps